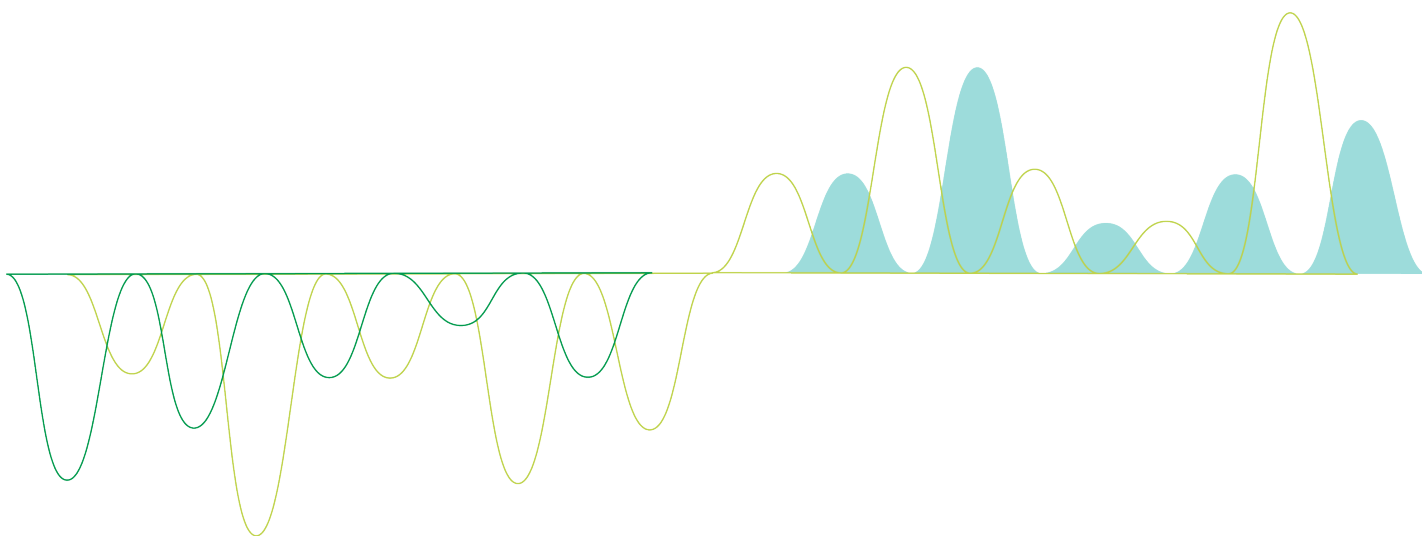


# Scriptsyntaxis en diagramfuncties

Qlik Sense®

November 2022

Copyright © 1993-2023} QlikTech International AB. Alle rechten voorbehouden.





---

<b>1 Wat is Qlik Sense?</b> .....	<b>16</b>
1.1 Wat kunt u doen in Qlik Sense? .....	16
1.2 Hoe werkt Qlik Sense? .....	16
Het app-model .....	16
De associatieve ervaring .....	16
Samenwerking en mobiliteit .....	16
1.3 Hoe implementeert u Qlik Sense? .....	16
Qlik Sense Desktop .....	17
Qlik Sense Enterprise .....	17
1.4 Een Qlik Sense-site besturen en beheren .....	17
1.5 Qlik Sense uitbreiden en aanpassen voor uw eigen doeleinden .....	17
Uitbreidingen en mashups bouwen .....	17
Clients bouwen .....	17
Serverhulpmiddelen bouwen .....	17
Verbinding maken met andere gegevensbronnen .....	17
<b>2 Overzicht scriptsyntaxis</b> .....	<b>18</b>
2.1 Inleiding tot scriptsyntaxis .....	18
2.2 Wat is het Backus-Naur-formalisme? .....	18
<b>2 Scriptinstructies en sleutelwoorden</b> .....	<b>20</b>
2.3 Scriptbesturingsopdrachten .....	20
Overzicht van scriptbesturingsopdrachten .....	20
Call .....	22
Do..loop .....	23
End .....	24
Exit .....	24
Exit script .....	24
For..next .....	25
For each..next .....	26
If..then..elseif..else..end if .....	29
Next .....	30
Sub..end sub .....	30
Switch..case..default..end switch .....	31
To .....	32
2.4 Scriptprefixen .....	32
Overzicht van scriptprefixen .....	32
Add .....	37
Buffer .....	38
Concatenate .....	40
Crosstable .....	45
First .....	55
Generic .....	57
Hierarchy .....	64
HierarchyBelongsTo .....	66
Inner .....	67
IntervalMatch .....	68
Join .....	71
Keep .....	81

---

Left .....	82
Mapping .....	83
Merge .....	85
NoConcatenate .....	90
Only .....	99
Outer .....	99
Gedeeltelijke lading .....	100
Replace .....	103
Right .....	105
Sample .....	106
Semantic .....	109
Unless .....	113
When .....	119
2.5 Reguliere scriptopdrachten .....	125
Overzicht van normale scriptopdrachten .....	125
Alias .....	131
AutoNumber .....	132
Binary .....	135
Comment field .....	137
Comment table .....	137
Connect .....	138
Declare .....	140
Derive .....	142
Directory .....	143
Disconnect .....	144
Drop .....	145
Drop table .....	146
Execute .....	147
Field/Fields .....	148
FlushLog .....	148
Force .....	148
From .....	150
Load .....	150
Let .....	169
Loosen Table .....	169
Map .....	170
NullAsNull .....	171
NullAsValue .....	171
Qualify .....	172
Rem .....	173
Rename .....	174
Search .....	175
Section .....	176
Select .....	177
Set .....	179
Sleep .....	180
SQL .....	180
SQLColumns .....	181

---

SQLTables .....	182
SQLTypes .....	182
Star .....	183
Store .....	185
Table/Tables .....	187
Tag .....	187
Trace .....	188
Unmap .....	189
Unqualify .....	189
Untag .....	190
2.6 Werkdirectory .....	191
Qlik Sense Desktop werkdirectory .....	191
Qlik Sense werkdirectory .....	191
<b>2 Werken met variabelen in de editor voor het laden van gegevens .....</b>	<b>192</b>
2.7 Overzicht .....	192
2.8 Een variabele definiëren .....	192
2.9 Een variabele verwijderen .....	193
2.10 De waarde van een variabele laden als een veldwaarde .....	193
2.11 Berekeningen met variabelen .....	193
2.12 Systeemvariabelen .....	194
Overzicht van systeemvariabelen .....	194
CreateSearchIndexOnReload .....	197
HidePrefix .....	198
HideSuffix .....	198
Include .....	198
OpenUrlTimeout .....	200
StripComments .....	200
Verbatim .....	200
2.13 Variabelen voor afhandeling van waarden .....	201
Overzicht van variabelen voor afhandeling van waarden .....	201
NullDisplay .....	201
NullInterpret .....	202
NullValue .....	202
OtherSymbol .....	202
2.14 Variabelen voor getalinterpretatie .....	203
Valutanotatie .....	203
Getalnotatie .....	203
Tijdnotatie .....	204
BrokenWeeks .....	205
DateFormat .....	206
DayNames .....	212
DecimalSep .....	217
FirstWeekDay .....	219
LongDayNames .....	223
LongMonthNames .....	226
MoneyDecimalSep .....	230
MoneyFormat .....	234

---

---

MoneyThousandSep .....	238
MonthNames .....	243
NumericalAbbreviation .....	248
ReferenceDay .....	249
ThousandSep .....	254
TimeFormat .....	260
TimestampFormat .....	260
2.15 Direct Discovery-variabelen .....	263
Direct Discovery-systeemvariabelen .....	263
Teradata-variabelen voor query banding .....	264
Direct Discovery-tekenvariabelen .....	265
Direct Discovery-variabelen voor getalinterpretatie .....	266
2.16 Foutvariabelen .....	267
Overzicht van foutvariabelen .....	267
ErrorMode .....	267
ScriptError .....	268
ScriptErrorCount .....	269
ScriptErrorList .....	269
<b>2 Scriptuitdrukkingen .....</b>	<b>270</b>
<b>3 Diagramuitdrukkingen .....</b>	<b>271</b>
3.1 Het aggregatiebereik definiëren .....	271
3.2 Set-analyse .....	274
Ingestelde uitdrukkingen .....	274
Voorbeelden .....	275
Natuurlijke sets .....	275
Set-Id's .....	278
Set-operatoren .....	279
Set-modificaties .....	280
Binnenste en buitenste set-uitdrukkingen .....	303
Zelfstudie - Een set-uitdrukking maken .....	305
Syntaxis voor set-uitdrukkingen .....	314
3.3 Algemene syntaxis voor diagramuitdrukkingen .....	314
3.4 Algemene syntaxis voor aggregaties .....	315
<b>4 Operatoren .....</b>	<b>316</b>
4.1 Bit-operatoren .....	316
4.2 Logische operatoren .....	317
4.3 Numerieke operatoren .....	318
4.4 Relationele operatoren .....	318
4.5 Tekenreeksoperatoren .....	320
& .....	320
like .....	320
<b>5 Script- en diagramfuncties .....</b>	<b>322</b>
5.1 Analytische verbindingen voor uitbreidingen aan de serverzijde (SSE) .....	322
5.2 Aggregatiefuncties .....	322
Aggregatiefuncties gebruiken in het script voor het laden van gegevens .....	323
Aggregatiefuncties gebruiken in diagramuitdrukkingen .....	323

---

---

Hoe aggregaties worden berekend .....	323
Aggregatie van sleutelvelden .....	324
Basisaggregatiefuncties .....	325
Teller-aggregatiefuncties .....	348
Financiële aggregatiefuncties .....	365
Statistische aggregatiefuncties .....	387
Statistische testfuncties .....	459
Aggregatiefuncties voor tekenreeksen .....	527
Functies voor synthetische dimensies .....	540
Geneste aggregaties .....	543
5.3 Aggr - diagramfunctie .....	543
Voorbeelden: Diagramuitdrukkingen die gebruikmaken van Aggr .....	546
5.4 Kleurfuncties .....	549
Vooraf gedefinieerde kleurfuncties .....	551
ARGB .....	552
RGB .....	552
HSL .....	554
5.5 Voorwaardenfuncties .....	555
Overzicht voorwaardenfuncties .....	555
alt .....	556
class .....	557
coalesce .....	559
if .....	560
match .....	563
mixmatch .....	566
pick .....	569
wildmatch .....	569
5.6 Tellerfuncties .....	572
Overzicht tellerfuncties .....	572
autonumber .....	574
autonumberhash128 .....	576
autonumberhash256 .....	578
IterNo .....	580
RecNo .....	581
RowNo .....	582
RowNo - diagramfunctie .....	583
5.7 Datum- en tijdfuncties .....	585
Overzicht van datum- en tijdfuncties .....	586
addmonths .....	594
addyears .....	604
age .....	612
converttolocaltime .....	613
day .....	616
dayend .....	622
daylightsaving .....	630
dayname .....	631
daynumberofquarter .....	633
daynumberofyear .....	639

## Contents

---

daystart	645
firstworkdate	653
GMT	654
hour	659
inday	662
indaytotime	671
inlunarweek	681
inlunarweektodate	693
inmonth	704
inmonths	712
inmonthstodate	726
inmonthtodate	739
inquarter	749
inquartertodate	763
inweek	775
inweektodate	792
inyear	806
inyeartodate	819
lastworkdate	832
localtime	842
lunarweekend	843
lunarweekname	855
lunarweekstart	868
makedate	880
maketime	886
makeweekdate	893
minute	901
month	906
monthend	913
monthname	922
monthsend	930
monthsname	943
monthsstart	956
monthstart	970
networkdays	980
now	990
quarterend	997
quartername	1010
quarterstart	1022
second	1034
setdateyear	1039
setdateyearmonth	1041
timezone	1043
today	1043
UTC	1049
week	1049
weekday	1066
weekend	1075



---

weekname .....	1087
weekstart .....	1101
weekyear .....	1113
year .....	1122
yearend .....	1129
yearname .....	1141
yearstart .....	1153
yeartodate .....	1165
5.8 Exponentiële en logaritmische functies .....	1181
5.9 Veldfuncties .....	1182
Tellerfuncties .....	1183
Veld- en selectiefuncties .....	1183
GetAlternativeCount - diagramfunctie .....	1184
GetCurrentSelections - diagramfunctie .....	1185
GetExcludedCount - diagramfunctie .....	1187
GetFieldSelections - diagramfunctie .....	1188
GetNotSelectedCount - diagramfunctie .....	1190
GetObjectDimension - diagramfunctie .....	1190
GetObjectField - diagramfunctie .....	1191
GetObjectMeasure - diagramfunctie .....	1192
GetPossibleCount - diagramfunctie .....	1193
GetSelectedCount - diagramfunctie .....	1194
5.10 Bestandsfuncties .....	1195
Overzicht van bestandsfuncties .....	1195
Attribute .....	1197
ConnectString .....	1204
FileName .....	1205
FileDir .....	1205
FileExtension .....	1205
FileName .....	1206
FilePath .....	1206
FileSize .....	1207
FileTime .....	1207
GetFolderPath .....	1208
QvdCreateTime .....	1209
QvdFieldName .....	1210
QvdNoOfFields .....	1211
QvdNoOfRecords .....	1212
QvdTableName .....	1213
5.11 Financiële functies .....	1214
Overzicht van financiële functies .....	1215
BlackAndSchole .....	1215
FV .....	1216
nPer .....	1217
Pmt .....	1218
PV .....	1219
Rate .....	1220
5.12 Opmaakfuncties .....	1221

---

Overzicht van opmaakfuncties .....	1221
ApplyCodepage .....	1223
Date .....	1224
Dual .....	1225
Interval .....	1227
Money .....	1228
Num .....	1229
Time .....	1232
Timestamp .....	1233
5.13 Algemene numerieke functies .....	1234
Overzicht van algemene numerieke functies .....	1234
Combinatie- en permutatiefuncties .....	1235
Modulofuncties .....	1235
Pariteitsfuncties .....	1236
Afrondingsfuncties .....	1236
BitCount .....	1236
Ceil .....	1237
Combin .....	1238
Div .....	1239
Even .....	1239
Fabs .....	1239
Fact .....	1240
Floor .....	1240
Fmod .....	1241
Frac .....	1242
Mod .....	1243
Odd .....	1244
Permut .....	1244
Round .....	1245
Sign .....	1246
5.14 Geospatiale functies .....	1247
Overzicht van geospatiale functies .....	1247
GeoAggrGeometry .....	1249
GeoBoundingBox .....	1250
GeoCountVertex .....	1250
GeoGetBoundingBox .....	1251
GeoGetPolygonCenter .....	1251
GeoInvProjectGeometry .....	1252
GeoMakePoint .....	1253
GeoProject .....	1253
GeoProjectGeometry .....	1254
GeoReduceGeometry .....	1255
5.15 Interpretatiefuncties .....	1256
Overzicht van interpretatiefuncties .....	1257
Date# .....	1258
Interval# .....	1259
Money# .....	1260
Num# .....	1261

---

Text .....	1262
Time# .....	1263
Timestamp# .....	1264
5.16 Interrecord-functies .....	1265
Rijfuncties .....	1265
Kolomfuncties .....	1266
Veldfuncties .....	1267
Draaitabelfuncties .....	1267
Interrecord-functies in het load-script voor gegevens .....	1268
Above - diagramfunctie .....	1268
Below - diagramfunctie .....	1273
Bottom - diagramfunctie .....	1277
Column - diagramfunctie .....	1281
Dimensionality - diagramfunctie .....	1283
Exists .....	1285
FieldIndex .....	1288
FieldValue .....	1290
FieldValueCount .....	1291
LookUp .....	1293
NoOfRows - diagramfunctie .....	1295
Peek .....	1297
Previous .....	1303
Top - diagramfunctie .....	1304
SecondaryDimensionality - diagramfunctie .....	1308
After - diagramfunctie .....	1309
Before - diagramfunctie .....	1310
First - diagramfunctie .....	1311
Last - diagramfunctie .....	1312
ColumnNo - diagramfunctie .....	1313
NoOfColumns - diagramfunctie .....	1314
5.17 Logische functies .....	1315
5.18 Toewijzingsfuncties .....	1316
Overzicht toewijzingsfuncties .....	1316
ApplyMap .....	1316
MapSubstring .....	1318
5.19 Rekenkundige functies .....	1320
5.20 NULL-functies .....	1321
Overzicht NULL-functies .....	1321
EmptyIsNull .....	1322
IsNull .....	1322
NULL .....	1323
5.21 Bereikfuncties .....	1324
Elementaire bereikfuncties .....	1324
Bereikfuncties voor tellers .....	1325
Statistische bereikfuncties .....	1326
Financiële bereikfuncties .....	1326
RangeAvg .....	1327
RangeCorrel .....	1329

---

---

RangeCount .....	1332
RangeFractile .....	1334
RangeIRR .....	1336
RangeKurtosis .....	1337
RangeMax .....	1338
RangeMaxString .....	1340
RangeMin .....	1342
RangeMinString .....	1344
RangeMissingCount .....	1345
RangeMode .....	1347
RangeNPV .....	1349
RangeNullCount .....	1350
RangeNumericCount .....	1352
RangeOnly .....	1353
RangeSkew .....	1354
RangeStdev .....	1355
RangeSum .....	1357
RangeTextCount .....	1359
RangeXIRR .....	1361
RangeXNPV .....	1362
5.22 Rangorde- en clusterfuncties .....	1363
Rangordefuncties in diagrammen .....	1363
Clusterfuncties in diagrammen .....	1363
Rank - diagramfunctie .....	1365
HRank - diagramfunctie .....	1369
Optimaliseren met k-means: Een praktijkvoorbeeld .....	1371
KMeans2D - diagramfunctie .....	1380
KMeansND - diagramfunctie .....	1393
KMeansCentroid2D - diagramfunctie .....	1406
KMeansCentroidND - diagramfunctie .....	1408
5.23 Statistische verdelingsfuncties .....	1409
Overzicht van statistische verdelingsfuncties .....	1409
BetaDensity .....	1412
BetaDist .....	1412
BetaInv .....	1413
BinomDist .....	1413
BinomFrequency .....	1414
BinomInv .....	1414
ChiDensity .....	1414
ChiDist .....	1415
ChiInv .....	1415
FDensity .....	1416
FDist .....	1417
FInv .....	1417
GammaDensity .....	1418
GammaDist .....	1418
GammaInv .....	1419
NormDist .....	1419

---

NormInv .....	1420
PoissonDist .....	1421
PoissonFrequency .....	1421
PoissonInv .....	1422
TDensity .....	1422
TDist .....	1423
TInv .....	1423
5.24 Tekenreeksfuncties .....	1424
Overzicht van tekenreeksfuncties .....	1424
Capitalize .....	1428
Chr .....	1428
Evaluate .....	1429
FindOneOf .....	1430
Hash128 .....	1431
Hash160 .....	1431
Hash256 .....	1432
Index .....	1433
IsJson .....	1434
JsonGet .....	1435
JsonSet .....	1436
KeepChar .....	1437
Left .....	1438
Len .....	1438
LevenshteinDist .....	1439
Lower .....	1440
LTrim .....	1441
Mid .....	1442
Ord .....	1443
PurgeChar .....	1444
Repeat .....	1444
Replace .....	1445
Right .....	1446
RTrim .....	1447
SubField .....	1448
SubStringCount .....	1450
TextBetween .....	1451
Trim .....	1452
Upper .....	1453
5.25 Systeemfuncties .....	1454
Overzicht van systeemfuncties .....	1454
EngineVersion .....	1456
IsPartialReload .....	1456
ProductVersion .....	1457
StateName - diagramfunctie .....	1457
5.26 Tabelfuncties .....	1457
Overzicht van tabelfuncties .....	1457
FieldName .....	1459
FieldNumber .....	1460

---

---

NoOfFields .....	1460
NoOfRows .....	1461
5.27 Trigonometrische en hyperbolische functies .....	1461
<b>6 Beperking van toegang tot bestandssysteem .....</b>	<b>1464</b>
6.1 Beveiligingsaspecten bij verbinding met op bestanden gebaseerde ODBC- en OLE DB-gegevensverbindingen .....	1464
6.2 Beperkingen in standaardmodus .....	1464
Systeemvariabelen .....	1465
Reguliere scriptopdrachten .....	1466
Scriptbesturingsopdrachten .....	1468
Bestandsfuncties .....	1468
Systeemfuncties .....	1470
6.3 Standaardmodus uitschakelen .....	1471
Qlik Sense .....	1471
Qlik Sense Desktop .....	1471
<b>6 Scripts op diagramniveau .....</b>	<b>1472</b>
6.4 Scriptbesturingsopdrachten .....	1472
Overzicht besturingsopdracht diagramaanpassing .....	1472
Call .....	1474
Do..loop .....	1475
End .....	1476
Exit .....	1476
Exit script .....	1476
For..next .....	1477
For each..next .....	1478
If..then..elseif..else..end if .....	1481
Next .....	1482
Sub..end sub .....	1482
Switch..case..default..end switch .....	1483
To .....	1484
6.5 Prefixen .....	1484
Overzicht prefixen diagramaanpassing .....	1484
Add .....	1485
Replace .....	1485
6.6 Reguliere scriptopdrachten .....	1486
Overzicht besturingsopdracht diagramaanpassing .....	1486
Load .....	1487
Let .....	1491
Set .....	1492
Put .....	1492
HCValue .....	1493
<b>7 QlikView-functies en -opdrachten die niet worden ondersteund in Qlik Sense .....</b>	<b>1495</b>
7.1 Scriptopdrachten die niet worden ondersteund in Qlik Sense .....	1495
7.2 Functies die niet worden ondersteund in Qlik Sense .....	1495
7.3 Prefixes die niet worden ondersteund in Qlik Sense .....	1495
<b>8 Functies en opdrachten die niet worden aanbevolen in Qlik Sense .....</b>	<b>1496</b>
8.1 Scriptopdrachten die niet worden aanbevolen in Qlik Sense .....	1496

---

8.2 Parameters voor scriptopdrachten die niet worden aanbevolen in Qlik Sense .....	1496
8.3 Functies die niet worden aanbevolen in Qlik Sense .....	1497
Kwalificatie ALL .....	1498

# 1 Wat is Qlik Sense?

Qlik Sense is een platform voor gegevensanalyse. Met Qlik Sense kunt u gegevens analyseren en zelf ontdekkingen doen in gegevens. U kunt kennis uitwisselen en gegevens analyseren in groepen en tussen organisaties. Met Qlik Sense kunt u vragen stellen en uw eigen vragen beantwoorden en uw eigen paden naar inzicht volgen. Met Qlik Sense kunnen u en uw collega's samen tot besluiten komen.

## 1.1 Wat kunt u doen in Qlik Sense?

De meeste BI-producten (Business Intelligence) kunnen u helpen vragen te beantwoorden die vooraf duidelijk zijn. Maar hoe zit het met uw vervolgvragen? De vragen die opkomen nadat iemand uw rapport heeft gelezen of uw visualisatie heeft gezien? Met de associatieve ervaring van Qlik Sense kunt u vraag na vraag beantwoorden en uw eigen pad naar inzicht volgen. Met Qlik Sense kunt u vrijelijk uw gegevens verkennen, met slechts enkele klikken, en bij elke stap meer te weten komen en de volgende stappen bedenken op basis van eerdere bevindingen.

## 1.2 Hoe werkt Qlik Sense?

Qlik Sense genereert weergaven van informatie voor u. Qlik Sense vereist geen vooraf gedefinieerde en statische rapporten en maakt u niet afhankelijk van andere gebruikers - alles wat u doet is simpelweg klikken en leren. Telkens wanneer u klikt, reageert Qlik Sense direct en werkt elke Qlik Sense-visualisatie en weergave in de app bij met de nieuw berekende set gegevens en visualisaties die specifiek zijn voor uw selecties.

### Het app-model

In plaats van enorme bedrijfstoepassingen te implementeren en beheren, kunt u uw eigen Qlik Sense-apps maken die u kunt hergebruiken, wijzigen en delen met anderen. Het app-model helpt u zelf de volgende vraag te stellen en te beantwoorden, zonder terug te hoeven gaan naar een specialist voor een nieuw rapport of een nieuwe visualisatie.

### De associatieve ervaring

Qlik Sense beheert automatisch alle relaties in de gegevens en presenteert informatie met behulp van een **green/white/gray** metafoor. Selecties worden groen gemarkeerd, bijbehorende gegevens worden in wit weergegeven en uitgesloten (gekoppelde) gegevens in grijs. Deze directe feedback stelt u in staat om na te denken over nieuwe vragen en om verder te verkennen en te ontdekken.

### Samenwerking en mobiliteit

Qlik Sense stelt u verder in staat samen te werken met collega's op welk tijdstip dan ook en waar deze zich ook bevinden. Alle Qlik Sense-mogelijkheden, met inbegrip van de associatieve ervaring en samenwerking, zijn beschikbaar op mobiele apparaten. Met Qlik Sense kunt u vragen stellen en uw eigen vragen en vervolgvragen beantwoorden, samen met uw collega's, waar u ook bent.

## 1.3 Hoe implementeert u Qlik Sense?

Er zijn twee versies van Qlik Sense om te implementeren, Qlik Sense Desktop en Qlik Sense Enterprise.



## Qlik Sense Desktop

Dit is een eenvoudig te installeren versie voor één gebruiker die gewoonlijk op een lokale computer wordt geïnstalleerd.

## Qlik Sense Enterprise

Deze versie wordt gebruikt voor het implementeren van Qlik Sense-sites. Een site is een verzameling van één of meer servers die zijn verbonden met een gemeenschappelijk logische bibliotheek of een centraal knooppunt.

## 1.4 Een Qlik Sense-site besturen en beheren

Met de Qlik Management Console kunt u Qlik Sense-sites op een eenvoudige en intuïtieve wijze beheren en bewaken. U kunt licenties, toegangs- en beveiligingsregels beheren, knooppunten en verbindingen met gegevensbronnen configureren en inhoud en gebruikers synchroniseren, naast nog vele andere activiteiten en hulpbronnen.

## 1.5 Qlik Sense uitbreiden en aanpassen voor uw eigen doeleinden

Qlik Sense biedt u flexibele API's en SDK's voor het ontwikkelen van uw eigen uitbreidingen en voor het aanpassen en integreren van Qlik Sense voor verschillende doeleinden, zoals:

### Uitbreidingen en mashups bouwen

Hier kunt u aan webontwikkeling doen met behulp van JavaScript om uitbreidingen te bouwen die aangepaste visualisaties vormen in Qlik Sense-apps of u gebruikt de API's van een mashup om websites te bouwen met Qlik Sense-inhoud.

### Clients bouwen

U kunt clients bouwen in .NET en Qlik Sense-objecten insluiten in uw eigen toepassingen. U kunt ook native clients bouwen in elke programmeertaal die WebSocket-communicatie kan verwerken met behulp van het Qlik Sense-clientprotocol.

### Serverhulpmiddelen bouwen

Met service- en gebruikersdirectory-API's kunt u uw eigen hulpmiddelen bouwen voor het besturen en beheren van Qlik Sense-sites.

### Verbinding maken met andere gegevensbronnen

Maak Qlik Sense-connectoren voor het ophalen van gegevens uit aangepaste gegevensbronnen.

## 2 Overzicht scriptsyntaxis

### 2.1 Inleiding tot scriptsyntaxis

In een script worden de namen gedefinieerd van de desbetreffende gegevensbron, tabellen en velden die bij de logica betrokken zijn. Daarnaast worden de velden in de definitie van de toegangsrechten in het script gedefinieerd. Een script bestaat uit een aantal opdrachten die achter elkaar worden uitgevoerd.

De syntaxis van de Qlik Sense-opdrachtregel en de scriptsyntaxis worden beschreven in een notatie die het Backus-Naur-formalisme of BNF-code wordt genoemd.

Als een nieuw Qlik Sense-bestand wordt gemaakt, worden de eerste regels met code automatisch gegenereerd. De standaardwaarden van deze variabelen voor getalinterpretatie worden afgeleid van de landinstellingen in het besturingssysteem.

Het script bestaat uit een aantal scriptopdrachten en sleutelwoorden die achter elkaar worden uitgevoerd. Alle scriptopdrachten moeten met een puntkomma ";" worden afgesloten.

U kunt uitdrukkingen en functies gebruiken in de **LOAD**-opdrachten om de geladen gegevens te transformeren.

Voor een tabelbestand met komma's, tabs of puntkomma's als scheidingsteken kan een **LOAD**-opdracht worden gebruikt. Met een **LOAD**-opdracht worden standaard alle velden in het bestand geladen.

Algemene databases kunnen worden geopend via ODBC- of OLE DB-databaseconnectoren. Daar worden standaard SQL-opdrachten voor gebruikt. De juiste SQL-syntaxis kan per ODBC-stuurprogramma verschillen.

Bovendien kunt u andere gegevensbronnen openen via aangepaste connectoren.

### 2.2 Wat is het Backus-Naur-formalisme?

De syntaxis van de Qlik Sense-opdrachtregel en de scriptsyntaxis worden beschreven in een notatie die het Backus-Naur-formalisme of BNF-code wordt genoemd.

De volgende tabel biedt een lijst met symbolen die worden gebruikt in BNF-code, met een beschrijving van hoe zij worden geïnterpreteerd:

Symbolen

Symbool	Beschrijving
	Logische OR: het symbool aan beide zijden kan worden gebruikt.
()	Haakjes geven prioriteit aan: ze worden gebruikt om de BNF-syntaxis te structureren.
[]	Vierkante haken: items tussen vierkante haakjes zijn optioneel.
{ }	Accolades: items tussen accolades kunnen nul of meer keren worden herhaald.

Symbol	Beschrijving
Symbol	Een niet-eindige syntactische categorie: een categorie die verder kan worden opgesplitst in andere symbolen. Bijvoorbeeld samenstellingen van de bovenstaande, andere niet-terminal symbolen, teksteekensreeksen, enzovoort.
::=	Geeft het begin aan van een blok dat een symbool definieert.
<b>LOAD</b>	Een eindsymbool dat bestaat uit een teksteekensreeks. Moet precies zoals aangegeven worden geschreven in het script.

Alle eindsymbolen worden afgedrukt in een **bold face** lettertype. Bijvoorbeeld: "(" moet worden geïnterpreteerd als een haakje dat prioriteit aangeeft, terwijl "(" moet worden geïnterpreteerd als een teken dat moet worden afgedrukt in het script.

### Voorbeeld:

De omschrijving van de alias-opdracht is:

```
alias fieldname as aliasname { , fieldname as aliasname }
```

Dit moet worden geïnterpreteerd als de teksteekensreeks "alias", gevolgd door een willekeurige veldnaam, gevolgd door de teksteekensreeks "as", gevolgd door een willekeurige aliasnaam. Elk aantal aanvullende combinaties van "fieldname as alias" kan worden gegeven, gescheiden door komma's.

De volgende opdrachten zijn correct:

```
alias a as first;  
alias a as first, b as second;  
alias a as first, b as second, c as third;
```

De volgende opdrachten zijn niet correct:

```
alias a as first b as second;  
alias a as first { , b as second };
```

## 2 Scriptinstructies en sleutelwoorden

Het Qlik Sense-script bestaat uit een aantal opdrachten. Een opdracht kan een normale scriptopdracht of een scriptbesturingsopdracht zijn. Bepaalde opdrachten kunnen worden voorafgegaan door prefixen.

Normale opdrachten worden over het algemeen gebruikt om gegevens op een bepaalde manier te manipuleren. Deze opdrachten kunnen worden geschreven op een willekeurig aantal regels in het script en moeten altijd worden afgesloten met een puntkomma (;).

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Elke zin van een besturingsopdracht moet op één scriptregel blijven en kan worden afgesloten met een puntkomma of een regeleinde.

Prefixen kunnen worden toegevoegd aan normale opdrachten, maar nooit aan besturingsopdrachten. De prefixen **when** en **unless** kunnen wel als suffix bij enkele specifieke clauses van besturingsopdrachten worden gebruikt.

In de volgende sectie vindt u een alfabetisch overzicht van alle scriptopdrachten, besturingsopdrachten en prefixen.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

### 2.3 Scriptbesturingsopdrachten

Het Qlik Sense-script bestaat uit een aantal opdrachten. Een opdracht kan een normale scriptopdracht of een scriptbesturingsopdracht zijn.

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Elke zin van een besturingsopdracht moet op één scriptregel blijven en kan worden afgesloten met een puntkomma of een regeleinde.

Prefixen zijn nooit van toepassing op besturingsopdrachten, met uitzondering van de prefixen **when** en **unless**, die bij een aantal specifieke besturingsopdrachten kunnen worden gebruikt.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken.

### Overzicht van scriptbesturingsopdrachten

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Call

De besturingsopdracht **call** roept een subroutine aan die door een vorige **sub**-opdracht is gedefinieerd.

```
Call name ( [ paramlist ] )
```

### Do..loop

De besturingsopdracht **do..loop** is een herhalingsconstructie voor gebruik in scripts waarmee één of meer opdrachten worden uitgevoerd totdat aan een logische voorwaarde wordt voldaan.

```
Do..loop [ ( while | until ) condition ] [statements]
[exit do [ ( when | unless ) condition ] [statements]
loop [ ( while | until ) condition ]
```

### Exit script

Met deze besturingsopdracht wordt de uitvoering van het script stopgezet. Deze opdracht kan overal in het script worden ingevoegd.

```
Exit script [ (when | unless) condition ]
```

### For each ..next

De besturingsopdracht **for each..next** is een herhalingsconstructie voor gebruik in scripts waarmee een of meer opdrachten worden uitgevoerd voor elke waarde in een door komma's gescheiden lijst. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de lijst.

```
For each..next var in list
[statements]
[exit for [ ( when | unless ) condition ]
[statements]
next [var]
```

### For..next

De besturingsopdracht **for..next** is een herhalingsconstructie voor gebruik in scripts met een teller. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de tellervariabele tussen de opgegeven onder- en bovengrens.

```
For..next counter = expr1 to expr2 [ stepexpr3 ]
[statements]
[exit for [ ( when | unless ) condition ]
[statements]
Next [counter]
```

### If..then

De besturingsopdracht **if..then** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van één of meer logische voorwaarden, op verschillende manieren kan verlopen.



*Omdat de instructie **if..then** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clausules (**if..then**, **elseif..then**, **else** en **end if**) de regelgrens overschrijden.*

```
If..then..elseif..else..end if condition then
[ statements ]
{ elseif condition then
[ statements ] }
[ else
```

```
[ statements ] ]  
end if
```

### Sub

De besturingsopdracht **sub..end sub** geeft een subroutine aan die vanuit een **call**-opdracht kan worden aangeroepen.

```
Sub..end sub name [ ( paramlist ) ] statements end sub
```

### Switch

De besturingsopdracht **switch** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van de waarde van een uitdrukking, op verschillende manieren kan verlopen.

```
Switch..case..default..end switch expression {case valuelist [ statements ]}  
[default statements] end switch
```

### Call

De besturingsopdracht **call** roept een subroutine aan die door een vorige **sub**-opdracht is gedefinieerd.

#### Syntaxis:

```
Call name ( [ paramlist ] )
```

#### Argumenten:

Argumenten

Argument	Beschrijving
name	De naam van de subroutine.
paramlist	Een door komma's gescheiden lijst met de werkelijke parameters die worden gezonden aan de subroutine. Elk item in de lijst kan een veldnaam, een variabele of een willekeurige uitdrukking zijn.

De subroutine die wordt opgeroepen door een **call**-opdracht moet op een eerdere positie in het script door een **sub** zijn gedefinieerd.

Parameters worden gekopieerd naar de subroutine en worden, als de parameter in de **call**-instructie een variabele is en geen uitdrukking, bij het beëindigen van de subroutine weer teruggekopieerd.

#### Beperkingen:

- Aangezien de **call**-opdracht een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag de opdracht niet op meerdere regels staan.
- Als u een subroutine definieert met `sub..end sub` binnen een besturingsopdracht, bijvoorbeeld `if..then`, kunt u de subroutine alleen oproepen binnen dezelfde besturingsopdracht.

### Voorbeeld:

In dit voorbeeld worden alle aan Qlik gerelateerde bestanden in een map en in de bijbehorende submap opgesomd en wordt bestandsinformatie opgeslagen in een tabel. Er wordt vanuit gegaan dat u een gegevensverbinding hebt gemaakt met de naam Apps in de map.

De subroutine DoDir wordt aangeroepen met de verwijzing naar de map , 'lib://Apps', als parameter. Binnen de subroutine, is er een recursieve aanroep, call DoDir (Dir), waarmee de functie recursief naar bestanden kijkt in submappen.

```
sub DoDir (Root)      For Each Ext in 'qvw', 'qvo', 'qvs', 'qvt', 'qvd', 'qvc', 'qvf'      For
Each File in filelist (Root&'\'*'.' &Ext)          LOAD          '$(File)' as Name,
      FileSize( '$(File)' ) as Size,          FileTime( '$(File)' ) as FileTime
autogenerate 1;      Next File      Next Ext      For Each Dir in dirlist (Root&'\'*' )
Call DoDir (Dir)      Next Dir End Sub      Call DoDir ('lib://Apps')
```

### Do..loop

De besturingsopdracht **do..loop** is een herhalingsconstructie voor gebruik in scripts waarmee één of meer opdrachten worden uitgevoerd totdat aan een logische voorwaarde wordt voldaan.

### Syntaxis:

```
Do [ ( while | until ) condition ] [statements]
[exit do [ ( when | unless ) condition ] [statements]
loop[ ( while | until ) condition ]
```



Omdat de instructie **do..loop** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clausules ervan (**do**, **exit do** en **loop**) de regelgrens overschrijden.

### Argumenten:

#### Argumenten

Argument	Beschrijving
condition	Een logische uitdrukking die resulteert in True of False.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.
while / until	De voorwaardelijke clausule <b>while</b> of <b>until</b> mag slechts eenmaal voorkomen in elke <b>do..loop</b> -opdracht, namelijk na <b>do</b> of na <b>loop</b> . Elke voorwaarde wordt alleen de eerste keer geïnterpreteerd wanneer deze wordt aangetroffen, maar wordt telkens in de lus geëvalueerd.

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
exit do	Als een clause <b>exit do</b> binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clause <b>loop</b> die het einde van de lus aangeeft. De clause <b>exit do</b> kan voorwaardelijk worden gemaakt met het optionele suffix <b>when</b> of <b>unless</b> .

### Voorbeeld:

```
// LOAD files file1.csv..file9.csv
Set a=1;
Do while a<10
LOAD * from file$(a).csv;
Let a=a+1;
Loop
```

## End

Het sleutelwoord **End** voor scripts wordt gebruikt om **If**-, **Sub**- en **Switch**-clausules af te sluiten.

## Exit

Het sleutelwoord **Exit** voor scripts maakt onderdeel uit van de **Exit Script**-opdracht, maar kan ook worden gebruikt voor het afsluiten van **Do**-, **For**- of **Sub**-clausules.

## Exit script

Met deze besturingsopdracht wordt de uitvoering van het script stopgezet. Deze opdracht kan overal in het script worden ingevoegd.

### Syntaxis:

```
Exit Script [ (when | unless) condition ]
```

Aangezien de **exit script**-opdracht een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag de opdracht niet op meerdere regels staan.

### Argumenten:

#### Argumenten

Argument	Beschrijving
condition	Een logische uitdrukking die resulteert in True of False.
when / unless	De opdracht <b>exit script</b> kan voorwaardelijk worden gemaakt met de optionele clause <b>when</b> of <b>unless</b> .

### Voorbeelden:

```
//Exit script
```



```
Exit Script;
```

```
//Exit script when a condition is fulfilled  
Exit Script when a=1
```

### For..next

De besturingsopdracht **for..next** is een herhalingsconstructie voor gebruik in scripts met een teller. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de tellervariabele tussen de opgegeven onder- en bovengrens.

#### Syntaxis:

```
For counter = expr1 to expr2 [ step expr3 ]  
[statements]  
[exit for [ ( when | unless ) condition ]  
[statements]  
Next [counter]
```

De uitdrukkingen *expr1*, *expr2* en *expr3* worden alleen de eerste keer dat de lus wordt ingevoerd geëvalueerd. De waarde van de tellervariabele kan worden gewijzigd door opdrachten binnen de lus. Dit is echter met het oog op de gebruikelijke manier van programmeren niet aan te raden.

Als een clause **exit for** binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clause **next** die het einde van de lus aangeeft. De clause **exit for** kan voorwaardelijk worden gemaakt met het optionele suffix **when** of **unless**.



*Omdat de instructie **for..next** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clauses ervan (**for..to..step**, **exit for** en **next**) de regelgrens overschrijden.*

#### Argumenten:

##### Argumenten

Argument	Beschrijving
counter	De naam van een variabele. Als <i>counter</i> wordt opgegeven na <b>next</b> moet de variabele dezelfde naam hebben als de variabele die wordt gevonden na de overeenkomende <b>for</b> .
expr1	Een uitdrukking die de eerste waarde van de variabele <i>counter</i> bepaalt waarvoor de lus wordt uitgevoerd.
expr2	Een uitdrukking die de laatste waarde van de variabele <i>counter</i> bepaalt waarvoor de lus wordt uitgevoerd.

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
expr3	Een uitdrukking die de waarde bepaalt van de stap waarmee de variabele <i>counter</i> toeneemt bij elke uitvoering van de lus.
condition	Een logische uitdrukking die resulteert in True of False.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.

### Example 1: Een reeks van bestanden laden

```
// LOAD files file1.csv..file9.csv
for a=1 to 9
    LOAD * from file$(a).csv;
next
```

### Example 2: Een willekeurig aantal bestanden laden

In dit voorbeeld gaan we ervan uit dat we de gegevensbestanden *x1.csv*, *x3.csv*, *x5.csv*, *x7.csv* en *x9.csv* hebben. Het laden wordt bij een willekeurig punt gestopt met de voorwaarde `if rand( )<0.5 then`.

```
for counter=1 to 9 step 2
    set filename=x$(counter).csv;
    if rand( )<0.5 then
        exit for unless counter=1
    end if
    LOAD a,b from $(filename);
next
```

## For each..next

De besturingsopdracht **for each..next** is een herhalingsconstructie voor gebruik in scripts waarmee een of meer opdrachten worden uitgevoerd voor elke waarde in een door komma's gescheiden lijst. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de lijst.

### Syntaxis:

Door een speciale syntaxis kunnen lijsten worden gegenereerd met namen van bestanden en mappen in de huidige map.

```
for each var in list
[statements]
[exit for [ ( when | unless ) condition ]
[statements]
next [var]
```

## 2 Scriptinstructies en sleutelwoorden

### Argumenten:

#### Argumenten

Argument	Beschrijving
var	De naam van een scriptvariabele die een nieuwe waarde uit de lijst ophaalt, telkens wanneer de lus wordt uitgevoerd. Als <b>var</b> wordt opgegeven na <b>next</b> moet de variabele dezelfde naam hebben als de variabele die wordt gevonden na de overeenkomende <b>for each</b> .

De waarde van de **var** kan worden gewijzigd door opdrachten binnen de lus. Dit is echter met het oog op de gebruikelijke manier van programmeren niet aan te raden.

Als een clause **exit for** binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clause **next** die het einde van de lus aangeeft. De clause **exit for** kan voorwaardelijk worden gemaakt met het optionele suffix **when** of **unless**.



*Omdat de instructie **for each..next** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clauses ervan (**for each**, **exit for** en **next**) de regelgrens overschrijden.*

### Syntaxis:

```
list := item { , item }  
item := constant | (expression) | filelist mask | dirlist mask |  
fieldvaluelist mask
```

#### Argumenten

Argument	Beschrijving
constant	Elk getal of elke tekenreeks. Een tekenreeks die rechtstreeks in het script wordt opgenomen moet tussen enkele aanhalingstekens worden geplaatst. Een tekenreeks zonder enkele aanhalingstekens wordt geïnterpreteerd als een variabele en vervolgens wordt de waarde van de variabele gebruikt. Getallen hoeven niet tussen enkele aanhalingstekens te worden geplaatst.
expression	Een willekeurige uitdrukking.
mask	Een masker voor de naam van een bestand of map die geldige tekens voor bestandsnamen plus de standaardjokertekens, * en ?, kan bevatten.  U kunt absolute bestandspaden of paden lib:// gebruiken.
condition	Een logische uitdrukking die resulteert in True of False.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
filelist mask	Deze syntaxis produceert een lijst met door komma's gescheiden namen van alle bestanden in de huidige map die overeenkomen met het bestandsnaammasker.   <i>Dit argument ondersteunt alleen bibliotheekverbindingen in de standaardmodus.</i>
dirlist mask	Deze syntaxis produceert een lijst met door komma's gescheiden namen van alle mappen in de huidige map die overeenkomen met het mapnaammasker.   <i>Dit argument ondersteunt alleen bibliotheekverbindingen in de standaardmodus.</i>
fieldvaluelist mask	Deze syntaxis itereert door de waarden van een veld die al in Qlik Sense zijn geladen.



*De Qlik Connectoren van internetopslagprovider en andere DataFiles-verbindingen bieden geen ondersteuning voor filtermaskeringen die jokertekens (\* en ?) gebruiken.*

### Example 1: Een lijst van bestanden laden

```
// LOAD the files 1.csv, 3.csv, 7.csv and xyz.csv for each a in 1,3,7,'xyz' LOAD * from  
file$(a).csv; next
```

### Example 2: Een lijst van bestanden maken op schijf

Via dit voorbeeld wordt een lijst met alle aan Qlik Sense gerelateerde bestanden in een map geladen.

```
sub DoDir (Root) for each Ext in 'qwv', 'qva', 'qvo', 'qvs', 'qvc', 'qvf', 'qvd'  
for each File in filelist (Root&'/*.' &Ext) LOAD '$(File)' as Name,  
FileSize( '$(File)' ) as Size, FileTime( '$(File)' ) as FileTime  
autogenerate 1; next File next Ext for each Dir in dirlist (Root&'/*' )  
call DoDir (Dir) next Dir end sub call DoDir ('lib://DataFiles')
```

### Example 3: Itereren door de waarden van een veld

Dit voorbeeld itereert door de waarden van geladen waarden van FIELD en genereert een nieuw veld, NEWFIELD. Voor elke waarde van FIELD, worden twee NEWFIELD-records gemaakt.

```
load * inline [ FIELD one two three ]; FOR Each a in FieldValueList('FIELD') LOAD '$(a)' &'-  
&RecNo() as NEWFIELD AutoGenerate 2; NEXT a
```

De resulterende tabel ziet er als volgt uit:

## 2 Scriptinstructies en sleutelwoorden

Example table

<b>NEWFIELD</b>
one-1
one-2
two-1
two-2
three-1
three-2

### If..then..elseif..else..end if

De besturingsopdracht **if..then** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van één of meer logische voorwaarden, op verschillende manieren kan verlopen.

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Gebruik in een diagramuitdrukking in plaats daarvan de **if** voorwaardelijke functie.

#### Syntaxis:

```
If condition then
  [ statements ]
{ elseif condition then
  [ statements ] }
[ else
  [ statements ] ]
end if
```

Omdat de instructie **if..then** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clausules (**if..then**, **elseif..then**, **else** en **end if**) de regelgrens overschrijden.

#### Argumenten:

Argumenten

Argument	Beschrijving
condition	Een logische uitdrukking die resulteert in True of False.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.

#### Example 1:

```
if a=1 then
    LOAD * from abc.csv;
    SQL SELECT e, f, g from tab1;
```

end if

### Example 2:

```
if a=1 then; drop table xyz; end if;
```

### Example 3:

```
if x>0 then
    LOAD * from pos.csv;
elseif x<0 then
    LOAD * from neg.csv;
else
    LOAD * from zero.txt;
end if
```

## Next

Het sleutelwoord **Next** voor scripts wordt gebruikt om **For**-lussen af te sluiten.

## Sub..end sub

De besturingsopdracht **sub..end sub** geeft een subroutine aan die vanuit een **call**-opdracht kan worden aangeroepen.

### Syntaxis:

```
Sub name [ ( paramlist ) ] statements end sub
```

De argumenten worden in de subroutine gekopieerd en, als de corresponderende parameter in de **call**-opdracht een variabelenaam is, weer teruggekopieerd bij het verlaten van de subroutine.

Als een subroutine meer formele parameters heeft dan er aan daadwerkelijke parameters door een **call**-opdracht worden doorgegeven, worden de overtollige parameters op NULL ingesteld en kunnen ze als lokale variabelen binnen de subroutine worden gebruikt.

### Argumenten:

Argumenten

Argument	Beschrijving
name	De naam van de subroutine.
paramlist	Een door komma's gescheiden lijst variabelen voor de formele parameters van de subroutine. Deze kunnen worden gebruikt als willekeurige variabelen binnen de subroutine.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.

### Beperkingen:

- Omdat de instructie **sub** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de twee mogelijke clauses (**sub** en **end sub**) de regelgrens overschrijden.
- Als u een subroutine definieert met `sub . . end sub` binnen een besturingsopdracht, bijvoorbeeld `if . . then`, kunt u de subroutine alleen oproepen binnen dezelfde besturingsopdracht.

### Example 1:

```
Sub INCR (I,J)
I = I + 1
Exit Sub when I < 10
J = J + 1
End Sub
Call INCR (X,Y)
```

### Example 2: - parameteroverdracht

```
Sub ParTrans (A,B,C)
A=A+1
B=B+1
C=C+1
End Sub
A=1
X=1
C=1
Call ParTrans (A, (X+1)*2)
```

Het resultaat van het bovenstaande is dat lokaal, binnen de subroutine, A wordt geïnitieerd op 1, B op 4 en C op NULL.

Bij het verlaten van de subroutine krijgt de globale variabele A de waarde 2 (teruggekopieerd van de subroutine). De tweede werkelijke parameter "(X+1)\*2" wordt niet teruggekopieerd omdat dit geen variabele is. Tot slot verandert de globale variabele C niet door de aanroep van de subroutine.

## Switch..case..default..end switch

De besturingsopdracht **switch** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van de waarde van een uitdrukking, op verschillende manieren kan verlopen.

### Syntaxis:

```
Switch expression {case valuelist [ statements ]} [default statements] end
switch
```



Omdat de instructie **switch** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clauses (**switch**, **case**, **default** en **end switch**) de regelgrens overschrijden.

### Argumenten:

Argumenten

Argument	Beschrijving
expression	Een willekeurige uitdrukking.
valuelist	Een door komma's gescheiden lijst waarden waarmee de waarde van de uitdrukking zal worden vergeleken. Uitvoering van het script wordt vervolgd met de opdrachten in de eerste groep waarvoor een waarde in valuelist is gevonden die gelijk is aan de waarde in de uitdrukking. Elke waarde in valuelist kan een willekeurige uitdrukking zijn. Als in geen van de <b>case</b> -opdrachten een overeenkomst is gevonden, worden de opdrachten achter <b>default</b> , indien gespecificeerd, uitgevoerd.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.

### Voorbeeld:

```
Switch I
Case 1
LOAD '$(I): CASE 1' as case autogenerate 1;
Case 2
LOAD '$(I): CASE 2' as case autogenerate 1;
Default
LOAD '$(I): DEFAULT' as case autogenerate 1;
End Switch
```

## To

Het sleutelwoord **To** voor scripts wordt gebruikt in verschillende scriptopdrachten.

## 2.4 Scriptprefixen

Prefixen kunnen worden toegevoegd aan normale opdrachten, maar nooit aan besturingsopdrachten. De prefixen **when** en **unless** kunnen wel als suffix bij enkele specifieke clausules van besturingsopdrachten worden gebruikt.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

## Overzicht van scriptprefixen

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Add

Het voorvoegsel **Add** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat er records aan een andere tabel moeten worden toegevoegd. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Add** kan ook worden



## 2 Scriptinstructies en sleutelwoorden

---

gebruikt in een **Map**-opdracht.

```
Add [only] [Concatenate [(tablename )]] (loadstatement | selectstatement)
Add [ Only ] mapstatement
```

### Buffer

QVD-bestanden kunnen automatisch worden gemaakt en beheerd met behulp van het prefix **buffer**. Dit prefix kan worden gebruikt bij de meeste **LOAD**- en **SELECT**-opdrachten in een script. Hiermee wordt aangegeven dat QVD-bestanden worden gebruikt als cache/buffer om het resultaat van de opdracht op te slaan.

```
Buffer [(option [ , option])] ( loadstatement | selectstatement )
option ::= incremental | stale [after] amount [(days | hours)]
```

### Concatenate

Als twee tabellen die u wilt aaneenschakelen verschillende sets velden hebben, kunt u aaneenschakeling toch afdwingen met het prefix **Concatenate**.

```
Concatenate [ (tablename ) ] ( loadstatement | selectstatement )
```

### Crosstable

Het laadvoorvoegsel **crosstable** wordt gebruikt om gestructureerde gegevens in "kruistabel" of "draaitabel" te transponeren. Gegevens die op deze manier zijn gestructureerd, worden vaak aangetroffen bij het werken met spreadsheetbronnen. De uitvoer en het doel van het laadvoorvoegsel **crosstable** is om dergelijke structuren om te zetten in een normaal kolomgeoriënteerd tabelequivalent, aangezien deze structuur over het algemeen beter geschikt is voor analyse in Qlik Sense.

```
Crosstable (attribute field name, data field name [ , n ] ) ( loadstatement |
selectstatement )
```

### First

Het prefix **First** voor een **LOAD**- of **SELECT (SQL)**-opdracht wordt gebruikt voor het laden van een vast maximaal aantal records uit een gegevensbrontabel.

```
First n ( loadstatement | selectstatement )
```

### Generic

Het laadvoorvoegsel **Generic** maakt conversie mogelijk van entiteit-attribuu-waarde gemodelleerde gegevens (EAV) naar een traditionele, genormaliseerde relationele tabelstructuur. EAV-modellering wordt ook wel "generieke gegevensmodellering" of "open schema" genoemd.

```
Generic ( loadstatement | selectstatement )
```

### Hierarchy

Het prefix **hierarchy** wordt gebruikt om een hiërarchietabel met bovenliggende-onderliggende objecten om te zetten in een tabel die bruikbaar is in een Qlik Sense-gegevensmodel. Het prefix kan voor een **LOAD**- of **SELECT**-opdracht worden geplaatst en gebruikt het resultaat van de laadopdracht als invoer voor een tabeltransformatie.

```
Hierarchy (NodeID, ParentID, NodeName, [ParentName], [PathSource],
[PathName], [PathDelimiter], [Depth]) (loadstatement | selectstatement)
```

### HierarchBelongsTo

Dit prefix wordt gebruikt om een hiërarchietabel met bovenliggende-onderliggende objecten om te zetten in een tabel die bruikbaar is in een Qlik Sense-gegevensmodel. Het prefix kan voor een **LOAD**- of **SELECT**-opdracht worden geplaatst en gebruikt het resultaat van de laadopdracht als invoer voor een tabeltransformatie.

```
HierarchyBelongsTo (NodeID, ParentID, NodeName, AncestorID, AncestorName, [DepthDiff]) (loadstatement | selectstatement)
```

### Inner

De prefixen **join** en **keep** kunnen worden voorafgegaan door het prefix **inner**.

Indien gebruikt voor **join**, geeft de opdracht aan dat een inner join moet worden gebruikt. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in beide tabellen voorkomen. Indien gebruikt voor **keep**, geeft de opdracht aan dat beide tabellen met onbewerkte gegevens worden gereduceerd tot de gemeenschappelijke doorsnede voordat ze worden opgeslagen in Qlik Sense.

```
Inner ( Join | Keep ) [ (tablename) ] (loadstatement | selectstatement )
```

### IntervalMatch

Het prefix **IntervalMatch** wordt gebruikt voor het maken van een tabel waarin discrete numerieke waarden worden gekoppeld aan een of meer numerieke intervallen, en waarin de waarden optioneel worden gekoppeld aan een of meer extra sleutels.

```
IntervalMatch (matchfield) (loadstatement | selectstatement )  
IntervalMatch (matchfield, keyfield1 [ , keyfield2, ... keyfield5 ] )  
(loadstatement | selectstatement )
```

### Join

Het prefix **join** koppelt de geladen tabel aan een bestaande benoemde tabel of de laatste eerder gemaakte gegevenstabel.

```
[Inner | Outer | Left | Right ] Join [ (tablename) ] ( loadstatement |  
selectstatement )
```

### Keep

Het prefix **keep** is vergelijkbaar met het prefix **join**. Net als bij het prefix **join** wordt de geladen tabel vergeleken met een bestaande benoemde tabel of de laatste eerder gemaakte gegevenstabel. Maar de geladen tabel wordt niet samengevoegd met een bestaande tabel, in plaats daarvan wordt een tabel of worden beide tabellen voordat ze worden opgeslagen in Qlik Sense gereduceerd op basis van de doorsnede van de tabelgegevens. De uitgevoerde vergelijking is equivalent aan een natuurlijke join voor alle gemeenschappelijke velden, net zoals bij een overeenkomstige join. De twee tabellen worden echter niet samengevoegd, maar als twee afzonderlijk benoemde tabellen in Qlik Sense bewaard.

```
(Inner | Left | Right) Keep [ (tablename) ] ( loadstatement | selectstatement  
)
```

### Left

De prefixen **Join** en **Keep** kunnen worden voorafgegaan door het prefix **left**.

Gebruikt voor **join** geeft het prefix aan dat er een join links moet plaatsvinden. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in de eerste tabel voorkomen. Als het prefix wordt gebruikt vóór **keep**, geeft het aan dat de tweede tabel met onbewerkte gegevens wordt gereduceerd tot de gemeenschappelijke doorsnede met de eerste tabel, voordat deze wordt opgeslagen in Qlik Sense.

```
Left ( Join | Keep ) [ (tablename) ] (loadstatement | selectstatement )
```

### Mapping

Het prefix **mapping** wordt gebruikt om een toewijzingstabel te maken die bijvoorbeeld kan worden gebruikt voor het vervangen van veldwaarden en veldnamen tijdens de uitvoering van het script.

```
Mapping ( loadstatement | selectstatement )
```

### Merge

Het voorvoegsel **Merge** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat de geladen tabel met een andere tabel moet worden samengevoegd. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd.

```
Merge [only] [(SequenceNoField [, SequenceNoVar])] On ListOfKeys [Concatenate [(TableName)]] (loadstatement | selectstatement)
```

### NoConcatenate

Het prefix **NoConcatenate** zorgt ervoor dat twee geladen tabellen met identieke veldensets worden behandeld als twee aparte interne tabellen, terwijl ze anders automatisch zouden worden aaneengeschakeld.

```
NoConcatenate ( loadstatement | selectstatement )
```

### Outer

Het expliciete prefix **Join** kan worden voorafgegaan door het prefix **Outer** om een outer join op te geven. In een outer join worden alle combinaties tussen de twee tabellen gegenereerd. De resulterende tabel bevat dus gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in één of beide tabellen voorkomen. Het trefwoord **Outer** is optioneel en is het standaard join type dat wordt gebruikt als geen join prefix is opgegeven.

```
Outer Join [ (tablename) ] (loadstatement | selectstatement )
```

### Partial reload

Bij een volledige lading worden altijd eerst alle tabellen in het bestaande gegevensmodel verwijderd en vervolgens wordt het load-script uitgevoerd.

Bij een *Gedeeltelijke lading (page 100)* gebeurt dit niet. Hierbij blijven alle tabellen in het gegevensmodel behouden en worden alleen de opdrachten **Load** en **Select** uitgevoerd voorafgegaan door het voorvoegsel **Add**, **Merge** of **Replace**. Andere gegevenstabellen worden niet door de instructie beïnvloed. Het **only**-

## 2 Scriptinstructies en sleutelwoorden

---

argument geeft aan dat de opdracht alleen tijdens gedeeltelijke ladingen moet worden uitgevoerd, en moet worden genegeerd tijdens volledige ladingen. In de volgende tabel vindt u een overzicht van de uitvoering van opdrachten voor gedeeltelijke en volledige ladingen.

### Replace

het voorvoegsel **Replace** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat de geladen tabel een andere tabel moet vervangen. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Replace** kan ook worden gebruikt in een **Map**-opdracht.

```
Replace [only] [Concatenate [(tablename) ]] (loadstatement | selectstatement)  
Replace [only] mapstatement
```

### Right

De prefixen **Join** en **Keep** kunnen worden voorafgegaan door het prefix **right**.

Als het prefix wordt gebruikt vóór **join**, geeft het aan dat er een join rechts moet plaatsvinden. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in de tweede tabel voorkomen. Als het prefix wordt gebruikt voor **keep**, geeft het aan dat de eerste tabel met onbewerkte gegevens wordt gereduceerd tot de gemeenschappelijke doorsnede met de tweede tabel, voordat deze wordt opgeslagen in Qlik Sense.

```
Right (Join | Keep) [(tablename)] (loadstatement | selectstatement )
```

### Sample

Als u het prefix **sample** voor een **LOAD**- of **SELECT**-opdracht plaatst, wordt er een willekeurige selectie records uit de gegevensbron geladen.

```
Sample p ( loadstatement | selectstatement )
```

### Semantic

Tabellen met relaties tussen records kunnen worden geladen met het prefix **semantic**. Dit kunnen bijvoorbeeld interne referenties binnen een tabel zijn waarbij een record wijst naar een andere, zoals bovenliggend, behoort tot of voorganger.

```
Semantic ( loadstatement | selectstatement )
```

### Unless

Met het prefix en suffix **unless** maakt u een conditionele clause waarmee wordt bepaald of een opdracht of exit-clausule wel of niet moet worden geëvalueerd. De opdracht kan worden beschouwd als een compact alternatief voor een volledige **if..end if**-opdracht.

```
(Unless condition statement | exitstatement Unless condition )
```

### When

Met het prefix en suffix **when** maakt u een conditionele clause waarmee wordt bepaald of een opdracht of exit-clausule wel of niet moet worden uitgevoerd. De opdracht kan worden beschouwd als een compact alternatief voor een volledige **if..end if**-opdracht.

```
( When condition statement | exitstatement when condition )
```

### Add

Het voorvoegsel **Add** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat er records aan een andere tabel moeten worden toegevoegd. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Add** kan ook worden gebruikt in een **Map**-opdracht.



*Om ervoor te zorgen dat een gedeeltelijke lading correct werkt, moet de app met gegevens worden geopend voordat een gedeeltelijke lading wordt geactiveerd.*

Voer een gedeeltelijke lading uit met de knop **Opnieuw laden**. U kunt ook de Qlik Engine JSON API gebruiken.

#### Syntaxis:

```
Add [only] [Concatenate [(tablename)]] (loadstatement | selectstatement)
```

```
Add [only] mapstatement
```

Tijdens een normale (niet-gedeeltelijke) lading, dient de constructie **Add LOAD** als een normale **LOAD**-instructie. Records worden gegenereerd en opgeslagen in een tabel.

Als het voorvoegsel **Concatenate** is gebruikt of is een tabel aanwezig met dezelfde verzameling van velden, worden de records toegevoegd aan de relevante bestaande tabel. Ander zorgt de constructie **Add LOAD** dat er een nieuwe tabel wordt gemaakt.

Bij een gedeeltelijke lading gebeurt hetzelfde. Het enige verschil is dat de bij de constructie **Add LOAD** nooit een nieuwe tabel wordt gemaakt. Er bestaat altijd een relevante tabel van de uitvoering van het vorige script waarmee de records moeten worden samengevoegd.

Er wordt geen controle op duplicaten uitgevoerd. Daarom bevat een instructie waarbij het prefix **Add** wordt gebruikt gewoonlijk een kwalificatie voor **distinct** of een **where**-clausule om duplicaten te voorkomen.

De instructie **Add Map...Using** zorgt ervoor dat de toewijzing ook plaatsvindt als het script gedeeltelijk wordt uitgevoerd.

#### Argumenten:

Argumenten

Argument	Beschrijving
only	Een optionele kwalificatie die aangeeft dat de opdracht alleen moet worden uitgevoerd tijdens het gedeeltelijk laden van gegevens. Het moet worden genegeerd tijdens een normale (niet-gedeeltelijke) lading.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
<p>Tab1:</p> <pre>LOAD Name, Number FROM Persons.csv;</pre> <p>Add LOAD Name, Number FROM newPersons.csv;</p>	<p>Tijdens normaal opnieuw laden worden gegevens geladen uit <i>Persons.csv</i> en opgeslagen in de Qlik Sense-tabel Tab1. Gegevens uit <i>NewPersons.csv</i> worden vervolgens samengevoegd met dezelfde Qlik Sense-tabel.</p> <p>Tijdens het gedeeltelijk opnieuw laden worden gegevens geladen uit <i>NewPersons.csv</i> en toegevoegd aan de Qlik Sense-tabel Tab1. Er wordt geen controle op duplicaten uitgevoerd.</p>
<p>Tab1:</p> <pre>SQL SELECT Name, Number FROM Persons.csv;</pre> <p>Add LOAD Name, Number FROM NewPersons.csv where not exists(Name);</p>	<p>Controle op duplicaten wordt uitgevoerd door te kijken of Name bestaat in de eerder geladen tabelgegevens.</p> <p>Tijdens normaal opnieuw laden worden gegevens geladen uit <i>Persons.csv</i> en opgeslagen in de Qlik Sense-tabel Tab1. Gegevens uit <i>NewPersons.csv</i> worden vervolgens samengevoegd met dezelfde Qlik Sense-tabel.</p> <p>Tijdens het gedeeltelijk opnieuw laden worden gegevens geladen uit <i>NewPersons.csv</i> die wordt toegevoegd aan de Qlik Sense-tabel Tab1. Controle op duplicaten wordt uitgevoerd door te kijken of Name bestaat in de eerder geladen tabelgegevens.</p>
<p>Tab1:</p> <pre>LOAD Name, Number FROM Persons.csv;</pre> <p>Add Only LOAD Name, Number FROM NewPersons.csv where not exists(Name);</p>	<p>Tijdens normaal opnieuw laden worden gegevens geladen uit <i>Persons.csv</i> en opgeslagen in de Qlik Sense-tabel Tab1. De opdracht voor het laden van <i>NewPersons.csv</i> wordt genegeerd.</p> <p>Tijdens het gedeeltelijk opnieuw laden worden gegevens geladen uit <i>NewPersons.csv</i> die wordt toegevoegd aan de Qlik Sense-tabel Tab1. Controle op duplicaten wordt uitgevoerd door te kijken of Name bestaat in de eerder geladen tabelgegevens.</p>

## Buffer

QVD-bestanden kunnen automatisch worden gemaakt en beheerd met behulp van het prefix **buffer**. Dit prefix kan worden gebruikt bij de meeste **LOAD**- en **SELECT**-opdrachten in een script. Hiermee wordt aangegeven dat QVD-bestanden worden gebruikt als cache/buffer om het resultaat van de opdracht op te slaan.

### Syntaxis:

```
Buffer [(option [ , option])] ( loadstatement | selectstatement )
option ::= incremental | stale [after] amount [(days | hours)]
```

Als deze opties niet zijn gedefinieerd, wordt de QVD-buffer die bij de eerste uitvoering van het script werd gemaakt, voor onbepaalde tijd gebruikt.

Het bufferbestand wordt opgeslagen in de submap *Buffers*, normaliter *C:\ProgramData\Qlik\Sense\Engine\Buffers* (serverinstallatie) of *C:\Gebruikers\{user}\Mijn Documenten\Qlik\Sense\Buffers* (Qlik Sense Desktop).

---

## 2 Scriptinstructies en sleutelwoorden

---

De naam van het QVD-bestand is een berekende naam, een 160-bits hexadecimale hash van de gehele volgende **LOAD** of **SELECT**-opdracht en andere onderscheidende informatie. Dit betekent dat de QVD-buffer ongeldig wordt door een wijziging in de volgende **LOAD**- of **SELECT**-opdracht.

Een QVD-buffer wordt normaal gesproken verwijderd als er tijdens de volledige uitvoering van het script in de app waarvoor de buffer werd gemaakt, niet langer naar wordt verwezen of als de app waarvoor de buffer werd gemaakt, niet meer bestaat.

### Argumenten:

#### Argumenten

Argument	Beschrijving
incrementeel	<p>Met de optie <code>incremental</code> wordt aangegeven dat alleen een gedeelte van een onderliggend bestand hoeft te worden gelezen. De vorige grootte van het bestand wordt opgeslagen in de XML-header van het QVD-bestand. Dit is vooral nuttig bij logbestanden. Alle records die eerder zijn geladen, worden gelezen uit het QVD-bestand. Nieuwe bestanden worden gelezen uit de originele bron en vervolgens wordt een bijgewerkt QVD-bestand gemaakt.</p> <p>De optie <code>incremental</code> kan alleen worden gebruikt met <b>LOAD</b>-opdrachten en tekstbestanden. Incrementeel laden kan niet worden gebruikt als oude gegevens zijn gewijzigd of verwijderd.</p>
stale [after] amount [(days   hours)]	<p><code>amount</code> is een getal dat de tijdsperiode aangeeft. Het gebruik van decimalen is toegestaan. Als er geen eenheid is aangegeven, wordt verondersteld dat het om dagen gaat.</p> <p>De optie <code>stale after</code> wordt meestal gebruikt voor DB-bronnen waarbij de gegevens niet zijn voorzien van een tijdsstempel. In plaats daarvan geeft men aan hoe oud een gebruikte QVD-momentopname mag zijn. Een <code>stale after</code>-clausule geeft een tijdsperiode aan. Als de QVD-buffer ouder is dan deze periode, wordt de buffer niet meer als geldig beschouwd. Voor deze tijd wordt de QVD-buffer gebruikt als gegevensbron, daarna wordt de oorspronkelijke gegevensbron gebruikt. Het QVD-bufferbestand wordt dan automatisch bijgewerkt en er begint een nieuwe periode.</p>

### Beperkingen:

Er zijn talloze beperkingen. De belangrijkste is dat er een **LOAD**- of een **SELECT**-opdracht voor een bestand moet voorkomen in de "basis" van een complexe opdracht.

#### Example 1:

```
Buffer SELECT * from MyTable;
```

#### Example 2:

```
Buffer (stale after 7 days) SELECT * from MyTable;
```

### Example 3:

```
Buffer (incremental) LOAD * from MyLog.log;
```

## Concatenate

`concatenate` is een load-prefix van het script waardoor een gegevensset kan worden bijgevoegd bij een bestaande tabel in het geheugen. Het wordt vaak gebruikt om verschillende sets transactiegegevens toe te voegen aan één centrale feitentabel, of veel voorkomende referentiegegevenssets van een specifiek type op te bouwen die voortkomen uit verschillende bronnen. Het is vergelijkbaar wat betreft functionaliteit met de SQL UNION-operator.

De resulterende tabel van een concatenate-bewerking omvat vervolgens de oorspronkelijke gegevensset met de nieuwe rijen gegevens die onderaan aan die tabel zijn toegevoegd. In de bron- en doeltabellen kunnen verschillende velden staan. Indien velden anders zijn, wordt de resulterende tabel verbreed om het gecombineerde resultaat van alle aanwezige velden in zowel de bron- als de doeltabel weer te geven.

### Syntaxis:

```
Concatenate [ (tablename ) ] ( loadstatement | selectstatement )
```

#### Argumenten

Argument	Beschrijving
tablename	De naam van een bestaande tabel. De genoemde tabel wordt het doel van de concatenate-bewerking en eventuele records van geladen gegevens worden aan die tabel toegevoegd. Als de parameter tablename niet wordt gebruikt, wordt de doeltabel de laatst geladen tabel vóór deze opdracht.
loadstatement/selectstatement	Het loadstatement/selectstatement-argument dat op het tablename-argument volgt, wordt aaneengeschakeld met de opgegeven tabel.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze



## 2 Scriptinstructies en sleutelwoorden

---

landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Funcatievoorbeeld

Voorbeeld	Resultaat
Concatenate (Transactions) Load ... ;	De gegevens die in de load-instructie onder het concatenate-prefix zijn geladen, worden toegevoegd aan de bestaande tabel in het geheugen, Transactions. Er wordt hierbij vanuit gegaan dat de tabel Transactions eerder dan dit punt in het load-script al is geladen.

### Voorbeeld 1 - Meerdere gegevenssets toegevoegd aan een doeltabel met de load-prefix voor aaneenschakelen

Load-script en resultaten

#### Overzicht

In dit voorbeeld laadt u twee scripts achter elkaar.

- Het eerste load-script bevat een aanvankelijke gegevensset met datums en bedragen, die naar de tabel Transactions wordt gestuurd.
- Het tweede load-script bevat:
  - Een tweede gegevensset die aan de eerste gegevensset wordt toegevoegd met behulp van de concatenate-prefix. Deze gegevensset heeft een extra veld, type, dat niet in de eerste gegevensset staat.
  - Het concatenate-prefix.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

#### Eerste load-script

```
Transactions:  
Load * Inline [
```

```
id, date, amount  
3750, 08/30/2018, 23.56  
3751, 09/07/2018, 556.31  
3752, 09/16/2018, 5.75  
3753, 09/22/2018, 125.00  
3754, 09/22/2018, 484.21  
3756, 09/22/2018, 59.18  
3757, 09/23/2018, 177.42  
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- amount

Resultatentabel eerste load-script

id	date	amount
3750	08/30/2018	23.56
3751	09/07/2018	556.31
3752	09/16/2018	5.75
3753	09/22/2018	125.00
3754	09/22/2018	484.21
3756	09/22/2018	59.18
3757	09/23/2018	177.42

De tabel toont de eerste gegevensset.

### Tweede load-script

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe.

```
concatenate(Transactions)
Load * Inline [
id, date, amount, type
3758, 10/01/2018, 164.27, Internal
3759, 10/03/2018, 384.00, External
3760, 10/06/2018, 25.82, Internal
3761, 10/09/2018, 312.00, Internal
3762, 10/15/2018, 4.56, Internal
3763, 10/16/2018, 90.24, Internal
3764, 10/18/2018, 19.32, External
];
```

### Resultaten

Laad de gegevens en ga naar het werkblad. Maak dit tabel als een dimensie.

- type

Resultatentabel tweede load-script

id	date	amount	type
3750	08/30/2018	23.56	-
3751	09/07/2018	556.31	-
3752	09/16/2018	5.75	-

id	date	amount	type
3753	09/22/2018	125.00	-
3754	09/22/2018	484.21	-
3756	09/22/2018	59.18	-
3757	09/23/2018	177.42	-
3758	10/01/2018	164.27	Intern
3759	10/03/2018	384.00	Extern
3760	10/06/2018	25.82	Intern
3761	10/09/2018	312.00	Intern
3762	10/15/2018	4.56	Intern
3763	10/16/2018	90.24	Intern
3764	10/18/2018	19.32	Extern

Voor de nulwaarden in het type-veld voor de eerste zeven geladen records was type nog niet was gedefinieerd.

### Voorbeeld 2 - Meerdere gegevenssets toegevoegd aan een doeltabel met impliciete aaneenschakeling

Load-script en resultaten

#### Overzicht

Een typische gebruikssituatie voor impliciete toevoeging van gegevens is wanneer u verschillende bestanden met identiek gestructureerde gegevens laadt en u ze allemaal aan een doeltabel wilt toevoegen.

Bijvoorbeeld door wildcards in bestandsnamen te gebruiken met een syntax zoals:

```
myTable:
Load * from [myFile_*.qvd] (qvd);
```

of in lussen met gebruik van constructies zoals:

```
for each file in filelist('myFile_*.qvd')

myTable:
Load * from [$(file)] (qvd);

next file
```



*Impliciete aaneenschakeling vindt plaats tussen twee geladen tabellen met identieke veldnamen, ook als ze niet na elkaar in het script zijn gedefinieerd. Dit kan ertoe leiden dat gegevens onbedoeld aan tabellen worden toegevoegd. Als u niet wilt dat een secundaire tabel met identieke velden op deze manier wordt toegevoegd, gebruikt u het load-prefix `NoConcatenate`. De tabel een andere tabelnaam geven is niet afdoende om impliciete aaneenschakeling te voorkomen. Ga voor meer informatie naar `NoConcatenate` (page 90).*

In dit voorbeeld laadt u twee scripts achter elkaar.

- Het eerste load-script bevat een aanvankelijke gegevensset met vier velden, die naar de tabel `Transactions` wordt gestuurd.
- Het tweede load-script bevat een gegevensset met dezelfde velden als de eerste gegevensset.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

### Eerste load-script

```
Transactions:
Load * Inline [
id, date, amount, type
3758, 10/01/2018, 164.27, Internal
3759, 10/03/2018, 384.00, External
3760, 10/06/2018, 25.82, Internal
3761, 10/09/2018, 312.00, Internal
3762, 10/15/2018, 4.56, Internal
3763, 10/16/2018, 90.24, Internal
3764, 10/18/2018, 19.32, External
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `amount`
- `type`

Resultatentabel eerste load-script

<b>id</b>	<b>date</b>	<b>type</b>	<b>amount</b>
3758	10/01/2018	Intern	164.27
3759	10/03/2018	Extern	384.00
3760	10/06/2018	Intern	25.82
3761	10/09/2018	Intern	312.00
3762	10/15/2018	Intern	4.56

## 2 Scriptinstructies en sleutelwoorden

---

id	date	type	amount
3763	10/16/2018	Intern	90.24
3764	10/18/2018	Extern	19.32

De tabel toont de eerste gegevensset.

### Tweede load-script

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe.

```
Load * Inline [  
id, date, amount, type  
3765, 11/03/2018, 129.40, Internal  
3766, 11/05/2018, 638.50, External  
];
```

### Resultaten

Laad de gegevens en ga naar het werkblad.

Resultatentabel tweede load-script

id	date	type	amount
3758	10/01/2018	Intern	164.27
3759	10/03/2018	Extern	384.00
3760	10/06/2018	Intern	25.82
3761	10/09/2018	Intern	312.00
3762	10/15/2018	Intern	4.56
3763	10/16/2018	Intern	90.24
3764	10/18/2018	Extern	19.32
3765	11/03/2018	Intern	129.40
3766	11/05/2018	Extern	638.50

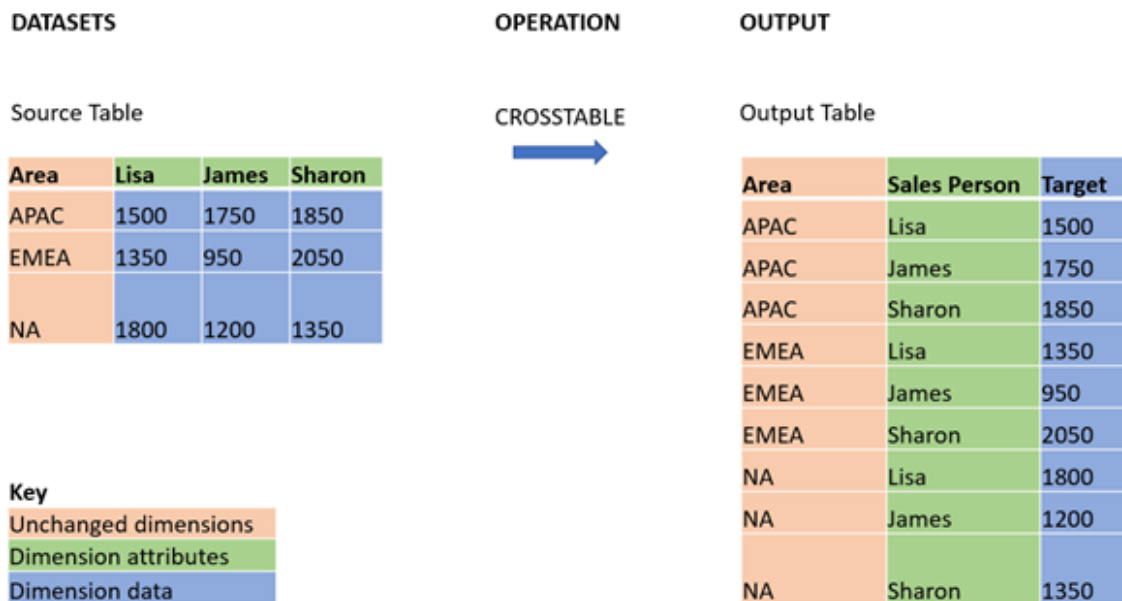
De tweede gegevensset is impliciet aaneengeschakeld met de eerste gegevensset omdat de sets identieke velden bevatten.

### Crosstable

Het laadvoorvoegsel **crosstable** wordt gebruikt om gestructureerde gegevens in "kruistabel" of "draaitabel" te transponeren. Gegevens die op deze manier zijn gestructureerd, worden vaak aangetroffen bij het werken met spreadsheetbronnen. De uitvoer en het doel van het laadvoorvoegsel **crosstable** is om dergelijke structuren om te zetten in een normaal kolomgeoriënteerd tabelequivalent, aangezien deze structuur over het algemeen beter geschikt is voor analyse in Qlik Sense.

## 2 Scriptinstructies en sleutelwoorden

Voorbeeld van gegevens die zijn gestructureerd als een kruistabel en de equivalente structuur ervan na een kruistabeltransformatie



### Syntaxis:

```
crosstable (attribute field name, data field name [ , n ] ) ( loadstatement | selectstatement )
```

#### Argumenten

Argument	Beschrijving
attribute field name	De gewenste uitvoerveldnaam die de horizontaal georiënteerde dimensie beschrijft die moet worden getransponeerd (de koprij).
data field name	De gewenste uitvoerveldnaam die de horizontaal georiënteerde gegevens beschrijft van de dimensie die moet worden getransponeerd (de matrix van gegevenswaarden onder de koprij).
n	Het aantal kwalificatievelden, of ongewijzigde dimensies, dat voor de tabel staat, die in generieke vorm moet worden veranderd. De standaardwaarde is 1.

Deze scriptfunctie is gerelateerd aan de volgende functies:

#### Gerelateerde functies

Functie	Interactie
<i>Generic</i> (page 57)	Een voorvoegsel voor het laden van de transformatie dat een gestructureerde gegevensset van entiteit-attribuut-waarde neemt en deze omzet in een normale relationele tabelstructuur, waarbij elk gevonden attribuut wordt gescheiden in een nieuw veld of nieuwe kolom met gegevens.

### Voorbeeld 1 - Omzetten van gedraaide verkoopgegevens (eenvoudig)

Load-scripts en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het eerste onderstaande load-script toe aan een nieuw tabblad.

Het eerste load-script bevat een gegevensset waarop het scriptvoorvoegsel `crosstable` later zal worden toegepast, waarbij de sectie die van toepassing is op `crosstable` wordt weggelaten. Dit betekent dat de syntaxis van opmerkingen is gebruikt om deze sectie in het load-script uit te schakelen.

Het tweede load-script is hetzelfde als het eerste, maar met de toepassing van `crosstable` zonder commentaar (ingeschakeld door de commentaarsyntaxis te verwijderen). De scripts worden op deze manier weergegeven om de waarde van deze scriptfunctie bij het transformeren van gegevens te benadrukken.

#### Eerste load-script (functie niet toegepast)

```
tmpData:
//Crosstable (MonthText, Sales)
Load * inline [
Product, Jan 2021, Feb 2021, Mar 2021, Apr 2021, May 2021, Jun 2021
A, 100, 98, 103, 63, 108, 82
B, 284, 279, 297, 305, 294, 292
C, 50, 53, 50, 54, 49, 51];

//Final:
//Load Product,
//Date(Date#(MonthText, 'MMM YYYY'), 'MMM YYYY') as Month,
//Sales

//Resident tmpData;

//Drop Table tmpData;
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Product
- Jan 2021
- Feb 2021
- Mar 2021
- Apr 2021
- May 2021
- Jun 2021

## 2 Scriptinstructies en sleutelwoorden

Resultatentabel

Product	Jan 2021	Feb 2021	Mrt 2021	Apr 2021	Mei 2021	Jun 2021
A	100	98	103	63	108	82
B	284	279	297	305	294	292
C	50	53	50	54	49	51

Met dit script kan een kruistabel met een kolom voor elke maand en één rij per product worden gemaakt. In de huidige indeling kunnen deze gegevens niet gemakkelijk worden geanalyseerd. Het is veel beter als alle getallen zich in één veld bevinden en alle maanden in een ander veld, in een tabel met drie kolommen. In de volgende sectie wordt uitgelegd hoe u deze transformatie naar de kruistabel kunt uitvoeren.

### Tweede load-script (functie toegepast)

Verwijder het commentaar van het script door de // te verwijderen. Het load-script zou er als volgt moeten uitzien:

```
tmpData:
Crosstable (MonthText, Sales)
Load * inline [
Product, Jan 2021, Feb 2021, Mar 2021, Apr 2021, May 2021, Jun 2021
A, 100, 98, 103, 63, 108, 82
B, 284, 279, 297, 305, 294, 292
C, 50, 53, 50, 54, 49, 51];

Final:
Load Product,
Date(Date#(MonthText, 'MMM YYYY'), 'MMM YYYY') as Month,
Sales

Resident tmpData;

Drop Table tmpData;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Product
- Month
- Sales

Resultatentabel

Product	Month	Sales
A	Jan 2021	100



## 2 Scriptinstructies en sleutelwoorden

---

Product	Month	Sales
A	Feb 2021	98
A	Mar 2021	103
A	Apr 2021	63
A	May 2021	108
A	Jun 2021	82
B	Jan 2021	284
B	Feb 2021	279
B	Mar 2021	297
B	Apr 2021	305
B	May 2021	294
B	Jun 2021	292
C	Jan 2021	50
C	Feb 2021	53
C	Mar 2021	50
C	Apr 2021	54
C	May 2021	49
C	Jun 2021	51

Wanneer het scriptvoorvoegsel is toegepast, wordt de kruistabel getransformeerd in een rechte tabel met één kolom voor `month` en nog een voor `sales`. Dit verbetert de leesbaarheid van de gegevens.

### Voorbeeld 2 - Omzetten van gedraaide verkoopdoelgegevens in een verticale tabelstructuur (tussenliggend)

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Doelen`.
- Het laadvoorvoegsel `crostable`, dat de gedraaide verkopersnamen omzet in een eigen veld, met het label `sales Person`.
- De bijbehorende verkoopdoelgegevens, die zijn gestructureerd in een veld met de naam `target`.

### Load-script

```
SalesTargets:
CROSTABLE([Sales Person],Target,1)
LOAD
*
INLINE [
Area, Lisa, James, Sharon
APAC, 1500, 1750, 1850
EMEA, 1350, 950, 2050
NA, 1800, 1200, 1350
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Area
- Sales Person

Voeg deze meting toe:

```
=Sum(Target)
```

Resultatentabel

Gebied	Verkoop-medewerker	=Sum(Target)
APAC	James	1750
APAC	Lisa	1500
APAC	Sharon	1850
EMEA	James	950
EMEA	Lisa	1350
EMEA	Sharon	2050
n.v.t.	James	1200
n.v.t.	Lisa	1800
n.v.t.	Sharon	1350

Als u de weergave van gegevens wilt repliceren als de gedraaide invoertabel, kunt u een equivalente draaitabel in een werkblad maken.

### Doe het volgende:

1. Kopieer en plak de tabel die u zojuist hebt gemaakt in het werkblad.
2. Sleep het diagramobject **Draaitabel** bovenop de nieuw gemaakte tabelkopie. Selecteer **Converteren**.

## 2 Scriptinstructies en sleutelwoorden

---

3. Klik op ✓ **Klaar met bewerken**.
4. Sleep het veld `sa1es Person` van de verticale kolomplank naar de horizontale kolomplank.

De volgende tabel toont de gegevens in de oorspronkelijke tabelvorm, zoals weergegeven in Qlik Sense:

Originele resultatentabel, zoals weergegeven in Qlik Sense

Gebied	Verkoop-medewerker	=Sum(Target)
Totalen	-	13800
APAC	James	1750
APAC	Lisa	1500
APAC	Sharon	1850
EMEA	James	950
EMEA	Lisa	1350
EMEA	Sharon	2050
n.v.t.	James	1200
n.v.t.	Lisa	1800
n.v.t.	Sharon	1350

De equivalente draaitabel ziet er als volgt uit, met de kolom voor de naam van elke verkoper in de grotere rij voor `sa1es Person`:

Equivalente draaitabel met het veld `sa1es Person` horizontaal gedraaid

Gebied	James	Lisa	Sharon
APAC	1750	1500	1850
EMEA	950	1350	2050
n.v.t.	1350	1350	1350

## 2 Scriptinstructies en sleutelwoorden

Voorbeeld van gegevens weergegeven als tabel en een equivalente draaitabel met het veld `Sales Person` horizontaal gedraaid

Table			
Area	Sales Person		Sum(Target)
Totals			13800
APAC	James		1750
APAC	Lisa		1500
APAC	Sharon		1850
EMEA	James		950
EMEA	Lisa		1350
EMEA	Sharon		2050
NA	James		1200
NA	Lisa		1800
NA	Sharon		1350

Pivot table			
Area	Sales Person		
	James	Lisa	Sharon
APAC	1750	1500	1850
EMEA	950	1350	2050
NA	1200	1800	1350

### Voorbeeld 3 - Omzetten van gedraaide verkoop- en doelgegevens in een verticale tabelstructuur (geavanceerd)

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met verkoop- en doelgegevens, geordend op gebied en maand van het jaar. Dit wordt geladen in een tabel genaamd `SalesAndTargets`.
- Het laadvoorvoegsel `CROSTABLE`. Dit wordt gebruikt om het draaien van de dimensie `Month Year` ongedaan te maken in een speciaal veld en om de matrix van verkoop- en doelbedragen om te zetten in een speciaal veld met de naam `Amount`.
- Een conversie van het veld `Month Year` van tekst naar een goede datum, met behulp van de tekst-naar-datum-conversiefunctie `date#`. Dit naar datum geconverteerde veld `Month Year` wordt teruggekoppeld naar de tabel `SalesAndTarget` via een laadvoorvoegsel `Join`.

#### Load-script

`SalesAndTargets:`

```
CROSTABLE(MonthYearAsText, Amount, 2)
```

```
LOAD
```

```
*
```

```
INLINE [
```

Area	Type	Jan-22	Feb-22	Mar-22	Apr-22	May-22	Jun-22	Jul-22	Aug-22	Sep-22	Oct-22	Nov-22	Dec-22
APAC	Target	425	425	425	425	425	425	425	425	425	425	425	425
APAC	Actual	435	434	397	404	458	447	413	458	385	421	448	397

## 2 Scriptinstructies en sleutelwoorden

```
EMEA Target 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5 362.5
EMEA Actual 363.5 359.5 337.5 361.5 341.5 337.5 379.5 352.5 327.5 337.5 360.5 334.5
NA Target 375 375 375 375 375 375 375 375 375 375 375 375 375 375
NA Actual 378 415 363 356 403 343 401 365 393 340 360 405
```

```
] (delimiter is '\t');
```

```
tmp:
```

```
LOAD DISTINCT MonthYearAsText,date#(MonthYearAsText,'MMM-YY') AS [Month Year]
RESIDENT SalesAndTargets;
```

```
JOIN (SalesAndTargets)
```

```
LOAD * RESIDENT tmp;
```

```
DROP TABLE tmp;
```

```
DROP FIELD MonthYearAsText;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Area
- Month Year

Maak de volgende meting, met het label Actual:

```
=Sum({<Type={'Actual'}>} Amount)
```

Maak ook deze meting, met het label Target:

```
=Sum({<Type={'Target'}>} Amount)
```

Resultatentabel (bijgesneden)

Gebied	Maand Jaar	Werkelijk	Doel
APAC	Jan-22	435	425
APAC	Feb-22	434	425
APAC	Mar-22	397	425
APAC	Apr-22	404	425
APAC	May-22	458	425
APAC	Jun-22	447	425
APAC	Jul-22	413	425
APAC	Aug-22	458	425
APAC	Sep-22	385	425
APAC	Oct-22	421	425
APAC	Nov-22	448	425

## 2 Scriptinstructies en sleutelwoorden

---

Gebied	Maand Jaar	Werkelijk	Doel
APAC	Dec-22	397	425
EMEA	Jan-22	363.5	362.5
EMEA	Feb-22	359.5	362.5

Als u de weergave van gegevens wilt repliceren als de gedraaide invoertabel, kunt u een equivalente draaitabel in een werkblad maken.

### Doe het volgende:

1. Kopieer en plak de tabel die u zojuist hebt gemaakt in het werkblad.
2. Sleep het diagramobject **Draaitabel** bovenop de nieuw gemaakte tabelkopie. Selecteer **Converteren**.
3. Klik op **✓ Klaar met bewerken**.
4. Sleep het veld `Month Year` van de verticale kolomplank naar de horizontale kolomplank.
5. Sleep het item `values` van de horizontale kolomplank naar de verticale kolomplank.

De volgende tabel toont de gegevens in de oorspronkelijke tabelvorm, zoals weergegeven in Qlik Sense:

Originele resultatentabel (bijgesneden), zoals weergegeven in Qlik Sense

Gebied	Maand Jaar	Werkelijk	Doel
Totalen	-	13812	13950
APAC	Jan-22	435	425
APAC	Feb-22	434	425
APAC	Mar-22	397	425
APAC	Apr-22	404	425
APAC	May-22	458	425
APAC	Jun-22	447	425
APAC	Jul-22	413	425
APAC	Aug-22	458	425
APAC	Sep-22	385	425
APAC	Oct-22	421	425
APAC	Nov-22	448	425
APAC	Dec-22	397	425
EMEA	Jan-22	363.5	362.5
EMEA	Feb-22	359.5	362.5

## 2 Scriptinstructies en sleutelwoorden

De equivalente draaitabel ziet er als volgt uit, met de kolom voor elke afzonderlijke maand van het jaar in de grotere rij voor Month Year:

Equivalente draaitabel (bijgesneden) met het veld Month Year horizontaal gedraaid

Gebied (Waarden)	Jan-22	Feb-22	Mar-22	Apr-22	May-22	Jun-22	Jul-22	Aug-22	Sep-22	Oct-22	Nov-22	Dec-22
APAC - Actueel	435	434	397	404	458	447	413	458	385	421	448	397
APAC - Doel	425	425	425	425	425	425	425	425	425	425	425	425
EMEA - Actueel	363.5	359.5	337.5	361.5	341.5	337.5	379.5	352.5	327.5	337.5	360.5	334.5
EMEA - Doel	362.5	362.5	362.5	362.5	362.5	362.5	362.5	362.5	362.5	362.5	362.5	362.5
N.v.t. - Actueel	378	415	363	356	403	343	401	365	393	340	360	405
N.v.t. - Doel	375	375	375	375	375	375	375	375	375	375	375	375

Voorbeeld van gegevens weergegeven als tabel en een equivalente draaitabel met het veld Month Year horizontaal gedraaid

Table				Pivot table													
Area	Q	Month Year	Q	Actual	Target												
Totals				13812	13990												
APAC		Jan-22	Q	435	425	Jan-22	Feb-22	Mar-22	Apr-22	May-22	Jun-22	Jul-22	Aug-22	Sep-22	Oct-22	Nov-22	Dec-22
APAC		Feb-22	Q	434	425	435	434	397	404	458	447	413	458	385	421	448	397
APAC		Mar-22	Q	397	425	425	425	425	425	425	425	425	425	425	425	425	425
APAC		Apr-22	Q	404	425	363.5	359.5	337.5	361.5	341.5	337.5	379.5	352.5	327.5	337.5	360.5	334.5
APAC		May-22	Q	458	425	362.5	362.5	362.5	362.5	362.5	362.5	362.5	362.5	362.5	362.5	362.5	362.5
APAC		Jun-22	Q	447	425	NA - Actual	378	415	363	356	403	343	401	365	393	340	360
APAC		Jul-22	Q	413	425	NA - Target	375	375	375	375	375	375	375	375	375	375	375
APAC		Aug-22	Q	458	425												
APAC		Sep-22	Q	385	425												
APAC		Oct-22	Q	421	425												
APAC		Nov-22	Q	448	425												

### First

Het voorvoegsel `First` voor een `LOAD`- of `SELECT`-opdracht (SQL) wordt gebruikt voor het laden van een vast maximaal aantal records uit een gegevensbrontabel. Een typisch gebruiksscenario voor het gebruik van het voorvoegsel `First` is wanneer u een kleine subset records wilt ophalen uit een grote en/or langzame stap voor het laden van gegevens. Zodra het gedefinieerde 'n' aantal records is geladen, wordt de laadstap voortijdig beëindigd en gaat de rest van de uitvoering van het script gewoon door.

### Syntaxis:

```
First n ( loadstatement | selectstatement )
```

## 2 Scriptinstructies en sleutelwoorden

---

### Argumenten

Argument	Beschrijving
n	Een willekeurige uitdrukking die resulteert in een geheel getal waarmee het maximale aantal te lezen records wordt aangegeven. n kan ook worden omsloten door haken: (n).
loadstatement   selectstatement	Het load statement/select statement dat volgt op het n-argument definieert de gespecificeerde tabel die moet worden geladen met het ingestelde maximale aantal records.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>FIRST 10 LOAD * from abc.csv;</code>	Dit voorbeeld haalt de eerste tien regels op uit een Excel-bestand.
<code>FIRST (1) SQL SELECT * from Orders;</code>	In dit voorbeeld wordt de eerste geselecteerde regel opgehaald uit de gegevensset <code>Orders</code> .

### Voorbeeld - De eerste vijf rijen laden

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums uit de eerste twee weken van 2020.
- De variabele `First` die de toepassing de opdracht geeft om alleen de eerste vijf records te laden.



### Load-script

```
Sales:
FIRST 5
LOAD
*
Inline [
date,sales
01/01/2020,6000
01/02/2020,3000
01/03/2020,6000
01/04/2020,8000
01/05/2020,5000
01/06/2020,7000
01/07/2020,3000
01/08/2020,5000
01/09/2020,9000
01/10/2020,5000
01/11/2020,7000
01/12/2020,7000
01/13/2020,7000
01/14/2020,7000
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg date toe als een veld en sum (sales) als een meting.

Resultatentabel

Date	sum(sales)
01/01/2020	6000
01/02/2020	3000
01/03/2020	6000
01/04/2020	8000
01/05/2020	5000

Het script laadt alleen de eerste vijf records van de tabel sales.


### Generic

Het laadvoorvoegsel **Generic** maakt conversie mogelijk van entiteit-attriboot-waarde gemodelleerde gegevens (EAV) naar een traditionele, genormaliseerde relationele tabelstructuur. EAV-modellering wordt ook wel "generieke gegevensmodellering" of "open schema" genoemd.

## 2 Scriptinstructies en sleutelwoorden

Voorbeeld van EAV-gemodelleerde gegevens en een equivalente gedenormaliseerde relationele tabel


Product ID	Attribute	Value
13	Status	Discontinued
13	Colour	Brown
20	Colour	White
13	Size	13-15
20	Size	16-18



Product ID	Status	Colour	Size
13	Discontinued	Brown	13-15
20		White	16-18

Voorbeeld van EAV-gemodelleerde gegevens en een equivalente set van genormaliseerde relationele tabellen

Product ID	Attribute	Value
13	Status	Discontinued
13	Colour	Brown
20	Colour	White
13	Size	13-15
20	Size	16-18



Product ID	Status
13	Discontinued

Product ID	Colour
13	Brown
20	White

Product ID	Size
13	13-15
20	16-18

Hoewel het technisch mogelijk is om EAV-gemodelleerde gegevens in Qlik te laden en te analyseren, is het vaak gemakkelijker om met een equivalente traditionele relationele gegevensstructuur te werken.

### Syntaxis:

```
Generic( loadstatement | selectstatement )
```

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

#### Gerelateerde onderwerpen

Onderwerp	Beschrijving
<i>Crosstable</i> (page 45)	Het laadvoorvoegsel <code>Crosstable</code> transformeert gegevens die horizontaal zijn georiënteerd in verticaal georiënteerde gegevens. Vanuit een puur functioneel perspectief voert het de tegenovergestelde transformatie uit naar het laadvoorvoegsel <code>generic</code> , hoewel de voorvoegsels doorgaans geheel andere gebruiksscenario's dienen.
<b>Generieke databases</b> in <i>Gegevens beheren</i>	EAV-gestructureerde gegevensmodellen worden hier verder beschreven.

### Voorbeeld 1 - Gestructureerde EAV-gegevens transformeren met het algemene laadvoorvoegsel

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat een gegevensset die wordt geladen in een tabel met de naam `Transactions`. De gegevensset bevat een datumveld. De standaard `MonthNames`-definitie wordt gebruikt.

#### Load-script

```
Products:
Generic
Load * inline [
Product ID, Attribute, Value
13, Status, Discontinued
13, Color, Brown
20, Color, White
13, Size, 13-15
20, Size, 16-18
2, Status, Discontinued
5, Color, Brown
2, Color, White
44, Color, Brown
45, Size, 16-18
45, Color, Brown
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `color`.

Voeg deze meting toe:

```
=Count([Product ID])
```

Nu kunt u het aantal producten op kleur inspecteren.

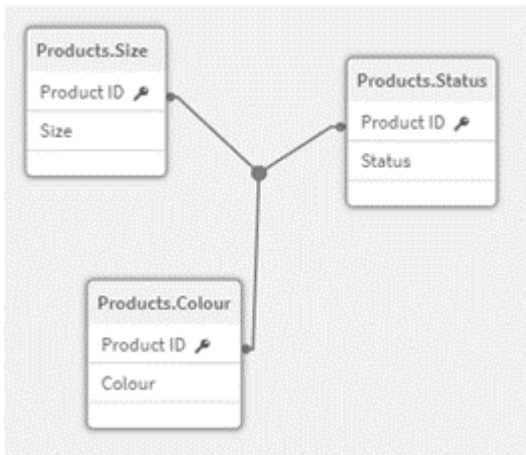
Resultatentabel

Kleur	=Count([Product ID])
Bruin	4
Wit	2

## 2 Scriptinstructies en sleutelwoorden

Let op de vorm van het gegevensmodel, waarbij elk attribuut is uitgesplitst in een aparte tabel die wordt genoemd volgens de oorspronkelijke tag van de doeltabel `Product`. Elke tabel heeft het attribuut als achtervoegsel. Een voorbeeld hiervan is `Product.colour`. De resulterende uitvoerrecords `Productattribuut` worden gekoppeld door de `Product ID`.

*Weergave van de gegevensmodelviewer van de resultaten*



Resulterende tabel met records: `Products.Status`

Product-id	Status
13	Stopgezet
2	Stopgezet

Resulterende tabel met records: `Products.Size`

Product-id	Grootte
13	13-15
20	16-18
45	16-18

Resulterende tabel met records: `Products.Color`

Product-id	Kleur
13	Bruin
5	Bruin
44	Bruin
45	Bruin

Product-id	Kleur
20	Wit
2	Wit

### Voorbeeld 2 - Gestructureerde EAV-gegevens analyseren zonder het algemene laadvoorvoegsel

Load-script en diagramuitdrukking

#### Overzicht

Dit voorbeeld laat zien hoe u gestructureerde EAV-gegevens in hun oorspronkelijke vorm kunt analyseren.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat een gegevensset die wordt geladen in een tabel met de naam `Products` in een EAV-structuur.

In dit voorbeeld tellen we nog steeds producten op kleurattribuut. Om op deze manier gestructureerde gegevens te analyseren, moet u filtering op uitdrukkningsniveau toepassen op producten met de attribuutwaarde `color`.

Bovendien zijn individuele attributen niet beschikbaar om te selecteren als dimensies of velden, waardoor het moeilijker wordt om te bepalen hoe effectieve visualisaties kunnen worden gemaakt.

#### Load-script

```
Products:
Load * Inline
[
Product ID, Attribute, value
13, Status, Discontinued
13, Color, Brown
20, Color, white
13, Size, 13-15
20, Size, 16-18
2, Status, Discontinued
5, Color, Brown
2, Color, white
44, Color, Brown
45, Size, 16-18
45, Color, Brown
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `value`.

Maak de volgende meting:

## 2 Scriptinstructies en sleutelwoorden

---

```
=Count({<Attribute={'Color'}>} [Product ID])
```

Nu kunt u het aantal producten op kleur inspecteren.

Resulterende tabel met records: Products.Status

Waarde	=Count({<Attribute={'Color'}>} [Product ID])
Bruin	4
Wit	2

### Voorbeeld 3 - Denormaliseren van de resulterende uitvoertabellen van een generieke lading (geavanceerd)

Load-script en diagramuitdrukking

#### Overzicht

In dit voorbeeld laten we zien hoe de genormaliseerde gegevensstructuur die wordt geproduceerd door het laadvoorvoegsel `Generic`, weer kan worden gededenormaliseerd naar een geconsolideerde `Product`-dimensietabel. Dit is een geavanceerde modelleringstechniek die kan worden gebruikt als onderdeel van het afstemmen van de prestaties van gegevensmodellen.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

#### Load-script

Products:

```
Generic
Load * inline [
Product ID, Attribute, Value
13, Status, Discontinued
13, Color, Brown
20, Color, White
13, Size, 13-15
20, Size, 16-18
2, Status, Discontinued
5, Color, Brown
2, Color, White
44, Color, Brown
45, Size, 16-18
45, Color, Brown
];
```

```
RENAME TABLE Products.Color TO Products;
```

```
OUTER JOIN (Products)
LOAD * RESIDENT Products.Size;
```

```
OUTER JOIN (Products)
```

## 2 Scriptinstructies en sleutelwoorden

---

```
LOAD * RESIDENT Products.Status;  
DROP TABLES Products.Size,Products.Status;
```

### Resultaten

Open de gegevensmodelviewer en let op de vorm van het resulterende gegevensmodel. Er is slechts één gedenormaliseerde tabel aanwezig. Het is een combinatie van de drie tussenliggende uitvoertabellen:

Products.Size, Products.Status en Products.Color.

Resulterend  
intern  
gegevensmodel

Products
Product-id
Status
Kleur
Grootte

Resulterende tabel met records: Products

Product-id	Status	Kleur	Grootte
13	Stopgezet	Bruin	13-15
20	-	Wit	16-18
2	Stopgezet	Wit	-
5	-	Bruin	-
44	-	Bruin	-
45	-	Bruin	16-18

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: color.

Voeg deze meting toe:

```
=Count([Product ID])
```

Resultatentabel

Kleur	=Count([Product ID])
Bruin	4
Wit	2

### Hierarchy

Het prefix **hierarchy** wordt gebruikt om een hiërarchietabel met bovenliggende-onderliggende objecten om te zetten in een tabel die bruikbaar is in een Qlik Sense-gegevensmodel. Het prefix kan voor een **LOAD**- of **SELECT**-opdracht worden geplaatst en gebruikt het resultaat van de laadopdracht als invoer voor een tabeltransformatie.

De prefix maakt een uitgebreide knooppuntentabel, die normaliter hetzelfde aantal records bevat als de invoertabel, maar daarnaast wordt elk niveau in de hiërarchie ook nog een opgeslagen in een apart veld. Het padveld kan worden gebruikt in een boomstructuur.

#### Syntaxis:

```
Hierarchy (NodeID, ParentID, NodeName, [ParentName, [PathSource, [PathName, [PathDelimiter, Depth]]]]) (loadstatement | selectstatement)
```

De invoertabel moet een aangrenzende knooppuntentabel zijn. Aangrenzende knooppuntentabellen zijn tabellen waarin elk record correspondeert met een knooppunt en een veld heeft dat de referentie naar het bovenliggende knooppunt bevat. In een dergelijke tabel wordt het knooppunt opgeslagen in slechts één record, maar kan het knooppunt wel meerdere onderliggende knooppunten hebben. De tabel kan natuurlijk aanvullende velden bevatten waarin kenmerken van de knooppunten worden beschreven.

De prefix maakt een uitgebreide knooppuntentabel, die normaliter hetzelfde aantal records bevat als de invoertabel, maar daarnaast wordt elk niveau in de hiërarchie ook nog een opgeslagen in een apart veld. Het padveld kan worden gebruikt in een boomstructuur.

Gewoonlijk heeft de interne tabel precies één record per knooppunt. In dat geval heeft de uitvoertabel hetzelfde aantal records. Soms zijn er echter knooppunten met meerdere bovenliggende knooppunten. Een knooppunt wordt dan gerepresenteerd door meerdere records in de invoertabel. In dat geval kan de uitvoertabel meer records hebben dan de invoertabel.

Alle knooppunten met een bovenliggende ID die niet is gevonden in de kolom node id (met inbegrip van knooppunten met ontbrekende bovenliggende ID) beschouwd worden als roots. Er worden dus alleen knooppunten geladen met een directe of indirecte verbinding met een root-knooppunt. Op deze wijze worden cirkelreferenties voorkomen.

U kunt aanvullende velden met de naam van het bovenliggende knooppunt en het pad of de diepte van het knooppunt maken.

#### Argumenten:

##### Argumenten

Argument	Beschrijving
NodeID	De naam van het veld dat de knooppunt-ID bevat. Dit veld moet bestaan in de invoertabel.



## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
ParentID	De naam van het veld dat de knooppunt-ID van het bovenliggende knooppunt bevat. Dit veld moet bestaan in de invoertabel.
NodeName	De naam van het veld dat de naam van het knooppunt bevat. Dit veld moet bestaan in de invoertabel.
ParentName	Een tekenreeks die wordt gebruikt als naam voor een nieuw veld <b>ParentName</b> . Als deze parameter wordt weggelaten, wordt dit veld niet gemaakt.
ParentSource	De naam van het veld dat de naam van het knooppunt bevat dat wordt gebruikt om het knooppuntpad te bouwen. Optionele parameter. Als deze parameter wordt weggelaten, wordt <b>NodeName</b> gebruikt.
PathName	Een tekenreeks die wordt gebruikt als naam voor het nieuwe veld <b>Path</b> . Hierin staat het pad van de root naar het knooppunt. Optionele parameter. Als deze parameter wordt weggelaten, wordt dit veld niet gemaakt.
PathDelimiter	Een tekenreeks die wordt gebruikt als scheidingsteken in het nieuwe veld <b>Path</b> . Optionele parameter. Als deze parameter wordt weggelaten, wordt '/' gebruikt.
Depth	Een tekenreeks die wordt gebruikt als naam voor het nieuwe veld <b>Depth</b> . Hiermee wordt aangegeven wat de diepte van het knooppunt in de hiërarchie is. Optionele parameter. Als deze parameter wordt weggelaten, wordt dit veld niet gemaakt.

### Voorbeeld:

```
Hierarchy(NodeID, ParentID, NodeName, ParentName, NodeName, PathName, '\', Depth) LOAD *
inline [
NodeID, ParentID, NodeName
1, 4, London
2, 3, Munich
3, 5, Germany
4, 5, UK
5, , Europe
];
```

Node ID	ParentID	NodeName	NodeName1	NodeName2	NodeName3	ParentName	PathName	Depth
1	4	London	Europe	UK	London	UK	Europe\UK\London	3
2	3	Munich	Europe	Germany	Munich	Germany	Europe\Germany\Munich	3
3	5	Germany	Europe	Germany	-	Europe	Europe\Germany	2
4	5	UK	Europe	UK	-	Europe	Europe\UK	2
5		Europe	Europe	-	-	-	Europe	1

### HierarchyBelongsTo

Dit prefix wordt gebruikt om een hiërarchietabel met bovenliggende-onderliggende objecten om te zetten in een tabel die bruikbaar is in een Qlik Sense-gegevensmodel. Het prefix kan voor een **LOAD**- of **SELECT**-opdracht worden geplaatst en gebruikt het resultaat van de laadopdracht als invoer voor een tabeltransformatie.

Met het prefix maakt u een tabel waarin alle relaties tussen voorgangers en onderliggende elementen van de hiërarchie staan. Met de voorgangervelden kunt u vervolgens hele bomen in de hiërarchie selecteren. De uitvoertabel bevat in de meeste gevallen meerdere records per knooppunt.

#### Syntaxis:

```
HierarchyBelongsTo (NodeID, ParentID, NodeName, AncestorID, AncestorName, [DepthDiff]) (loadstatement | selectstatement)
```

De invoertabel moet een aangrenzende knooppuntentabel zijn. Aangrenzende knooppuntentabellen zijn tabellen waarin elk record correspondeert met een knooppunt en een veld heeft dat de referentie naar het bovenliggende knooppunt bevat. In een dergelijke tabel wordt het knooppunt opgeslagen in slechts één record, maar kan het knooppunt wel meerdere onderliggende knooppunten hebben. De tabel kan natuurlijk aanvullende velden bevatten waarin kenmerken van de knooppunten worden beschreven.

Met het prefix maakt u een tabel waarin alle relaties tussen voorgangers en onderliggende elementen van de hiërarchie staan. Met de voorgangervelden kunt u vervolgens hele bomen in de hiërarchie selecteren. De uitvoertabel bevat in de meeste gevallen meerdere records per knooppunt.

U kunt een extra veld maken waarin het diepteververschil van de knooppunten wordt aangegeven.

#### Argumenten:

##### Argumenten

Argument	Beschrijving
NodeID	De naam van het veld dat de knooppunt-ID bevat. Dit veld moet bestaan in de invoertabel.
ParentID	De naam van het veld dat de knooppunt-ID van het bovenliggende knooppunt bevat. Dit veld moet bestaan in de invoertabel.
NodeName	De naam van het veld dat de naam van het knooppunt bevat. Dit veld moet bestaan in de invoertabel.
AncestorID	Een tekenreeks die wordt gebruikt voor het geven van een naam aan het nieuwe veld ancestor-ID. Hierin staat de ID van het voorgangerknooppunt.
AncestorName	Een tekenreeks die wordt gebruikt voor het geven van een naam aan het nieuwe veld ancestor-ID. Hierin staat de naam van het voorgangerknooppunt.

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
DepthDiff	Een tekenreeks die wordt gebruikt als naam voor het nieuwe veld <b>DepthDiff</b> . Hiermee wordt aangegeven wat de diepte van het knooppunt in de hiërarchie in relatie tot het voorgangerknooppunt. Optionele parameter. Als deze parameter wordt weggelaten, wordt dit veld niet gemaakt.

### Voorbeeld:

```
HierarchyBelongsTo (NodeID, AncestorID, NodeName, AncestorID, AncestorName, DepthDiff) LOAD *
inline [
NodeID, AncestorID, NodeName
1, 4, London
2, 3, Munich
3, 5, Germany
4, 5, UK
5, , Europe
];
```

### Results

NodeID	AncestorID	NodeName	AncestorName	DepthDiff
1	1	London	London	0
1	4	London	UK	1
1	5	London	Europe	2
2	2	Munich	Munich	0
2	3	Munich	Germany	1
2	5	Munich	Europe	2
3	3	Germany	Germany	0
3	5	Germany	Europe	1
4	4	UK	UK	0
4	5	UK	Europe	1
5	5	Europe	Europe	0

## Inner

De prefixen **join** en **keep** kunnen worden voorafgegaan door het prefix **inner**. Indien gebruikt voor **join**, geeft de opdracht aan dat een inner join moet worden gebruikt. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in beide tabellen voorkomen. Indien gebruikt voor **keep**, geeft de opdracht aan dat beide tabellen met onbewerkte gegevens worden gereduceerd tot de gemeenschappelijke doorsnede voordat ze worden opgeslagen in Qlik Sense.

## 2 Scriptinstructies en sleutelwoorden

### Syntaxis:

```
Inner ( Join | Keep ) [ (tablename) ] (loadstatement | selectstatement )
```

### Argumenten:

Argumenten

Argument	Beschrijving
tablename	De benoemde tabel die moet worden vergeleken met de geladen tabel.
loadstatementof selectstatement	De <b>LOAD</b> of <b>SELECT</b> -opdracht voor de geladen tabel.

### Voorbeeld

#### Load-script

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
Table1: Load * inline [ Column1, Column2 A, B 1, aa 2, cc 3, ee ]; Table2: Inner Join Load *  
inline [ Column1, Column3 A, C 1, xx 4, yy ];
```

### Resultaat

Resultaattabel

Column1	Column2	Column3
A	B	C
1	aa	xx

### Uitleg

Dit voorbeeld toont de uitvoer voor inner join waarbij alleen waarden die aanwezig zijn in de eerste (links) en de tweede (rechts) tabellen zijn samengevoegd.

## IntervalMatch

Het prefix **IntervalMatch** wordt gebruikt voor het maken van een tabel waarin discrete numerieke waarden worden gekoppeld aan een of meer numerieke intervallen, en waarin de waarden optioneel worden gekoppeld aan een of meer extra sleutels.

### Syntaxis:

```
IntervalMatch (matchfield) (loadstatement | selectstatement )  
IntervalMatch (matchfield, keyfield1 [ , keyfield2, ... keyfield5 ] )  
(loadstatement | selectstatement )
```

## 2 Scriptinstructies en sleutelwoorden

---

Het prefix **IntervalMatch** moet voor een **LOAD**- of **SELECT**-opdracht worden geplaatst waarmee de intervallen worden geladen. Het veld met de discrete gegevenspunten (Tijd in het voorbeeld verderop) en extra sleutels moet al in Qlik Sense geladen zijn vóór de opdracht met het prefix **IntervalMatch**. Het prefix leest dit veld niet zelf uit de databasetabel. Het prefix zet de geladen tabel met intervallen en sleutels om in een tabel die een extra kolom bevat: de discrete numerieke gegevenspunten. Tevens wordt het aantal records zodanig uitgebreid dat de nieuwe tabel één record per mogelijke combinatie van discreet gegevenspunt, interval en waarde van het sleutelveld of de sleutelvelden bevat.

De intervallen kunnen elkaar overlappen en de discrete waarden worden aan alle relevante intervallen gekoppeld.

Als het prefix **IntervalMatch** wordt uitgebreid met sleutelvelden, wordt het gebruikt voor het maken van een tabelkoppeling van discrete numerieke waarden met een of meer numerieke intervallen, terwijl tegelijkertijd de waarden van een of meer extra sleutels worden gekoppeld.

Om te voorkomen dat ongedefinieerde intervallimieten worden genegeerd, moet u mogelijk **NULL**-waarden toewijzen aan andere velden met de onderste en bovenste limieten voor het interval. Dit kan worden afgehandeld met de opdracht **NullAsValue** of door een expliciete test waarbij **NULL**-waarden worden vervangen door een numerieke waarde ruim voor of na een van de discrete numerieke gegevenspunten.

### Argumenten:

Argumenten

Argument	Beschrijving
matchfield	Het veld met de discrete numerieke waarden voor koppeling aan intervallen.
keyfield	Velden met de extra kenmerken die bij de transformatie moeten worden gekoppeld.
loadstatement orselectstatement	Moet resulteren in een tabel waarvan het eerste veld de laagste limiet van elk interval bevat en het twee veld de hoogste limiet. Als sleutelkoppeling wordt gebruikt, moeten het derde veld en de volgende velden de sleutelvelden bevatten van de opdracht <b>IntervalMatch</b> . De intervallen zijn altijd gesloten, dat wil zeggen dat de eindpunten in het interval zijn opgenomen. Niet-numerieke limieten zorgen ervoor dat het interval buiten beschouwing wordt geladen (ongedefinieerd).

### Example 1:

In de twee onderstaande tabellen wordt in de eerste een aantal discrete gebeurtenissen weergegeven en worden in de tweede de begin- en eindtijden voor de productie van verschillende bestellingen gedefinieerd. Met behulp van het prefix **IntervalMatch** kan een logische verbinding tot stand worden gebracht tussen de twee tabellen om bijvoorbeeld uit te zoeken welke bestellingen last hadden van storingen en welke bestellingen door welke ploegen zijn verwerkt.

EventLog:

```
LOAD * Inline [  
Time, Event, Comment  
00:00, 0, Start of shift 1  
01:18, 1, Line stop
```

## 2 Scriptinstructies en sleutelwoorden

---

```
02:23, 2, Line restart 50%
04:15, 3, Line speed 100%
08:00, 4, Start of shift 2
11:43, 5, End of production
];
```

```
OrderLog:
LOAD * INLINE [
Start, End, Order
01:00, 03:35, A
02:30, 07:58, B
03:04, 10:27, C
07:23, 11:43, D
];
```

```
//Link the field Time to the time intervals defined by the fields Start and End.
Inner Join IntervalMatch ( Time )
LOAD Start, End
Resident OrderLog;
```

De tabel **OrderLog** bevat nu een extra kolom: *Time*. Het aantal records is eveneens uitgebreid.

Table with additional column

Time	Start	End	Order
00:00	-	-	-
01:18	01:00	03:35	A
02:23	01:00	03:35	A
04:15	02:30	07:58	B
04:15	03:04	10:27	C
08:00	03:04	10:27	C
08:00	07:23	11:43	D
11:43	07:23	11:43	D

### Example 2: (met keyfield)

Hetzelfde voorbeeld als boven, waarbij *ProductionLine* is toegevoegd als sleutelveld.

```
EventLog:
LOAD * Inline [
Time, Event, Comment, ProductionLine
00:00, 0, Start of shift 1, P1
01:00, 0, Start of shift 1, P2
01:18, 1, Line stop, P1
02:23, 2, Line restart 50%, P1
04:15, 3, Line speed 100%, P1
08:00, 4, Start of shift 2, P1
09:00, 4, Start of shift 2, P2
11:43, 5, End of production, P1
11:43, 5, End of production, P2
```

## 2 Scriptinstructies en sleutelwoorden

---

```
];
```

```
OrderLog:
```

```
LOAD * INLINE [  
Start, End, Order, ProductionLine  
01:00, 03:35, A, P1  
02:30, 07:58, B, P1  
03:04, 10:27, C, P1  
07:23, 11:43, D, P2  
];
```

```
//Link the field Time to the time intervals defined by the fields Start and End and match the values
```

```
// to the key ProductionLine.
```

```
Inner Join
```

```
IntervalMatch ( Time, ProductionLine )
```

```
LOAD Start, End, ProductionLine
```

```
Resident OrderLog;
```

Het volgende tabelvak kan nu worden gemaakt:

Tablebox example

ProductionLine	Time	Event	Comment	Order	Start	End
P1	00:00	0	Start of shift 1	-	-	-
P2	01:00	0	Start of shift 1	-	-	-
P1	01:18	1	Line stop	A	01:00	03:35
P1	02:23	2	Line restart 50%	A	01:00	03:35
P1	04:15	3	Line speed 100%	B	02:30	07:58
P1	04:15	3	Line speed 100%	C	03:04	10:27
P1	08:00	4	Start of shift 2	C	03:04	10:27
P2	09:00	4	Start of shift 2	D	07:23	11:43
P1	11:43	5	End of production	-	-	-
P2	11:43	5	End of production	D	07:23	11:43

### Join

Het prefix **join** koppelt de geladen tabel aan een bestaande benoemde tabel of de laatste eerder gemaakte gegevenstabel.

Het effect van het samenvoegen van gegevens is dat de doeltabel wordt uitgebreid met een extra set velden of attributen, namelijk die welke nog niet in de doeltabel aanwezig zijn. Alle gemeenschappelijke veldnamen tussen de brongegevensset en de doeltabel worden gebruikt om uit te zoeken hoe de nieuwe binnenkomende records moeten worden gekoppeld. Dit wordt gewoonlijk een 'natuurlijke join' genoemd.

---

## 2 Scriptinstructies en sleutelwoorden

---

Een Qlik-samenvoegbewerking kan ertoe leiden dat de resulterende doeltabel meer of minder records heeft dan waarmee hij begon, afhankelijk van de uniekheid van de join-koppeling en het type join dat wordt gebruikt.

Er zijn vier soorten joins:

### Left join

Left joins zijn het meest voorkomende join-type. Als u bijvoorbeeld een transactiegegevensset hebt en deze wilt combineren met een referentiegegevensset, gebruikt u doorgaans een `Left Join`. U zou eerst de transactietabel laden en vervolgens de referentiegegevensset laden terwijl u deze via een `Left Join`-voorvoegsel aan de reeds geladen transactietabel toevoegt. Een `Left Join` zou alle transacties ongewijzigd laten en de aanvullende referentiegegevensvelden toevoegen waar een overeenkomst wordt gevonden.

### Inner join

Als u twee datasets hebt waarbij u alleen om resultaten geeft als er een overeenkomende koppeling is, overweeg dan om een `Inner Join` te gebruiken. Hierdoor worden alle records verwijderd uit zowel de geladen brongegevens als de doeltabel als er geen overeenkomst wordt gevonden. Als gevolg hiervan kan uw doeltabel minder records bevatten dan voordat de samenvoegbewerking plaatsvond.

### Outer join

Als u zowel de doelrecords als alle inkomende records moet bewaren, gebruikt u een `Outer Join`. Als er geen overeenkomst wordt gevonden, wordt elke set records nog steeds bewaard, terwijl de velden aan de andere kant van de join niet ingevuld blijven (null). Outer joins hebben over het algemeen weinig praktische toepassingen.

### Right join

Dit type join houdt alle records die op het punt staan te worden geladen, terwijl de records in de tabel waarop de join is gericht, worden beperkt tot alleen die records waarbij er een koppelingsovereenkomst is in de inkomende records. Dit is een niche-jointype dat soms wordt gebruikt om een reeds vooraf geladen tabel met records in te korten tot een vereiste subset.

Als het sleutelwoord type wordt weggelaten, is `Inner Join` het standaard join-type.



## 2 Scriptinstructies en sleutelwoorden

Voorbeelden van resultaatsets van verschillende typen samenvoegbewerkingen

DATASETS	OPERATION	OUTPUT																		
<p>Target Table</p> <table border="1"> <thead> <tr> <th>Trade ID</th> <th>Asset Class</th> </tr> </thead> <tbody> <tr> <td>101533</td> <td>Fixed Income</td> </tr> <tr> <td>606601</td> <td>Commodities</td> </tr> </tbody> </table>	Trade ID	Asset Class	101533	Fixed Income	606601	Commodities	<p>LEFT JOIN</p> <p>➔</p>	<table border="1"> <thead> <tr> <th>Trade ID</th> <th>Asset Class</th> <th></th> </tr> </thead> <tbody> <tr> <td>101533</td> <td>Fixed Income</td> <td>LSE</td> </tr> <tr> <td>606601</td> <td>Commodities</td> <td></td> </tr> </tbody> </table>	Trade ID	Asset Class		101533	Fixed Income	LSE	606601	Commodities				
Trade ID	Asset Class																			
101533	Fixed Income																			
606601	Commodities																			
Trade ID	Asset Class																			
101533	Fixed Income	LSE																		
606601	Commodities																			
	<p>INNER JOIN</p> <p>➔</p>	<table border="1"> <thead> <tr> <th>Trade ID</th> <th>Asset Class</th> <th></th> </tr> </thead> <tbody> <tr> <td>101533</td> <td>Fixed Income</td> <td>LSE</td> </tr> </tbody> </table>	Trade ID	Asset Class		101533	Fixed Income	LSE												
Trade ID	Asset Class																			
101533	Fixed Income	LSE																		
<p>Incoming Dataset</p> <table border="1"> <thead> <tr> <th>Trade ID</th> <th>Exchange</th> </tr> </thead> <tbody> <tr> <td>101533</td> <td>LSE</td> </tr> <tr> <td>79052</td> <td>Hong Kong</td> </tr> </tbody> </table>	Trade ID	Exchange	101533	LSE	79052	Hong Kong	<p>OUTER JOIN</p> <p>➔</p>	<table border="1"> <thead> <tr> <th>Trade ID</th> <th>Asset Class</th> <th></th> </tr> </thead> <tbody> <tr> <td>101533</td> <td>Fixed Income</td> <td>LSE</td> </tr> <tr> <td>606601</td> <td>Commodities</td> <td></td> </tr> <tr> <td>79052</td> <td></td> <td>Hong Kong</td> </tr> </tbody> </table>	Trade ID	Asset Class		101533	Fixed Income	LSE	606601	Commodities		79052		Hong Kong
Trade ID	Exchange																			
101533	LSE																			
79052	Hong Kong																			
Trade ID	Asset Class																			
101533	Fixed Income	LSE																		
606601	Commodities																			
79052		Hong Kong																		
	<p>RIGHT JOIN</p> <p>➔</p>	<table border="1"> <thead> <tr> <th>Trade ID</th> <th>Asset Class</th> <th></th> </tr> </thead> <tbody> <tr> <td>101533</td> <td>Fixed Income</td> <td>LSE</td> </tr> <tr> <td>79052</td> <td></td> <td>Hong Kong</td> </tr> </tbody> </table>	Trade ID	Asset Class		101533	Fixed Income	LSE	79052		Hong Kong									
Trade ID	Asset Class																			
101533	Fixed Income	LSE																		
79052		Hong Kong																		



Als er geen gemeenschappelijke veldnamen zijn tussen de bron en het doel van een samenvoegbewerking, resulteert de join in een cartesiaans product van alle rijen - dit wordt een 'cross join' genoemd.

Voorbeeld resultaatset van een 'cross join'-bewerking

DATASETS	OPERATION	OUTPUT																																		
<p>Target Table</p> <table border="1"> <thead> <tr> <th>Trade ID</th> <th>Base Currency</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>101533</td> <td>EUR</td> <td>1250</td> </tr> <tr> <td>606601</td> <td>EUR</td> <td>1650</td> </tr> </tbody> </table>	Trade ID	Base Currency	Amount	101533	EUR	1250	606601	EUR	1650	<p>JOIN (any type)</p> <p>➔</p>	<table border="1"> <thead> <tr> <th>Trade ID</th> <th>Base Currency</th> <th>Amount</th> <th>Target Currency</th> <th>Rate</th> </tr> </thead> <tbody> <tr> <td>101533</td> <td>EUR</td> <td>1250</td> <td>USD</td> <td>1.08</td> </tr> <tr> <td>101533</td> <td>EUR</td> <td>1250</td> <td>GBP</td> <td>0.84</td> </tr> <tr> <td>606601</td> <td>EUR</td> <td>1650</td> <td>USD</td> <td>1.08</td> </tr> <tr> <td>606601</td> <td>EUR</td> <td>1650</td> <td>GBP</td> <td>0.84</td> </tr> </tbody> </table>	Trade ID	Base Currency	Amount	Target Currency	Rate	101533	EUR	1250	USD	1.08	101533	EUR	1250	GBP	0.84	606601	EUR	1650	USD	1.08	606601	EUR	1650	GBP	0.84
Trade ID	Base Currency	Amount																																		
101533	EUR	1250																																		
606601	EUR	1650																																		
Trade ID	Base Currency	Amount	Target Currency	Rate																																
101533	EUR	1250	USD	1.08																																
101533	EUR	1250	GBP	0.84																																
606601	EUR	1650	USD	1.08																																
606601	EUR	1650	GBP	0.84																																
<p>Incoming Dataset</p> <table border="1"> <thead> <tr> <th>Target Currency</th> <th>Rate</th> </tr> </thead> <tbody> <tr> <td>USD</td> <td>1.08</td> </tr> <tr> <td>GBP</td> <td>0.84</td> </tr> </tbody> </table>	Target Currency	Rate	USD	1.08	GBP	0.84																														
Target Currency	Rate																																			
USD	1.08																																			
GBP	0.84																																			

### Syntaxis:

```
[inner | outer | left | right ]Join [ (tablename ) ] ( loadstatement | selectstatement )
```

## 2 Scriptinstructies en sleutelwoorden

### Argumenten

Argument	Beschrijving
tablename	De benoemde tabel die moet worden vergeleken met de geladen tabel.
loadstatementof selectstatement	De <b>LOAD</b> of <b>SELECT</b> -opdracht voor de geladen tabel.

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

### Gerelateerde onderwerpen

Onderwerp	Beschrijving
<b>Tabellen samenvoegen met Join en Keep</b> in <i>Gegevens beheren</i>	In dit onderwerp vindt u meer uitleg over de concepten 'samenvoegen' en 'bewaren' van gegevenssets.
<i>Keep (page 81)</i>	Het laadvoorvoegsel <code>keep</code> is vergelijkbaar met het voorvoegsel <code>join</code> , maar combineert de bron- en doelgegevenssets niet. In plaats daarvan trimt het elke gegevensset op basis van het type bewerking dat is aangenomen (binnen, buiten, links of rechts).

### Voorbeeld 1 - Left join: Een doeltabel verrijken met een referentiegegevensset

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wijzigingsrecords vertegenwoordigt, die in een tabel met de naam `changes` wordt geladen. Bevat een sleutelveld `Status-id`.
- Een tweede gegevensset die wijzigingsstatussen vertegenwoordigt, die wordt geladen en gecombineerd met de originele wijzigingsrecords door deze te combineren met een linker `join`-laadvoorvoegsel.

Deze left join zorgt ervoor dat de wijzigingsrecords intact blijven terwijl er statusattributen worden toegevoegd waar een match in de inkomende statusrecords wordt gevonden op basis van een gemeenschappelijk `Status-id`.

#### Load-script

Changes:

```
Load * inline [
```

## 2 Scriptinstructies en sleutelwoorden

```
Change ID      Status ID      Scheduled Start Date      Scheduled End Date      Business Impact
10030 4        19/01/2022      23/02/2022      None
10015 3        04/01/2022      15/02/2022      Low
10103 1        02/04/2022      29/05/2022      Medium
10185 2        23/06/2022      08/09/2022      None
10323 1        08/11/2022      26/11/2022      High
10326 2        11/11/2022      05/12/2022      None
10138 2        07/05/2022      03/08/2022      None
10031 3        20/01/2022      25/03/2022      Low
10040 1        29/01/2022      22/04/2022      None
10134 1        03/05/2022      08/07/2022      Low
10334 2        19/11/2022      06/02/2023      Low
10220 2        28/07/2022      06/09/2022      None
10264 1        10/09/2022      17/10/2022      Medium
10116 1        15/04/2022      24/04/2022      None
10187 2        25/06/2022      24/08/2022      Low
] (delimiter is '\t');
```

Status:

Join (Changes)

Load \* inline [

```
Status ID      Status      Sub Status
```

```
1      Open      Not Started
```

```
2      Open      Started
```

```
3      Closed   Complete
```

```
4      Closed   Cancelled
```

] (delimiter is '\t');

### Resultaten

Open de gegevensmodelviewer en let op de vorm van het gegevensmodel. Er is slechts één gedenormaliseerde tabel aanwezig. Het is een combinatie van alle originele wijzigingsrecords, met de overeenkomende statusattributen samengevoegd met elk wijzigingsrecord.

Resultierend intern  
gegevensmodel

<b>Wijzigingen</b>
Wijzigings-id
Status-id
Geplande begindatum
Geplande einddatum
Bedrijfsimpact
Status
Substatus

Als u het voorbeeldvenster in de gegevensmodelviewer uitvouwt, ziet u een deel van deze volledige resultaten set, georganiseerd in een tabel:

## 2 Scriptinstructies en sleutelwoorden

Voorbeeld van de tabel Wijzigingen in de gegevensmodelviewer

Wijzigings-id	Status-id	Geplande begindatum	Geplande einddatum	Bedrijfsimpact	Status	Substatus
10015	3	04/01/2022	15/02/2022	Laag	Gesloten	Voltooid
10030	4	19/01/2022	23/02/2022	Geen	Gesloten	Geannuleerd
10031	3	20/01/2022	25/03/2022	Laag	Gesloten	Voltooid
10040	1	29/01/2022	22/04/2022	Geen	Openen	Niet gestart
10103	1	02/04/2022	29/05/2022	Gemiddeld	Openen	Niet gestart
10116	1	15/04/2022	24/04/2022	Geen	Openen	Niet gestart
10134	1	03/05/2022	08/07/2022	Laag	Openen	Niet gestart
10138	2	07/05/2022	03/08/2022	Geen	Openen	Gestart
10185	2	23/06/2022	08/09/2022	Geen	Openen	Gestart
10187	2	25/06/2022	24/08/2022	Laag	Openen	Gestart
10220	2	28/07/2022	06/09/2022	Geen	Openen	Gestart
10264	1	10/09/2022	17/10/2022	Gemiddeld	Openen	Niet gestart
10323	1	08/11/2022	26/11/2022	Hoog	Openen	Niet gestart
10326	2	11/11/2022	05/12/2022	Geen	Openen	Gestart
10334	2	19/11/2022	06/02/2023	Laag	Openen	Gestart

Ga terug naar de editor voor laden van gegevens. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: status.

Voeg deze meting toe:

=Count([Change ID])

Nu kunt u het aantal wijzigingen per status bekijken.

Resultatentabel

Status	=Count([Change ID])
Openen	12
Gesloten	3

### Voorbeeld 2 - Inner join: Alleen overeenkomende records combineren

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

## 2 Scriptinstructies en sleutelwoorden

---

Het load-script bevat:

- Een gegevensset die wijzigingsrecords vertegenwoordigt, die in een tabel met de naam changes wordt geladen.
- Een tweede gegevensset die wijzigingsrecords uit het bronsysteem JIRA weergeeft. Deze wordt geladen en gecombineerd met de oorspronkelijke records door deze samen te voegen met een Inner Join-laadvoorvoegsel.

Dit Inner Join zorgt ervoor dat alleen de vijf wijzigingsrecords die in beide gegevenssets voorkomen, worden bewaard.

### Load-script

Changes:

```
Load * inline [  
Change ID      Status ID      Scheduled Start Date      Scheduled End Date      Business Impact  
10030 4      19/01/2022      23/02/2022      None  
10015 3      04/01/2022      15/02/2022      Low  
10103 1      02/04/2022      29/05/2022      Medium  
10185 2      23/06/2022      08/09/2022      None  
10323 1      08/11/2022      26/11/2022      High  
10326 2      11/11/2022      05/12/2022      None  
10138 2      07/05/2022      03/08/2022      None  
10031 3      20/01/2022      25/03/2022      Low  
10040 1      29/01/2022      22/04/2022      None  
10134 1      03/05/2022      08/07/2022      Low  
10334 2      19/11/2022      06/02/2023      Low  
10220 2      28/07/2022      06/09/2022      None  
10264 1      10/09/2022      17/10/2022      Medium  
10116 1      15/04/2022      24/04/2022      None  
10187 2      25/06/2022      24/08/2022      Low  
] (delimiter is '\t');
```

JIRA\_changes:

Inner Join (Changes)

Load

```
[Ticket ID] AS [Change ID],  
[Source System]
```

inline

```
[  
Ticket ID      Source System  
10030 JIRA  
10323 JIRA  
10134 JIRA  
10334 JIRA  
10220 JIRA  
10187 JIRA  
] (delimiter is '\t');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Source System
- Change ID
- Business Impact

Nu kunt u de vijf resulterende records inspecteren.

Resultatentabel

Bronstelsysteem	Wijzigings-id	Bedrijfsimpact
JIRA	10030	Geen
JIRA	10134	Laag
JIRA	10220	Geen
JIRA	10323	Hoog
JIRA	10334	Laag

### Voorbeeld 3 - Outer join: Overlappende recordsets combineren

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wijzigingsrecords vertegenwoordigt, die in een tabel met de naam Changes wordt geladen.
- Een tweede gegevensset die wijzigingsrecords uit het bronsysteem JIRA weergeeft, die wordt geladen en gecombineerd met de oorspronkelijke records door deze samen te voegen met een outer Join-laadvoorvoegsel.

Dit zorgt ervoor dat alle overlappende wijzigingsrecords van beide gegevenssets behouden blijven.

#### Load-script

```
// 8 Change records
```

```
Changes:
```

```
Load * inline [
```

```
Change ID      Status ID      Scheduled Start Date      Scheduled End Date      Business Impact
10030 4      19/01/2022      23/02/2022      None
10015 3      04/01/2022      15/02/2022      Low
10138 2      07/05/2022      03/08/2022      None
10031 3      20/01/2022      25/03/2022      Low
10040 1      29/01/2022      22/04/2022      None
10134 1      03/05/2022      08/07/2022      Low
```

## 2 Scriptinstructies en sleutelwoorden

---

```
10334 2      19/11/2022    06/02/2023    Low
10220 2      28/07/2022    06/09/2022    None
] (delimiter is '\t');
```

```
// 6 Change records
```

```
JIRA_changes:
Outer Join (Changes)
Load
  [Ticket ID] AS [Change ID],
  [Source System]
inline
[
Ticket ID      Source System
10030 JIRA
10323 JIRA
10134 JIRA
10334 JIRA
10220 JIRA
10597 JIRA
] (delimiter is '\t');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Source System
- Change ID
- Business Impact

Nu kunt u de 10 resulterende records inspecteren.

Resultatentabel

Bronstelsysteem	Wijzigings-id	Bedrijfsimpact
JIRA	10030	Geen
JIRA	10134	Laag
JIRA	10220	Geen
JIRA	10323	-
JIRA	10334	Laag
JIRA	10597	-
-	10015	Laag
-	10031	Laag
-	10040	Geen
-	10138	Geen

### Voorbeeld 4 - Right join: Een doeltabel inkorten door een secundaire mastergegevensset

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wijzigingsrecords vertegenwoordigt, die in een tabel met de naam Changes wordt geladen.
- Een tweede gegevensset die wijzigingsrecords uit het bronsysteem Teamwork weergeeft. Deze wordt geladen en gecombineerd met de oorspronkelijke records door deze samen te voegen met een Right Join-laadvoorvoegsel.

Dit zorgt ervoor dat alleen Teamwork-wijzigingsrecords worden bewaard, terwijl er geen Teamwork-records verloren gaan als de doeltabel geen overeenkomende Change ID heeft.

#### Load-script

Changes:

```
Load * inline [  
Change ID      Status ID      Scheduled Start Date      Scheduled End Date      Business Impact  
10030 4        19/01/2022      23/02/2022      None  
10015 3        04/01/2022      15/02/2022      Low  
10103 1        02/04/2022      29/05/2022      Medium  
10185 2        23/06/2022      08/09/2022      None  
10323 1        08/11/2022      26/11/2022      High  
10326 2        11/11/2022      05/12/2022      None  
10138 2        07/05/2022      03/08/2022      None  
10031 3        20/01/2022      25/03/2022      Low  
10040 1        29/01/2022      22/04/2022      None  
10134 1        03/05/2022      08/07/2022      Low  
10334 2        19/11/2022      06/02/2023      Low  
10220 2        28/07/2022      06/09/2022      None  
10264 1        10/09/2022      17/10/2022      Medium  
10116 1        15/04/2022      24/04/2022      None  
10187 2        25/06/2022      24/08/2022      Low  
] (delimiter is '\t');
```

Teamwork\_changes:

Right Join (Changes)

Load

[Ticket ID] AS [Change ID],

[Source System]

inline

[

Ticket ID Source System

10040 Teamwork



## 2 Scriptinstructies en sleutelwoorden

---

```
10015 Teamwork
10103 Teamwork
10031 Teamwork
50231 Teamwork
] (delimiter is '\t');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Source System
- Change ID
- Business Impact

Nu kunt u de vijf resulterende records inspecteren.

Resultatentabel

Bronstelsysteem	Wijzigings-id	Bedrijfsimpact
Teamwerk	10015	Laag
Teamwerk	10031	Laag
Teamwerk	10040	Geen
Teamwerk	10103	Gemiddeld
Teamwerk	50231	-

### Keep

Het prefix **keep** is vergelijkbaar met het prefix **join**. Net als bij het prefix **join** wordt de geladen tabel vergeleken met een bestaande benoemde tabel of de laatste eerder gemaakte gegevenstabel. Maar de geladen tabel wordt niet samengevoegd met een bestaande tabel, in plaats daarvan wordt een tabel of worden beide tabellen voordat ze worden opgeslagen in Qlik Sense gereduceerd op basis van de doorsnede van de tabelgegevens. De uitgevoerde vergelijking is equivalent aan een natuurlijke join voor alle gemeenschappelijke velden, net zoals bij een overeenkomstige join. De twee tabellen worden echter niet samengevoegd, maar als twee afzonderlijk benoemde tabellen in Qlik Sense bewaard.

#### Syntaxis:

```
(inner | left | right) keep [(tablename ) ]( loadstatement | selectstatement )
```

Het prefix **keep** moet worden voorafgegaan door een van de volgende prefixen **inner**, **left** of **right**.

Het expliciete prefix **join** in de scripttaal van Qlik Sense zorgt voor een volledige koppeling van de twee tabellen. Het resultaat is één tabel. Dergelijke samenvoegingen resulteren vaak in zeer grote tabellen. Een van de belangrijkste functies in Qlik Sense is de mogelijkheid om meerdere tabellen met elkaar te associëren in plaats van ze samen te voegen. Dit vereist veel minder geheugen, versnelt de verwerking en biedt een

## 2 Scriptinstructies en sleutelwoorden

enorme flexibiliteit. Expliciete joins kunnen dan ook beter worden vermeden in Qlik Sense-scripts. De functionaliteit van `keep` is ontworpen om het aantal gevallen waar expliciete joins moeten worden gebruikt te verkleinen.

### Argumenten:

Argumenten

Argument	Beschrijving
tablename	De benoemde tabel die moet worden vergeleken met de geladen tabel.
loadstatementof selectstatement	De <b>LOAD</b> of <b>SELECT</b> -opdracht voor de geladen tabel.

### Voorbeeld:

```
Inner Keep LOAD * from abc.csv;
Left Keep SELECT * from table1;
tab1:
LOAD * from file1.csv;
tab2:
LOAD * from file2.csv;
...
Left Keep (tab1) LOAD * from file3.csv;
```

## Left

De prefixen **Join** en **Keep** kunnen worden voorafgegaan door het prefix **left**.

Gebruikt voor **join** geeft het prefix aan dat er een join links moet plaatsvinden. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelande veldwaarden in de eerste tabel voorkomen. Als het prefix wordt gebruikt vóór **keep**, geeft het aan dat de tweede tabel met onbewerkte gegevens wordt gereduceerd tot de gemeenschappelijke doorsnede met de eerste tabel, voordat deze wordt opgeslagen in Qlik Sense.



Was u op zoek naar de tekenreeksfunctie met dezelfde naam? Zie: [Left \(page 1438\)](#)

### Syntaxis:

```
Left ( Join | Keep ) [ (tablename) ] (loadstatement | selectstatement)
```

### Argumenten:

Argumenten

Argument	Beschrijving
tablename	De benoemde tabel die moet worden vergeleken met de geladen tabel.

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
loadstatementof selectstatement	De <b>LOAD</b> of <b>SELECT</b> -opdracht voor de geladen tabel.

Voorbeeld

### Load-script

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
Table1: Load * inline [ Column1, Column2 A, B 1, aa 2, cc 3, ee ]; Table2: Left Join Load *  
inline [ Column1, Column3 A, C 1, xx 4, yy ];
```

### Resultaat

Resultaattabel

Column1	Column2	Column3
A	B	C
1	aa	xx
2	cc	-
3	ee	-

### Uitleg

Dit voorbeeld toont de uitvoer voor left join waarbij alleen waarden die aanwezig zijn in de eerste (links) tabel zijn samengevoegd.

## Mapping

Het prefix **mapping** wordt gebruikt om een toewijzingstabel te maken die bijvoorbeeld kan worden gebruikt voor het vervangen van veldwaarden en veldnamen tijdens de uitvoering van het script.

### Syntaxis:

```
Mapping( loadstatement | selectstatement )
```

Het prefix **mapping** kan voor een **LOAD**- of **SELECT**-opdracht worden gezet en gebruikt het resultaat van de laadopdracht als toewijzingstabel. Toewijzing vormt een efficiënt manier om veldwaarden te vervangen tijdens de uitvoering van het script, zoals het vervangen van US, U.S. of Amerika door USA. Een toewijzingstabel bestaat uit twee kolommen, de eerste met vergelijkingswaarden en de tweede met de gewenste toewijzingswaarden. Toewijzingstabellen worden tijdelijk in het geheugen opgeslagen en automatisch gewist na uitvoering van het script.

De inhoud van de toewijzingstabel is toegankelijk via onder andere de opdracht **Map ... Using**, de opdracht **Rename Field**, de functie **Applymap()** of de functie **Mapsubstring()**.

### Voorbeeld:

In dit voorbeeld laden wij een lijst met verkopers waarbij een landcode het land aangeeft waar zijn wonen. Wij gebruiken een tabel waarin een landcode wordt toegewezen aan een land ter vervanging van de landcode door de landnaam. Er zijn slechts drie landen gedefinieerd in de toewijzingstabel, andere landcode worden toegewezen aan 'Rest of the world'.

```
// Load mapping table of country codes:
map1:
mapping LOAD *
inline [
CCode, Country
Sw, Sweden
Dk, Denmark
No, Norway
] ;
// Load list of salesmen, mapping country code to country
// If the country code is not in the mapping table, put Rest of the world
Salespersons:
LOAD *,
ApplyMap('map1', CCode,'Rest of the world') As Country
inline [
CCode, Salesperson
Sw, John
Sw, Mary
Sw, Per
Dk, Preben
Dk, Olle
No, Ole
sf, Risttu] ;
// we don't need the CCode anymore
Drop Field 'CCode';
```

De resulterende tabel ziet er als volgt uit:

Mapping table

Salesperson	Country
John	Sweden
Mary	Sweden
Per	Sweden
Preben	Denmark
Olle	Denmark
Ole	Norway
Risttu	Rest of the world

### Merge

Het voorvoegsel **Merge** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat de geladen tabel met een andere tabel moet worden samengevoegd. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd.

Deze instructie wordt meestal gebruikt als u een wijzigingslogboek laadt en inserts, updates en deletes wilt toepassen op een bestaande tabel.



*Om ervoor te zorgen dat een gedeeltelijke lading correct werkt, moet de app met gegevens worden geopend voordat een gedeeltelijke lading wordt geactiveerd.*

Voer een gedeeltelijke lading uit met de knop **Opnieuw laden**. U kunt ook de Qlik Engine JSON API gebruiken.

#### Syntaxis:

```
Merge [only] [(SequenceNoField [, SequenceNoVar])] On ListOfKeys [Concatenate [(TableName)]] (loadstatement | selectstatement)
```

#### Argumenten:

Argumenten

Argument	Beschrijving
only	Een optionele kwalificatie die aangeeft dat de opdracht alleen moet worden uitgevoerd tijdens het gedeeltelijk laden van gegevens. De opdracht wordt genegeerd tijdens het normaal (niet-gedeeltelijk) laden van gegevens.
SequenceNoField	De naam van het veld dat een tijdstempel bevat of een volgnummer dat de volgorde van de bewerkingen definieert.
SequenceNoVar	De naam van de variabele die wordt toegewezen aan de maximumwaarde voor SequenceNoField van de tabel die wordt samengevoegd.
ListOfKeys	Een door komma's gescheiden lijst met veldnamen waarin de primaire sleutel is opgegeven.
Operation	Het eerste veld van de LOAD-instructie moet de bewerking als tekstreks bevatten: 'Insert', 'Update' of 'Delete'. 'i', 'u' en 'd' worden ook geaccepteerd.

### Algemene functionaliteit

Tijdens een normale (niet-gedeeltelijke) lading zal de constructie **Merge LOAD** worden uitgevoerd als een normale **Load**-instructie, maar zorgt dit ook voor de verwijdering van verouderde records en records die zijn gemarkeerd voor verwijdering. Het eerste veld van de **Load**-instructie moet informatie over de bewerking bevatten: Insert, Update of Delete.

Voor elke geladen record, wordt de identificatie van een record vergeleken met eerder geladen records en alleen het laatste record (op basis van het volgnummer) wordt behouden. Als het laatste record is gemarkeerd met Delete, worden er geen record behouden.

### Doeltabel

Welke tabel moet worden gewijzigd, wordt bepaald door de reeks velden. Als een tabel met dezelfde reeks velden (met uitzondering van het eerste veld; de bewerking) al bestaat, wordt dit de tabel die u moet wijzigen. U kunt ook het prefix **Concatenate** gebruiken om de tabel op te geven. Als de doeltabel niet is bepaald, wordt het resultaat van de **Merge LOAD**-constructie opgeslagen in een nieuwe tabel.

Als het prefix **Concatenate** wordt gebruikt, bevat de resulterende tabel een reeks velden die overeenkomt met de verbintenis van de bestaande tabel en de invoer voor de samenvoeging. De doeltabel kan daardoor meer tabellen krijgen dan het wijzigingslogboek dat is gebruik als invoer voor de samenvoeging.

Bij een gedeeltelijke lading gebeurt hetzelfde als bij een volledige lading. Het enige verschil is dat bij een gedeeltelijke lading vrijwel nooit een nieuwe tabel wordt gemaakt. Tenzij u de **Only**-clausule hebt gebruikt, bestaat er altijd een doeltabel met dezelfde reek velden van de vorige uitvoering van het script.

### Volgnummer

Als het geladen wijzigingslogboek een samengevoegd logboek is, dan bevat deze wijzigingen die al zijn geladen, de parameter **SequenceNoVar** kan worden gebruikt in een **Where**-clausule om een limiet in te stellen voor de invoergegevens. De constructie **Merge LOAD** kan dan alleen worden gebruikt voor laadrecords waarbij het veld **SequenceNoField** groter is dan **SequenceNoVar**. Na voltooiing wijst de constructie **Merge LOAD** een nieuwe waarde toe aan de **SequenceNoVar** met de maximumwaarde die te zien is in het veld **SequenceNoField**.

### Bewerkingen

De **Merge LOAD** kan minder velden bevatten dan de doeltabel. De verschillende bewerkingen gaan anders met missende velden om:

**Insert:** Ontbrekende velden in de **Merge LOAD**, die wel bestaan in de doeltabel, krijgen een NULL-waarde in de doeltabel.

**Delete:** Ontbrekende velden hebben geen invloed op het resultaat. De relevante records worden toch verwijderd.

**Update:** Velden die worden weergegeven in de **Merge LOAD** worden bijgewerkt in de doeltabel. Ontbrekende velden worden niet gewijzigd. Dit betekent dat de twee volgende opdrachten niet identiek zijn:

## 2 Scriptinstructies en sleutelwoorden

---

- Merge on Key Concatenate Load 'U' as Operation, Key, F1, Null() as F2 From ...;
- Merge on Key Concatenate Load 'U' as Operation, Key, F1 From ...;

Met de eerste opdracht worden de weergegeven records bijgewerkt en F2 gewijzigd in NULL. Bij de tweede opdracht wordt F2 niet gewijzigd, maar blijven de waarden van de doeltabel behouden.

Voorbeelden

### Voorbeeld 1 Eenvoudige samenvoeging met opgegeven tabel

In dit voorbeeld wordt een uitgelijnde tabel met vier rijen met de naam Persons geladen. **Merge** verandert de tabel dan als volgt:

- De rij *Mary*, 4 wordt toegevoegd.
- De rij *Steven*, 3 wordt verwijderd.
- Het getal 5 wordt toegewezen aan *Jake*.

De variabele *LastChangeDate* wordt ingesteld op de maximumwaarde in de kolom *ChangeDate* nadat **Merge** is uitgevoerd.

### Load-script

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
Set DateFormat='D/M/YYYY';
Persons:
load * inline [
Name, Number
Jake, 3
Jill, 2
Steven, 3
];
```

```
Merge (ChangeDate, LastChangeDate) on Name Concatenate(Persons)
LOAD * inline [
Operation, ChangeDate, Name, Number
Insert, 1/1/2021, Mary, 4
Delete, 1/1/2021, Steven,
Update, 2/1/2021, Jake, 5
];
```

### Resultaat

Voorafgaand aan de lading **Merge Load** verschijnt de volgende tabel:

Resulting table

Name	Number
Jake	3
Jill	2
Steven	3

## 2 Scriptinstructies en sleutelwoorden

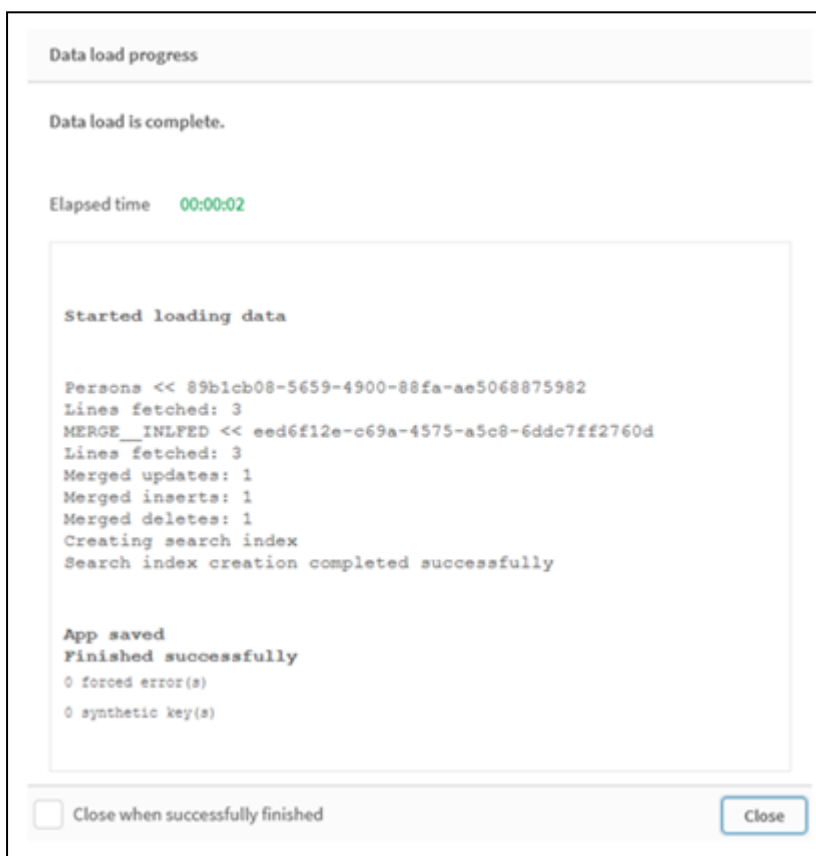
Na de **Merge Load** verschijnt de volgende tabel:

Resulting table

ChangeDate	Name	Number
2/1/2021	Jake	5
-	Jill	2
1/1/2021	Mary	4

Als de gegevens zijn geladen toont het dialoogvenster **Voortgang laden van gegevens** de bewerkingen die worden uitgevoerd:

*Dialoogvenster Voortgang laden van gegevens*



### Voorbeeld 2: Load-script voor gegevens met ontbrekende velden

In dit voorbeeld worden dezelfde gegevens als hierboven geladen, maar nu met een id voor elke persoon.

**Merge** verandert de tabel als volgt:

- De rij *Mary*, 4 wordt toegevoegd.
- De rij *Steven*, 3 wordt verwijderd.



## 2 Scriptinstructies en sleutelwoorden

---

- Het getal 5 wordt toegewezen aan *Jake*.
- Het getal 6 wordt toegewezen aan *Jill*.

### Load-script

Hier gebruiken we twee **Merge Load**-opdrachten, een voor 'Invoegen' en 'Verwijderen' en een voor het 'Bijwerken'.

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
Set DateFormat='D/M/YYYY';
```

```
Persons:
```

```
Load * Inline [  
PersonID, Name, Number  
1, Jake, 3  
2, Jill, 2  
3, Steven, 3  
];
```

```
Merge (ChangeDate, LastChangeDate) on PersonID Concatenate(Persons)
```

```
Load * Inline [  
Operation, ChangeDate, PersonID, Name, Number  
Insert, 1/1/2021, 4, Mary, 4  
Delete, 1/1/2021, 3, Steven,  
];
```

```
Merge (ChangeDate, LastChangeDate) on PersonID Concatenate(Persons)
```

```
Load * Inline [  
Operation, ChangeDate, PersonID, Number  
Update, 2/1/2021, 1, 5  
Update, 3/1/2021, 2, 6  
];
```

### Resultaat

Na de **Merge Load**-opdrachten verschijnt de volgende tabel:

Resulting table

PersonID	ChangeDate	Name	Number
1	2/1/2021	Jake	5
2	3/1/2021	Jill	6
4	1/1/2021	Mary	4

De tweede **Merge**-opdracht bevat niet het veld **Name** en daardoor zijn de namen mogelijk niet gewijzigd.

### Voorbeeld 3: Load-script voor gegevens - Gedeeltelijke lading met een Where-clausule met ChangeDate

In het volgende voorbeeld, geeft het argument **Only** op dat de instructie **Merge** alleen wordt uitgevoerd tijdens een gedeeltelijke lading. Updates worden gefilterd op basis van de eerder vastgelegde LastChangeDate. Nadat **Merge** is voltooid, wordt de variabele LastChangeDate toegewezen aan de maximumwaarde van de kolom ChangeDate die tijdens het samenvoegen is verwerkt

#### Load-script

```
Merge only (ChangeDate, LastChangeDate) on Name Concatenate(Persons)
LOAD Operation, ChangeDate, Name, Number
from [lib://ChangeFilesFolder/BulkChangesInPersonsTable.csv] (txt)
where ChangeDate >= $(LastChangeDate);
```

### NoConcatenate

Het prefix **NoConcatenate** zorgt ervoor dat twee geladen tabellen met identieke veldensets worden behandeld als twee aparte interne tabellen, terwijl ze anders automatisch zouden worden aaneengeschakeld.

#### Syntaxis:

```
NoConcatenate ( loadstatement | selectstatement )
```

Als een tabel met een identiek aantal velden en overeenkomende veldnamen eerder in het script in een tabel is geladen, zal Qlik Sense standaard deze twee tabellen automatisch aaneenschakelen. Dit gebeurt ook als de tweede tabel een andere naam heeft.

Maar als het script-prefix **NoConcatenate** is ingevoegd vóór de load-opdracht of de selecteeropdracht van de tweede tabel, dan worden deze twee tabellen afzonderlijk geladen.

Een typische gebruikssituatie voor **NoConcatenate** is wanneer u mogelijk een tijdelijke kopie van een tabel moet maken om een aantal tijdelijke transformaties in die kopie uit te voeren, terwijl u een kopie van de oorspronkelijke gegevens bewaart. **NoConcatenate** zorgt ervoor dat u die kopie kunt maken zonder deze impliciet weer toe te voegen aan de brontabel.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/YYYY. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## 2 Scriptinstructies en sleutelwoorden

### Functievoorbeeld

Voorbeeld	Resultaat
Source: LOAD A,B from file1.csv; CopyOfSource: NoConcatenate LOAD A,B resident Source;	Er wordt een tabel met A en B als metingen geladen. Een tweede tabel met dezelfde velden wordt afzonderlijk geladen door gebruik van de NoConcatenate-variabele.

### Voorbeeld 1 - Impliciete aaneenschakeling

Load-script en resultaten

#### Overzicht

In dit voorbeeld voegt u twee load-scripts achter elkaar toe.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een eerste gegevensset met datums en bedragen die naar de tabel `Transactions` wordt gestuurd.

#### Eerste load-script

Transactions:

```
LOAD
```

```
*
```

```
Inline [
```

```
id, date, amount
```

```
1, 08/30/2018, 23.56
```

```
2, 09/07/2018, 556.31
```

```
3, 09/16/2018, 5.75
```

```
4, 09/22/2018, 125.00
```

```
5, 09/22/2018, 484.21
```

```
6, 09/22/2018, 59.18
```

```
7, 09/23/2018, 177.42
```

```
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- amount

Eerste resultatentabel

id	date	amount
1	08/30/2018	23.56

id	date	amount
2	09/07/2018	556.31
3	09/16/2018	5.75
4	09/22/2018	125.00
5	09/22/2018	484.21
6	09/22/2018	59.18
7	09/23/2018	177.42

### Tweede load-script

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een tweede gegevensset met identieke velden wordt naar de tabel sales gestuurd.

sales:

LOAD

\*

Inline [

id, date, amount

8, 10/01/2018, 164.27

9, 10/03/2018, 384.00

10, 10/06/2018, 25.82

11, 10/09/2018, 312.00

12, 10/15/2018, 4.56

13, 10/16/2018, 90.24

14, 10/18/2018, 19.32

];

### Resultaten

Laad de gegevens en ga naar de tabel.

Tweede resultatentabel

id	date	amount
1	08/30/2018	23.56
2	09/07/2018	556.31
3	09/16/2018	5.75
4	09/22/2018	125.00
5	09/22/2018	484.21
6	09/22/2018	59.18
7	09/23/2018	177.42

## 2 Scriptinstructies en sleutelwoorden

---

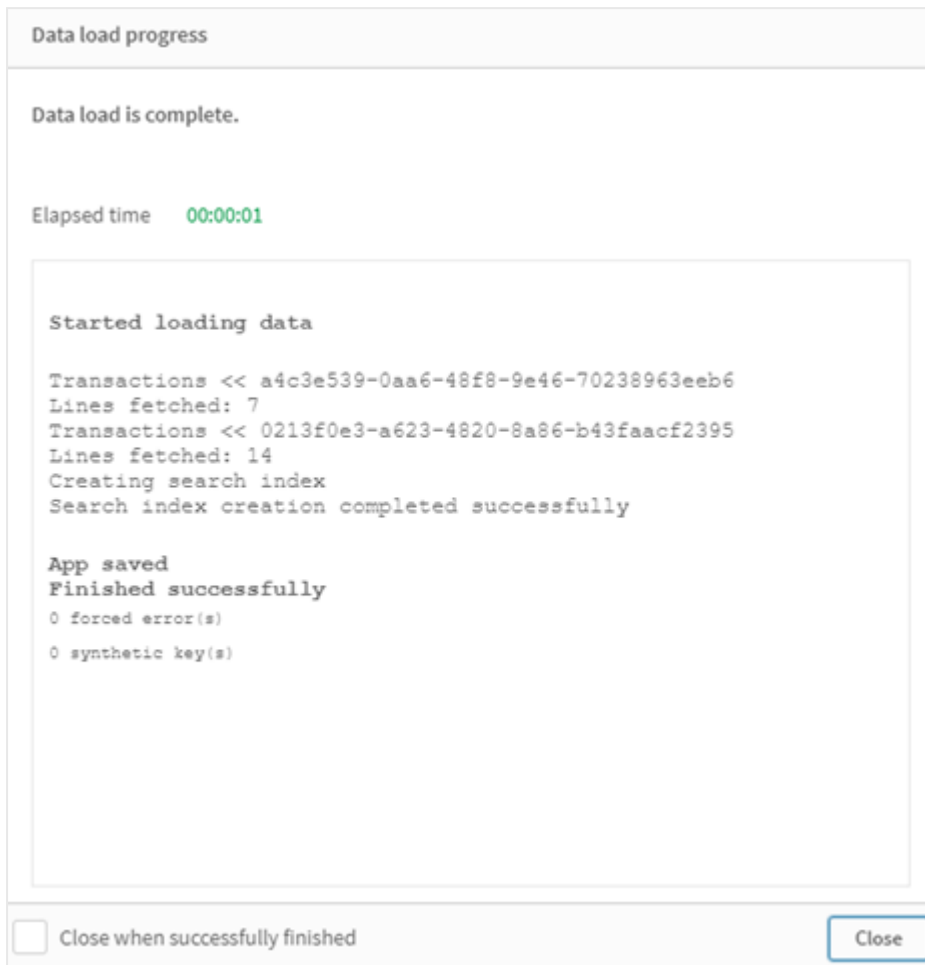
id	date	amount
8	10/01/2018	164.27
9	10/03/2018	384.00
10	10/06/2018	25.82
11	10/09/2018	312.00
12	10/15/2018	4.56
13	10/16/2018	90.24
14	10/18/2018	19.32

Wanneer het script wordt uitgevoerd, wordt de tabel `sales` impliciet aaneengeschakeld met de bestaande `Transactions`-tabel omdat de twee gegevensset een identiek aantal velden delen met identieke veldnamen. Dit gebeurt ondanks het feit dat het tweede tabelnaamlabel de resultaten set 'sales' probeert te noemen.

Als u naar het logbestand **Voortgang laden van gegevens** kijkt, kunt u zien dat de verkoopgegevensset impliciet aaneengeschakeld wordt.

## 2 Scriptinstructies en sleutelwoorden

Het logbestand *Voortgang laden van gegevens* toont dat transactiegegevens impliciet aaneengeschakeld worden.



### Voorbeeld 2 - Scenario voor gebruikssituatie

Load-script en resultaten

#### Overzicht

In deze gebruikssituatie hebt u:

- Een transactiegegevensset met:
  - id
  - date
  - bedrag (in GBP)
- Een valutatablel met:
  - Wisselkoers voor USD naar GBP
- Een tweede transactiegegevensset met:
  - id

## 2 Scriptinstructies en sleutelwoorden

---

- date
- bedrag (in USD)

U laadt vijf scripts achter elkaar.

- Het eerste load-script bevat een aanvankelijke gegevensset met datums en bedragen in GBP, die naar de tabel `Transactions` wordt gestuurd.
- Het tweede load-script bevat:
  - Een tweede gegevensset met datums en bedragen in USD, die naar de tabel `Transactions_in_USD` wordt gestuurd.
  - Het `noconcatenate`-prefix dat vóór de load-instructie van de `Transactions_in_USD`-gegevensset is geplaatst om impliciete aaneenschakeling te voorkomen.
- Het derde load-script bevat het `join`-prefix dat wordt gebruikt om een wisselkoers op te zetten tussen GBP en USD in de `Transactions_in_USD`-tabel.
- Het vierde load-script bevat het `concatenate`-prefix dat de `Transactions_in_USD` toevoegt aan de aanvankelijke `Transactions`-tabel.
- Het vijfde load-script bevat de `drop table`-opdracht die de `Transactions_in_USD`-tabel verwijdert waaruit gegevens zijn aaneengeschakeld met de `Transactions`-tabel.

### Eerste load-script

`Transactions:`

```
Load * Inline [  
id, date, amount  
1, 12/30/2018, 23.56  
2, 12/07/2018, 556.31  
3, 12/16/2018, 5.75  
4, 12/22/2018, 125.00  
5, 12/22/2018, 484.21  
6, 12/22/2018, 59.18  
7, 12/23/2018, 177.42  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- amount

Resultaten eerste load-script

id	date	amount
1	12/30/2018	23.56
2	12/07/2018	556.31

## 2 Scriptinstructies en sleutelwoorden

---

<b>id</b>	<b>date</b>	<b>amount</b>
3	12/16/2018	5.75
4	12/22/2018	125.00
5	12/22/2018	484.21
6	12/22/2018	59.18
7	12/23/2018	177.42

De tabel toont de eerste gegevensset met bedragen in GBP.

### Tweede load-script

```
Transactions_in_USD:  
NoConcatenate  
Load * Inline [  
id, date, amount  
8, 01/01/2019, 164.27  
9, 01/03/2019, 384.00  
10, 01/06/2019, 25.82  
11, 01/09/2019, 312.00  
12, 01/15/2019, 4.56  
13, 01/16/2019, 90.24  
14, 01/18/2019, 19.32  
];
```

### Resultaten

Laad de gegevens en ga naar de tabel.

#### Resultaten tweede load-script

<b>id</b>	<b>date</b>	<b>amount</b>
1	12/30/2018	23.56
2	12/07/2018	556.31
3	12/16/2018	5.75
4	12/22/2018	125.00
5	12/22/2018	484.21
6	12/22/2018	59.18
7	12/23/2018	177.42
8	01/01/2019	164.27
9	01/03/2019	384.00
10	01/06/2019	25.82
11	01/09/2019	312.00



## 2 Scriptinstructies en sleutelwoorden

---

id	date	amount
12	01/15/2019	4.56
13	01/16/2019	90.24
14	01/18/2019	19.32

U ziet dat de tweede gegevensset uit de Transactions\_in\_USD-tabel is toegevoegd.

### Derde load-script

Dit load-script koppelt een wisselkoers van USD naar GBP aan de Transactions\_in\_USD-tabel.

```
Join (Transactions_in_USD)
Load * Inline [
rate
0.7
];
```

### Resultaten

Laad de gegevens en ga naar de gegevensmodelviewer. Selecteer de Transactions\_in\_USD-tabel en u zult zien dat elk bestaand record een koersveldwaarde heeft van 0,7.

### Vierde load-script

Met gebruik van een resident-load zal dit load-script de Transactions\_in\_USD-tabel aaneenschakelen met de Transactions-tabel nadat de bedragen zijn omgezet in USD.

```
Concatenate (Transactions)
LOAD
id,
date,
amount * rate as amount
Resident Transactions_in_USD;
```

### Resultaten

Laad de gegevens en ga naar de tabel. U ziet nieuwe invoeren met bedragen in GBP van regels acht t/m veertien.

Resultaten vierde load-script

id	date	amount
1	12/30/2018	23.56
2	12/07/2018	556.31
3	12/16/2018	5.75
4	12/22/2018	125.00

<b>id</b>	<b>date</b>	<b>amount</b>
5	12/22/2018	484.21
6	12/22/2018	59.18
7	12/23/2018	177.42
8	01/01/2019	114.989
8	01/01/2019	164.27
9	01/03/2019	268.80
9	01/03/2019	384.00
10	01/06/2019	18.074
10	01/06/2019	25.82
11	01/09/2019	218.40
11	01/09/2019	312.00
12	01/15/2019	3.192
12	01/15/2019	4.56
13	01/16/2019	63.168
13	01/16/2019	90.24
14	01/18/2019	13.524
14	01/18/2019	19.32

### Vijfde load-script

Dit load-script verwijdert de dubbele invoeren uit de resultatentabel van het vierde load-script en laat alleen invoeren met bedragen in GBP staan.

```
drop tables Transactions_in_USD;
```

### Resultaten

Laad de gegevens en ga naar de tabel.

Resultaten vijfde load-script

<b>id</b>	<b>date</b>	<b>amount</b>
1	12/30/2018	23.56
2	12/07/2018	556.31
3	12/16/2018	5.75
4	12/22/2018	125.00
5	12/22/2018	484.21

## 2 Scriptinstructies en sleutelwoorden

---

id	date	amount
6	12/22/2018	59.18
7	12/23/2018	177.42
8	01/01/2019	114.989
9	01/03/2019	268.80
10	01/06/2019	18.074
11	01/09/2019	218.40
12	01/15/2019	3.192
13	01/16/2019	63.168
14	01/18/2019	13.524

Nadat het vijfde load-script is geladen, toont de resultatentabel alle veertien transacties die in beide transactiegegevenssets stonden. Maar voor transacties 8-14 zijn de bedragen omgezet in GBP.

Als we het `noconcatenate-prefix` verwijderen dat werd gebruikt vóór de `Transactions_in_USD` in het tweede load-script, geeft het script de volgende foutmelding weer: "Tabel `Transactions_in_USD` niet gevonden". Dit komt doordat de `Transactions_in_USD`-tabel automatisch had zullen worden aaneengeschakeld met de oorspronkelijke `Transactions`-tabel.

### Only

Het sleutelwoord **Only** in scripts wordt gebruikt als aggregatiefunctie of als onderdeel van de syntaxis in de prefixen **Add**, **Replace** en **Merge** voor gedeeltelijk opnieuw laden.

### Outer

Het expliciete prefix **Join** kan worden voorafgegaan door het prefix **Outer** om een outer join op te geven. In een outer join worden alle combinaties tussen de twee tabellen gegenereerd. De resulterende tabel bevat dus gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in één of beide tabellen voorkomen. Het trefwoord **Outer** is optioneel en is het standaard join type dat wordt gebruikt als geen join prefix is opgegeven.

#### Syntaxis:

```
Outer Join [ (tablename) ] (loadstatement |selectstatement )
```

#### Argumenten:

##### Argumenten

Argument	Beschrijving
tablename	De benoemde tabel die moet worden vergeleken met de geladen tabel.
loadstatementof selectstatement	De <b>LOAD</b> of <b>SELECT</b> -opdracht voor de geladen tabel.

## 2 Scriptinstructies en sleutelwoorden

---

Voorbeeld

### Load-script

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
Table1: Load * inline [ Column1, Column2 A, B 1, aa 2, cc 3, ee ]; Table2: Outer Join Load * inline [ Column1, Column3 A, C 1, xx 4, yy ];
```

Resultaattabel

Column1	Column2	Column3
A	B	C
1	aa	xx
2	cc	-
3	ee	-
4	-	yy

### Uitleg

In dit voorbeeld worden de twee tabellen, Table1 en Table2 samengevoegd in één tabel met de naam Table1. In dit soort gevallen wordt vaak het prefix **outer** gebruikt om verschillende tabellen samen te voegen tot één tabel om aggregaties uit te voeren voor de waarden van één tabel.

### Gedeeltelijke lading

Bij een volledige lading worden altijd eerst alle tabellen in het bestaande gegevensmodel verwijderd en vervolgens wordt het load-script uitgevoerd.

Bij een gedeeltelijke lading gebeurt dit niet. Hierbij blijven alle tabellen in het gegevensmodel behouden en worden alleen de opdrachten **Loaden Select** uitgevoerd voorafgegaan door het voorvoegsel **Add, Merge of Replace**. Andere gegevenstabellen worden niet door de instructie beïnvloed. Het **only**-argument geeft aan dat de opdracht alleen tijdens gedeeltelijke ladingen moet worden uitgevoerd, en moet worden genegeerd tijdens volledige ladingen. In de volgende tabel vindt u een overzicht van de uitvoering van opdrachten voor gedeeltelijke en volledige ladingen.

Opdracht	Volledige lading	Gedeeltelijke lading
Load ...	Opdracht wordt uitgevoerd	Opdracht wordt niet uitgevoerd
Add/Replace/Merge Load ...	Opdracht wordt uitgevoerd	Opdracht wordt uitgevoerd
Add/Replace/Merge Only Load ...	Opdracht wordt niet uitgevoerd	Opdracht wordt uitgevoerd

Gedeeltelijk opnieuw laden heeft verschillende voordelen ten opzichte van volledig opnieuw laden:

## 2 Scriptinstructies en sleutelwoorden

- Sneller, omdat alleen recent gewijzigde gegevens hoeven te worden geladen. Bij grote gegevensverzamelingen is het verschil aanzienlijk.
- Er wordt minder geheugen verbruikt, omdat er minder gegevens worden geladen.
- Betrouwbaarder, omdat query's naar brongegevens sneller verlopen, waardoor het risico op netwerkproblemen wordt verminderd.



*Om ervoor te zorgen dat een gedeeltelijke lading correct werkt, moet de app met gegevens worden geopend voordat een gedeeltelijke lading wordt geactiveerd.*

Voer een gedeeltelijke lading uit met de knop **Opnieuw laden**. U kunt ook de Qlik Engine JSON API gebruiken.

### Beperking

Bij een gedeeltelijke lading kunnen waarden van de gegevens worden verwijderd. Dit wordt echter niet weergegeven in de lijst met bijzondere waarden. Dit is een lijst die intern wordt bijgehouden. Dat betekent dat de lijst, na gedeeltelijk opnieuw laden, alle bijzondere waarden zal bevatten die in het veld hebben bestaan sinds de laatste volledige lading. Dit kan meer zijn dan wat er op dit moment bestaat na de gedeeltelijke lading. Dit treft de uitvoer van de functies `FieldValueCount()` en `FieldValue()`. De functie `FieldValueCount()` zou mogelijk een aantal kunnen retourneren dat groter is dan het huidige aantal veldwaarden.

Voorbeeld

### Voorbeeld 1

#### Load-script

Voeg het voorbeeldscript toe aan uw app en voer een gedeeltelijke lading uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

T1:

```
Add only Load distinct recno()+10 as Num autogenerate 10;
```

### Resultaat

Resulting table

Num	Count(Num)
11	1
12	1
13	1
14	1
15	1
16	1

## 2 Scriptinstructies en sleutelwoorden

---

Num	Count(Num)
17	1
18	1
19	1
20	1

### Uitleg

De opdracht wordt alleen tijdens een gedeeltelijke lading uitgevoerd. Als het prefix 'distinct' wordt weggelaten, zal het aantal van het veld **Num** na elke opeenvolgende gedeeltelijke lading worden verhoogd.

### Voorbeeld 2

#### Load-script

Voeg het voorbeeldscript toe aan uw app. Voer een volledige lading uit en bekijk het resultaat. Voer vervolgens een gedeeltelijke lading uit en bekijk het resultaat. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om de resultaten te bekijken.

T1:

```
Load recno() as ID, recno() as Value autogenerated 10;
```

T1:

```
Replace only Load recno() as ID, repeat(recno(),3) as Value autogenerated 10;
```

### Resultaat

Output table after full reload

ID	Value
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

## 2 Scriptinstructies en sleutelwoorden

Output table after partial reload

ID	Value
1	111
2	222
3	333
4	444
5	555
6	666
7	777
8	888
9	999
10	101010

### Uitleg

De eerste tabel is geladen tijdens een volledige lading en de tweede tabel vervangt de eerste tabel tijdens een gedeeltelijke lading.

## Replace

Het sleutelwoord **Replace** voor scripts wordt gebruikt als tekenreeksfunctie of als prefix bij gedeeltelijk opnieuw laden.

### Replace

het voorvoegsel **Replace** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat de geladen tabel een andere tabel moet vervangen. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Replace** kan ook worden gebruikt in een **Map**-opdracht.



*Om ervoor te zorgen dat een gedeeltelijke lading correct werkt, moet de app met gegevens worden geopend voordat een gedeeltelijke lading wordt geactiveerd.*

Voer een gedeeltelijke lading uit met de knop **Opnieuw laden**. U kunt ook de Qlik Engine JSON API gebruiken.

### Syntaxis:

```
Replace [only] [Concatenate[(tablename)]] (loadstatement | selectstatement)
```

```
Replace [only] mapstatement
```

## 2 Scriptinstructies en sleutelwoorden

Tijdens een normale (niet-gedeeltelijke) lading, dient de constructie **Replace LOAD** als een normale **LOAD**-instructie, maar wordt voorafgegaan door **Drop Table**. Eerst wordt de oude tabel verwijderd vervolgens worden de records gegenereerd en opgeslagen als een nieuwe tabel.

Als het voorvoegsel **Concatenate** is gebruikt of er is een tabel aanwezig met dezelfde verzameling van velden, is dit de relevante tabel die wordt verwijderd. Anders is er geen tabel om te negeren en is de constructie **Replace LOAD** identiek aan een normale **LOAD**.

Bij een gedeeltelijke lading gebeurt hetzelfde. Het enige verschil is dat er altijd een tabel uit het vorige uitgevoerde script is om te verwijderen. Bij de constructie **Replace LOAD** wordt altijd eerste de oude tabel verwijderd en wordt er vervolgens een nieuwe tabel gemaakt.

De instructie **Replace Map...Using** zorgt ervoor dat de toewijzing ook plaatsvindt als het script gedeeltelijk wordt uitgevoerd.

### Argumenten:

#### Argumenten

Argument	Beschrijving
only	Een optionele kwalificatie die aangeeft dat de opdracht alleen moet worden uitgevoerd tijdens het gedeeltelijk laden van gegevens. Het moet worden genegeerd tijdens een normale (niet-gedeeltelijke) lading.

### Voorbeelden en resultaten:

Voorbeeld	Resultaat
Tab1: Replace LOAD * from File1.csv;	Tijdens zowel volledig als gedeeltelijk opnieuw laden wordt de Qlik Sense-tabel Tab1 aanvankelijk verwijderd. Daarna worden er nieuwe gegevens uit File1.csv geladen en opgeslagen in Tab1.
Tab1: Replace only LOAD * from File1.csv;	Tijdens normaal opnieuw laden wordt deze opdracht genegeerd.  Tijdens gedeeltelijk opnieuw laden wordt elke Qlik Sense-tabel die eerst Tab1 heette, aanvankelijk verwijderd. Daarna worden er nieuwe gegevens uit File1.csv geladen en opgeslagen in Tab1.
Tab1: LOAD a,b,c from File1.csv; Replace LOAD a,b,c from File2.csv;	Tijdens normaal opnieuw laden wordt File1.csv eerst in de Qlik Sense-tabel Tab1 gelezen, maar meteen daarna verwijderd en vervangen door nieuwe gegevens, geladen uit File2.csv. Alle gegevens uit File1.csv zijn verloren gegaan.  Tijdens gedeeltelijk opnieuw laden wordt de gehele Qlik Sense-tabel Tab1 aanvankelijk verwijderd. Daarna wordt deze vervangen door nieuwe gegevens, geladen uit File2.csv.



## 2 Scriptinstructies en sleutelwoorden

Voorbeeld	Resultaat
Tab1: LOAD a,b,c from File1.csv; Replace only LOAD a,b,c from File2.csv;	Tijdens normaal opnieuw laden worden gegevens geladen uit File1.csv en opgeslagen in de Qlik Sense-tabel Tab1. File2.csv wordt genegeerd.  Tijdens gedeeltelijk opnieuw laden wordt de gehele Qlik Sense-tabel Tab1 aanvankelijk verwijderd. Daarna wordt deze vervangen door nieuwe gegevens, geladen uit File2.csv. Alle gegevens uit File1.csv zijn verloren gegaan.

### Right

De prefixen **Join** en **Keep** kunnen worden voorafgegaan door het prefix **right**.

Als het prefix wordt gebruikt vóór **join**, geeft het aan dat er een join rechts moet plaatsvinden. De resulterende tabel bevat dan alleen gecombineerde veldwaarden uit de tabellen met onbewerkte gegevens waarvoor de koppelende veldwaarden in de tweede tabel voorkomen. Als het prefix wordt gebruikt voor **keep**, geeft het aan dat de eerste tabel met onbewerkte gegevens wordt gereduceerd tot de gemeenschappelijke doorsnede met de tweede tabel, voordat deze wordt opgeslagen in Qlik Sense.



Was u op zoek naar de tekenreeksfunctie met dezelfde naam? Zie: [Right \(page 1446\)](#)

#### Syntaxis:

```
Right (Join | Keep) [(tablename)] (loadstatement | selectstatement )
```

#### Argumenten:

##### Argumenten

Argument	Beschrijving
tablename	De benoemde tabel die moet worden vergeleken met de geladen tabel.
loadstatementof selectstatement	De <b>LOAD</b> of <b>SELECT</b> -opdracht voor de geladen tabel.

#### Voorbeeld

##### Load-script

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
Table1: Load * inline [ Column1, Column2 A, B 1, aa 2, cc 3, ee ]; Table2: Right Join Load *  
inline [ Column1, Column3 A, C 1, xx 4, yy ];
```

### Resultaat

Resultaattabel

Column1	Column2	Column3
A	B	C
1	aa	xx
4	-	yy

### Uitleg

Dit voorbeeld toont de uitvoer voor `right join` waarbij alleen waarden die aanwezig zijn in de tweede (rechts) tabel zijn samengevoegd.

## Sample

Als u het prefix **sample** voor een **LOAD**- of **SELECT**-opdracht plaatst, wordt er een willekeurige selectie records uit de gegevensbron geladen.

### Syntaxis:

```
Sample p ( loadstatement | selectstatement )
```

De uitdrukking die wordt beoordeeld, definieert niet het percentage records uit de gegevensset dat wordt geladen in de Qlik Sense-applicatie, maar de waarschijnlijkheid dat elk record dat wordt gelezen in de applicatie zal worden geladen. Met andere woorden: door een waarde van  $p = 0.5$  te specificeren, wil dat niet zeggen dat 50% van het totale aantal records wordt geladen, maar dat voor elk bestaand record er een kans van 50% is dat het in de Qlik Sense-applicatie zal worden geladen.

Argumenten

Argument	Beschrijving
p	Een willekeurige uitdrukking die wordt geëvalueerd als een getal dat groter dan 0 en kleiner dan of gelijk is aan 1. Het getal geeft de waarschijnlijkheid aan dat een bepaalde record wordt gelezen.  Alle records worden gelezen maar slechts enkele daarvan worden in Qlik Sense geladen.

## Wanneer gebruiken

Een voorbeeld is handig wanneer u wilt dat er uit een grote tabel voorbeeldgegevens worden gehaald om de aard van gegevens, verdeling of veldinhoud te begrijpen. Omdat dit een subset van gegevens ophaalt, worden gegevens sneller geladen zodat scripts ook sneller kunnen worden getest. Anders dan de functie `First`, haalt de functie `sample` gegevens uit de hele tabel in plaats van beperkt te zijn tot de eerste paar rijen. Dit kan in bepaalde gevallen een nauwkeurigere afspiegeling van gegevens opleveren.

De volgende voorbeelden geven twee mogelijke toepassingen weer van het `sample`-scriptprefix:

```
sample 0.15 SQL SELECT * from Longtable;  
sample(0.15) LOAD * from Longtab.csv;
```

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Voorbeeld van een inline-tabel

Load-script en resultaten

#### Overzicht

In dit voorbeeld laadt het script vanuit een inline-tabel een voorbeeldgegevensset uit een gegevensset met zeven records in de tabel `Transactions`.

#### Load-script

```
Transactions:
SAMPLE 0.3
LOAD
*
Inline [
id, date, amount
1, 08/30/2018, 23.56
2, 09/07/2018, 556.31
3, 09/16/2018, 5.75
4, 09/22/2018, 125.00
5, 09/22/2018, 484.21
6, 09/22/2018, 59.18
7, 09/23/2018, 177.42
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `amount`

Voeg de volgende meting toe:

## 2 Scriptinstructies en sleutelwoorden

---

=sum(amount)8

Resultatentabel

id	date	=Sum(amount)
2	09/07/2018	556.31
4	09/22/2018	125
1	08/30/2018	23.56
3	09/16/2018	5.75

In de iteratie van de load die in dit voorbeeld wordt gebruikt, zijn alle zeven records gelezen, maar zijn er slechts vier in de gegevenstabel geladen. Elke keer dat de load opnieuw wordt uitgevoerd, kan dit resulteren in een ander aantal en in dat er een andere set records in de applicatie wordt geladen.

### Voorbeeld 2 - Voorbeeld uit een automatisch gegenereerde tabel

Load-script en resultaten

#### Overzicht

In dit voorbeeld met `Autogenerate`, wordt een gegevensset van 100 records gemaakt met de velden `date`, `id` en `amount`. Maar het prefix `sample` wordt gebruikt met een waarde van 0,1.

#### Load-script

```
sampleData:
sample 0.1
LOAD
RecNo() AS id,
MakeDate(2013, Ceil(Rand() * 12), Ceil(Rand() * 29)) as date,
Rand() * 1000 AS amount
```

```
Autogenerate(100);
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- amount

Voeg de volgende meting toe:

Resultatentabel

id	date	=Sum(amount)
48	9/28/2013	763

id	date	=Sum(amount)
20	5/15/2013	752
19	11/8/2013	657
25	3/24/2013	522
27	8/23/2013	389
81	6/1/2013	53
100	8/15/2013	17

In de iteratie van de load die in dit voorbeeld wordt gebruikt, zijn zeven records geladen uit de aangemaakte gegevensset. Wederom geldt dat elke keer dat de load opnieuw wordt uitgevoerd, dit kan resulteren in een ander aantal en in dat er een andere set records in de applicatie wordt geladen.

### Semantic

Het laadvoorvoegsel `semantic` maakt een speciaal type veld dat kan worden gebruikt in Qlik Sense om relationele gegevens te verbinden en te beheren, zoals boomstructuren, zelfverwijzende bovenliggende/onderliggende gestructureerde gegevens en/of gegevens die kunnen worden beschreven als een diagram.

Merk op dat de lading `semantic` op dezelfde manier kan werken als de voorvoegsels *Hierarchy* (page 64) en *HierarchyBelongsTo* (page 66). Alle drie de voorvoegsels kunnen worden gebruikt als bouwstenen in effectieve front-end-oplossingen voor het doorkruisen van relationele gegevens.

#### Syntaxis:

```
Semantic( loadstatement | selectstatement)
```

Een semantische lading verwacht een invoer die precies drie of vier velden breed is met een strikte definitie van wat elk geordend veld vertegenwoordigt, zoals weergegeven in de onderstaande tabel:

#### Semantische ladingsvelden

Veldnaam	Veldbeschrijving
1e veld:	Deze tag is een weergave van het eerste van twee objecten waartussen een relatie bestaat.
2e veld:	Deze tag wordt gebruikt om de 'voorwaartse' relatie tussen het eerste en tweede object te beschrijven. Als het eerste object een onderliggend object is en het tweede object een bovenliggend object, kunt u een relatietabblad maken met de vermelding 'bovenliggend' of 'bovenliggend van' alsof u de relatie van onderliggend tot bovenliggend volgt.
3e veld:	Deze tag is een weergave van het tweede van twee objecten waartussen een relatie bestaat.
4e veld:	Dit veld is optioneel. Deze tag beschrijft de 'achterwaartse' of 'omgekeerde' relatie

Veldnaam	Veldbeschrijving
	tussen het eerste en tweede object. Als het eerste object een onderliggend object is en het tweede object een bovenliggend object, kan een relatietabblad 'onderliggend' of 'onderliggend van' vermelden, alsof u de relatie van bovenliggend tot onderliggend volgt. Als u geen vierde veld toevoegt, wordt de tweede veldtag gebruikt om de relatie in beide richtingen te beschrijven. In dat geval wordt automatisch een pijlsymbool toegevoegd als onderdeel van de tag.

De volgende code is een voorbeeld van het voorvoegsel `semantic`.

```
Semantic
Load
Object,
'Parent' AS Relationship,
NeighbouringObject AS Object,
'Child' AS Relationship
from graphdata.csv;
```



*Het is toegestaan en gebruikelijk om het derde veld hetzelfde te labelen als het eerste veld. Dit creëert een zelfverwijzende opzoekactie, zodat u object(en) stap voor stap naar het/de verwante object(en) kunt volgen. Als het 3e veld niet dezelfde naam heeft, dan is het eindresultaat een eenvoudige opzoekactie van (een) object(en) naar zijn directe relationele buur/buren slechts één stap verwijderd, wat een uitvoer is van weinig praktisch nut.*

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

#### Gerelateerde functies

Functies	Interactie
<i>Hierarchy (page 64)</i>	Het laadvoorvoegsel Hiërarchie wordt gebruikt om knooppunten in bovenliggende/onderliggende gegevensstructuren en andere diagramachtige gegevensstructuren te verdelen en te ordenen en deze om te zetten in tabellen.

### Functies

*HierarchyBelongsTo*  
(page 66)

### Interactie

Het laadvoorvoegsel *HierarchyBelongsTo* wordt gebruikt om de voorlopers van bovenliggende/onderliggende gegevensstructuren en andere diagramachtige gegevensstructuren te vinden en te ordenen en deze om te zetten in tabellen.

## Voorbeeld - Een speciaal veld maken voor het verbinden van relaties met behulp van het semantische voorvoegsel

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die geografierelatierecords vertegenwoordigt, die in een tabel met de naam *GeographyTree* wordt geladen.
  - Elk item heeft een *id* aan het begin van de regel en een *ParentID* aan het einde van de regel.
- Het voorvoegsel *semantic* dat een speciaal gedragsveld met het label *relation* zal toevoegen.

### Load-script

*GeographyTree*:

```
LOAD
  ID,
  Geography,
  if(ParentID='',null(),ParentID) AS ParentID
```

```
INLINE [
ID,Geography,ParentID
1,world
2,Europe,1
3,Asia,1
4,North America,1
5,South America,1
6,UK,2
7,Germany,2
8,Sweden,2
9,South Korea,3
10,North Korea,3
11,China,3
12,London,6
13,Birmingham,6
];
```

*SemanticTable*:

```
Semantic Load
  ID as ID,
  'Parent' as Relation,
```

## 2 Scriptinstructies en sleutelwoorden

---

```
ParentID as ID,  
'Child' as Relation  
resident GeographyTree;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- Id
- Geography

Maak vervolgens een filtervak met `relation` als dimensie. Klik op **Klaar met bewerken**.

Resultatentabel

Id	Geografie
1	Wereld
2	Europa
3	Azië
4	Noord-Amerika
5	Zuid-Amerika
6	VK
7	Germany
8	Sweden
9	Zuid-Korea
10	Noord-Korea
11	China
12	Londen
13	Birmingham

Filtervak

### Relatie

Onderliggend

Bovenliggend

Klik op **Europa** vanuit de dimensie `Geography` in de tabel en klik op **Onderliggend** vanuit de dimensie `relation` in het filtervak. Noteer het verwachte resultaat in de tabel:



Resultatentabel met  
'onderliggende  
objecten' van Europa

Id	Geografie
6	VK
7	Germany
8	Sweden

Als u nogmaals op **Onderliggend** klikt, worden plaatsen weergegeven die 'onderliggende objecten' van het VK zijn, een stap verder.

Resultatentabel met  
'onderliggende objecten'  
van het VK

Id	Geografie
12	Londen
13	Birmingham

### Unless

Met het prefix en suffix **unless** maakt u een conditionele clausule waarmee wordt bepaald of een opdracht of exit-clausule wel of niet moet worden geëvalueerd. De opdracht kan worden beschouwd als een compact alternatief voor een volledige **if..end if**-opdracht.

#### Syntaxis:

```
(Unless condition statement | exitstatement Unless condition )
```

De **statement** of de **exitstatement** wordt alleen uitgevoerd als de **condition** resulteert in False.

Het prefix **unless** kan worden gebruikt bij opdrachten die al een of meer andere opdrachten bevatten, inclusief aanvullende prefixen **when** of **unless**.

Argumenten

Argument	Beschrijving
condition	Een logische uitdrukking die resulteert in True of False.
statement	Elke Qlik Sense-scriptopdracht behalve besturingsopdrachten.
exitstatement	Een <b>exit for-</b> , <b>exit do-</b> of <b>exit sub-</b> clausule of een <b>exit script</b> -opdracht.

### Wanneer gebruiken

De `unless`-opdracht retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als een voorwaarde wanneer de gebruiker delen van het script voorwaardelijk wil laden of uitsluiten.

De volgende regels tonen drie voorbeelden van hoe de `unless`-functie kan worden gebruikt:

```
exit script unless A=1;
```

```
unless A=1 LOAD * from myfile.csv;
unless A=1 when B=2 drop table Tab1;
```

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Unless-prefix

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Het maken van variabele A, met een waarde van 1.
- Een gegevensset die in de tabel Transacties wordt geladen, tenzij de variabele A = 2.

#### Load-script

```
LET A = 1;

UNLESS A = 2

Transactions:
LOAD
*
Inline [
id, date, amount
1, 08/30/2018, 23.56
2, 09/07/2018, 556.31
3, 09/16/2018, 5.75
4, 09/22/2018, 125.00
5, 09/22/2018, 484.21
6, 09/22/2018, 59.18
7, 09/23/2018, 177.42
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- amount

Resultatentabel

id	date	amount
1	08/30/2018	23.56
2	09/07/2018	556.31
3	09/16/2018	5.75
4	09/22/2018	125.00
5	09/22/2018	484.21
6	09/22/2018	59.18
7	09/23/2018	177.42

Omdat variabele A is toegewezen aan de waarde 1 aan het begin van het script, wordt de voorwaarde na het `unless`-prefix beoordeeld en wordt het resultaat `FALSE` geretourneerd. Het resultaat is dat het script de Load-opdracht blijft uitvoeren. In de resultatentabel worden alle records uit de `Transactions`-tabel weergegeven.

Als deze variabelewaarde is ingesteld op gelijk aan 2, dan worden er geen gegevens in het gegevensmodel geladen.

### Voorbeeld 2 - Unless-suffix

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script begint met het laden van een eerste gegevensset in de tabel `Transactions`. Het script wordt vervolgens beëindigd tenzij er minder dan 10 records in de `Transactions`-tabel staan.

Als deze voorwaarde niet resulteert in de beëindiging van het script, wordt er nog een set met transacties aaneengeschakeld in de `Transactions`-tabel en wordt dit proces herhaald.

#### Load-script

```
Transactions:  
LOAD  
*
```

## 2 Scriptinstructies en sleutelwoorden

---

```
Inline [  
id, date, amount  
1, 08/30/2018, 23.56  
2, 09/07/2018, 556.31  
3, 09/16/2018, 5.75  
4, 09/22/2018, 125.00  
5, 09/22/2018, 484.21  
6, 09/22/2018, 59.18  
7, 09/23/2018, 177.42  
];
```

```
exit script unless NoOfRows('Transactions') < 10 ;
```

```
Concatenate  
LOAD  
*  
Inline [  
id, date, amount  
8, 10/01/2018, 164.27  
9, 10/03/2018, 384.00  
10, 10/06/2018, 25.82  
11, 10/09/2018, 312.00  
12, 10/15/2018, 4.56  
13, 10/16/2018, 90.24  
14, 10/18/2018, 19.32  
];
```

```
exit script unless NoOfRows('Transactions') < 10 ;
```

```
Concatenate  
LOAD  
*  
Inline [  
id, date, amount  
15, 10/01/2018, 164.27  
16, 10/03/2018, 384.00  
17, 10/06/2018, 25.82  
18, 10/09/2018, 312.00  
19, 10/15/2018, 4.56  
20, 10/16/2018, 90.24  
21, 10/18/2018, 19.32  
];
```

```
exit script unless NoOfRows('Transactions') < 10 ;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- amount

Resultatentabel

id	date	amount
1	08/30/2018	23.56
2	09/07/2018	556.31
3	09/16/2018	5.75
4	09/22/2018	125.00
5	09/22/2018	484.21
6	09/22/2018	59.18
7	09/23/2018	177.42
8	10/01/2018	164.27
9	10/03/2018	384.00
10	10/06/2018	25.82
11	10/09/2018	312.00
12	10/15/2018	4.56
13	10/16/2018	90.24
14	10/18/2018	19.32

Er zijn zeven records in elk van de drie gegevenssets van het load-script.

De eerste gegevensset (met transactie id 1 t/m 7) wordt in de applicatie geladen. De voorwaarde `unless` controleert of er minder dan 10 rijen in de tabel `Transactions` staan. Dit resulteert in `TRUE` en daarom wordt de tweede gegevensset (met transactie id 8 t/m 14) in de applicatie geladen. De tweede voorwaarde `unless` controleert of er minder dan 10 records in de tabel `Transactions` staan. Dit resulteert in `FALSE` en dus wordt het script beëindigd.

### Voorbeeld 3 - Meerdere Unless-prefixes

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

In dit voorbeeld wordt een gegevensset met één transactie gemaakt als de tabel `Transactions`. Er wordt een `for-lus` geactiveerd waarin twee geneste `unless`-opdrachten het volgende controleren:

1. Tenzij (`unless`) er meer dan 100 records in de `Transactions`-tabel staan
2. Tenzij (`unless`) het aantal records in de `Transactions`-tabel een veelvoud is van 6

## 2 Scriptinstructies en sleutelwoorden

---

Als deze voorwaarden `FALSE` zijn, worden er nog eens zeven records gegenereerd en aaneengeschakeld met de bestaande `Transactions`-tabel. Dit proces wordt herhaald tot een van de twee transacties de waarde `TRUE` retourneert.

### Load-script

```
Transactions:
Load
    0 as id
Autogenerate 1;

For i = 1 to 100
    unless NoOfRows('Transactions') > 100 unless mod(NoOfRows('Transactions'),6) = 0
        Concatenate
            Load
if(isnull(Peek(id)),1,peek(id)+1) as id
                Autogenerate 7;
    next i
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `id`.

Resultatentab

id
0
1
2
3
4
5
Nog 30 rijen

De geneste `unless`-opdrachten die optreden in de `for`-lus controleren het volgende:

1. Staan er meer dan 100 rijen in de `Transactions`-tabel?
2. Is het totale aantal records in de `Transactions`-tabel een veelvoud van 6?

Wanneer beide `unless`-opdrachten een waarde van `FALSE` retourneren, worden er nog eens zeven records gegenereerd en aaneengeschakeld met de bestaande `Transactions`-tabel.

Deze opdrachten retourneren vijf keer een waarde van `FALSE` waarna er in totaal 36 rijen gegevens in de `Transactions`-tabel staan.

Hierna retourneert de tweede `unless`-opdracht een waarde van `TRUE`. Daarom wordt de `load`-instructie hierna niet meer uitgevoerd.

### When

Met het prefix en suffix **when** maakt u een conditionele clausule waarmee wordt bepaald of een opdracht of exit-clausule wel of niet moet worden uitgevoerd. De opdracht kan worden beschouwd als een compact alternatief voor een volledige **if..end if**-opdracht.

#### Syntaxis:

```
(when condition statement | exitstatement when condition )
```

#### Retourgegevenstypen: Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

De **statement** of de **exitstatement** wordt alleen uitgevoerd als de voorwaarde resulteert in TRUE.

Het prefix **when** kan worden gebruikt bij opdrachten die al een of meer andere opdrachten bevatten, inclusief aanvullende prefixen **when** of **unless**.

#### Wanneer gebruiken

De **when**-opdracht retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als een voorwaarde wanneer de gebruiker delen van een script wil laden of uitsluiten.

#### Argumenten

Argument	Beschrijving
condition	Een logische uitdrukking die resulteert in TRUE of FALSE
statement	Elke Qlik Sense-scriptopdracht behalve besturingsopdrachten.
exitstatement	Een <b>exit for-</b> , <b>exit do-</b> of <b>exit sub-</b> clausule of een <b>exit script</b> -opdracht.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## 2 Scriptinstructies en sleutelwoorden

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>exit script when A=1;</code>	Wanneer de uitdrukking <code>A=1</code> resulteert in <code>TRUE</code> , wordt het script beëindigd.
<code>when A=1 LOAD * from myfile.csv;</code>	Wanneer de uitdrukking <code>A=1</code> resulteert in <code>TRUE</code> , wordt het bestand <code>myfile.csv</code> geladen.
<code>when A=1 unless B=2 drop table Tab1;</code>	Wanneer de opdracht <code>A=1</code> resulteert in <code>TRUE</code> en als <code>B=2</code> resulteert in <code>FALSE</code> , dan wordt de <code>Tab1</code> -tabel verwijderd.

### Voorbeeld 1 - When-prefix

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums en bedragen die naar de tabel `Transactions` wordt gestuurd.
- De `Let`-opdracht die aangeeft dat de variabele `A` is gemaakt en de waarde `1` heeft.
- De `when`-voorwaarde levert de voorwaarde dat als `A` gelijk is aan `1`, het script zal blijven laden.

#### Load-script

```
LET A = 1;

WHEN A = 1

Transactions:
LOAD
*
Inline [
id, date, amount
1, 08/30/2018, 23.56
2, 09/07/2018, 556.31
3, 09/16/2018, 5.75
4, 09/22/2018, 125.00
5, 09/22/2018, 484.21
6, 09/22/2018, 59.18
7, 09/23/2018, 177.42
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`



- amount

Resultatentabel

id	date	amount
1	08/30/2018	23.56
2	09/07/2018	556.31
3	09/16/2018	5.75
4	09/22/2018	125.00
5	09/22/2018	484.21
6	09/22/2018	59.18
7	09/23/2018	177.42

Omdat variabele `A` is toegewezen aan de waarde `1` aan het begin van het script, wordt de voorwaarde na het `when`-prefix beoordeeld en wordt het resultaat `TRUE` geretourneerd. Vanwege het resultaat `TRUE`, blijft het script de `load`-instructie uitvoeren. Alle records uit de resultatentabel worden weergegeven.

Als deze variabelewaarde was ingesteld op niet-gelijk aan `1`, dan zouden er geen gegevens in het gegevensmodel worden geladen.

### Voorbeeld 2 - When-suffix

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Drie gegevenssets met datums en bedragen die naar de tabel `Transactions` worden gestuurd.
  - De eerste gegevensset bevat transacties 1-7.
  - De tweede gegevensset bevat transacties 8-14.
  - De derde gegevensset bevat transacties 15-21.
- Een `when`-voorwaarde die bepaalt of de tabel `Transactions` meer dan tien rijen bevat. Als een van de `when`-opdrachten resulteren in `TRUE`, wordt het load-script beëindigd. Deze voorwaarde wordt aan het eind van elk van de drie gegevenssets geplaatst.

#### Load-script

```
Transactions:
LOAD
*
Inline [
id, date, amount
1, 08/30/2018, 23.56
2, 09/07/2018, 556.31
```

## 2 Scriptinstructies en sleutelwoorden

---

```
3, 09/16/2018, 5.75
4, 09/22/2018, 125.00
5, 09/22/2018, 484.21
6, 09/22/2018, 59.18
7, 09/23/2018, 177.42
];
```

```
exit script when NoOfRows('Transactions') > 10 ;
```

```
Concatenate
LOAD
*
Inline [
id, date, amount
8, 10/01/2018, 164.27
9, 10/03/2018, 384.00
10, 10/06/2018, 25.82
11, 10/09/2018, 312.00
12, 10/15/2018, 4.56
13, 10/16/2018, 90.24
14, 10/18/2018, 19.32
];
```

```
exit script when NoOfRows('Transactions') > 10 ;
```

```
Concatenate
LOAD
*
Inline [
id, date, amount
15, 10/01/2018, 164.27
16, 10/03/2018, 384.00
17, 10/06/2018, 25.82
18, 10/09/2018, 312.00
19, 10/15/2018, 4.56
20, 10/16/2018, 90.24
21, 10/18/2018, 19.32
];
```

```
exit script when NoOfRows('Transactions') > 10 ;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- amount

Resultatentabel

id	date	amount
1	08/30/2018	23.56
2	09/07/2018	556.31
3	09/16/2018	5.75
4	09/22/2018	125.00
5	09/22/2018	484.21
6	09/22/2018	59.18
7	09/23/2018	177.42
8	10/01/2018	164.27
9	10/03/2018	384.00
10	10/06/2018	25.82
11	10/09/2018	312.00
12	10/15/2018	4.56
13	10/16/2018	90.24
14	10/18/2018	19.32

Elk van de drie gegevenssets bevat zeven transacties. De eerste gegevensset bevat transacties 1 t/m 7 en wordt in de applicatie geladen. De `when`-voorwaarde na deze load-instructie resulteert in `FALSE` omdat er minder dan tien rijen in de `Transactions`-tabel staan. Het load-script gaat verder naar de volgende gegevensset.

De tweede gegevensset bevat transacties 8 t/m 14 en wordt in de applicatie geladen. De tweede `when`-voorwaarde resulteert in `TRUE` omdat er meer dan tien rijen in de `Transactions`-tabel staan. Daarom wordt het script beëindigd.

### Voorbeeld 3 - Meerdere `When`-prefixes

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met één transactie wordt gemaakt als de tabel `Transactions`.
- Er wordt een `For`-lus geactiveerd die twee geneste `when`-voorwaarden bevat die resulteren in een van de volgende opties:

## 2 Scriptinstructies en sleutelwoorden

---

1. Er staan minder dan 100 records in de Transactions-tabel.
2. Het aantal records in de Transactions-tabel is geen veelvoud van 6.

### Load-script

```
Transactions:
Load
    0 as id
Autogenerate 1;

For i = 1 to 100
    when NoOfRows('Transactions') < 100 when mod(NoOfRows('Transactions'),6) <> 0
        Concatenate
            Load
                if(isnull(Peek(id)),1,peek(id)+1) as id
            Autogenerate 7;
next i
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- id

De resultatentabel toont alleen de eerste vijf transactie-id's, maar het load-script maakt 36 rijen en wordt vervolgens beëindigd zodra aan de when-voorwaarde is voldaan.

Resultatentabel

id
0
1
2
3
4
5
Nog 30 rijen

De geneste when-voorwaarden in de For-lus resulteren in de volgende vragen:

- Staan er minder dan 100 rijen in de Transactions-tabel?
- Is het totale aantal records in de Transactions-tabel geen veelvoud van 6?

Wanneer beide when-voorwaarden een waarde van TRUE retourneren, worden er nog eens zeven records gegenereerd en aaneengeschakeld met de bestaande Transactions-tabel.

De when-voorwaarden retourneren vijf keer de waarde TRUE. Als dat gebeurt, zijn er in totaal 36 rijen gegevens in de Transactions-tabel.

Wanneer 36 rijen gegevens worden gemaakt in de Transactions-tabel, retourneert de tweede when-opdracht de waarde FALSE en wordt de load-instructie hierna niet meer uitgevoerd.

### 2.5 Reguliere scriptopdrachten

Normale opdrachten worden over het algemeen gebruikt om gegevens op een bepaalde manier te manipuleren. Deze opdrachten kunnen worden geschreven op een willekeurig aantal regels in het script en moeten altijd worden afgesloten met een puntkomma (;).

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

#### Overzicht van normale scriptopdrachten

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

##### Alias

De opdracht **alias** wordt gebruikt om een alias in te stellen. Dit betekent dat een veld een andere naam krijgt als het voorkomt in het script dat volgt.

```
Alias fieldname as aliasname {,fieldname as aliasname}
```

##### Autonumber

Deze instructie genereert een uniek geheel getal voor elke afzonderlijk geëvalueerde waarde in een veld die tijdens de uitvoering van een script wordt aangetroffen.

```
AutoNumber fields [Using namespace] ]
```

##### Binary

De opdracht **binary** wordt gebruikt voor het laden van de gegevens vanuit een ander QlikView-document, met inbegrip van sectietoegangsgegevens.

```
Binary [path] filename
```

##### comment

Hiermee kunt u de opmerkingen voor velden (metagegevens) uit databases en spreadsheets weergeven. Veldnamen die niet in de app aanwezig zijn, worden genegeerd. Als meerdere exemplaren van een veldnaam worden aangetroffen, wordt de laatste waarde gebruikt.

```
Comment field *fieldlist using mapname
```

```
Comment field fieldname with comment
```

##### comment table

Hiermee kunt u de opmerkingen voor tabellen (metagegevens) uit databases of spreadsheets weergeven.

```
Comment table tablelist using mapname
```

## 2 Scriptinstructies en sleutelwoorden

```
Comment table tablename with comment
```

### Connect



*Deze functionaliteit is niet beschikbaar in Qlik Sense SaaS.*

Met de opdracht **CONNECT** wordt de toegang van Qlik Sense tot een algemene database via de OLE DB/ODBC-interface vastgelegd. Voor ODBC moet de gegevensbron eerst worden opgegeven met behulp van ODBC-beheer.

```
ODBC Connect TO connect-string [ ( access_info ) ]
OLEDB CONNECT TO connect-string [ ( access_info ) ]
CUSTOM CONNECT TO connect-string [ ( access_info ) ]
LIB CONNECT TO connection
```

### Declare

De opdracht **Declare** wordt gebruikt voor het maken van velddefinities, waarbij u relaties kunt definiëren tussen velden of functies. Een reeks van velddefinities kan worden gebruikt om automatisch afgeleide velden te genereren, die kunnen worden gebruikt als dimensies. Zo kunt u bijvoorbeeld een agendadefinitie maken en die gebruiken om gerelateerde dimensies te genereren, zoals jaar, maand, week en dag, vanuit een datumveld.

```
definition_name:
Declare [Field[s]] Definition [Tagged tag_list ]
[Parameters parameter_list ]
Fields field_list
[Groups group_list ]

<definition name>:
Declare [Field][s] Definition
Using <existing_definition>
[With <parameter_assignment> ]
```

### Derive

De opdracht **Derive** wordt gebruikt voor het genereren van afgeleide velden op basis van een velddefinitie die is gemaakt met een opdracht **Declare**. U kunt opgeven voor welke gegevensvelden velden moeten worden afgeleid of u kunt deze expliciet of impliciet afleiden op basis van veldtags.

```
Derive [Field[s]] From [Field[s]] field_list Using definition
Derive [Field[s]] From Explicit [Tag[s]] (tag_list) Using definition
Derive [Field[s]] From Implicit [Tag[s]] Using definition
```

### Directory

De opdracht **Directory** legt vast in welke directory moet worden gezocht naar gegevensbestanden in volgende **LOAD**-opdrachten, totdat een nieuwe **Directory**opdracht wordt gegeven.

```
Directory [path]
```

## 2 Scriptinstructies en sleutelwoorden

### Disconnect

De opdracht **Disconnect** verbreekt de huidige ODBC-/OLE DB-/aangepaste verbinding. Deze opdracht is optioneel.

```
Disconnect
```

### drop field

Een of meer Qlik Sense-velden kunnen op elk moment tijdens de uitvoering van een script worden verwijderd uit het gegevensmodel, en dus uit het geheugen, met behulp van de opdracht **drop field**.



Zowel **drop field** als **drop fields** zijn toegestane vormen en hebben hetzelfde effect. Als geen tabel wordt opgegeven, wordt het veld uit alle tabellen waarin het voorkomt verwijderd.

```
Drop field fieldname [ , fieldname2 ...] [from tablename1 [ , tablename2 ...]]
drop fields fieldname [ , fieldname2 ...] [from tablename1 [ , tablename2 ...]]
```

### drop table

Een of meer interne tabellen in Qlik Sense kunnen op elk moment tijdens de uitvoering van een script worden verwijderd uit het gegevensmodel, en dus uit het geheugen, met behulp van de opdracht **drop table**.



De vormen **drop table** en **drop tables** worden beide geaccepteerd.

```
Drop table tablename [ , tablename2 ...]
drop tables [ tablename [ , tablename2 ...]]
```

### Execute

De **Execute**-opdracht wordt gebruikt voor het uitvoeren van andere programma's terwijl gegevens worden geladen in Qlik Sense. Bijvoorbeeld voor het uitvoeren van noodzakelijke conversies.

```
Execute commandline
```

### FlushLog

Met de opdracht **FlushLog** dwingt u af dat Qlik Sense de inhoud van de scriptbuffer naar het scriptlogbestand schrijft.

```
FlushLog
```

### Force

Met de opdracht **force** kunt u afdwingen dat Qlik Sense veldnamen en veldwaarden van de daaropvolgende **LOAD**- en **SELECT**-opdrachten interpreteert als geschreven met alleen hoofdletters, met alleen kleine letters, met beginhoofdletters of zoals ze worden weergegeven (gemengd). Met deze opdracht kunt u veldwaarden van tabellen interpreteren volgens verschillende conventies.

```
Force ( capitalization | case upper | case lower | case mixed )
```

### LOAD

Met de opdracht **LOAD** worden velden geladen uit een bestand, uit gegevens die in het script zijn gedefinieerd, uit een eerder geladen tabel, van een webpagina, uit het resultaat van een daaropvolgende **SELECT**-opdracht of door gegevens automatisch te genereren. Het is ook mogelijk om gegevens via analytische verbindingen op te halen.

```
Load [ distinct ] *fieldlist  
[( from file [ format-spec ] |  
from_field fieldsource [format-spec]  
inline data [ format-spec ] |  
resident table-label |  
autogenerate size )]  
[ where criterion | while criterion ]  
[ group_by groupbyfieldlist ]  
[order_by orderbyfieldlist ]  
[extension pluginname.functionname (tabledescription) ]
```

### Let

De opdracht **let** is een aanvulling op de opdracht **set** voor het definiëren van scriptvariabelen. In tegenstelling tot de opdracht **set** evalueert de opdracht **let** de uitdrukking rechts van '=' tijdens runtime van het script voordat deze wordt toegewezen aan de variabele.

```
Let variablename=expression
```

### Loosen Table

Een of meer interne gegevenstabellen van Qlik Sense kunt u tijdens de uitvoering van het script expliciet declareren als losjes gekoppeld met behulp van de opdracht **Loosen Table**. Als een tabel losjes is gekoppeld, zijn alle associaties tussen veldwaarden in de tabel verwijderd. Een vergelijkbaar effect kan worden verkregen door elk veld van de losjes gekoppelde tabel als onafhankelijke, niet-verbonden tabel te laden. Losjes gekoppeld kan handig zijn tijdens het testen om tijdelijk verschillende onderdelen van de gegevensstructuur te isoleren. Een losjes gekoppelde tabel kan worden geïdentificeerd in de tabelweergave aan de hand van de stippellijnen. Het gebruik van een of meer **Loosen Table**-opdrachten in het script zorgt dat Qlik Sense eventuele instellingen negeert voor losjes gekoppelde tabellen voordat het script wordt uitgevoerd.

```
tablename [ , tablename2 ...]  
Loosen Tables tablename [ , tablename2 ...]
```

### Map ... using

De opdracht **map ... using** wordt gebruikt voor de toewijzing van een bepaalde veldwaarde of uitdrukking aan de waarden van een specifieke toewijzingstabel. De toewijzingstabel wordt gemaakt met behulp van de opdracht **Mapping**.

```
Map *fieldlist Using mapname
```

### NullAsNull

De opdracht **NullAsNull** schakelt de conversie van NULL-waarden naar tekenreekswaarden uit die eerder is ingesteld via een **NullAsValue**-opdracht.



## 2 Scriptinstructies en sleutelwoorden

---

```
NullAsNull *fieldlist
```

### NullAsValue

De opdracht **NullAsValue** legt vast voor welke velden NULL moet worden omgezet in een waarde.

```
NullAsValue *fieldlist
```

### Qualify

De opdracht **Qualify** wordt gebruikt om de kwalificatie van veldnamen in te schakelen. Dit houdt in dat veldnamen de tabelnaam als prefix krijgen.

```
Qualify *fieldlist
```

### Rem

Met de opdracht **rem** kunt u opmerkingen of commentaar aan het script toevoegen, of scriptopdrachten tijdelijk deactiveren zonder ze te verwijderen.

```
Rem string
```

### Rename Field

Met deze scriptfunctie wordt de naam van een of meer bestaande Qlik Sense-velden gewijzigd nadat deze zijn geladen.

```
Rename field (using mapname | oldname to newname{ , oldname to newname })
```

```
Rename Fields (using mapname | oldname to newname{ , oldname to newname })
```

### Rename Table

Met deze scriptfunctie wordt de naam van een of meer bestaande interne Qlik Sense-tabellen gewijzigd nadat deze zijn geladen.

```
Rename table (using mapname | oldname to newname{ , oldname to newname })
```

```
Rename Tables (using mapname | oldname to newname{ , oldname to newname })
```

### Section

Met de opdracht **section** kan worden aangegeven of de daarop volgende **LOAD**- en **SELECT**-opdrachten moeten worden beschouwd als gegevens of als een definitie van de toegangsrechten.

```
Section (access | application)
```

### Select

U selecteert velden uit een ODBC-gegevensbron of van een OLE DB-provider met de standaard **SELECT**-opdrachten van SQL. Of de **SELECT**-opdrachten worden geaccepteerd, hangt af van het gebruikte ODBC-stuurprogramma of de gebruikte OLE DB-provider.

```
Select [all | distinct | distinctrow | top n [percent] ] *fieldlist
```

```
From tablelist
```

```
[Where criterion ]
```

```
[Group by fieldlist [having criterion ] ]
```

## 2 Scriptinstructies en sleutelwoorden

---

```
[Order by fieldlist [asc | desc] ]  
[ (Inner | Left | Right | Full)Join tablename on fieldref = fieldref ]
```

### Set

Met de opdracht **set** legt u scriptvariabelen vast. Deze kunnen worden gebruikt om tekenreeksen, paden, stations en dergelijke te vervangen.

```
Set variablename=string
```

### Sleep

Met de opdracht **sleep** onderbreekt u de uitvoering van een script voor een bepaalde tijd.

```
Sleep n
```

### SQL

Met de opdracht **SQL** kunt u via een SQL- of ODBC-verbinding een willekeurige OLE DB-opdracht verzenden.

```
SQL sql_command
```

### SQLColumns

Met de opdracht **sqlcolumns** wordt een set velden geretourneerd waarin de kolommen worden beschreven van een ODBC- of OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

```
SQLColumns
```

### SQLTables

Met de opdracht **sqltables** wordt een set velden geretourneerd waarin de tabellen worden beschreven van een ODBC- of OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

```
SQLTables
```

### SQLTypes

Met de opdracht **sqltypes** wordt een set velden geretourneerd waarin de typen worden beschreven van een ODBC- of OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

```
SQLTypes
```

### Star

Met de opdracht **star** kunt u een tekenreeks opgeven die de set van alle waarden van een veld in een database vertegenwoordigt. De opdracht beïnvloedt de **LOAD**- en **SELECT**-opdrachten die volgen.

```
Star is [ string ]
```

### Store

De **Store**-instructie maakt een QVD-, CSV- of text-bestand.

```
Store [ *fieldlist from] table into filename [ format-spec ];
```

### Tag

Met de scriptopdracht kunnen tags aan één of meer velden of tabellen worden toegewezen. Als wordt geprobeert om een tag toe te wijzen aan een veld of tabel die niet bestaat in de app, wordt de tag genegeerd. Als conflicterende exemplaren van een veld- of tagnaam zijn aangetroffen, wordt de laatste waarde gebruikt.

```
Tag[field|fields] fieldlist with tagname
Tag [field|fields] fieldlist using mapname
Tag table tablelist with tagname
```

### Trace

Met de opdracht **trace** wordt eventueel een tekenreeks naar het venster **Voortgang scriptuitvoering** en in het scriptlogbestand geschreven. Dit is zeer handig bij het opsporen van fouten. Met behulp van \$-uitbreidingen van variabelen die zijn berekend vóór de **trace**-opdracht, kunt u het bericht aanpassen.

```
Trace string
```

### Unmap

Met de opdracht **Unmap** wordt de toewijzing van veldwaarden die is vastgelegd door een eerdere opdracht **Map ... Using**, uitgeschakeld voor velden die daarna worden geladen.

```
Unmap *fieldlist
```

### Unqualify

Met de opdracht **Unqualify** wordt de kwalificatie van veldnamen die eerder was ingeschakeld met de opdracht **Qualify**, uitgeschakeld.

```
Unqualify *fieldlist
```

### Untag

Met de scriptopdracht kunnen tags van één of meer velden of tabellen worden verwijderd. Als wordt geprobeert om een tag te verwijderen van een veld of tabel die niet bestaat in de app, wordt de tag niet verwijderd.

```
Untag[field|fields] fieldlist with tagname
Tag [field|fields] fieldlist using mapname
Tag table tablelist with tagname
```

### Alias

De opdracht **alias** wordt gebruikt om een alias in te stellen. Dit betekent dat een veld een andere naam krijgt als het voorkomt in het script dat volgt.

#### Syntaxis:

```
alias fieldname as aliasname {,fieldname as aliasname}
```

### Argumenten:

Argumenten

Argument	Beschrijving
fieldname	De naam van het veld in de brongegevens
aliasname	Een aliasnaam die u ervoor in de plaats wilt gebruiken

### Voorbeelden en resultaten:

Voorbeeld	Resultaat
Alias ID_N as NameID;	
Alias A as Name, B as Number, C as Date;	De naamswijzigingen die via deze opdracht worden gedefinieerd, worden gebruikt in alle volgende <b>SELECT</b> - en <b>LOAD</b> -opdrachten. Een nieuwe alias voor een veldnaam kan worden gedefinieerd met een nieuwe <b>alias</b> -opdracht verderop in het script.

## AutoNumber

Deze instructie genereert een uniek geheel getal voor elke afzonderlijk geëvalueerde waarde in een veld die tijdens de uitvoering van een script wordt aangetroffen.

Het is ook mogelijk om de functie *autonumber* (page 574) te gebruiken in een **LOAD**-instructie, maar dit heeft enige beperkingen als u een geoptimaliseerde lading wilt gebruiken. U kunt een geoptimaliseerde lading maken door de gegevens eerst op te halen vanuit een **QVD**-bestand. Vervolgens kunt u de instructie **AutoNumber** gebruiken om de waarden naar symbolleutels te converteren.

### Syntaxis:

```
AutoNumber *fieldlist [Using namespace] ]
```

### Argumenten:

Argumenten

Argument	Beschrijving
*fieldlist	Een door komma's gescheiden lijst met de velden waarin de waarden moeten worden vervangen door een uniek geheel getal.  U kunt de jokertekens ? en * in de veldnamen gebruiken om alle velden met een overeenkomstige naam toe te voegen. U kunt ook * gebruiken om alle velden toe te voegen. U moet de veldnamen tussen aanhalingstekens plaatsen als u jokertekens gebruikt.

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
namespace	<b>Gebruik van namespace</b> is optioneel. U kunt deze optie gebruiken als u een naamruimte wilt maken waarin identieke waarden in verschillende velden dezelfde sleutel delen.  Als u deze optie niet gebruikt, wordt voor alle velden een afzonderlijke sleutelindex gebruikt.

### Beperkingen:

Als u verschillende **LOAD**-instructies gebruikt in het script, moet de **AutoNumber**-instructie achter de laatste **LOAD**-instructie worden geplaatst.

Voorbeeld - script met AutoNumber

### Scriptvoorbeeld

In dit voorbeeld worden de gegevens eerst geladen zonder de **AutoNumber**-instructie. Vervolgens wordt de **AutoNumber**-instructie toegevoegd om het effect ervan te laten zien.

### Gebruikte gegevens in voorbeeld

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om het onderstaande scriptvoorbeeld te maken. Laat de **AutoNumber**-instructie eerst uitgeschakeld in de vorm van een commentaarregel.

```
RegionSales: LOAD *, Region &'|'& Year &'|'& Month as KeyToOtherTable INLINE [ Region, Year,
Month, Sales North, 2014, May, 245 North, 2014, May, 347 North, 2014, June, 127 S
June, 645 South, 2013, May, 367 South, 2013, May, 221 ];
&'|'& Year &'|'& Month as KeyToOtherTable INLINE [Region, Year, Month, Budget North, 2014,
May, 200 North, 2014, May, 350 North, 2014, June, 150 South, 2014, June,
500 South, 2013, May, 300 South, 2013, May, 200 ]; //AutoNumber KeyToOtherTable;
```

### Visualisaties maken

Maak twee tabelvisualisaties in een Qlik Sense-werkblad. Voeg **KeyToOtherTable**, **Region**, **Year**, **Month** en **Sales** toe als dimensies aan de eerste tabel. Voeg **KeyToOtherTable**, **Region**, **Year**, **Month** en **Budget** toe als dimensies aan de tweede tabel.

### Resultaat

Tabel RegionSales

KeyToOtherTable	Region	Year	Month	Sales
North 2014 June	North	2014	June	127
North 2014 May	North	2014	May	245
North 2014 May	North	2014	May	347
South 2013 May	South	2013	May	221

## 2 Scriptinstructies en sleutelwoorden

KeyToOtherTable	Region	Year	Month	Sales
South 2013 May	South	2013	May	367
South 2014 June	South	2014	June	645

Tabel Budget

KeyToOtherTable	Region	Year	Month	Budget
North 2014 June	North	2014	June	150
North 2014 May	North	2014	May	200
North 2014 May	North	2014	May	350
South 2013 May	South	2013	May	200
South 2013 May	South	2013	May	300
South 2014 June	South	2014	June	500

### Uitleg

In het voorbeeld ziet u een composietveld **KeyToOtherTable** waarmee de twee tabellen worden gekoppeld. **AutoNumber** wordt niet gebruikt. Kijk naar de lengte van de **KeyToOtherTable**-waarden.

### AutoNumber-instructie toevoegen

Verwijder de commentaaraanduiding van de **AutoNumber**-instructie in het load-script.

```
AutoNumber KeyToOtherTable;
```

### Resultaat

Tabel RegionSales

KeyToOtherTable	Region	Year	Month	Sales
1	North	2014	June	127
1	North	2014	May	245
2	North	2014	May	347
3	South	2013	May	221
4	South	2013	May	367
4	South	2014	June	645

## 2 Scriptinstructies en sleutelwoorden

---

Tabel Budget

KeyToOtherTable	Region	Year	Month	Budget
1	North	2014	June	150
1	North	2014	May	200
2	North	2014	May	350
3	South	2013	May	200
4	South	2013	May	300
4	South	2014	June	500

### Uitleg

De **KeyToOtherTable**-veldwaarden zijn vervangen door unieke gehele getalswaarden en daardoor is de lengte van de veldwaarden gereduceerd, wat geheugen bespaart. **AutoNumber** heeft een effect op de sleutelvelden in beide tabellen en de tabellen blijven gekoppeld. Het voorbeeld is kort vanwege demonstratiedoeleinden, maar zou relevant zijn bij een tabel met een groot aantal rijen.

### Binary

De opdracht **binary** wordt gebruikt voor het laden van de gegevens vanuit een andere Qlik Sense-app of een QlikView-document, met inbegrip van sectietoegangsgegevens. Er zijn geen andere elementen van de app toegevoegd, zoals werkbladen, presentaties, visualisaties, masteritems of variabelen.

Slechts één **binary** opdracht is toegestaan in het script. De **binary** opdracht moet de eerste opdracht van het script zijn, zelfs nog vóór de SET-opdrachten die zich gewoonlijk aan het begin van het script bevinden.

### Syntaxis:

```
binary [path] filename
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
path	<p>Het pad naar het bestand, dat een verwijzing zou moeten zijn naar een mapgegevensverbinding. Dit is een vereiste als het bestand zich niet in de werkdirectory van Qlik Sense bevindt.</p> <p><b>Voorbeeld: 'lib://Table Files/'</b></p> <p>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:</p> <ul style="list-style-type: none"><li>• absoluut</li></ul> <p><b>Voorbeeld: c:\data</b></p> <ul style="list-style-type: none"><li>• relatief ten opzicht van de app die deze scriptregel bevat.</li></ul> <p><b>Voorbeeld: data</b></p>
filename	De naam van het bestand, inclusief de bestandsextensie .qvw of .qvf.

### Beperkingen:

U kunt **binary** niet gebruiken om gegevens te laden vanuit een app op dezelfde Qlik Sense Enterprise-implementatie door naar de app-ID te verwijzen. U kunt alleen laden vanuit een .qvf-bestand.

### Voorbeelden

Tekenreeks	Beschrijving
<code>Binary lib://DataFolder/customer.qvw;</code>	In dit voorbeeld moet het bestand zich in de <b>map</b> gegevensverbinding bevinden. Dit kan bijvoorbeeld een map zijn die uw beheerder op de Qlik Sense-server maakt. Klik op <b>Nieuwe verbinding maken</b> in de editor voor laden van gegevens en selecteer vervolgens <b>map</b> onder <b>Bestandslocaties</b> .
<code>Binary customer.qvf;</code>	In dit voorbeeld moet het bestand zich in de werkdirectory van Qlik Sense bevinden.
<code>Binary c:\qv\customer.qvw;</code>	Dit voorbeeld waarbij een absoluut bestandspad wordt gebruikt, werkt alleen in de bestaande scriptmodus.



### Comment field

Hiermee kunt u de opmerkingen voor velden (metagegevens) uit databases en spreadsheets weergeven. Veldnamen die niet in de app aanwezig zijn, worden genegeerd. Als meerdere exemplaren van een veldnaam worden aangetroffen, wordt de laatste waarde gebruikt.

#### Syntaxis:

```
comment [fields] *fieldlist using mapname
comment [field] fieldname with comment
```

De toewijzingstabel moet uit twee kolommen bestaan, de eerste met veldnamen en de tweede met de opmerkingen.

#### Argumenten:

##### Argumenten

Argument	Beschrijving
<i>*fieldlist</i>	Een met door komma's gescheiden lijst velden die van commentaar moeten worden voorzien. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens.
<i>mapname</i>	De naam van een toewijzingstabel die eerder is gelezen in een opdracht mapping <b>LOAD</b> of mapping <b>SELECT</b> .
<i>fieldname</i>	De naam van het veld waaraan de opmerking moet worden toegevoegd.
<i>comment</i>	De opmerking die aan het veld moet worden toegevoegd.

#### Example 1:

```
commentmap:
mapping LOAD * inline [
a,b
Alpha,This field contains text values
Num,This field contains numeric values
];
comment fields using commentmap;
```

#### Example 2:

```
comment field Alpha with AFieldContainingCharacters;
comment field Num with '*A field containing numbers';
comment Gamma with 'Mickey Mouse field';
```

### Comment table

Hiermee kunt u de opmerkingen voor tabellen (metagegevens) uit databases of spreadsheets weergeven.

## 2 Scriptinstructies en sleutelwoorden

De tabelnamen die niet in de app aanwezig zijn, worden genegeerd. Als meerdere exemplaren van een tabelnaam zijn aangetroffen, wordt de laatste waarde gebruikt. Met het trefwoord kunnen opmerkingen uit een gegevensbron worden gelezen.

### Syntaxis:

```
comment [tables] tablelist using mapname
comment [table] tablename with comment
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
<i>tablelist</i>	(table{,table})
<i>mapname</i>	De naam van een toewijzingstabel die eerder is gelezen in een opdracht mapping <b>LOAD</b> of mapping <b>SELECT</b> .
<i>tablename</i>	De naam van de tabel waaraan de opmerking moet worden toegevoegd.
<i>comment</i>	De opmerking die aan de tabel moet worden toegevoegd.

### Example 1:

```
Commentmap:
mapping LOAD * inline [
a,b
Main,This is the fact table
Currencies, Currency helper table
];
comment tables using Commentmap;
```

### Example 2:

```
comment table Main with 'Main fact table';
```

## Connect

Met de opdracht **CONNECT** wordt de toegang van Qlik Sense tot een algemene database via de OLE DB/ODBC-interface vastgelegd. Voor ODBC moet de gegevensbron eerst worden opgegeven met behulp van ODBC-beheer.



*Deze functionaliteit is niet beschikbaar in Qlik Sense SaaS.*



*Deze opdracht ondersteunt alleen mapgegevensverbindingen in de standaardmodus.*

### Syntaxis:

```
ODBC CONNECT TO connect-string
OLEDB CONNECT TO connect-string
```

## 2 Scriptinstructies en sleutelwoorden

```
CUSTOM CONNECT TO connect-string  
LIB CONNECT TO connection
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
connect-string	<p><code>connect-string ::= datasourcename { ; conn-spec-item }</code></p> <p>De verbindingstekenreeks bestaat uit de naam van de gegevensbron en een optionele lijst met een of meer specificaties voor de verbinding. Als er spaties voorkomen in de naam van de gegevensbron of als specificaties voor de verbinding worden opgegeven, moet de verbindingstekenreeks tussen aanhalingstekens staan.</p> <p><b>datasourcename</b> moet een gedefinieerde ODBC-gegevensbron zijn of een tekenreeks waarmee een OLE DB-provider wordt gedefinieerd.</p> <p><code>conn-spec-item ::= DBQ=database_specifier   DriverID=driver_specifier   UID=userid   PWD=password</code></p> <p>Welke specificaties voor de verbinding mogelijk zijn, is afhankelijk van de databases. Voor sommige databases zijn andere specificaties mogelijk dan de hierboven genoemde. Voor OLE DB zijn sommige van de verbindingsspecifieke items verplicht en niet optioneel.</p>
connection	<p>De naam van een gegevensverbinding die is opgeslagen in de editor voor het laden van gegevens.</p>

Als de **ODBC** wordt geplaatst voor **CONNECT**, wordt de ODBC-interface gebruikt, anders wordt OLE DB gebruikt.

Met **LIB CONNECT TO** wordt verbinding gemaakt met een database waarin een gegevensverbinding is opgeslagen die in de editor voor het laden van gegevens werd gemaakt.

### Example 1:

```
ODBC CONNECT TO 'Sales  
DBQ=C:\Program Files\Access\Samples\Sales.mdb';
```

De door deze opdracht gedefinieerde gegevensbron wordt gebruikt door daaropvolgende **Select (SQL)**-opdrachten, tot een nieuwe **CONNECT**-opdracht wordt gemaakt.

### Example 2:

```
LIB CONNECT TO 'DataConnection';
```

### Connect32

Deze opdracht wordt op dezelfde wijze gebruikt als de **CONNECT**-opdracht, alleen wordt een 64-bits systeem gedwongen om een 32-bits ODBC/OLE DB-provider te gebruiken. Niet van toepassing op custom connect.

### Connect64

Deze opdracht wordt op dezelfde wijze gebruikt als de **CONNECT**-opdracht, alleen wordt het gebruik van een 64-bits provider afgedwongen. Niet van toepassing op custom connect.

### Declare

De opdracht **Declare** wordt gebruikt voor het maken van velddefinities, waarbij u relaties kunt definiëren tussen velden of functies. Een reeks van velddefinities kan worden gebruikt om automatisch afgeleide velden te genereren, die kunnen worden gebruikt als dimensies. Zo kunt u bijvoorbeeld een agendadefinitie maken en die gebruiken om gerelateerde dimensies te genereren, zoals jaar, maand, week en dag, vanuit een datumveld.


U kunt **Declare** gebruiken om een nieuwe velddefinitie op te zetten of om een velddefinitie te maken op basis van een reeds bestaande definitie.

### Een nieuwe velddefinitie opzetten

#### Syntaxis:

```
definition_name:  
Declare [Field[s]] Definition [Tagged tag_list ]  
[Parameters parameter_list ]  
Fields field_list
```

#### Argumenten:

Argument	Beschrijving
definition_name	<p>Naam van de velddefinitie, afgesloten met een dubbele punt.</p> <div style="border: 1px solid gray; padding: 5px;"> <i>Maak geen gebruik van autoCalendar als naam voor velddefinities, aangezien deze naam is gereserveerd voor automatisch gegenereerde agendasjablonen.</i></div> <p><b>Voorbeeld:</b></p> <p>calendar:</p>
tag_list	<p>Een door komma's gescheiden lijst met tags om toe te passen op velden die zijn afgeleid van de velddefinitie. Het toepassen van tags is optioneel, maar als u geen tags toepast die worden gebruikt voor het specificeren van de sorteervolgorde, zoals \$date, \$numeric of \$text, wordt het afgeleide veld standaard gesorteerd op laadvolgorde.</p> <p><b>Voorbeeld:</b></p> <p>'\$date'Thank you for bringing this to our attention, and apologies for the inconvenience.</p>

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
parameter_list	<p>Een door komma's gescheiden lijst met parameters. Een parameter wordt gedefinieerd in de vorm <code>name=value</code> en krijgt een beginwaarde toegewezen, die kan worden vervangen als een velddefinitie opnieuw wordt gebruikt. Optioneel.</p> <p><b>Voorbeeld:</b></p> <pre>first_month_of_year = 1</pre>
field_list	<p>Een door komma's gescheiden lijst met velden die worden gegenereerd als de velddefinitie wordt gebruikt. Een veld wordt gedefinieerd in de vorm <code>&lt;expression&gt; As field_name tagged tag</code>. Gebruik <code>\$1</code> om te verwijzen naar het gegevensveld op basis waarvan de afgeleide velden moeten worden gegenereerd.</p> <p><b>Voorbeeld:</b></p> <pre>Year(\$1) As Year tagged ('\$numeric')</pre>

### Voorbeeld:

Calendar:

```
DECLARE FIELD DEFINITION TAGGED '$date'
  Parameters
    first_month_of_year = 1
  Fields
    Year($1) As Year Tagged ('$numeric'),
    Month($1) as Month Tagged ('$numeric'),
    Date($1) as Date Tagged ('$date'),
    week($1) as week Tagged ('$numeric'),
    weekday($1) as weekday Tagged ('$numeric'),
    DayNumberOfYear($1, first_month_of_year) as DayNumberOfYear Tagged ('$numeric')
;
```

De agenda is nu gedefinieerd en u kunt deze toepassen op de datumvelden die zijn geladen, in dit geval `OrderDate` en `ShippingDate`, met behulp van een clause **Derive**.

### Een bestaande velddefinitie opnieuw gebruiken

#### Syntaxis:

```
<definition name>:
Declare [Field][s] Definition
Using <existing_definition>
[With <parameter_assignment> ]
```

### Argumenten:

Argument	Beschrijving
definition_name	Naam van de velddefinitie, afgesloten met een dubbele punt.  <b>Voorbeeld:</b>  MyCalendar:
existing_definition	De velddefinitie die opnieuw moet worden gebruikt bij het maken van de nieuwe velddefinitie. De nieuwe velddefinitie functioneert op dezelfde wijze als de definitie waarop deze is gebaseerd, behalve als u parameter_assignment gebruikt om een waarde te wijzigen die wordt gebruikt in de velduitdrukkingen.  <b>Voorbeeld:</b>  Using Calendar
parameter_assignment	Een door komma's gescheiden lijst met parametertoewijzingen. Een parametertoewijzing wordt gedefinieerd in de vorm name=value en vervangt de parameterwaarde die is ingesteld in de basisvelddefinitie. Optioneel.  <b>Voorbeeld:</b>  first_month_of_year = 4

### Voorbeeld:

In dit voorbeeld gebruiken we de agendadefinitie opnieuw die we in het voorafgaande voorbeeld hebben gemaakt. In dit geval willen we een fiscaal jaar gebruiken dat begint in april. Dit wordt gedaan door de waarde 4 toe te wijzen aan de parameter first\_month\_of\_year, hetgeen van invloed is op het veld DayNumberOfYear dat is wordt gedefinieerd.

In het voorbeeld wordt ervan uitgegaan dat u de voorbeeldgegevens en velddefinitie uit het voorafgaande voorbeeld gebruikt.

```
MyCalendar:  
DECLARE FIELD DEFINITION USING Calendar WITH first_month_of_year=4;  
  
DERIVE FIELDS FROM FIELDS OrderDate,ShippingDate USING MyCalendar;
```

Als u het gegevensscript opnieuw hebt geladen, zijn de gegenereerde velden beschikbaar in de werkbladeditor, met de namen OrderDate.MyCalendar.\* en ShippingDate.MyCalendar.\*.

## Derive

De opdracht **Derive** wordt gebruikt voor het genereren van afgeleide velden op basis van een velddefinitie die is gemaakt met een opdracht **Declare**. U kunt opgeven voor welke gegevensvelden velden moeten worden afgeleid of u kunt deze expliciet of impliciet afleiden op basis van veldtags.

### Syntaxis:

```
Derive [Field[s]] From [Field[s]] field_list Using definition
Derive [Field[s]] From Explicit [Tag[s]] tag_list Using definition
Derive [Field[s]] From Implicit [Tag[s]] Using definition
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
definition	Naam van de velddefinitie die moet worden gebruikt bij het afleiden van velden.  <b>Voorbeeld:</b> calendar
field_list	Een door komma's gescheiden lijst met gegevensvelden waaruit de afgeleide velden moeten worden gegenereerd, op basis van de velddefinitie. De gegevensvelden moeten velden zijn die u al hebt geladen in het script.  <b>Voorbeeld:</b> orderdate, shippingdate
tag_list	Een door komma's gescheiden lijst met tags. Afgeleide velden worden gegenereerd voor alle gegevensvelden met enige van de tags in de lijst. De lijst met tags moet tussen ronde haakjes worden geplaatst.  <b>Voorbeeld:</b> ('\$date', '\$timestamp')

### Voorbeelden:

- Velden afleiden voor specifieke gegevensvelden.  
In dit geval geven we de velden OrderDate en ShippingDate op.  
DERIVE FIELDS FROM FIELDS OrderDate,ShippingDate USING Calendar;
- Velden afleiden voor alle velden met een specifieke tag.  
In dit geval leiden we velden af die zijn gebaseerd op Calendar voor alle velden met een tag \$date.  
DERIVE FIELDS FROM EXPLICIT TAGS ('\$date') USING Calendar;
- Velden afleiden voor alle velden met de velddefinitietag.  
In dit geval leiden we velden af voor alle gegevensvelden met dezelfde tag als de velddefinitie Calendar. In dit geval is dit \$date.  
DERIVE FIELDS FROM IMPLICIT TAG USING Calendar;

## Directory

De opdracht **Directory** legt vast in welke directory moet worden gezocht naar gegevensbestanden in volgende **LOAD**-opdrachten, totdat een nieuwe **Directory**opdracht wordt gegeven.

### Syntaxis:

```
Directory [path]
```

## 2 Scriptinstructies en sleutelwoorden

Als de opdracht **Directory** wordt opgegeven zonder een **path** of als de opdracht wordt weggelaten, zoekt Qlik Sense in de werkdirectory van Qlik Sense.

### Argumenten:

#### Argumenten

Argument	Beschrijving
<b>path</b>	<p>Een tekst die kan worden geïnterpreteerd als het pad naar het data-bestand.</p> <p>Het pad is het pad naar het bestand. Het is:</p> <ul style="list-style-type: none"><li>• absoluut <b>Voorbeeld: <i>c:\data</i></b></li><li>• relatief ten opzichte van de werkmap van de Qlik Sense-app. <b>Voorbeeld: <i>data</i></b></li><li>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet. <b>Voorbeeld: <i>http://www.qlik.com</i></b></li></ul>

### Voorbeelden:

```
DIRECTORY C:\userfiles\data; // OR -> DIRECTORY data\
```

```
LOAD * FROM  
[data1.csv] // ONLY THE FILE NAME CAN BE SPECIFIED HERE (WITHOUT THE FULL PATH)  
(ansi, txt, delimiter is ',', embedded labels);
```

```
LOAD * FROM  
[data2.txt] // ONLY THE FILE NAME CAN BE SPECIFIED HERE UNTIL A NEW DIRECTORY STATEMENT IS  
MADE  
(ansi, txt, delimiter is '\t', embedded labels);
```

## Disconnect

De opdracht **Disconnect** verbreekt de huidige ODBC-/OLE DB-/aangepaste verbinding. Deze opdracht is optioneel.

### Syntaxis:

```
Disconnect
```

De verbinding wordt automatisch afgebroken wanneer een nieuwe **connect**-opdracht wordt uitgevoerd of wanneer het script volledig is uitgevoerd.



### Voorbeeld:

```
Disconnect;
```

## Drop

Het sleutelwoord **Drop** voor scripts kan worden gebruikt om tabellen of velden uit de database te verwijderen.

### Drop field

Een of meer Qlik Sense-velden kunnen op elk moment tijdens de uitvoering van een script worden verwijderd uit het gegevensmodel, en dus uit het geheugen, met behulp van de opdracht **drop field**.



*Zowel **drop field** als **drop fields** zijn toegestane vormen en hebben hetzelfde effect. Als geen tabel wordt opgegeven, wordt het veld uit alle tabellen waarin het voorkomt verwijderd.*

### Syntaxis:

```
Drop field fieldname { , fieldname2 ...} [from tablename1 { , tablename2 ...}]  
Drop fields fieldname { , fieldname2 ...} [from tablename1 { , tablename2 ...}]
```

### Voorbeelden:

```
Drop field A;  
Drop fields A,B;  
Drop field A from X;  
Drop fields A,B from X,Y;
```

## Drop table

Een of meer interne tabellen in Qlik Sense kunnen op elk moment tijdens de uitvoering van een script worden verwijderd uit het gegevensmodel, en dus uit het geheugen, met behulp van de opdracht **drop table**.

### Syntaxis:

```
drop table tablename {, tablename2 ...}  
drop tables tablename {, tablename2 ...}
```



*De vormen **drop table** en **drop tables** worden beide geaccepteerd.*

De volgende items gaan als gevolg hiervan verloren:

- De feitelijke tabel(len).
- Alle velden die geen onderdeel vormen van de resterende tabellen.

## 2 Scriptinstructies en sleutelwoorden

- Veldwaarden in resterende velden die uitsluitend uit de te wissen tabel(len) afkomstig zijn.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
<pre>drop table Orders, Salesmen, T456a;</pre>	Deze regel verwijdert drie tabellen uit het geheugen.
<pre>Tab1: Load * Inline [ Customer, Items, UnitPrice Bob, 5, 1.50 ];  Tab2: LOAD Customer, Sum( Items * UnitPrice ) as Sales resident Tab1 group by Customer;  drop table Tab1;</pre>	Nadat de tabel <i>Tab2</i> is gemaakt, wordt de tabel <i>Tab1</i> verwijderd.

### Drop table

Een of meer interne tabellen in Qlik Sense kunnen op elk moment tijdens de uitvoering van een script worden verwijderd uit het gegevensmodel, en dus uit het geheugen, met behulp van de opdracht **drop table**.

#### Syntaxis:

```
drop table tablename {, tablename2 ...}  
drop tables tablename {, tablename2 ...}
```



*De vormen **drop table** en **drop tables** worden beide geaccepteerd.*

De volgende items gaan als gevolg hiervan verloren:

- De feitelijke tabel(len).
- Alle velden die geen onderdeel vormen van de resterende tabellen.
- Veldwaarden in resterende velden die uitsluitend uit de te wissen tabel(len) afkomstig zijn.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
<pre>drop table Orders, Salesmen, T456a;</pre>	Deze regel verwijdert drie tabellen uit het geheugen.

## 2 Scriptinstructies en sleutelwoorden

Voorbeeld	Resultaat
<pre>Tab1: Load * Inline [ Customer, Items, UnitPrice Bob, 5, 1.50 ];  Tab2: LOAD Customer, Sum( Items * UnitPrice ) as Sales resident Tab1 group by Customer;  drop table Tab1;</pre>	Nadat de tabel <i>Tab2</i> is gemaakt, wordt de tabel <i>Tab1</i> verwijderd.

### Execute

De **Execute**-opdracht wordt gebruikt voor het uitvoeren van andere programma's terwijl gegevens worden geladen in Qlik Sense. Bijvoorbeeld voor het uitvoeren van noodzakelijke conversies.



*Deze functionaliteit is niet beschikbaar in Qlik Sense SaaS.*



*Deze opdracht wordt niet ondersteund in de standaardmodus.*

### Syntaxis:

```
execute commandline
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
<i>commandline</i>	Een tekst die door het besturingssysteem als een opdrachtregel kan worden geïnterpreteerd. U kunt verwijzen naar een absoluut bestandspad of naar een mappad lib://.

Als u **Execute** wilt gebruiken, moet aan de volgende voorwaarden worden voldaan:

- U moet werken in de bestaande modus (van toepassing voor Qlik Sense en Qlik Sense Desktop).
- U moet `OverrideScriptSecurity` instellen op 1 in `Settings.ini` (van toepassing voor Qlik Sense). `Settings.ini` bevindt zich in `C:\ProgramData\Qlik\Sense\Engine\` en is gewoonlijk een leeg bestand.



*Als u `OverrideScriptSecurity` zo instelt, dat **Execute** wordt ingeschakeld, kan elke gebruiker bestanden uitvoeren op de server. Een gebruiker kan bijvoorbeeld een uitvoerbaar bestand aan een app toevoegen en vervolgens het bestand uitvoeren in het load-script voor gegevens.*

### Doe het volgende:

1. Maak een kopie van *Settings.ini* en open het in een teksteditor.
2. Controleer of het bestand [*Settings 7*] bevat op de eerste regel.
3. Voeg een nieuwe regel in en typ *OverrideScriptSecurity=1*.
4. Voeg een lege regel in aan het einde van het bestand.
5. Sla het bestand op.
6. Vervang *Settings.ini* door uw bewerkte bestand.
7. Start Qlik Sense Engine Service (QES) opnieuw op.



*Als Qlik Sense wordt uitgevoerd als een service, werken sommige opdrachten mogelijk niet zoals verwacht.*

### Voorbeeld:

```
Execute C:\Program Files\Office12\Excel.exe;
```

```
Execute lib://win\notepad.exe // win is a folder connection referring to c:\windows
```

## Field/Fields

De sleutelwoorden **Field** en **Fields** voor scripts worden gebruikt in opdrachten **Declare**, **Derive**, **Drop**, **Comment**, **Rename** en **Tag/Untag**.

## FlushLog

Met de opdracht **FlushLog** dwingt u af dat Qlik Sense de inhoud van de scriptbuffer naar het scriptlogbestand schrijft.

### Syntaxis:

```
FlushLog
```

De inhoud van de buffer wordt naar het logbestand geschreven. Deze opdracht kan nuttig zijn bij de foutsproing, aangezien u gegevens ontvangt die anders verloren zouden zijn gegaan als de uitvoering van een script mislukt.

### Voorbeeld:

```
FlushLog;
```

## Force

Met de opdracht **force** kunt u afdwingen dat Qlik Sense veldnamen en veldwaarden van de daaropvolgende **LOAD**- en **SELECT**-opdrachten interpreteert als geschreven met alleen hoofdletters, met alleen kleine letters, met beginhoofdletters of zoals ze worden

## 2 Scriptinstructies en sleutelwoorden

weergegeven (gemengd). Met deze opdracht kunt u veldwaarden van tabellen interpreteren volgens verschillende conventies.

### Syntaxis:

```
Force ( capitalization | case upper | case lower | case mixed )
```

Als er niets is gespecificeerd, wordt force mixed case verondersteld. De force-opdracht is geldig tot een nieuwe force-opdracht wordt gegeven.

De opdracht **force** werkt niet in de toegangssectie: alle veldwaarden die zijn geladen zijn niet hoofdlettergevoelig.

### Voorbeelden en resultaten

Voorbeeld	Resultaat
<p>Dit voorbeeld laat zien hoe beginhoofdletters kunnen worden afgedwongen.</p> <pre>FORCE Capitalization; Capitalization: LOAD * Inline [ ab Cd eF GH ];</pre>	<p>De tabel <b>Capitalization</b> bevat de volgende waarden:</p> <p>Ab Cd eF Gh</p> <p>Alle waarden beginnen met een hoofdletter.</p>
<p>Dit voorbeeld laat zien hoe hoofdlettergebruik kan worden afgedwongen.</p> <pre>FORCE Case Upper; CaseUpper: LOAD * Inline [ ab Cd eF GH ];</pre>	<p>De tabel <b>CaseUpper</b> bevat de volgende waarden:</p> <p>AB CD EF GH</p> <p>Alle waarden zijn in hoofdletters.</p>
<p>Dit voorbeeld laat zien hoe het gebruik van kleine letters kan worden afgedwongen.</p> <pre>FORCE Case Lower; CaseLower: LOAD * Inline [ ab Cd eF GH ];</pre>	<p>De tabel <b>CaseLower</b> bevat de volgende waarden:</p> <p>ab cd ef gh</p> <p>Alle waarden zijn in kleine letters.</p>

Voorbeeld	Resultaat
<p>Dit voorbeeld laat zien hoe het gebruik van een combinatie van kleine letters en hoofdletters kan worden afgedwongen.</p> <pre>FORCE Case Mixed; CaseMixed: LOAD * Inline [ ab Cd eF GH ];</pre>	<p>De tabel <b>CaseMixed</b> bevat de volgende waarden:</p> <pre>ab Cd eF GH</pre> <p>Alle waarden worden weergegeven zoals zij voorkomen in het script.</p>

Zie ook:

### From

Het sleutelwoord **From** voor scripts wordt gebruikt in opdrachten **Load** om te verwijzen naar een bestand en in opdrachten **Select** om te verwijzen naar een databasetabel of -weergave.

### Load

Met de opdracht **LOAD** worden velden geladen uit een bestand, uit gegevens die in het script zijn gedefinieerd, uit een eerder geladen tabel, van een webpagina, uit het resultaat van een daaropvolgende **SELECT**-opdracht of door gegevens automatisch te genereren. U kunt tevens gegevens van analytische verbindingen inladen.

**Syntaxis:**

```
LOAD [ distinct ] fieldlist
[( from file [ format-spec ] |
from_field fieldsource [format-spec]|
inline data [ format-spec ] |
resident table-label |
autogenerate size ) |extension pluginname.functionname([script]
tabledescription)]
[ where criterion | while criterion ]
[ group by groupbyfieldlist ]
[order by orderbyfieldlist ]
```

## 2 Scriptinstructies en sleutelwoorden

---

### Argumenten:


#### Argumenten

Argument	Beschrijving
distinct	<p>U kunt <b>distinct</b> als predicaat gebruiken wanneer u alleen unieke records wilt laden. Als er dubbele records zijn, wordt het eerste exemplaar geladen.</p> <p>Als u voorgaande loads gebruikt, moet u <b>distinct</b> in de eerste laad instructie plaatsen, aangezien <b>distinct</b> alleen van invloed is op de bestemmingstabel.</p>

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
fieldlist	<p><i>fieldlist</i> ::= ( *   <i>field</i> { , *   <i>field</i> } )</p> <p>Een lijst van de velden die geladen moeten worden. Gebruik een * als veldlijst om alle velden in de tabel aan te duiden.</p> <p><i>field</i> ::= ( <i>fieldref</i>   <i>expression</i> ) [<b>as</b> <i>aliasname</i> ]</p> <p>De velddefinitie moet altijd een literal, een referentie naar een bestaand veld of een uitdrukking bevatten.</p> <p><i>fieldref</i> ::= ( <i>fieldname</i>  @<i>fieldnumber</i>  @<i>startpos</i>:<i>endpos</i> [ <b>I</b>   <b>U</b>   <b>R</b>   <b>B</b>   <b>T</b> ] )</p> <p><i>fieldname</i> is een tekst die identiek is aan een veldnaam in de tabel. Houd er rekening mee dat de veldnaam tussen rechte dubbele aanhalingstekens of vierkante haken moet staan als deze bijvoorbeeld spaties bevat. Soms zijn veldnamen niet expliciet beschikbaar. In dat geval wordt een andere notatie gebruikt:</p> <p>@<i>fieldnumber</i> geeft het veldnummer in een tabelbestand met scheidingstekens aan. Het moet een positief geheel getal zijn, voorafgegaan door "@". De nummering is altijd vanaf 1 optellend tot het aantal velden.</p> <p>@<i>startpos</i>:<i>endpos</i> geeft de begin- en eindposities van een veld aan in een bestand met een vaste recordlengte. De posities moeten beide positieve gehele getallen zijn. De twee getallen moeten worden voorafgegaan door "@" en gescheiden door een dubbele punt. De nummering is altijd vanaf 1 optellend tot het aantal posities. In het laatste veld wordt n gebruikt als eindpositie.</p> <ul style="list-style-type: none"><li>• Als @<i>startpos</i>:<i>endpos</i> direct wordt gevolgd door de tekens <b>I</b> of <b>U</b>, worden de gelezen bytes geïnterpreteerd als een binair ondertekend (<b>I</b>) of niet-ondertekend (<b>U</b>) geheel getal (Intel-bytevolgorde). Het aantal gelezen posities moet 1, 2 of 4 zijn.</li><li>• Als @<i>startpos</i>:<i>endpos</i> direct wordt gevolgd door het teken <b>R</b>, worden de bytes geïnterpreteerd als een binair reëel getal (IEEE 32-bits of 64-bits drijvende komma). Het aantal gelezen posities moet 4 of 8 zijn.</li><li>• Als @<i>startpos</i>:<i>endpos</i> direct wordt gevolgd door het teken <b>B</b>, worden de gelezen bytes geïnterpreteerd als een BCD (Binary Coded Decimal) volgens de COMP-3-standaard. U kunt een willekeurig aantal bytes opgeven.</li></ul> <p><i>expression</i> kan een numerieke functie of een tekenreeksfunctie zijn, gebaseerd op een of meer andere velden in dezelfde tabel. Zie de syntaxis van uitdrukkingen voor meer informatie.</p> <p><b>as</b> wordt gebruikt om een nieuwe naam aan een veld toe te wijzen.</p>



Argument	Beschrijving
from	<p><b>from</b> wordt gebruikt als gegevens moeten worden geladen vanuit een bestand via een map- of webbestandsgegevensverbinding.</p> <p><i>file ::= [ path ] filename</i></p> <p><b>Voorbeeld: 'lib://Table Files'</b></p> <p>Als er geen pad is opgegeven, zoekt Qlik Sense het bestand in de map die is opgegeven in de opdracht <b>Directory</b>. Als er geen opdracht <b>Directory</b> is, zoekt Qlik Sense in de werkdirectory, <i>C:\Users\{user}\Documents\Qlik\Sense\Apps</i>.</p> <div style="border: 1px solid gray; padding: 5px;"><p> <i>In een Qlik Sense-serverinstallatie wordt de werkdirectory opgegeven in Qlik Sense Repository Service. Standaard is dit C:\ProgramData\Qlik\Sense\Apps.</i></p></div> <p>De <i>filename</i> kan de standaard jokertekens van DOS bevatten ( * en ? ). Daarmee worden alle overeenkomstige bestanden in de opgegeven map geladen.</p> <p><i>format-spec ::= ( fspec-item { , fspec-item } )</i></p> <p>De opmaakspecificatie bestaat uit een lijst met verschillende opmaakspecificaties, tussen haakjes.</p> <p><b>Bestaande scriptingmodus</b></p> <p>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:</p> <ul style="list-style-type: none"><li>• absoluut</li></ul> <p><b>Voorbeeld: c:\data</b></p> <ul style="list-style-type: none"><li>• relatief ten opzichte van de werkmap van de Qlik Sense-app.</li></ul> <p><b>Voorbeeld: data</b></p> <ul style="list-style-type: none"><li>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.</li></ul> <p><b>Voorbeeld: http://www.qlik.com</b></p>

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
from_field	<p><b>from_field</b> wordt gebruikt als gegevens moeten worden geladen uit een eerder geladen veld.</p> <p><i>fieldsource::=(tablename, fieldname)</i></p> <p>De veldnaam is de naam van de eerder geladen <i>tablename</i> en <i>fieldname</i>.</p> <p><i>format-spec ::= ( fspec-item {, fspec-item } )</i></p> <p>De opmaakspecificatie bestaat uit een lijst met verschillende opmaakspecificaties, tussen haakjes.</p>
inline	<p><b>inline</b> wordt gebruikt als gegevens moeten worden getypt in het script en niet uit een bestand moeten worden geladen.</p> <p><i>data ::= [ text ]</i></p> <p>Gegevens die u invoert met een <b>inline</b>-clausule moeten tussen dubbele aanhalingstekens of tussen vierkante haakjes worden geplaatst. De tekst ertussen wordt net zo geïnterpreteerd als de inhoud van een bestand. Waar u bijvoorbeeld een nieuwe regel in een tekstbestand zou invoegen, moet u dat ook doen in de tekst van een <b>inline</b>-clausule. U drukt dus op Enter bij het typen van het script. Het aantal kolommen wordt bepaald door de eerste regel.</p> <p><i>format-spec ::= ( fspec-item {, fspec-item } )</i></p> <p>De opmaakspecificatie bestaat uit een lijst met verschillende opmaakspecificaties, tussen haakjes.</p>
resident	<p><b>resident</b> wordt gebruikt als gegevens moeten worden geladen uit een eerder geladen tabel.</p> <p><i>table label</i> is een label dat vooraf gaat aan de <b>LOAD</b>- of <b>SELECT</b>-opdracht(en) waarmee de oorspronkelijke tabel is gemaakt. Het label moet worden opgegeven met een puntkomma aan het eind.</p>
autogenerate	<p><b>autogenerate</b> wordt gebruikt als gegevens automatisch moeten worden gegenereerd door Qlik Sense.</p> <p><i>size ::= number</i></p> <p><i>Number</i> is een geheel getal dat het aantal te genereren records aangeeft.</p> <p>De veldenlijst mag geen uitdrukkingen bevatten die gegevens uit een externe gegevensbron of een eerder geladen tabel vereisen, tenzij u verwijst naar een enkele veldwaarde in een eerder geladen tabel met de functie <b>Peek</b>.</p>

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
extension	<p>U kunt gegevens via analytische verbindingen inladen. Als u een functie wilt aanroepen die in de SSE-plugin is gedefinieerd, moet u de clause <b>extension</b> gebruiken of een script evalueren.</p> <p>U kunt één tabel naar de SSE-plugin verzenden, waarop één gegevenstabel wordt teruggegeven. Als door de plugin niet de namen worden aangegeven van de velden die worden teruggegeven, krijgen de velden de namen Field1, Field2 enzovoort.</p> <pre>Extension pluginname.functionname( tabledescription );</pre> <ul style="list-style-type: none"><li>Gegevens inladen met een functie in een SSE-plugin <i>tabledescription ::= (table { ,tablefield} )</i> Als u geen tabelvelden aangeeft, worden de velden gebruikt in de volgorde waarin deze zijn ingeladen.</li><li>Gegevens inladen door een script in een SSE-plugin te evalueren <i>tabledescription ::= ( script, table { ,tablefield} )</i></li></ul> <p><b>Afhandeling van gegevenssoorten in de tabelvelddefinitie</b></p> <p>Gegevenssoorten in analytische verbindingen worden automatisch gedetecteerd. Als de gegevens geen getalswaarden bevatten en ten minste één tekstonekenreeks die niet leeg is, wordt het veld als tekstveld beschouwd. In alle andere gevallen wordt het veld als getalsveld beschouwd.</p> <p>U kunt de gegevenssoort afdwingen door een veldnaam op te nemen met <b>String()</b> of <b>Mixed()</b> voor gemengd.</p> <ul style="list-style-type: none"><li>Met <b>String()</b> maakt u het veld een tekstveld. Als het veld numeriek is, wordt het tekstgedeelte van de dubbele waarde geëxtraheerd. Er wordt niet geconverteerd.</li><li>Met <b>Mixed()</b> maakt u het veld een veld met dubbele waarden.</li></ul> <p><b>String()</b> en <b>Mixed()</b> kunnen niet buiten tabelvelddefinities van <b>extension</b> worden gebruikt, en u kunt geen andere Qlik Sense -functies in een tabelvelddefinitie gebruiken.</p> <p><b>Meer over analytische verbindingen</b></p> <p>U moet analytische verbindingen configureren voordat u deze kunt gebruiken.</p>
where	<p><b>where</b> is een clause die wordt gebruikt om aan te geven of een record wel of niet in de selectie moet worden opgenomen. De selectie wordt opgenomen als <i>criterion</i> de waarde True heeft.</p> <p><i>criterion</i> is een logische uitdrukking.</p>

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
while	<p><b>while</b> is een clause die aangeeft of een record herhaaldelijk moet worden gelezen. Hetzelfde record wordt gelezen zolang <i>criterion</i> de waarde True heeft. Een <b>while</b>-clause is vaak pas nuttig wanneer u deze samen met de functie <b>IterNo( )</b> gebruikt.</p> <p><i>criterion</i> is een logische uitdrukking.</p>
group by	<p><b>group by</b> is een clause die wordt gebruikt om te definiëren over welke velden de gegevens moeten worden geaggregeerd (gegroepeerd). De aggregatievelden moeten op de een of andere manier in de geladen uitdrukkingen worden opgenomen. U mag geen andere velden dan de aggregatievelden gebruiken buiten aggregatiefuncties in de geladen uitdrukkingen.</p> <p><i>groupbyfieldlist ::= (fieldname { ,fieldname } )</i></p>
order by	<p><b>order by</b> is een clause die wordt gebruikt om de records van een residentie tabel te sorteren voordat ze worden verwerkt door de <b>load</b>-opdracht. De residentie tabel kan worden gesorteerd op één of meer velden in oplopende of aflopende volgorde. Het sorteren gebeurt primair op numerieke waarde en secundair op nationale sorteervolgorde. U kunt deze clause alleen gebruiken wanneer de gegevensbron een residentie tabel is.</p> <p>De volgordevelden geven aan op welk veld de residentie tabel wordt gesorteerd. U kunt het veld opgeven met z'n naam of met z'n nummer in de residentie tabel (het eerste veld is nummer 1).</p> <p><i>orderbyfieldlist ::= fieldname [ sortorder ] { , fieldname [ sortorder ] }</i></p> <p><i>sortorder</i> is <i>asc</i> voor oplopend of <i>desc</i> voor aflopend. Als u geen <i>sortorder</i> opgeeft, wordt <i>asc</i> verondersteld.</p> <p><i>fieldname, path, filename</i> en <i>aliasname</i> zijn tekenreeksen die aangeven waar de respectievelijke namen voor staan. Elk veld in de brontabel kan worden gebruikt als <i>fieldname</i>. Velden die zijn gemaakt via de <i>as</i>-clause (<i>aliasname</i>) vallen echter buiten het bereik en kunnen niet worden gebruikt binnen dezelfde <b>load</b>-opdracht.</p>

Als u geen gegevensbron opgeeft door middel van de clause **from**, **inline**, **resident**, **from\_field**, **extension** of **autogenerate**, worden gegevens geladen uit de resultaten van de direct volgende **SELECT**- of **LOAD**-opdracht. Deze opdracht mag geen prefix hebben.

### Voorbeelden:

Verschillende bestandstypen laden

Laad een bestand met gegevens met scheidingstekens met standaardopties:

```
LOAD * from data1.csv;
```

## 2 Scriptinstructies en sleutelwoorden

---

Laad een bestand met gegevens met scheidingstekens via een bibliotheekverbinding (DataFiles):

```
LOAD * from 'lib://DataFiles/data1.csv';
```

Laad alle bestanden met gegevens met scheidingstekens via een bibliotheekverbinding (DataFiles):

```
LOAD * from 'lib://DataFiles/*.csv';
```

Laad een bestand met gegevens met scheidingstekens, waarbij een komma is opgegeven als scheidingsteken en ingesloten labels worden gebruikt:

```
LOAD * from 'c:\userfiles\data1.csv' (ansi, txt, delimiter is ',', embedded labels);
```

Laad een bestand met gegevens met scheidingstekens, waarbij een tab is opgegeven als scheidingsteken en ingesloten labels worden gebruikt:

```
LOAD * from 'c:\userfiles\data2.txt' (ansi, txt, delimiter is '\t', embedded labels);
```

Laad een dif-bestand met ingesloten kopteksten:

```
LOAD * from file2.dif (ansi, dif, embedded labels);
```

Laad drie velden vanuit een bestand met vaste recordlengte zonder kopteksten:

```
LOAD @1:2 as ID, @3:25 as Name, @57:80 as City from data4.fix (ansi, fix, no labels, header is 0, record is 80);
```

Laad een QVX-bestand, waarbij een absoluut pad wordt opgegeven:

```
LOAD * from c:\qdssamples\xyz.qvx (qvx);
```

Webbestanden inladen

Inladen via de standaard-URL die in de gegevensverbinding voor het webbestand is ingesteld:

```
LOAD * from [lib://MyWebFile];
```

Inladen via een specifieke URL in plaats van via de URL die in de gegevensverbinding voor het webbestand is ingesteld:

```
LOAD * from [lib://MyWebFile] (URL is 'http://localhost:8000/foo.bar');
```

Inladen via een specifieke URL die met een dollarteken-uitbreiding in een variabele is ingesteld:

```
SET dynamicURL = 'http://localhost/foo.bar';  
LOAD * from [lib://MyWebFile] (URL is '$(dynamicURL)');
```

Selecteren van bepaalde velden, velden hernoemen en velden berekenen

Laad slechts drie specifieke velden vanuit een bestand met gegevens met scheidingstekens:

```
LOAD FirstName, LastName, Number from data1.csv;
```

Hernoem het eerste veld tot A en het tweede veld tot B bij het laden van een bestand zonder labels:

```
LOAD @1 as A, @2 as B from data3.txt (ansi, txt, delimiter is '\t', no labels);
```

Laad Name als een samenvoeging van FirstName, een spatie en LastName:

## 2 Scriptinstructies en sleutelwoorden

---

```
LOAD FirstName&' '&LastName as Name from data1.csv;
```

Laad Quantity, Price en Value (het product van Quantity en Price):

```
LOAD Quantity, Price, Quantity*Price as Value from data1.csv;
```

Bepaalde records selecteren

Laad alleen unieke records. Dubbele records worden verwijderd:

```
LOAD distinct FirstName, LastName, Number from data1.csv;
```

Laad alleen records waarbij het veld Litres een waarde van meer dan nul heeft:

```
LOAD * from Consumption.csv where Litres>0;
```

Gegevens niet in bestand en automatisch gegenereerde gegevens laden

Laad een tabel met uitgelijnde gegevens en twee velden met de naam CatID en Category:

```
LOAD * Inline  
[CatID, Category  
0,Regular  
1,occasional  
2,Permanent];
```

Laad een tabel met uitgelijnde gegevens en drie velden met de naam UserID, Password en Access:

```
LOAD * Inline [UserID, Password, Access  
A, ABC456, User  
B, VIP789, Admin];
```

Laad een tabel met 10.000 rijen. Veld A bevat het nummer van de ingelezen record (1,2,3,4,5...) en veld B bevat een willekeurig getal tussen 0 en 1:

```
LOAD RecNo( ) as A, rand( ) as B autogenerate(10000);
```



*De haakjes na autogenerate zijn toegestaan maar niet verplicht.*

Gegevens laden uit een eerder geladen tabel

Als eerste laden we een bestand met een tabel met scheidingstekens en geven dat de naam tab1:

```
tab1:  
SELECT A,B,C,D from 'lib://DataFiles/data1.csv';
```

Laad velden vanuit de reeds geladen tabel tab1 als tab2:

```
tab2:  
LOAD A,B,month(C),A*B+D as E resident tab1;
```

Laad velden uit reeds geladen tabel tab1 maar alleen records waarbij A groter is dan B:

```
tab3:  
LOAD A,A+B+C resident tab1 where A>B;
```

Laad velden uit al geladen tabel tab1 geordend volgens A:

## 2 Scriptinstructies en sleutelwoorden

---

```
LOAD A,B*C as E resident tab1 order by A;
```

Laad velden vanuit reeds geladen tabel tab1, geordend volgens het eerste veld en vervolgens het tweede veld:

```
LOAD A,B*C as E resident tab1 order by 1,2;
```

Laad veld vanuit reeds geladen tabel tab1 geordend volgens C aflopend, daarna B in oplopende volgorde, en dan het eerste veld in aflopende volgorde:

```
LOAD A,B*C as E resident tab1 order by C desc, B asc, 1 desc;
```

Gegevens laden uit eerder geladen velden

Laad veld Types uit eerder geladen tabel Characters als A:

```
LOAD A from_field (Characters, Types);
```

Gegevens laden uit een volgende tabel (voorafgaande load)

Laad A, B en berekende velden X en Y vanuit Table1 die is geladen in de volgende opdracht **SELECT**:

```
LOAD A, B, if(C>0,'positive','negative') as X, weekday(D) as Y;  
SELECT A,B,C,D from Table1;
```

Gegevens groeperen

Laad velden gegroepeerd (geaggregeerd) op ArtNo:

```
LOAD ArtNo, round(Sum(TransAmount),0.05) as ArtNoTotal from table.csv group by ArtNo;
```

Laad velden gegroepeerd (geaggregeerd) op Week en ArtNo:

```
LOAD Week, ArtNo, round(Avg(TransAmount),0.05) as weekArtNoAverages from table.csv group by  
Week, ArtNo;
```

Herhaaldelijk lezen van één record

In dit voorbeeld hebben we een invoerbestand, Grades.csv, met de cijfers voor elke student gecompriemd in één veld:

```
Student,Grades  
Mike,5234  
John,3345  
Pete,1234  
Paul,3352
```

De cijfers, op een schaal van 1-5, staan voor Math, English, Science en History. We kunnen de cijfers scheiden tot afzonderlijke waarden. Hiervoor lezen we elke record meerdere keren met een **while**-clause. We gebruiken de functie **IterNo( )** als een teller. Bij elke lezing wordt het cijfer uit de functie **Mid** gehaald en opgeslagen in Grade en wordt het vak geselecteerd op basis van de functie **pick** en opgeslagen in Subject. De laatste **while**-clause bevat de test waarmee wordt gecontroleerd of alle cijfers zijn gelezen (in dit geval vier per leerling). Dat houdt in dat de volgende leerlingrecord moet worden gelezen.

MyTab:

```
LOAD Student,  
mid(Grades,IterNo( ),1) as Grade,
```

## 2 Scriptinstructies en sleutelwoorden

---

```
pick(IterNo( ), 'Math', 'English', 'Science', 'History') as Subject from Grades.csv  
while IsNum(mid(Grades,IterNo(),1));
```

Het resultaat is een tabel met de volgende gegevens:

Student	Subject	Grade
John	English	3
John	History	5
John	Math	3
John	Science	4
Mike	English	2
Mike	History	4
Mike	Math	5
Mike	Science	3
Paul	English	3
Paul	History	2
Paul	Math	3
Paul	Science	5
Pete	English	2
Pete	History	4
Pete	Math	1
Pete	Science	3

Inladen via analytische verbindingen

De onderstaande voorbeeldgegevens worden gebruikt.

values:

Load

Rand() as A,

Rand() as B,

Rand() as C

AutoGenerate(50);

### Gegevens inladen met een functie

In deze voorbeelden nemen we aan dat we een plugin met de naam *P* voor analytische verbinding hebben, die een aangepaste functie *Calculate(Parameter1, Parameter2)* bevat. Deze functie geeft de tabel *Results* terug met de velden *Field1* en *Field2*.

```
Load * Extension P.Calculate( values{A, C} );
```

Hiermee laadt u alle gegevens in die worden teruggegeven nadat de velden A en C naar de functie zijn verzonden.

```
Load Field1 Extension P.Calculate( values{A, C} );
```

Hiermee laadt u alleen veld Field1 in de velden A en C naar de functie zijn verzonden.

```
Load * Extension P.Calculate( values );
```

Hiermee laadt u alle gegevens in die worden teruggegeven nadat de velden A en B naar de functie zijn verzonden. Als velden niet gespecificeerd zijn, worden A en B gebruikt als deze het eerst zijn in de volgorde in de tabel.



## 2 Scriptinstructies en sleutelwoorden

```
Load * Extension P.Calculate( values {C, C});
```

Hiermee laadt u alle velden in die worden teruggegeven nadat het veld C naar beide parameters van de functie is verzonden.

```
Load * Extension P.Calculate( values {String(A), Mixed(B)});
```

Hiermee laadt u alle velden in die worden teruggegeven nadat het veld A geforceerd als tekenreeks en het veld B geforceerd als numeriek naar de functie zijn verzonden.

### Gegevens inladen door een script te evalueren

```
Load A as A_echo, B as B_echo Extension R.ScriptEval( 'q;', values{A, B} );
```

Laad de tabel die door het script q wordt teruggegeven als de waarden van A en B worden verzonden.

```
Load * Extension R.ScriptEval( '$(My_R_Script)', values{A, B} );
```

Hiermee laadt u de tabel in die door het script wordt teruggegeven in de variabele My\_R\_Script als de waarden A en B worden verzonden.

```
Load * Extension R.ScriptEval( '$(My_R_Script)', values{B as D, *} );
```

Hiermee laadt u de tabel in die door het script wordt teruggegeven in de variabele My\_R\_Script als de waarden van B worden herbenaemd tot D, A en C. Als u \* gebruikt, verzendt u de overgebleven ongekoppelde velden.



*Bestandsextensies van DataFiles-verbindingen zijn hoofdlettergevoelig. Bijvoorbeeld: .qvd.*

### Opmaakspecificaties

Elke indelingsspecificatie definieert een bepaalde eigenschap van het tabelbestand:

```
fspec-item ::= [ ansi | oem | mac | UTF-8 | Unicode | txt | fix | dif | biff | ooxml | html | xml | kml | qvd | qvx | delimiter is char | no eof | embedded labels | explicit labels | no labels | table is [tablename] | header is n | header is line | header is n lines | comment is string | record is n | record is line | record is n lines | no quotes | msq | URL is string | userAgent is string ]
```

### Tekenset

Tekenset is een bestandsspecificatie voor de **LOAD**-opdracht die de tekenset definieert die wordt gebruikt in het bestand.

De specificaties **ansi**, **oem** en **mac** werden gebruikt in QlikView en werken nog steeds. Zij worden echter niet gegenereerd bij het maken van de **LOAD**-opdracht met Qlik Sense.

### Syntaxis:

```
utf8 | unicode | ansi | oem | mac | codepage is
```

### Argumenten:

Argumenten

Argument	Beschrijving
<b>utf8</b>	UTF-8-tekenset
<b>unicode</b>	Unicode-tekenset
<b>ansi</b>	Windows, codepagina 1252
<b>oem</b>	DOS, OS/2, AS400 en andere
<b>mac</b>	Codepagina 10000
<b>codepage is</b>	Met de specificatie <b>codepage</b> is het mogelijk om elke Windows-codepagina te gebruiken als <i>N</i> .

### Beperkingen:

Conversie van de **oem**-tekenreeks is niet geïmplementeerd voor MacOS. Als geen tekenset is opgegeven, wordt onder Windows codepagina 1252 gebruikt.

### Voorbeeld:

```
LOAD * from a.txt (utf8, txt, delimiter is ',' , embedded labels)
LOAD * from a.txt (unicode, txt, delimiter is ',' , embedded labels)
LOAD * from a.txt (codepage is 10000, txt, delimiter is ',' , no labels)
```

---

### Zie ook:

p *Load (page 150)*


### Tabelopmaak

De tabelindeling is een bestandsspecificatie voor de **LOAD**-opdracht waarmee het bestandstype wordt vastgelegd. Als er niets is gespecificeerd, wordt een *.txt*-bestand verondersteld.

Tabelopmaaktypen

Type	Beschrijving
txt	In een tekstbestand met scheidingstekens worden de kolommen in de tabel gescheiden door een bepaald teken.

## 2 Scriptinstructies en sleutelwoorden

Type	Beschrijving
fix	<p>In een bestand met een vaste recordlengte is elke kolom precies een bepaald aantal tekens breed.</p> <p>Veel bestanden met vaste recordlengte bevatten records die via een einde regel van elkaar zijn gescheiden, maar er zijn meer geavanceerde opties om de recordgrootte in bytes op te geven of om records meer dan één regel lang te laten zijn met <b>Record is</b>.</p> <div style="border: 1px solid gray; padding: 5px;"><p> <i>Als de gegevens tekens met meerdere bytes bevatten, kunnen afbreekposities in velden onjuist uitgelijnd raken aangezien de indeling is gebaseerd op een vaste lengte in bytes.</i></p></div>
dif	In een <i>.dif</i> -bestand (Data Interchange Format) wordt een speciale tabelindeling gebruikt.
biff	Qlik Sense kan ook gegevens in standaard Excel-bestanden interpreteren, door middel van de <i>biff</i> -indeling (Binary Interchange File Format).
ooxml	Excel 2007 en hogere versies gebruiken de indeling ooxml <i>.xlsx</i> .
html	Als de tabel onderdeel vormt van een html-pagina of -bestand, moet html worden gebruikt.
xml	xml (Extensible Markup Language) is een veelgebruikte opmaaktaal voor de representatie van gegevensstructuren in tekstindeling.
qvd	De <i>qvd</i> -indeling is de eigen indeling voor QVD-bestanden die zijn geëxporteerd uit een Qlik Sense-app.
qvx	<i>qvx</i> is a bestands-/stream-indeling voor snelle uitvoer naar Qlik Sense.

### Delimiter is

Voor tabelbestanden met scheidingstekens kan een willekeurig scheidingsteken worden opgegeven via de specificatie **delimiter is**. Deze specificatie is alleen relevant voor *.txt*-bestanden met scheidingstekens.

#### Syntaxis:

```
delimiter is char
```

#### Argumenten:

##### Argumenten

Argument	Beschrijving
char	Hiermee wordt een enkel teken van de 127 ASCII-tekenen gespecificeerd.

Daarnaast kunnen de volgende waarden worden gebruikt:

## 2 Scriptinstructies en sleutelwoorden

---

### Optionele waarden

Value	Beschrijving
't'	geeft een tabteken weer, met of zonder aanhalingstekens.
'\'	geeft een backslash (\) weer.
'spaces'	geeft alle combinaties van een of meer spaties weer. Niet-afdrukbare tekens met een ASCII-waarde onder de 32, met uitzondering van CR en LF, worden als spaties geïnterpreteerd.

Als er niets is gespecificeerd, wordt **delimiter is ','** verondersteld.

#### Voorbeeld:

```
LOAD * from a.txt (utf8, txt, delimiter is ',' , embedded labels);
```

---

#### Zie ook:

p *Load (page 150)*

### No eof

Met de specificatie **no eof** geeft u aan dat het teken voor het einde van het bestand moet worden genegeerd bij het laden van **.txt**-bestanden met scheidingstekens.

#### Syntaxis:

```
no eof
```

Als de specificatie **no eof** wordt gebruikt, worden tekens met codepunt 26, die gewoonlijk het eind van het bestand aangeven, genegeerd en kunnen deze deel uitmaken van een veldwaarde.

Dit is alleen relevant voor tekstbestanden met scheidingstekens.

#### Voorbeeld:

```
LOAD * from a.txt (txt, utf8, embedded labels, delimiter is ' ', no eof);
```

---

#### Zie ook:

p *Load (page 150)*

### Labels

**Labels** is een bestandsspecificatie voor de opdracht **LOAD** waarmee wordt gedefinieerd waar in een bestand de veldnamen te vinden zijn.

#### Syntaxis:

```
embedded labels|explicit labels|no labels
```

---

## 2 Scriptinstructies en sleutelwoorden

---

De veldnamen worden in verschillende plaatsen in het bestand aangetroffen. Als de eerste record de veldnamen bevat, moet u **embedded labels** gebruiken. Als er geen veldnamen worden aangetroffen, moet u **no labels** gebruiken. In *dif*-bestanden wordt soms een afzonderlijke koptekstsectie met uitdrukkelijke veldnamen gebruikt. In dat geval moet u **explicit labels** gebruiken. Als er niets wordt opgegeven, wordt aangenomen dat **embedded labels** wordt gebruikt, ook voor *dif*-bestanden.

### Example 1:

```
LOAD * from a.txt (unicode, txt, delimiter is ',' , embedded labels)
```

### Example 2:

```
LOAD * from a.txt (codePage is 1252, txt, delimiter is ',' , no labels)
```

---

### Zie ook:

p *Load (page 150)*

### Header is

Hiermee wordt de grootte van de koptekst in tabelbestanden opgegeven. U kunt een willekeurige koptekstlengte opgeven met behulp van de specificatie **header is**. Een koptekst is een tekstgedeelte dat niet in Qlik Sense wordt gebruikt.

### Syntaxis:

```
header is n  
header is line  
header is n lines
```

De lengte van de koptekst kan worden opgegeven in bytes (**header is n**) of in regels (**header is line** of **header is n lines**). **n** moet een positief geheel getal zijn dat de lengte van de koptekst aangeeft. Als dit niet is gespecificeerd, wordt **header is 0** verondersteld. De specificatie **header is** is uitsluitend relevant voor tabelbestanden.

### Voorbeeld:

Dit is een voorbeeld van een gegevensbrontabel die een tekstregel voor de koptekst bevat die niet als gegevens moet worden geïnterpreteerd in Qlik Sense.

```
*Header line  
col1,col2  
a,B  
c,D
```

Bij gebruik van de specificatie **header is 1 lines** wordt de eerste regel niet geladen als gegevens. In het voorbeeld geeft de specificatie **embedded labels** aan dat Qlik Sense de eerste niet-uitgesloten regel moet interpreteren als regel met veldlabels.

```
LOAD col1, col2  
FROM 'lib://files/header.txt'  
(txt, embedded labels, delimiter is ',', msq, header is 1 lines);
```

Het resultaat is een tabel met twee velden: Col1 en Col2.

### Zie ook:

p *Load (page 150)*

### Record is

Voor bestanden met een vaste recordlengte moet de recordlengte worden opgegeven met de specificatie **record is**.

### Syntaxis:

```
Record is n
Record is line
Record is n lines
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
n	Geeft de recordlengte in bytes.
line	Geeft de recordlengte op als één regel.
n lines	Geeft de recordlengte in regels waarbij n een positief geheel getal is dat de recordlengte voorstelt.

### Beperkingen:

De specificatie **record is** is alleen relevant voor **fix**-bestanden.

### Zie ook:

p *Load (page 150)*

### Quotes

**Quotes** is een bestandsspecificatie voor de opdracht **LOAD** waarmee wordt gedefinieerd of aanhalingstekens kunnen worden gebruikt en wat de prioriteit is tussen aanhalingstekens en scheidingstekens. Uitsluitend voor tekstbestanden.

### Syntaxis:

```
no quotes
msq
```

Als de specificatie wordt weggelaten, kunnen standaard de aanhalingstekens " " of ' ' worden gebruikt, maar alleen als ze de eerste en laatste niet-blanco tekens in een veldwaarde zijn.

### Argumenten:

#### Argumenten

Argument	Beschrijving
no quotes	Wordt gebruikt als aanhalingstekens niet worden geaccepteerd in een tekstbestand.
msq	Wordt gebruikt voor het opgeven van aanhalingstekens in moderne stijl, waardoor meerregelige inhoud in velden mogelijk wordt. Velden die tekens bevatten die het einde van een regel aangeven, moeten tussen dubbele aanhalingstekens worden geplaatst.  Eén beperking van de optie msq is dat enkele tekens die een dubbel aanhalingsteken aangeven (") die als eerste of laatste teken voorkomen in veldinhoud worden geïnterpreteerd als het begin of einde van meerregelige inhoud. Dit kan tot onvoorspelbare resultaten leiden in de gegevensset die wordt geladen. In dat geval moet u standaard aanhalingstekens gebruiken door de specificatie weg te laten.

### XML

Deze scriptspecificatie wordt gebruikt voor het laden van xml-bestanden. Geldige opties voor deXML specificatie worden vermeld in de syntaxis.



*U kunt geen DTD-bestanden laden in Qlik Sense.*

### Syntaxis:

```
xmlsimple
```

### Zie ook:

p *Load (page 150)*

### KML

Deze scriptspecificatie wordt gebruikt voor het laden van KML-bestanden voor een kaartvisualisatie.

### Syntaxis:

```
kml
```

Het KML-bestand kan gebiedsgegevens (bijvoorbeeld landen of regio's) aangeven die worden weergegeven door polygonen, lijngegevens (bijvoorbeeld spoorwegen of wegen) of puntgegevens (bijvoorbeeld steden of plaatsen) die worden weergegeven door punten in de vorm [long, lat].

### URL is

De scriptspecificatie wordt tijdens het laden van een webbestand gebruikt voor het instellen van de URL van een gegevensverbinding hiervoor.

### Syntaxis:

```
URL is string
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
string	Hiermee wordt de URL van het in te laden bestand gespecificeerd. Dit gaat voor de URL die in de gebruikte webbestandverbinding is ingesteld.

### Beperkingen:

De specificatie **URL is** is alleen relevant voor webbestanden. U moet een bestaande gegevensverbinding voor een webbestand gebruiken.

### Zie ook:

p *Load (page 150)*

### userAgent is

Deze scriptspecificatie wordt gebruikt om de browsergebruikersagent in te stellen als er een webbestand wordt ingeladen.

### Syntaxis:

```
userAgent is string
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
string	Hiermee wordt de tekenreeks van de browsergebruikersagent weergegeven. Hiermee negeert u de standaardbrowsergebruikersagent "Mozilla/5.0".

### Beperkingen:

De specificatie **userAgent is** is alleen relevant voor webbestanden.

### Zie ook:

p *Load (page 150)*



### Let

De opdracht **let** is een aanvulling op de opdracht **set** voor het definiëren van scriptvariabelen. In tegenstelling tot de opdracht **set** evalueert de opdracht **let** de uitdrukking rechts van '=' tijdens runtime van het script voordat deze wordt toegewezen aan de variabele.

#### Syntaxis:

```
Let variablename=expression
```

Voorbeelden en resultaten:

Voorbeeld	Resultaat
Set x=3+4;	\$(x) wordt geëvalueerd als ' 3+4 '
Let y=3+4;	\$(y) wordt geëvalueerd als ' 7 '
z=\$(y)+1;	\$(z) wordt geëvalueerd als ' 8 '
	Let op het verschil tussen de <b>Set</b> - en <b>Let</b> -instructies. De <b>Set</b> -instructie wijst de reeks '3+4' toe aan de variabele, terwijl de <b>Let</b> -instructie de reeks evalueert en 7 toewijst aan de variabele.
Let T=now( );	\$(T) krijgt de waarde van de huidige tijd.

### Loosen Table

Een of meer interne gegevenstabellen van Qlik Sense kunt u tijdens de uitvoering van het script expliciet declareren als losjes gekoppeld met behulp van de opdracht **Loosen Table**. Als een tabel losjes is gekoppeld, zijn alle associaties tussen veldwaarden in de tabel verwijderd. Een vergelijkbaar effect kan worden verkregen door elk veld van de losjes gekoppelde tabel als onafhankelijke, niet-verbonden tabel te laden. Losjes gekoppeld kan handig zijn tijdens het testen om tijdelijk verschillende onderdelen van de gegevensstructuur te isoleren. Een losjes gekoppelde tabel kan worden geïdentificeerd in de tabelweergave aan de hand van de stippellijnen. Het gebruik van een of meer **Loosen Table**-opdrachten in het script zorgt dat Qlik Sense eventuele instellingen negeert voor losjes gekoppelde tabellen voordat het script wordt uitgevoerd.

#### Syntaxis:

```
Loosen Tabletablename [ , tablename2 ...]
```

```
Loosen Tablestablename [ , tablename2 ...]
```

U kunt de syntaxis **Loosen Table** of **Loosen Tables** gebruiken.



Als er in Qlik Sense circulaire referenties in de gegevensstructuur worden gevonden die niet kunnen worden verbroken door tabellen die interactief of expliciet in het script tot losjes gekoppeld zijn benoemd, worden een of meer extra tabellen gedwongen losjes gekoppeld, totdat er geen circulaire referenties meer bestaan. Als dit gebeurt, wordt er een dialoogvenster weergegeven met een **waarschuwing dat er een circulaire referentie is gevonden**.

### Voorbeeld:

```
Tab1:  
SELECT * from Trans;  
Loosen Table Tab1;
```

## Map

De opdracht **map ... using** wordt gebruikt voor de toewijzing van een bepaalde veldwaarde of uitdrukking aan de waarden van een specifieke toewijzingstabel. De toewijzingstabel wordt gemaakt met behulp van de opdracht **Mapping**.

### Syntaxis:

```
Map fieldlist Using mapname
```

De automatische toewijzing vindt plaats voor velden die zijn geladen na de opdracht **Map ... Using** tot aan het einde van het script of totdat een opdracht **Unmap** wordt aangetroffen.

De toewijzing gebeurt als laatste in de keten van gebeurtenissen die tot de opslag van het veld in de interne tabel in Qlik Sense leiden. Toewijzing vindt dus niet steeds plaats wanneer een veldnaam wordt aangetroffen als onderdeel van een uitdrukking, maar wanneer de waarde wordt opgeslagen onder de veldnaam in de interne tabel. Als op het niveau van de uitdrukking toewijzing vereist is, moet echter de functie **Applymap()** worden gebruikt.

### Argumenten:

#### Argumenten

Argument	Beschrijving
<i>fieldlist</i>	Een door komma's gescheiden lijst met de velden die vanaf dit punt in het script moeten worden toegewezen. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens.
<i>mapname</i>	De naam van een toewijzingstabel die eerder is gelezen in een opdracht <b>mapping load</b> of <b>mapping select</b> .

## 2 Scriptinstructies en sleutelwoorden

Voorbeelden en resultaten:

Voorbeeld	Resultaat
Map Country Using Cmap;	Maakt het toewijzen van het veld Country mogelijk met de toewijzing Cmap.
Map A, B, C Using X;	Maakt het toewijzen van de velden A, B en C mogelijk met de toewijzing X.
Map * Using GenMap;	Maakt het toewijzen van alle velden mogelijk met GenMap.

### NullAsNull

De opdracht **NullAsNull** schakelt de conversie van NULL-waarden naar tekenreekswaarden uit die eerder is ingesteld via een **NullAsValue**-opdracht.

#### Syntaxis:

```
NullAsNull *fieldlist
```

De opdracht **NullAsValue** fungeert als schakeloptie en kan verschillende keren worden in- of uitgeschakeld in het script, met behulp van de opdracht **NullAsValue** of **NullAsNull**.

#### Argumenten:

Argumenten

Argument	Beschrijving
*fieldlist	Een door komma's gescheiden lijst met velden waarvoor <b>NullAsNull</b> moet worden ingeschakeld. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens.

#### Voorbeeld:

```
NullAsNull A,B;  
LOAD A,B from x.csv;
```

### NullAsValue

De opdracht **NullAsValue** legt vast voor welke velden NULL moet worden omgezet in een waarde.

#### Syntaxis:

```
NullAsValue *fieldlist
```

In Qlik Sense wordt standaard aangenomen dat NULL-waarden ontbrekende of ongedefinieerde entiteiten zijn. In sommige databasecontexten moeten NULL-waarden echter als speciale waarden worden beschouwd in plaats van ontbrekende waarden. Het feit dat NULL-waarden gewoonlijk niet mogen worden gekoppeld aan andere NULL-waarden, is te omzeilen door middel van de opdracht **NullAsValue**.

## 2 Scriptinstructies en sleutelwoorden

De opdracht **NullAsValue** fungeert als schakeloptie en werkt bij opeenvolgende laadopdrachten. Deze opdracht kan met behulp van de opdracht **NullAsNull** worden uitgeschakeld.

### Argumenten:

#### Argumenten

Argument	Beschrijving
*fieldlist	Een door komma's gescheiden lijst met velden waarvoor <b>NullAsValue</b> moet worden ingeschakeld. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens.

### Voorbeeld:

```
NullAsValue A,B;  
Set NullValue = 'NULL';  
LOAD A,B from x.csv;
```

## Qualify

De opdracht **Qualify** wordt gebruikt om de kwalificatie van veldnamen in te schakelen. Dit houdt in dat veldnamen de tabelnaam als prefix krijgen.

### Syntaxis:

```
Qualify *fieldlist
```

De automatische join van velden met dezelfde naam in verschillende tabellen kan worden voorkomen met behulp van de opdracht **qualify**. Hiermee wordt de veldnaam gekwalificeerd met de tabelnaam ervan. Bij kwalificatie worden de veldnamen gewijzigd wanneer ze in een tabel worden aangetroffen. De nieuwe naam heeft de vorm *tablename.fieldname*. *Tablename* is gelijk aan het label van de huidige tabel of, als er geen label is, aan de naam die verschijnt achter **from** in **LOAD**- en **SELECT**-opdrachten.

De kwalificatie vindt plaats voor alle velden die worden geladen na de opdracht **qualify**.

Kwalificatie is standaard altijd uitgeschakeld aan het begin van een script. Kwalificatie van een specifiek veld kan op elk moment worden geactiveerd met behulp van de instructie **qualify**. Kwalificatie kan op elk moment worden uitgeschakeld met behulp van de instructie **Unqualify**.



*De opdracht **qualify** moet niet worden gebruikt in combinatie met gedeeltelijk opnieuw laden.*

### Argumenten:

#### Argumenten

Argument	Beschrijving
*fieldlist	Een door komma's gescheiden lijst met velden waarvoor kwalificatie moet worden ingeschakeld. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens.

### Example 1:

```
Qualify B;  
LOAD A,B from x.csv;  
LOAD A,B from y.csv;
```

De twee tabellen **x.csv** en **y.csv** zijn alleen gekoppeld via **A**. Dit leidt tot drie velden: , , A, x.B, y.B.

### Example 2:

In een onbekende database is het vaak handig om in het begin alleen een enkel veld of een paar velden te koppelen, zoals in het volgende voorbeeld:

```
qualify *;  
unqualify TransID;  
SQL SELECT * from tab1;  
SQL SELECT * from tab2;  
SQL SELECT * from tab3;
```

Alleen **TransID** zal worden gebruikt voor associaties tussen de tabellen *tab1*, *tab2* en *tab3*.

## Rem

Met de opdracht **rem** kunt u opmerkingen of commentaar aan het script toevoegen, of scriptopdrachten tijdelijk deactiveren zonder ze te verwijderen.

### Syntaxis:

```
Rem string
```

Alles wat tussen **rem** en de volgende puntkomma ; staat, wordt als opmerking beschouwd.

Er zijn twee alternatieve manieren om opmerkingen in uw script op te nemen:

1. U kunt een opmerking op elke willekeurige plaats in het script opnemen - behalve tussen twee aanhalingstekens - door het betreffende deel tussen /\* en \*/ te plaatsen.
2. Als u // in het script typt, wordt alle tekst rechts daarvan op dezelfde rij beschouwd als een opmerking. (Let op de uitzondering //: die als deel van een internetadres kan worden gebruikt.)

### Argumenten:

#### Argumenten

Argument	Beschrijving
string	Een willekeurige tekst.

### Voorbeeld:

```
Rem ** This is a comment **;  
/* This is also a comment */  
// This is a comment as well
```

## Rename

Het sleutelwoord **Rename** voor scripts kan worden gebruikt om de naam te wijzigen van tabellen of velden die al zijn geladen.

### Rename field

Met deze scriptfunctie wordt de naam van een of meer bestaande Qlik Sense-velden gewijzigd nadat deze zijn geladen.



*Het wordt niet aanbevolen om een variabele dezelfde naam te geven als een veld of functie in Qlik Sense.*

U kunt de syntaxis **rename field** of **rename fields** gebruiken.

### Syntaxis:

```
Rename Field (using mapname | oldname to newname{ , oldname to newname })  
Rename Fields (using mapname | oldname to newname{ , oldname to newname })
```

### Argumenten:

Argument	Beschrijving
mapname	De naam van de eerder geladen toewijzingstabel met een of meer oude en nieuwe veldnaamparen.
oldname	De oude veldnaam.
newname	De nieuwe veldnaam.

### Beperkingen:

U kunt de namen van twee velden niet zo wijzigen dat deze dezelfde naam hebben.

### Example 1:

```
Rename Field XAZ0007 to Sales;
```

### Example 2:

```
FieldMap:  
Mapping SQL SELECT oldnames, newnames from datadictionary;  
Rename Fields using FieldMap;
```

### Rename table

Met deze scriptfunctie wordt de naam van een of meer bestaande interne Qlik Sense-tabellen gewijzigd nadat deze zijn geladen.

U kunt de syntaxis **rename table** of **rename tables** gebruiken.

### Syntaxis:

```
Rename Table (using mapname | oldname to newname{ , oldname to newname })  
Rename Tables (using mapname | oldname to newname{ , oldname to newname })
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
mapname	De naam van de eerder geladen toewijzingstabel met een of meer oude en nieuwe tabelnaamparen.
oldname	De oude tabelnaam.
newname	De nieuwe tabelnaam.

### Beperkingen:

Twee tabellen met verschillende namen kunnen niet dezelfde naam krijgen. Het script genereert een fout als u probeert de naam van een tabel te wijzigen zodat deze dezelfde naam krijgt als een bestaande tabel.

### Example 1:

```
Tab1:  
SELECT * from Trans;  
Rename Table Tab1 to Xyz;
```

### Example 2:

```
TabMap:  
Mapping LOAD oldnames, newnames from tabnames.csv;  
Rename Tables using TabMap;
```

## Search

De **Search**-opdracht wordt gebruikt voor het opnemen of uitsluiten van velden bij slimme zoekacties.

## 2 Scriptinstructies en sleutelwoorden

### Syntaxis:

```
Search Include *fieldlist
```

```
Search Exclude *fieldlist
```

U kunt verschillende Search-opdrachten gebruiken om uw selectie van op te nemen velden te verfijnen. De opdrachten worden geëvalueerd van boven naar beneden.

### Argumenten:

#### Argumenten

Argument	Beschrijving
*fieldlist	Een door komma's gescheiden lijst van op te nemen of uit te sluiten velden in zoekopdrachten in slimme zoekacties. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens.

### Voorbeeld:

#### Zoekvoorbeelden

Opdracht	Beschrijving
Search Include *;	Alle velden opnemen in zoekopdrachten in slimme zoekacties.
Search Exclude [*ID];	Alle velden die eindigen op ID uitsluiten van zoekopdrachten in slimme zoekacties.
Search Exclude '*ID';	Alle velden die eindigen op ID uitsluiten van zoekopdrachten in slimme zoekacties.
Search Include ProductID;	Het veld ProductID opnemen in zoekopdrachten in slimme zoekacties.

Het gecombineerde resultaat van deze drie opdrachten in deze volgorde is dat alle velden die eindigen op ID behalve ProductID worden uitgesloten van zoekopdrachten in slimme zoekacties.

## Section

Met de opdracht **section** kan worden aangegeven of de daarop volgende **LOAD**- en **SELECT**-opdrachten moeten worden beschouwd als gegevens of als een definitie van de toegangsrechten.

### Syntaxis:

```
Section (access | application)
```

Als er niets is gespecificeerd, wordt **section application** verondersteld. De definitie van **section** is geldig totdat er een nieuwe **section**-opdracht wordt gemaakt.



### Voorbeeld:

```
section access;  
section application;
```

## Select

U selecteert velden uit een ODBC-gegevensbron of van een OLE DB-provider met de standaard **SELECT**-opdrachten van SQL. Of de **SELECT**-opdrachten worden geaccepteerd, hangt af van het gebruikte ODBC-stuurprogramma of de gebruikte OLE DB-provider. Voor het gebruik van de **SELECT** -opdracht is een open gegevensverbinding naar de bron vereist.

### Syntaxis:

```
Select [all | distinct | distinctrow | top n [percent] ] fieldlist  
  
From tablelist  
  
[where criterion ]  
  
[group by fieldlist [having criterion ] ]  
  
[order by fieldlist [asc | desc] ]  
  
[ (Inner | Left | Right | Full) join tablename on fieldref = fieldref ]
```

Verder kunnen diverse **SELECT**-instructies soms tot één instructie worden samengevoegd door middel van de operator **union**:

```
selectstatement Union selectstatement
```

Omdat de **SELECT**-opdracht wordt geïnterpreteerd door het ODBC-stuurprogramma of de OLE DB-provider, zijn er soms afwijkingen van de algemene SQL-syntaxis, afhankelijk van de mogelijkheden van de ODBC-stuurprogramma's of OLE DB-provider, bijvoorbeeld:

- **as** is niet altijd toegestaan, *aliasname* moet dan direct achter *fieldname* worden geplaatst.
- Soms is **as** juist verplicht bij het gebruik van *aliasname*.
- **distinct**, **as**, **where**, **group by**, **order by** of **union** worden soms niet ondersteund.
- Het ODBC-stuurprogramma accepteert niet altijd alle verschillende soorten aanhalingstekens.



*Deze beschrijving van de SQL-opdracht **SELECT** is niet volledig! **SELECT**-opdrachten kunnen bijvoorbeeld worden genest of worden samengevoegd tot een enkele **SELECT**-opdracht, en het aantal toegestane functies in een uitdrukking is soms heel groot.*

### Argumenten:

#### Argumenten

Argument	Beschrijving
distinct	<b>distinct</b> wordt gebruikt om waarden die dubbel voorkomen in de geselecteerde velden, maar één keer te laden.
distinctrow	<b>distinctrow</b> wordt gebruikt om records die dubbel voorkomen in de brontabel, maar één keer te laden.
fieldlist	<p><b>fieldlist ::= (*  field ) {, field }</b>            Een lijst van de te selecteren velden. Gebruik een sterretje * als veldlijst voor alle velden in de tabel.</p> <p><b>fieldlist ::= field {, field }</b>            Een lijst van een of meer velden, gescheiden door komma's.</p> <p><b>field ::= ( fieldref   expression ) [as aliasname ]</b>            De uitdrukking kan bijvoorbeeld een numerieke of tekenreeksfunctie zijn, gebaseerd op een of meer andere velden. Enkele van de meestal geaccepteerde operatoren en functies zijn: +, -, *, /, &amp; (aaneengeschakelde tekenreeks), sum(fieldname), count (fieldname), avg(fieldname)(average), month(fieldname), etc. Zie de documentatie van het ODBC-stuurprogramma voor meer informatie.</p> <p><b>fieldref ::= [ tablename. ] fieldname</b>  <b>tablename</b> en <b>fieldname</b> zijn teksttekenreeksen voor de tabelnaam en veldnaam. Ze moeten tussen rechte dubbele aanhalingstekens worden geplaatst als ze bijvoorbeeld spaties bevatten.            De <b>as</b>-clausule wordt gebruikt om een nieuwe naam toe te wijzen aan het veld.</p>
from	<p><b>tablelist ::= table {, table }</b>            De lijst van tabellen waaruit de velden worden geselecteerd.</p> <p><b>table ::= tablename [ [as ] aliasname ]</b>  <b>tablename</b> mag tussen aanhalingstekens worden geplaatst, maar dat hoeft niet.</p>
where	<p><b>where</b> is een clause die wordt gebruikt om aan te geven of een record wel of niet in de selectie moet worden opgenomen.</p> <p><b>criterion</b> is een logische uitdrukking, die zeer complex kan zijn. Enkele geaccepteerde operatoren zijn: numerieke operatoren en functies, =, &lt;&gt; of #(niet gelijk), &gt;, &gt;=, &lt;, &lt;=, <b>and</b>, <b>or</b>, <b>not</b>, <b>exists</b>, <b>some</b>, <b>all</b>, <b>in</b> en ook nieuwe <b>SELECT</b>-instructies. Zie de documentatie van het ODBC-stuurprogramma of de OLE DB-provider voor meer informatie.</p>
group by	Met <b>group by</b> -clausule worden diverse records geaggregeerd (gegroepeerd) tot een enkele record. Binnen een groep moeten alle records voor een bepaald veld dezelfde waarde hebben, anders kan het veld alleen binnen een uitdrukking worden gebruikt, bijvoorbeeld als som of gemiddelde. De op een of meer velden gebaseerde uitdrukking wordt gedefinieerd in de uitdrukking van het veldsymbool.

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
having	Met <b>having</b> worden groepen gekwalificeerd, net zoals bij de <b>where</b> -clausule records worden gekwalificeerd.
order by	<b>order by</b> is een clausule die de sorteervolgorde aangeeft van de tabel die door de <b>SELECT</b> -opdracht ontstaat.
join	<b>join</b> verenigt verschillende tabellen tot één tabel. Veld- en tabelnamen moeten tussen aanhalingstekens worden geplaatst als ze spaties of letters uit de nationale tekenset bevatten. Als het script automatisch door Qlik Sense wordt gegenereerd, worden de aanhalingstekens gebruikt die worden geaccepteerd door het ODBC-stuurprogramma dat of de OLE DB-provider die is opgegeven in de gegevensbrondefinities van de opdracht <b>Connect</b> .

### Example 1:

```
SELECT * FROM `Categories`;
```

### Example 2:

```
SELECT `Category ID`, `Category Name` FROM `Categories`;
```

### Example 3:

```
SELECT `Order ID`, `Product ID`,  
`Unit Price` * Quantity * (1-Discount) as NetSales  
FROM `Order Details`;
```

### Example 4:

```
SELECT `Order Details`.`Order ID`,  
Sum(`Order Details`.`Unit Price` * `Order Details`.Quantity) as `Result`  
FROM `Order Details`, Orders  
where Orders.`Order ID` = `Order Details`.`Order ID`  
group by `Order Details`.`Order ID`;
```

## Set

Met de opdracht **set** legt u scriptvariabelen vast. Deze kunnen worden gebruikt om tekenreeksen, paden, stations en dergelijke te vervangen.

### Syntaxis:

```
Set variablename=string
```

### Example 1:

```
Set FileToUse=Data1.csv;
```

### Example 2:

```
Set Constant="My string";
```

### Example 3:

```
Set BudgetYear=2012;
```

## Sleep

Met de opdracht **sleep** onderbreekt u de uitvoering van een script voor een bepaalde tijd.

### Syntaxis:

```
Sleep n
```

### Argumenten:

Argument	Beschrijving
n	Wordt opgegeven in milliseconden, waarbij <i>n</i> een positief geheel getal is dat niet groter is dan 3600000 (1 uur). De waarde mag ook een uitdrukking zijn.

### Example 1:

```
Sleep 10000;
```

### Example 2:

```
Sleep t*1000;
```

## SQL

Met de opdracht **SQL** kunt u via een SQL- of ODBC-verbinding een willekeurige OLE DB-opdracht verzenden.

### Syntaxis:

```
SQL sql_command
```

Als u SQL-opdrachten verzendt waarmee de database wordt bijgewerkt, wordt er een foutcode geretourneerd, als Qlik Sense de ODBC-verbinding in de modus alleen-lezen heeft geopend.

De syntaxis:

```
SQL SELECT * from tab1;
```

## 2 Scriptinstructies en sleutelwoorden

---

is toegestaan en wordt voor de consistentie als voorkeurssyntaxis voor **SELECT** aangeduid. Het prefix SQL voor de **SELECT**-opdracht blijft echter optioneel.

### Argumenten:

Argument	Beschrijving
<i>sql_command</i>	Een geldige SQL-opdracht.

### Example 1:

```
SQL leave;
```

### Example 2:

```
SQL Execute <storedProc>;
```

## SQLColumns

Met de opdracht **sqlcolumns** wordt een set velden geretourneerd waarin de kolommen worden beschreven van een ODBC- of OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

### Syntaxis:

```
SQLcolumns
```

Als u de velden combineert met de velden die worden verkregen met de opdrachten **sqltables** en **sqltypes**, krijgt u een goed overzicht van een bepaalde database. De twaalf standaardvelden zijn:

TABLE\_QUALIFIER

TABLE\_OWNER

TABLE\_NAME

COLUMN\_NAME

DATA\_TYPE

TYPE\_NAME

PRECISION

LENGTH

SCALE

RADIX

NULLABLE

REMARKS

In een ODBC-referentiehandleiding vindt u een gedetailleerde omschrijving van deze velden.

### Voorbeeld:

```
Connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';
SQLColumns;
```



*Sommige ODBC-stuurprogramma's ondersteunen deze opdracht mogelijk niet. Sommige ODBC-stuurprogramma's produceren mogelijk extra velden.*

## SQLTables

Met de opdracht **sqltables** wordt een set velden geretourneerd waarin de tabellen worden beschreven van een ODBC- of OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

### Syntaxis:

```
SQLTables
```

Als u de velden combineert met de velden die worden verkregen met de opdrachten **sqlcolumns** en **sqltypes**, krijgt u een goed overzicht van een bepaalde database. De vijf standaardvelden zijn:

TABLE\_QUALIFIER

TABLE\_OWNER

TABLE\_NAME

TABLE\_TYPE

REMARKS

In een ODBC-referentiehandleiding vindt u een gedetailleerde omschrijving van deze velden.

### Voorbeeld:

```
Connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';
SQLTables;
```



*Sommige ODBC-stuurprogramma's ondersteunen deze opdracht mogelijk niet. Sommige ODBC-stuurprogramma's produceren mogelijk extra velden.*

## SQLTypes

Met de opdracht **sqltypes** wordt een set velden geretourneerd waarin de typen worden beschreven van een ODBC- of OLE DB-gegevensbron waarmee via **connect** verbinding is gemaakt.

### Syntaxis:

```
SQLTypes
```

## 2 Scriptinstructies en sleutelwoorden

---

Als u de velden combineert met de velden die worden verkregen met de opdrachten **sqlcolumns** en **sqltables**, krijgt u een goed overzicht van een bepaalde database. De vijftien standaardvelden zijn:

TYPE\_NAME  
DATA\_TYPE  
PRECISION  
LITERAL\_PREFIX  
LITERAL\_SUFFIX  
CREATE\_PARAMS  
NULLABLE  
CASE\_SENSITIVE  
SEARCHABLE  
UNSIGNED\_ATTRIBUTE  
MONEY  
AUTO\_INCREMENT  
LOCAL\_TYPE\_NAME  
MINIMUM\_SCALE  
MAXIMUM\_SCALE

In een ODBC-referentiehandleiding vindt u een gedetailleerde omschrijving van deze velden.

### Voorbeeld:

```
Connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';  
SQLTypes;
```



*Sommige ODBC-stuurprogramma's ondersteunen deze opdracht mogelijk niet. Sommige ODBC-stuurprogramma's produceren mogelijk extra velden.*

### Star

Met de opdracht **star** kunt u een tekenreeks opgeven die de set van alle waarden van een veld in een database vertegenwoordigt. De opdracht beïnvloedt de **LOAD**- en **SELECT**-opdrachten die volgen.

### Syntaxis:

```
Star is[ string ]
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
string	<p>Een willekeurige tekst. Houd er rekening mee dat de tekenreeks tussen aanhalingstekens moet worden geplaatst als deze spaties bevat.</p> <p>Als er niets is aangegeven, wordt <b>star is</b>; verondersteld, d.w.z. dat er geen sterretje beschikbaar is tenzij er expliciet een is opgegeven. Deze definitie is geldig totdat er een nieuwe <b>star</b>-opdracht wordt gemaakt.</p>

Het wordt niet aangeraden om de instructie **Star is** te gebruiken in het gegevensdeel van het script (onder **Section Application**) als er gebruik wordt gemaakt van sectietoegang. Het sterretje wordt echter volledig ondersteund voor de beveiligde velden in het gedeelte **Section Access** van het script. In dit geval hoeft u de expliciete instructie **Star is** niet te gebruiken, aangezien deze altijd impliciet is in sectietoegang.

### Beperkingen

- U kunt het sterretje niet gebruiken in sleutelvelden, oftewel velden waarmee tabellen worden gekoppeld.
- U kunt het sterretje niet gebruiken in velden die worden beïnvloed door de instructie **Unqualify**, aangezien dit mogelijk van invloed is op velden waarmee tabellen worden gekoppeld.
- U kunt het sterretje niet gebruiken in niet-logische tabellen, zoals tabellen voor het laden van informatie of voor het laden van toewijzingen.
- Wanneer u het sterretje gebruikt in een beperkend veld (een veld met een koppeling naar de gegevens) in sectietoegang, stelt het de waarden voor die in dit veld voorkomen in de sectietoegang. Het sterretje stelt geen andere waarden voor die mogelijk voorkomen in de gegevens maar niet worden vermeld in de sectietoegang.
- U kunt het sterretje niet gebruiken in velden die worden beïnvloed door enige vorm van gegevensreductie buiten het gebied **Section Access**.

### Voorbeeld

Het onderstaande voorbeeld is een samenvatting van een load-script voor gegevens met sectietoegang.

```
star is *;
```

```
Section Access;
```

```
LOAD * INLINE [
```

```
ACCESS, USERID, OMIT
```

```
ADMIN, ADMIN,
```

```
USER, USER1, SALES
```



## 2 Scriptinstructies en sleutelwoorden

---

```
USER, USER2, WAREHOUSE
```

```
USER, USER3, EMPLOYEES
```

```
USER, USER4, SALES
```

```
USER, USER4, WAREHOUSE
```

```
USER, USER5, *
```

```
];
```

```
Section Application;
```

```
LOAD * INLINE [
```

```
SALES, WAREHOUSE, EMPLOYEES, ORDERS
```

```
1, 2, 3, 4
```

```
];
```

Het volgende is van toepassing:

- Het symbool *Star* is \*.
- De gebruiker *ADMIN* ziet alle velden. Er wordt niets weggelaten.
- De gebruiker *USER1* kan het veld *SALES* niet bekijken.
- De gebruiker *USER2* kan het veld *WAREHOUSE* niet bekijken.
- De gebruiker *USER3* kan het veld *EMPLOYEES* niet bekijken.
- De gebruiker *USER4* wordt tweemaal toegevoegd aan de oplossing om twee velden voor deze gebruiker WEG TE LATEN, namelijk *SALES* en *WAREHOUSE*.
- *USER5* heeft een '\*' wat betekent dat alle getoonde velden in OMIT niet beschikbaar zijn. Gebruiker *USER5* kan de velden *SALES*, *WAREHOUSE* en *EMPLOYEES* dus niet zien, maar deze gebruiker kan wel het veld *ORDERS* zien.

### Store

De **Store**-instructie maakt een QVD-, CSV- of text-bestand.

#### Syntaxis:

```
Store [ fieldlist from] table into filename [ format-spec ];
```

Met deze opdracht wordt een expliciet met de naam QVD, CSV of TXT gemaakt.

De opdracht kan alleen velden exporteren uit één gegevenstabel. Als velden uit diverse tabellen moeten worden geëxporteerd, moet eerst een expliciete join in het script worden opgegeven om de te exporteren gegevenstabel te maken.

## 2 Scriptinstructies en sleutelwoorden

De tekstwaarden worden in de indeling UTF-8 naar het CSV-bestand geëxporteerd. U kunt een scheidingsteken opgeven, zie **LOAD**. De **store**-opdracht voor een CSV-bestand ondersteunt geen BIFF-export.

### Argumenten:

#### Opdrachtargumenten opslaan

Argument	Beschrijving
<i>fieldlist</i> ::= ( *   <i>field</i> ) { , <i>field</i> }	<p>Een lijst van de te selecteren velden. Met een * als veldlijst worden alle velden aangeduid.</p> <p><i>field</i>::= <i>fieldname</i> [<b>as</b> <i>aliasname</i> ]</p> <p><i>fieldname</i> is een tekst die identiek is aan een veldnaam in <i>table</i>. (Houd er rekening mee dat de veldnaam tussen rechte dubbele aanhalingstekens of vierkante haken moet staan als deze spaties of andere niet-standaardtekens bevat.)</p> <p><i>aliasname</i> is een alternatieve naam voor het veld, te gebruiken in het resulterende QVD- of CSV-bestand.</p>
<i>table</i>	<p>Een scriptlabel representeert een reeds geladen tabel die als gegevensbron wordt gebruikt.</p>
<i>filename</i>	<p>De naam van het doelbestand inclusief een geldig pad naar een bestaande mapgegevensverbinding.</p> <p><b>Voorbeeld: 'lib://Table Files/target.qvd'</b></p> <p>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:</p> <ul style="list-style-type: none"><li>• absoluut</li></ul> <p><b>Voorbeeld: c:\data\sales.qvd</b></p> <ul style="list-style-type: none"><li>• relatief ten opzichte van de werkmap van de Qlik Sense-app.</li></ul> <p><b>Voorbeeld: data\sales.qvd</b></p> <p>Als er geen pad is opgegeven, bewaart Qlik Sense het bestand in de map die wordt gespecificeerd in de <b>Directory</b>-instructie. Als er geen <b>Directory</b>-instructie is opgegeven, bewaart Qlik Sense het bestand in de werkmap, <i>C:\Users\{user}\Documents\Qlik\Sense\Apps</i>.</p>

## 2 Scriptinstructies en sleutelwoorden

Argument	Beschrijving
<code>format-spec ::= ( ( txt   qvd ) )</code>	De opmaakspecificatie bestaat uit de tekst <b>txt</b> voor tekstbestanden, of de tekst <b>qvd</b> voor qvd-bestanden. Als de opmaakspecificatie wordt weggelaten, wordt <b>qvd</b> verondersteld.

### Voorbeelden:

```
store mytable into xyz.qvd (qvd);
store * from mytable into 'lib://FolderConnection/myfile.qvd';
store Name, RegNo from mytable into xyz.qvd;
store Name as a, RegNo as b from mytable into 'lib://FolderConnection/myfile.qvd';
store mytable into myfile.txt (txt);
store * from mytable into 'lib://FolderConnection/myfile.qvd';
```



*Bestandsextensies van DataFiles-verbindingen zijn hoofdlettergevoelig. Bijvoorbeeld: .qvd.*

## Table/Tables

De sleutelwoorden **Table** en **Tables** voor scripts worden gebruikt in opdrachten **Drop**, **Comment** en **Rename**, alsmede als indelingsspecificatie in **Load**-opdrachten.

## Tag

Met de scriptopdracht kunnen tags aan één of meer velden of tabellen worden toegewezen. Als wordt geprobeert om een tag toe te wijzen aan een veld of tabel die niet bestaat in de app, wordt de tag genegeerd. Als conflicterende exemplaren van een veld- of tagnaam zijn aangetroffen, wordt de laatste waarde gebruikt.

### Syntaxis:

```
Tag [field|fields] fieldlist with tagname
```

```
Tag [field|fields] fieldlist using mapname
```

```
Tag table tablelist with tagname
```

### Argumenten

Argument	Beschrijving
fieldlist	Aan één of meerdere velden moet een tag worden toegewezen in een door komma's gescheiden lijst.
mapname	De naam van een toewijzingstabel die eerder is gelezen in een <b>mapping Load</b> - of <b>mapping Select</b> -instructie.

Argument	Beschrijving
tablelist	Een door komma's gescheiden lijst van de tabellen waaraan een tag moet worden toegewezen.
tagname	De naam van de tag die moet worden toegepast op het veld.

### Example 1:

```
tagmap:  
mapping LOAD * inline [  
a,b  
Alpha,MyTag  
Num,MyTag  
];  
tag fields using tagmap;
```

### Example 2:

```
tag field Alpha with 'MyTag2';
```

## Trace

Met de opdracht **trace** wordt eventueel een tekenreeks naar het venster **Voortgang scriptuitvoering** en in het scriptlogbestand geschreven. Dit is zeer handig bij het opsporen van fouten. Met behulp van \$-uitbreidingen van variabelen die zijn berekend vóór de **trace**-opdracht, kunt u het bericht aanpassen.

### Syntaxis:

```
Trace string
```

### Example 1:

De volgende instructie kan worden gebruikt meteen na de Load-instructie waarmee de 'Main'-tabel wordt geladen.

```
Trace Main table loaded;
```

Hiermee wordt de tekst 'Main table loaded' weergegeven in het scriptuitvoeringsvenster en in het logbestand.

### Example 2:

De volgende instructies kunnen worden gebruikt meteen na de Load-instructie waarmee de 'Main'-tabel wordt geladen.

```
Let MyMessage = NoOfRows('Main') & ' rows in Main table';
```

```
Trace $(MyMessage);
```

Hiermee wordt in het scriptuitvoeringsvenster en in het logbestand een tekst weergegeven die het aantal rijen aangeeft, bijvoorbeeld '265,391 rows in Main table'.

### Unmap

Met de opdracht **Unmap** wordt de toewijzing van veldwaarden die is vastgelegd door een eerdere opdracht **Map ... Using**, uitgeschakeld voor velden die daarna worden geladen.

#### Syntaxis:

```
Unmap *fieldlist
```

#### Argumenten:

##### Argumenten

Argument	Beschrijving
*fieldlist	Een door komma's gescheiden lijst met velden die vanaf dit punt in het script niet meer moeten worden toegewezen. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens.

#### Voorbeelden en resultaten:

Voorbeeld	Resultaat
Unmap Country;	De toewijzing van het veld Country wordt uitgeschakeld.
Unmap A, B, C;	De toewijzing van de velden A, B en C wordt uitgeschakeld.
Unmap * ;	De toewijzing van alle velden wordt uitgeschakeld.

### Unqualify

Met de opdracht **Unqualify** wordt de kwalificatie van veldnamen die eerder was ingeschakeld met de opdracht **Qualify**, uitgeschakeld.

#### Syntaxis:

```
Unqualify *fieldlist
```

#### Argumenten:

##### Argumenten

Argument	Beschrijving
*fieldlist	Een door komma's gescheiden lijst met velden waarvoor kwalificatie moet worden ingeschakeld. Met een * als veldlijst worden alle velden aangeduid. De jokertekens * en ? zijn toegestaan in de veldnamen. Het gebruik van aanhalingstekens rond veldnamen kan nodig zijn bij het gebruik van jokertekens.  In de documentatie voor de opdracht <b>Qualify</b> vindt u meer informatie.

### Example 1:

In een onbekende database is het vaak handig om in het begin alleen een enkel veld of een paar velden te koppelen, zoals in het volgende voorbeeld:

```
qualify *;  
unqualify TransID;  
SQL SELECT * from tab1;  
SQL SELECT * from tab2;  
SQL SELECT * from tab3;
```

Eerst wordt de kwalificatie voor alle velden uitgeschakeld.

Vervolgens wordt de kwalificatie uitgeschakeld voor **TransID**.

Alleen **TransID** zal worden gebruikt voor associaties tussen de tabellen *tab1*, *tab2* en *tab3*. Alle andere velden worden gekwalificeerd met de tabelnaam.

## Untag

Met de scriptopdracht kunnen tags van één of meer velden of tabellen worden verwijderd. Als wordt geprobeert om een tag te verwijderen van een veld of tabel die niet bestaat in de app, wordt de tag niet verwijderd.

### Syntaxis:

```
Untag [field|fields] fieldlist with tagname
```

```
Untag [field|fields] fieldlist using mapname
```

```
Untag table tablelist with tagname
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
fieldlist	Eén of meerdere velden waarvan een label moet worden verwijderd in een door komma's gescheiden lijst.
mapname	De naam van een toewijzingstabel die eerder is geladen in een instructie mapping <b>LOAD</b> of mapping <b>SELECT</b> .
tablelist	Een door komma's gescheiden lijst van de tabellen waaruit een tag moet worden verwijderd.
tagname	De naam van de tag met extra info die moet worden verwijderd uit het veld.

### Example 1:

```
tagmap:  
mapping LOAD * inline [  
a,b  
Alpha,MyTag
```

```
Num,MyTag  
];  
Untag fields using tagmap;
```

### Example 2:

```
Untag field Alpha with MyTag2;
```

## 2.6 Werkdirectory

Als u verwijst naar een bestand in een scriptopdracht en het pad wordt weggelaten, zoekt Qlik Sense in deze volgorde naar het bestand:

1. De directory die is opgegeven door een opdracht **Directory** (alleen ondersteund in de bestaande scriptmodus).
2. Als er geen opdracht **Directory** is, zoekt Qlik Sense in de werkdirectory.

### Qlik Sense Desktop werkdirectory

In Qlik Sense Desktop is `C:\Users\{user}\Documents\Qlik\Sense\Apps` de werkdirectory.

### Qlik Sense werkdirectory

In een Qlik Sense-serverinstallatie wordt de werkdirectory opgegeven in Qlik Sense Repository Service. Standaard is dit `C:\ProgramData\Qlik\Sense\Apps`. Raadpleeg de Help van Qlik Management Console voor meer informatie.

## 2 Werken met variabelen in de editor voor het laden van gegevens

Een variabele in Qlik Sense is een container waarin een statische waarde of een berekening wordt opgeslagen, bijvoorbeeld een numerieke of alfanumerieke waarde. Als u de variabele in de app gebruikt, wordt een eventuele wijziging die wordt aangebracht in de variabele overall toegepast waar de variabele wordt gebruikt. U kunt variabelen definiëren in het overzicht Variabelen of in het script met de editor voor het laden van gegevens. U kunt de waarde van variabelen instellen met de instructies **Let** of **Set** in het load-script voor gegevens.



*U kunt tevens werken met de Qlik Sense-variabelen uit het overzicht van variabelen bij de bewerking van een werkblad.*

### 2.7 Overzicht

Als het eerste teken van een variabelewaarde het isgelijktteken '=' is, probeert Qlik Sense de waarde als formule (Qlik Sense-uitdrukking) te interpreteren. Vervolgens wordt niet de formuletekst, maar het resultaat weergegeven of geretourneerd.

Bij het gebruik van de variabele wordt deze vervangen door de waarde ervan. Variabelen kunnen in het script worden gebruikt voor uitbreiding met een dollarteken en in verschillende besturingsopdrachten. Dat is zeer handig als dezelfde tekenreeks vaak in het script wordt herhaald, bijvoorbeeld een pad.

Bepaalde speciale systeemvariabelen worden aan het begin van de scriptuitvoering door Qlik Sense ingesteld, ongeacht hun eerdere waarden.

### 2.8 Een variabele definiëren

Variabelen bieden de mogelijkheid om statische waarden of het resultaat van een berekening op te slaan. Gebruik de volgende syntaxis bij het definiëren van een variabele:

```
set variablename = string
```

of

```
let variable = expression
```

De **Set**-instructie wordt gebruikt voor tekenreekstoeewijzing. Het wijst de tekst rechts van het isgelijktteken toe aan de variabele. De **Let**-instructie evalueert een uitdrukking rechts van het isgelijktteken tijdens de runtime van het script en wijst het resultaat toe aan de uitdrukking van de variabele.

Variabelen zijn hoofdlettergevoelig.



## 2 Werken met variabelen in de editor voor het laden van gegevens

---



Het wordt niet aanbevolen om een variabele dezelfde naam te geven als een veld of functie in Qlik Sense.

### Voorbeelden:

```
set x = 3 + 4; // de variabele zal de tekenreeks '3 + 4' ophalen als de waarde.
```

```
let x = 3 + 4; // retourneert 7 als de waarde.
```

```
set x = Today(); // retourneert 'Today()' als de waarde.
```

```
let x = Today(); // retourneert de datum van vandaag als de waarde, bijvoorbeeld '9/27/2021'.
```

## 2.9 Een variabele verwijderen

Als u een variabele uit het script verwijdert en de gegevens opnieuw laadt, blijft de variabele in de app. Als u de variabele volledig uit de app wilt verwijderen, moet u de variabele ook uit het variabelendialoogvenster verwijderen.

## 2.10 De waarde van een variabele laden als een veldwaarde

Als u de waarde van een variabele wil laden als een veldwaarde in een **LOAD**-opdracht en het resultaat van de dollaruitbreiding is tekst in plaats van numeriek of een uitdrukking, moet u de uitgebreide variabele tussen enkele aanhalingstekens zetten.

### Voorbeeld:

In dit voorbeeld wordt de systeemvariabele die de lijst met scriptfouten bevat in een tabel geladen. Zoals u ziet, zijn voor uitbreiding van in de clause geen aanhalingstekens vereist, terwijl bij de uitbreiding van aanhalingstekens zijn vereist. `ScriptErrorCountScriptErrorList`

```
IF $(ScriptErrorCount) >= 1 THEN  
  
    LOAD '$(ScriptErrorList)' AS Error AutoGenerate 1; END IF
```

## 2.11 Berekeningen met variabelen

Er zijn diverse manieren om variabelen te gebruiken met berekende waarden in Qlik Sense, en het resultaat is afhankelijk van de wijze waarop u de variabele definieert en aanroept in een uitdrukking.

In dit voorbeeld laden we enkele inline-gegevens:

```
LOAD * INLINE [  
    Dim, Sales  
    A, 150  
    A, 200  
    B, 240  
    B, 230  
    C, 410
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

c, 330

];

Eerst definiëren we twee variabelen:

```
Let vSales = 'Sum(Sales)' ;
```

```
Let vSales2 = '=Sum(Sales)' ;
```

In de tweede variabele plaatsen we een isgelijkteken voor de uitdrukking. Hierdoor wordt de variabele berekend voordat de variabele wordt uitgebreid en de uitdrukking wordt geëvalueerd.

Als u de variabele vSales zonder toevoeging gebruikt, is het resultaat de tekenreeks Sum(Sales) en wordt er geen berekening uitgevoerd.

Als u een dollar-tekenuitbreiding toevoegt en \$(vSales) in de uitdrukking aanroept, wordt de variabele uitgebreid en wordt de som van Sales weergegeven.

En tot slot als u \$(vSales2) aanroept, wordt de variabele eerst berekend en daarna uitgebreid. Dit betekent dat het weergegeven resultaat de totale som van Sales is. Het verschil tussen=\$(vSales) en=\$(vSales2) metinguitdrukkingen wordt duidelijk uit de tabel met de resultaten:

Resultaten

Dim	\$(vSales)	\$(vSales2)
A	350	1560
B	470	1560
C	740	1560

Zoals u ziet, is het resultaat van \$(vSales) de deelsom voor een dimensiewaarde, terwijl het resultaat van \$(vSales2) de totale som is.

De volgende scriptvariabelen zijn beschikbaar:

- *Foutvariabelen (page 267)*
- *Variabelen voor getalinterpretatie (page 203)*
- *Systeemvariabelen (page 194)*
- *Variabelen voor afhandeling van waarden (page 201)*

### 2.12 Systeemvariabelen

Systeemvariabelen waarvan sommige door het systeem zijn gedefinieerd, geven informatie over het systeem en de Qlik Sense-app.

#### Overzicht van systeemvariabelen

Sommige functies worden na het overzicht nader beschreven. Bij deze functies kunt u ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Floppy

Retourneert de stationsletter van het eerst gevonden diskettestation. Dit is normaal gesproken *a:*. Dit is een door het systeem gedefinieerde variabele.

#### Floppy



*Deze variabele wordt niet ondersteund in de standaardmodus.*

### CD

Retourneert de stationsletter van het eerst gevonden cd-rom-station. Als er geen cd-rom-station is gevonden, wordt *c:* geretourneerd. Dit is een door het systeem gedefinieerde variabele.

#### CD



*Deze variabele wordt niet ondersteund in de standaardmodus.*

### Include

De variabele **Include/Must\_Include** specificeert een bestand dat tekst bevat die in het script moet worden opgenomen en moet worden geëvalueerd als scriptcode. Deze wordt niet gebruikt om gegevens toe te voegen. U kunt delen van uw scriptcode opslaan in een apart tekstbestand en dit in meerdere apps hergebruiken. Dit is een door de gebruiker gedefinieerde variabele.

```
$(Include=filename)
```

```
$(Must_Include=filename)
```

### HidePrefix

Alle veldnamen die beginnen met deze teksttekenreeks worden op dezelfde manier als de systeemvelden verborgen. Dit is een door de gebruiker gedefinieerde variabele.

#### HidePrefix

### HideSuffix

Alle veldnamen die eindigen met deze teksttekenreeks worden op dezelfde manier als de systeemvelden verborgen. Dit is een door de gebruiker gedefinieerde variabele.

#### HideSuffix

### QvPath

Retourneert het pad naar het Qlik Sense-programma. Dit is een door het systeem gedefinieerde variabele.

#### QvPath



*Deze variabele wordt niet ondersteund in de standaardmodus.*

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### QvRoot

Retourneert de hoofdmap van het Qlik Sense-programma. Dit is een door het systeem gedefinieerde variabele.

#### QvRoot



*Deze variabele wordt niet ondersteund in de standaardmodus.*

### QvWorkPath

Retourneert het pad naar de huidige Qlik Sense-app. Dit is een door het systeem gedefinieerde variabele.

#### QvWorkPath



*Deze variabele wordt niet ondersteund in de standaardmodus.*

### QvWorkRoot

Retourneert de hoofdmap van de huidige Qlik Sense-app. Dit is een door het systeem gedefinieerde variabele.

#### QvWorkRoot



*Deze variabele wordt niet ondersteund in de standaardmodus.*

### StripComments

Als deze variabele wordt ingesteld op 0, worden de opmerkingen in het script tussen `/*..*/` en na `//` niet verwijderd. Als deze variabele niet is ingesteld, worden opmerkingen altijd verwijderd.

#### StripComments

### Verbatim

Normaliter worden alle veldwaarden automatisch ontdaan van voorafgaande of opvolgende spaties (ASCII 32) voordat ze in de Qlik Sense-database worden geladen. Als u deze variabele instelt op 1, worden de spaties niet verwijderd. Tabs (ASCII 9) en harde spaties (ANSI 160) worden nooit verwijderd.

#### Verbatim

### OpenUrlTimeout

Deze variabele bepaalt de time-out in seconden die Qlik Sense moet aanhouden bij het ophalen van gegevens uit URL-bronnen (bijvoorbeeld -pagina's). HTML pagina's). Als deze variabele niet is ingesteld, is de time-out ongeveer 20 minuten.

#### OpenUrlTimeout

### WinPath

Retourneert het pad naar Windows. Dit is een door het systeem gedefinieerde variabele.

#### WinPath

## 2 Werken met variabelen in de editor voor het laden van gegevens

---



*Deze variabele wordt niet ondersteund in de standaardmodus.*

### **WinRoot**

Retourneert de hoofdmap van Windows. Dit is een door het systeem gedefinieerde variabele.

### **WinRoot**



*Deze variabele wordt niet ondersteund in de standaardmodus.*

### **CollationLocale**

Geeft op welke landinstellingen moeten worden gebruikt voor sorteervolgorde en zoekafstemming. De waarde is de cultuurnaam van een landinstelling, bijvoorbeeld 'en-US'. Dit is een door het systeem gedefinieerde variabele.

### **CollationLocale**

### **CreateSearchIndexOnReload**

Deze variabele wordt gebruikt om aan te geven of zoekindexbestanden moeten worden gemaakt wanneer gegevens opnieuw worden geladen.

### **CreateSearchIndexOnReload**

## CreateSearchIndexOnReload

Deze variabele wordt gebruikt om aan te geven of zoekindexbestanden moeten worden gemaakt wanneer gegevens opnieuw worden geladen.

### **Syntaxis:**

### **CreateSearchIndexOnReload**

U kunt instellen of u wilt dat zoekindexbestanden worden gemaakt tijdens het opnieuw laden van gegevens of na de eerste zoekopdracht van de gebruiker. Het voordeel van het maken van zoekindexbestanden tijdens het opnieuw laden van gegevensbestanden is dat u niet zoals de eerste gebruiker van de zoekopdracht moet wachten. De afweging is dat het langer duurt om gegevens opnieuw te laden wanneer een zoekindex wordt gemaakt.

Als deze variabele wordt weggelaten, worden tijdens het opnieuw laden van de gegevens geen zoekindexbestanden gemaakt.



*Voor sessieapps worden geen zoekindexbestanden gemaakt tijdens het opnieuw laden van gegevens, ongeacht de instelling van deze variabele.*

### **Example 1: Zoekindexbestanden maken tijdens het opnieuw laden van gegevens**

```
set CreateSearchIndexOnReload=1;
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Example 2: Zoekindexbestanden maken na de eerste zoekopdracht

```
set CreateSearchIndexOnReload=0;
```

### HidePrefix

Alle veldnamen die beginnen met deze teksttekenreeks worden op dezelfde manier als de systeemvelden verborgen. Dit is een door de gebruiker gedefinieerde variabele.

#### Syntaxis:

```
HidePrefix
```

#### Voorbeeld:

```
set HidePrefix='_ ' ;
```

Bij gebruik van deze opdracht worden de veldnamen die beginnen met een liggend streepje niet getoond in de lijst met veldnamen als de systeemvelden verborgen zijn.

### HideSuffix

Alle veldnamen die eindigen met deze teksttekenreeks worden op dezelfde manier als de systeemvelden verborgen. Dit is een door de gebruiker gedefinieerde variabele.

#### Syntaxis:

```
HideSuffix
```

#### Voorbeeld:

```
set HideSuffix='% ' ;
```

Bij gebruik van deze opdracht worden de veldnamen die eindigen met een procentteken niet getoond in de lijst met veldnamen als de systeemvelden verborgen zijn.

### Include

De variabele **Include/Must\_Include** specificeert een bestand dat tekst bevat die in het script moet worden opgenomen en moet worden geëvalueerd als scriptcode. Deze wordt niet gebruikt om gegevens toe te voegen. U kunt delen van uw scriptcode opslaan in een apart tekstbestand en dit in meerdere apps hergebruiken. Dit is een door de gebruiker gedefinieerde variabele.



*Deze variabele ondersteunt alleen mapgegevensverbindingen in de standaardmodus.*

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Syntaxis:

```
$(Include=filename)
```

```
$(Must_Include=filename)
```

Er zijn twee versies van de variabele:

- **Include** genereert geen fout als het bestand niet wordt gevonden. De bewerking mislukt zonder melding.
- **Must\_Include** genereert een fout als het bestand niet wordt gevonden.

Als u geen pad opgeeft, is de bestandsnaam gerelateerd aan de werkdirectory voor de Qlik Sense-app. U kunt ook een absoluut bestandspad opgeven of een pad naar een mapverbinding lib://. Voeg geen spatie toe voor of na het isgelijkteken.



*De constructie **set Include =filename** is niet van toepassing.*

### Voorbeelden:

```
$(Include=abc.txt);
```

```
$(Must_Include=lib://DataFiles/abc.txt);
```

## Beperkingen

### Beperkte cross-compatibiliteit tussen UTF-8 gecodeerde bestanden onder Windows versus Linux.

Het gebruik van UTF-8 met BOM (Byte Order Mark) is optioneel. BOM kan interfereren met het gebruik van UTF-8 in software die geen niet-ASCII bytes verwacht aan het begin van een bestand, maar die anders wel met de tekststroom overweg kan.

- Windows systemen gebruiken BOM in UTF-8 om aan te geven dat een bestand UTF-8 gecodeerd is, ook al is er geen dubbelzinnigheid in de byte-opslag.
- Unix/Linux gebruiken UTF-8 voor Unicode, maar gebruiken de BOM niet omdat dit interfereert met de syntaxis voor opdrachtbestanden.

Dit heeft een aantal consequenties voor Qlik Sense.

- In Windows wordt elk bestand dat begint met een UTF-8 BOM beschouwd als een UTF-8 scriptbestand. Anders wordt ANSI-codering verondersteld.
- In Linux is de standaard 8 bit codepagina UTF-8. Daarom werkt UTF-8, ook al bevat het geen BOM.

Zodoende kan de overdraagbaarheid niet worden gegarandeerd. Het is niet altijd mogelijk om onder Windows een bestand te maken dat door Linux kan worden weergegeven en omgekeerd. Er is geen onderlinge compatibiliteit tussen de twee systemen wat betreft UTF-8 gecodeerde bestanden, omdat de BOM verschillend wordt behandeld.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### OpenUrlTimeout

Deze variabele bepaalt de time-out in seconden die Qlik Sense moet aanhouden bij het ophalen van gegevens uit URL-bronnen (bijvoorbeeld -pagina's). HTML pagina's). Als deze variabele niet is ingesteld, is de time-out ongeveer 20 minuten.

**Syntaxis:**

```
OpenUrlTimeout
```

**Voorbeeld:**

```
set OpenUrlTimeout=10;
```

### StripComments

Als deze variabele wordt ingesteld op 0, worden de opmerkingen in het script tussen `/*..*/` en na `//` niet verwijderd. Als deze variabele niet is ingesteld, worden opmerkingen altijd verwijderd.

**Syntaxis:**

```
StripComments
```

Bepaalde databasestuurprogramma's geven tussen de tekens `/*..*/` hints voor het optimaliseren van **SELECT**-opdrachten. Als dit het geval is, moeten de opmerkingen niet worden verwijderd voordat de **SELECT**-opdracht naar het databasestuurprogramma wordt gestuurd.



*Het wordt aanbevolen deze variabele meteen na de betreffende opdrachten terug te zetten op 1.*

**Voorbeeld:**

```
set StripComments=0;  
SQL SELECT * /* <optimization directive> */ FROM Table ;  
set StripComments=1;
```

### Verbatim

Normaliter worden alle veldwaarden automatisch ontdaan van voorafgaande of opvolgende spaties (ASCII 32) voordat ze in de Qlik Sense-database worden geladen. Als u deze variabele instelt op 1, worden de spaties niet verwijderd. Tabs (ASCII 9) en harde spaties (ANSI 160) worden nooit verwijderd.

**Syntaxis:**

```
Verbatim
```



## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Voorbeeld:

```
set verbatim = 1;
```

## 2.13 Variabelen voor afhandeling van waarden

In dit hoofdstuk worden variabelen beschreven die worden gebruikt voor de afhandeling van NULL en andere waarden.

### Overzicht van variabelen voor afhandeling van waarden

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### **NullDisplay**

Het gedefinieerde symbool vervangt alle NULL-waarden van ODBC en connectoren op het laagste gegevensniveau. Dit is een door de gebruiker gedefinieerde variabele.

```
NullDisplay
```

#### **NullInterpret**

Als het gedefinieerde symbool in een tekstbestand, Excel-bestand of inline-uitdrukking staat, wordt het geïnterpreteerd als NULL. Dit is een door de gebruiker gedefinieerde variabele.

```
NullInterpret
```

#### **NullValue**

Als de opdracht **NullAsValue** wordt gebruikt, vervangt het gedefinieerde symbool alle NULL-waarden in de velden waarvoor **NullAsValue** is opgegeven, door de opgegeven tekenreeks.

```
NullValue
```

#### **OtherSymbol**

Geeft aan dat een symbool voor een **LOAD/SELECT**-uitdrukking moet worden behandeld zoals 'alle andere waarden'. Dit is een door de gebruiker gedefinieerde variabele.

```
OtherSymbol
```

### NullDisplay

Het gedefinieerde symbool vervangt alle NULL-waarden van ODBC en connectoren op het laagste gegevensniveau. Dit is een door de gebruiker gedefinieerde variabele.

#### **Syntaxis:**

```
NullDisplay
```

#### **Voorbeeld:**

```
set NullDisplay='<NULL>';
```

### NullInterpret

Als het gedefinieerde symbool in een tekstbestand, Excel-bestand of inline-uitdrukking staat, wordt het geïnterpreteerd als NULL. Dit is een door de gebruiker gedefinieerde variabele.

#### Syntaxis:

```
NullInterpret
```

#### Voorbeelden:

```
set NullInterpret=' ';  
set NullInterpret =;
```

retourneert geen NULL-waarden voor lege waarden Excel, maar wel voor een CSV-tekstbestand.

```
set NullInterpret ='';
```

retourneert NULL-waarden voor lege waarden in Excel.

### NullValue

Als de opdracht **NullAsValue** wordt gebruikt, vervangt het gedefinieerde symbool alle NULL-waarden in de velden waarvoor **NullAsValue** is opgegeven, door de opgegeven tekenreeks.

#### Syntaxis:

```
NullValue
```

#### Voorbeeld:

```
NullAsValue Field1, Field2;  
set NullValue='<NULL>';
```

### OtherSymbol

Geeft aan dat een symbool voor een **LOAD/SELECT**-uitdrukking moet worden behandeld zoals 'alle andere waarden'. Dit is een door de gebruiker gedefinieerde variabele.

#### Syntaxis:

```
OtherSymbol
```

#### Voorbeeld:

```
set otherSymbol='+';  
LOAD * inline  
[X, Y  
a, a  
b, b];  
LOAD * inline  
[X, Z  
a, a  
+, c];
```

De veldwaarde Y='b' is nu gekoppeld aan Z='c' via het andere symbool.

### 2.14 Variabelen voor getalinterpretatie

Variabelen voor getalinterpretatie worden door het systeem vastgesteld. De variabelen staan aan het begin van het load-script en passen instellingen voor de getalnotatie toe op het moment dat het script wordt uitgevoerd. Ze kunnen worden verwijderd, bewerkt of gedupliceerd.

Variabelen voor getalinterpretatie worden automatisch gegenereerd volgens de huidige landinstellingen van het besturingssysteem wanneer een nieuwe app wordt gemaakt. In Qlik Sense Desktop is dit afhankelijk van de instellingen van het besturingssysteem van de computer. In Qlik Sense is dit op basis van de instellingen van het besturingssysteem van de server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

#### Valutanotatie

##### **MoneyDecimalSep**

Het gedefinieerde scheidingsteken voor decimalen vervangt het decimaalteken voor valuta dat is ingesteld door uw landinstellingen.

```
MoneyDecimalSep
```

##### **MoneyFormat**

Het gedefinieerde symbool vervangt het valutasympool dat is ingesteld door uw landinstellingen.

```
MoneyFormat
```

##### **MoneyThousandSep**

Het gedefinieerde scheidingsteken voor duizendtallen vervangt het cijfergroeperingssymbool voor valuta dat is ingesteld door uw landinstellingen.

```
MoneyThousandSep
```

#### Getalnotatie

##### **DecimalSep**

Het gedefinieerde scheidingsteken voor decimalen vervangt het decimaalteken dat is ingesteld door uw landinstellingen.

```
DecimalSep
```

##### **ThousandSep**

Het gedefinieerde scheidingsteken voor duizendtallen vervangt het cijfergroeperingssymbool van het besturingssysteem (landinstellingen).

```
ThousandSep
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### **NumericalAbbreviation**

De numerieke afkorting bepaalt welke afkorting u moet gebruiken als het voorvoegsel dat de schaal van numerieke waarden aangeeft, bijvoorbeeld M voor mega of een miljoen ( $10^6$ ), en  $\mu$  voor micro ( $10^{-6}$ ).

**NumericalAbbreviation**

### **Tijdnotatie**

#### **DateFormat**

Deze omgevingsvariabele definieert de datumopmaak die standaard wordt gebruikt in de app. De opmaak wordt gebruikt om datums te interpreteren en op te maken. Als de variabele niet is gedefinieerd, wordt de datumopmaak van de regionale instellingen van het besturingssysteem opgehaald wanneer het script wordt uitgevoerd.

**DateFormat**

#### **TimeFormat**

De gedefinieerde notatie vervangt de tijdnotatie van het besturingssysteem (landinstellingen).

**TimeFormat**

#### **TimestampFormat**

De gedefinieerde notatie vervangt de datum- en tijdnotatie van het besturingssysteem (landinstellingen).

**TimestampFormat**

#### **MonthNames**

De gedefinieerde notatie vervangt de naamgevingsconventie voor maanden in de landinstellingen.

**MonthNames**

#### **LongMonthNames**

De gedefinieerde notatie vervangt de lange naamgevingsconventie voor maanden in de landinstellingen.

**LongMonthNames**

#### **DayNames**

De gedefinieerde notatie vervangt de naamgevingsconventie voor weekdays die zijn ingesteld door uw landinstellingen.

**DayNames**

#### **LongDayNames**

De gedefinieerde notatie vervangt de lange naamgevingsconventie voor weekdays in de landinstellingen.

**LongDayNames**

#### **FirstWeekDay**

Geheel getal dat definieert welke dag moet worden gebruikt als eerste dag van de week.

**FirstWeekDay**

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### BrokenWeeks

Deze instelling definieert of weken worden gesplitst of niet.

#### **BrokenWeeks**

### ReferenceDay

De instelling definieert welke dag in januari is ingesteld als referentiedag voor het definiëren van week 1.

#### **ReferenceDay**

### FirstMonthOfYear

De instelling definieert welke maand moet worden gebruikt als eerste maand van het jaar. Deze kan worden gebruikt voor het definiëren van boekjaren die een maandelijkse verschuiving hebben en bijvoorbeeld op 1 april starten.



*Deze instelling wordt momenteel niet gebruikt, maar is gereserveerd voor toekomstig gebruik.*

Geldige instellingen zijn 1 (januari) t/m 12 (december). De standaardinstelling is 1.

### Syntaxis:

#### **FirstMonthOfYear**

### Voorbeeld:

```
set FirstMonthOfYear=4; //Sets the year to start in April
```

## BrokenWeeks

Deze instelling definieert of weken worden gesplitst of niet.

### Syntaxis:

#### **BrokenWeeks**

Standaard wordt in Qlik Sense-functies gebruikgemaakt van ongebroken weken. Dit betekent dat:

- In sommige jaren, week 1 begint in december en in andere jaren week 52 of 53 doorloopt tot in januari.
- Week 1 altijd ten minste 4 dagen heeft in januari.

Het alternatief is om gebroken weken te gebruiken:

- Week 52 of 53 lopen niet door tot in januari.
- Week 1 begint op 1 januari en is, in de meeste gevallen, geen volle week.

De volgende waarden kunnen worden gebruikt:

- 0 (= ongebroken weken gebruiken)
- 1 (= gebroken weken gebruiken)

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

#### Voorbeelden:

```
Set BrokenWeeks=0; //(use unbroken weeks)
Set BrokenWeeks=1; //(use broken weeks)
```

### DateFormat

Deze omgevingsvariabele definieert de datumnotatie die als standaard wordt gebruikt in de app en op datumretourfuncties zoals `date()` en `date#()`. De notatie wordt gebruikt om datums te interpreteren en op te maken. Als de variabele niet is gedefinieerd, wordt de datumnotatie die is ingesteld door uw regionale instellingen opgehaald wanneer het script wordt uitgevoerd.

#### Syntaxis:

##### DateFormat

Voorbeelden functie DateFormat	
Voorbeeld	Resultaat
<pre>Set DateFormat='M/D/YY'; //(US format)</pre>	Dit gebruik van de functie <code>DateFormat</code> definieert de datum als de Amerikaanse notatie, maand/dag/jaar.
<pre>Set DateFormat='DD/MM/YY'; //( UK date format)</pre>	Dit gebruik van de functie <code>DateFormat</code> definieert de datum als de Britse notatie, dag/maand/jaar.
<pre>Set DateFormat='YYYY/MM/DD'; //(ISO date format)</pre>	Dit gebruik van de functie <code>DateFormat</code> definieert de datum als de ISO-notatie, jaar/maand/dag.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Standaardsysteemvariabelen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling van datums.
- De functie `DateFormat` die de Amerikaanse datumnotatie gebruikt.

In dit voorbeeld wordt een gegevensset geladen in de tabel 'Transactions'. Deze bevat een veld `date`. De Amerikaanse `DateFormat`-definitie wordt gebruikt. Dit patroon wordt gebruikt voor de omzetting van impliciete tekst naar een datum wanneer de tekstdatums worden geladen.

#### Load-script

```
Set DateFormat='MM/DD/YYYY';
```

```
Transactions:
LOAD
date,
month(date) as month,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

- date
- month

Maak deze meting:

```
=sum(amount)
```

Resultatentabel		
date	month	=sum(amount)
01/01/2022	Jan	1000
02/01/2022	Feb	2123
03/01/2022	Mar	4124
04/01/2022	Apr	2431

De definitie `DateFormat MM/DD/YYYY` wordt gebruikt voor de impliciete omzetting van tekst in datums en dit is waarom het veld `date` als een datum wordt geïnterpreteerd. Dezelfde notatie wordt gebruikt voor het weergeven van de datum, zoals in de resultatentabel wordt getoond.

### Voorbeeld 2 - Systeemvariabele wijzigen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensverzameling uit het vorige voorbeeld.
- De `DateFormat` functie die de notatie `'DD/MM/JJJJ'` gebruikt.

#### Load-script

```
SET DateFormat='DD/MM/YYYY';
Transactions:
LOAD
date,
month(date) as month,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```



## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- month

Maak deze meting:

=sum(amount)

Resultatentabel		
date	month	=sum(amount)
01/01/2022	Jan	1000
02/01/2022	Jan	2123
03/01/2022	Jan	4124
04/01/2022	Jan	2431

Omdat de definitie `DateFormat` is ingesteld op 'DD/MM/JJJJ' ziet u dat de twee getallen achter het eerste '/'-symbool zijn geïnterpreteerd als maand, waardoor alle records afkomstig zijn uit de maand januari.

### Voorbeeld 3 - Interpretatie van datum

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met datums in numerieke indeling.
- De variabele `DateFormat` die de notatie 'DD-MM-JJJJ' gebruikt.
- De `date()` variabele.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
date(numerical_date),
month(date(numerical_date)) as month,
id,
amount
Inline
[
numerical_date,id,amount
43254,1,1000
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
43255,2,2123
43256,3,4124
43258,4,2431
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- month

Maak deze meting:

```
=sum(amount)
```

Resultatentabel		
date	month	=sum(amount)
06/03/2022	Jun	1000
06/04/2022	Jun	2123
06/05/2022	Jun	4124
06/07/2022	Jun	2431

In het load-script gebruikt u de `date()` functie om de numerieke datum om te zetten in een datumnotatie. Omdat u in de functie geen specifieke notatie als tweede argument opgeeft, wordt de `dateFormat` gebruikt. Dit resulteert in een datumveld met de notatie 'MM/DD/YYYY'.

### Voorbeeld 4 - Notatie buitenlandse datum

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling van datums.
- De variabele `dateFormat`, die de notatie 'DD/MM/YYYY' gebruikt maar niet wordt omsloten door slashes.

#### Load-script

```
// SET dateFormat='DD/MM/YYYY';
```

```
Transactions:
Load
date,
month(date) as month,
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
id,  
amount  
Inline  
[  
date, id, amount  
22-05-2022, 1, 1000  
23-05-2022, 2, 2123  
24-05-2022, 3, 4124  
25-05-2022, 4, 2431  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- month

Maak deze meting:

```
=sum(amount)
```

Resultatentabel		
date	month	=sum(amount)
22-05-2022	-	1000
23-05-2022	-	2123
24-05-2022	-	4124
25-05-2022	-	2431

In het initiële load-script wordt `dateFormat` gebruikt als de standaard 'MM/DD/YYYY'. Omdat het veld `date` in de gegevensverzameling `Transacties` deze notatie niet heeft, wordt het veld niet als datum geïnterpreteerd. Dit is te zien in de resultatentabel waarin de veldwaarden `month` null zijn.

U kunt de geïnterpreteerde gegevenstypen verifiëren in de gegevensmodelviewer door de eigenschappen van 'Labels' van het veld `date` te controleren:

## 2 Werken met variabelen in de editor voor het laden van gegevens

Voorbeeld van de tabel *Transactions*. Let op de 'Labels' voor het veld *date* die aangeven dat de tekstuele invoergegevens niet impliciet zijn geconverteerd naar een datum/tijdstempel.

date		Transactions			
Density	100%	date	month	id	amount
Subset ratio	100%	22-05-2022	-	1	1000
Has duplicates	false	23-05-2022	-	2	2123
Total distinct values	4	24-05-2022	-	3	4124
Present distinct values	4	25-05-2022	-	4	2431
Non-null values	4				
Tags	Sascii Stext				

Dit kan worden opgelost door de systeemvariabele `DateFormat` in te schakelen:

```
// SET DateFormat='DD/MM/YYYY';
```

Verwijder de dubbele slashes en laad de gegevens opnieuw.

Voorbeeld van de tabel *Transactions*. Let op de 'Labels' voor het veld *date* die aangeven dat de tekstuele invoergegevens impliciet zijn geconverteerd naar een datum/tijdstempel.

date		Transactions			
Density	100%	date	month	id	amount
Subset ratio	100%	22-05-2022	May	1	1000
Has duplicates	false	23-05-2022	May	2	2123
Total distinct values	4	24-05-2022	May	3	4124
Present distinct values	4	25-05-2022	May	4	2431
Non-null values	4				
Tags	Snumeric Sinteger Stimestamp Sdate				

### DayNames

De gedefinieerde notatie vervangt de naamgevingsconventie voor weekdays die zijn ingesteld door uw landinstellingen.

#### Syntaxis:

##### DayNames

Bij het wijzigen van de variabele is een puntkomma ; vereist om de afzonderlijke waarden te scheiden.

Voorbeelden van functie `DayName`

#### Functievoorbeeld

```
set
```

#### Resultaatdefinitie

Dit gebruik van de functie `Dagnamen` definieert

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Functievoorbeeld

```
DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
```

```
Set DayNames='M;Tu;W;Th;F;Sa;Su';
```

### Resultaatdefinitie

dagnamen in hun afgekorte vorm.

Dit gebruik van de functie Dagnamen definieert dagnamen op hun eerste letters.

De functie `DayNames` wordt vaak gebruikt in combinatie met de volgende functies:

#### Gerelateerde functies

##### Functie

##### Interactie

*weekday (page 1066)*

Scriptfunctie om `DayNames` als veldwaarden te retourneren.

*Date (page 1224)*

Scriptfunctie om `DayNames` als veldwaarden te retourneren.

*LongDayNames (page 223)*

Lange vormwaarden van `DayNames`.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 - Standaardstelselvariabelen

Load-script en resultaten

### Overzicht

In dit voorbeeld zijn de datums in de gegevensset ingesteld in de notatie MM/DD/JJJJ.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums, die wordt geladen in een tabel met de naam `Transactions`.
- Een veld `date`.
- De standaard `DayNames`-definitie.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Load-script

```
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
```

```
Transactions:
```

```
LOAD
date,
weekDay(date) as dayname,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- dayname

Maak deze meting:

```
sum(amount)
```

Resultatentabel		
date	dayname	sum(amount)
01/01/2022	Za	1000
02/01/2022	Di	2123
03/01/2022	Di	4124
04/01/2022	Vr	2431

In het load-script wordt de functie `weekDay` gebruikt met het veld `date` als het opgegeven argument. In de resultatentabel geeft de uitvoer van deze functie `weekDay` de dagen van de week weer in de notatie van de definitie `DayNames`.

### Voorbeeld 2 - Systeemvariabele wijzigen

Load-script en resultaten

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad. Dezelfde gegevensset en het scenario uit het eerste voorbeeld worden gebruikt.

Aan het begin van het script wordt de definitie `DayNames` echter gewijzigd om de afgekorte dagen van de week in het Afrikaans te gebruiken.

### Load-script

```
SET DayNames='Ma;Di;wo;Do;Vr;Sa;SO';
```

```
Transactions:
Load
date,
weekDay(date) as dayname,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- dayname

Maak deze meting:

```
sum(amount)
```

Resultatentabel		
date	dayname	sum(amount)
01/01/2022	Za	1000
02/01/2022	Di	2123
03/01/2022	Di	4124
04/01/2022	Vr	2431

In de resultatentabel geeft de uitvoer van deze functie `weekDay` de dagen van de week weer in de notatie van de definitie `DayNames`.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

Het is belangrijk om te onthouden dat als de taal voor de `DayNames` wordt gewijzigd zoals in dit voorbeeld, de `LongDayNames` nog steeds de dagen van de week in het Engels bevat. Dit zou ook moeten worden gewijzigd als beide variabelen in de toepassing worden gebruikt.

### Voorbeeld 3 - Datumfunctie

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums, die wordt geladen in een tabel met de naam `Transactions`.
- Een veld `date`.
- De standaard `DayNames`-definitie.

#### Load-script

```
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
```

```
Transactions:
```

```
Load
date,
Date(date,'www') as dayname,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `dayname`

Maak deze meting:

```
sum(amount)
```



## 2 Werken met variabelen in de editor voor het laden van gegevens

---

Resultatentabel		
date	dayname	sum(amount)
01/01/2022	Za	1000
02/01/2022	Di	2123
03/01/2022	Di	4124
04/01/2022	Vr	2431

De standaard `DayNames`-definitie wordt gebruikt. In het load-script wordt de functie `date` gebruikt met het veld `date` als het eerste argument. Het tweede argument is `www`. Deze opmaak converteert het resultaat naar de waarden die zijn opgeslagen in de definitie `DayNames`. Dit wordt weergegeven in de uitvoer van de resultatentabel.

### DecimalSep

Het gedefinieerde scheidingsteken voor decimalen vervangt het decimaalteken dat is ingesteld door uw landinstellingen.

Qlik Sense interpreteert tekst automatisch als getallen wanneer een herkenbaar nummerpatroon wordt aangetroffen. De systeemvariabelen `ThousandSep` en `DecimalSep` bepalen de samenstelling van de patronen die worden toegepast bij het parseren van tekst als getallen. De variabelen `ThousandSep` en `DecimalSep` stellen het standaardpatroon voor getalnotatie in bij het visualiseren van numerieke inhoud in front-end diagrammen en tabellen. Dat wil zeggen dat het rechtstreeks van invloed is op de opties voor **Getalnotatie** voor elke front-end-uitdrukking.

Uitgaande van een duizendtal scheidingsteken van komma ',' en een decimaal scheidingsteken van '.', zijn dit voorbeelden van patronen die impliciet zouden worden geconverteerd naar numerieke equivalente waarden:

0,000.00

0000.00

0,000

Dit zijn voorbeelden van patronen die als tekst ongewijzigd zouden blijven; dat wil zeggen, niet geconverteerd naar numeriek:

0.000,00

0,00

#### Syntaxis:

`DecimalSep`

Voorbeelden van functies

**Voorbeeld**

**Resultaat**

`set DecimalSep='.';` Hiermee stelt u '.' in als het scheidingsteken voor decimalen.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Voorbeeld

```
set DecimalSep=',';
```

### Resultaat

Hiermee stelt u ',' in als het scheidingsteken voor decimalen.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld - Effect van het instellen van variabelen voor het scheiden van getallen op verschillende invoergegevens

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset van sommen en datums met de sommen in verschillende opmaakpatronen.
- Een tabel met de naam `Transactions`.
- De variabele `DecimalSep` die is ingesteld op '.'.
- De variabele `ThousandSep` die is ingesteld op ','.
- De variabele `Delimiter` die is ingesteld als het teken '|' om de verschillende velden op een regel te scheiden.

### Load-script

```
set ThousandSep=',';
set DecimalSep='.';
```

```
Transactions:
Load date,
id,
amount as amount
Inline
[
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
date|id|amount
01/01/2022|1|1.000-45
01/02/2022|2|23.344
01/03/2022|3|4124,35
01/04/2022|4|2431.36
01/05/2022|5|4,787
01/06/2022|6|2431.84
01/07/2022|7|4132.5246
01/08/2022|8|3554.284
01/09/2022|9|3.756,178
01/10/2022|10|3,454.356
] (delimiter is '|');
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: amount.

Maak deze meting:

```
=sum(amount)
```

Amount	Resultatentabel	=Sum(amount)
Totalen		20814.7086
1.000-45		
3.756,178		
4124,35		
	23.344	23.344
	2431.36	2431.36
	2431.84	2431.84
	3,454.356	3454.356
	3554.284	3554.284
	4132.5246	4132.5246
	4,787	4787

Elke waarde die niet als getal wordt geïnterpreteerd, blijft als tekst en wordt standaard links uitgelijnd. Alle succesvol geconverteerde waarden worden rechts uitgelijnd, waarbij het oorspronkelijke invoerformaat behouden blijft.

De uitdrukingskolom toont het numerieke equivalent, dat standaard is opgemaakt met alleen een scheidingsteken voor decimalen '.'. Dit kan worden overschreven met de vervolgkeuzelijstinstelling **Getalnotatie** in de uitdrukingsconfiguratie.

### FirstWeekDay

Geheel getal dat definieert welke dag moet worden gebruikt als eerste dag van de week.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Syntaxis:

#### FirstWeekDay

Standaard definieert de Qlik Sense-systeemvariabele `FirstWeekDay=6`. Dit betekent dat zondag de eerste dag van de week is.

Waarden die kunnen worden ingesteld voor FirstWeekDay

Value	Dag
0	Maandag
1	Dinsdag
2	Woensdag
3	Donderdag
4	Vrijdag
5	Zaterdag
6	Zondag

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Maakt gebruik van de standaardwaarde (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

In dit voorbeeld gebruikt het load-script de standaardwaarde van de Qlik Sense-systeemvariabele, `FirstWeekDay=6`. Deze gegevens bevatten gegevens voor de eerste 14 dagen van 2020.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Load-script

// Example 1: Load Script using the default value of FirstWeekDay=6, i.e. Sunday

```
SET FirstWeekDay = 6;
```

```
Sales:
```

```
LOAD
```

```
    date,  
    sales,  
    week(date) as week,  
    weekday(date) as weekday
```

```
Inline [
```

```
date,sales
```

```
01/01/2021,6000
```

```
01/02/2021,3000
```

```
01/03/2021,6000
```

```
01/04/2021,8000
```

```
01/05/2021,5000
```

```
01/06/2020,7000
```

```
01/07/2020,3000
```

```
01/08/2020,5000
```

```
01/09/2020,9000
```

```
01/10/2020,5000
```

```
01/11/2020,7000
```

```
01/12/2020,7000
```

```
01/13/2020,7000
```

```
01/14/2020,7000
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- week
- weekday

Resultatentabel

Date	week	weekday
01/01/2021	1	Wo
01/02/2021	1	Do
01/03/2021	1	Vr
01/04/2021	1	Za
01/05/2021	2	Zo
01/06/2020	2	Ma

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

Date	week	weekday
01/07/2020	2	Di
01/08/2020	2	Wo
01/09/2020	2	Do
01/10/2020	2	Vr
01/11/2020	2	Za
01/12/2020	3	Zo
01/13/2020	3	Ma
01/14/2020	3	Di

Omdat de standaardinstellingen worden gebruikt, wordt de `FirstWeekDay` systeemvariabele ingesteld op 6. In de resultatentabel is te zien dat elke nieuwe week op zondag begint (5 en 12 januari).

### Voorbeeld 2 - De FirstWeekDay variabele wijzigen (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

In dit voorbeeld bevatten de gegevens de eerste 14 dagen van 2020. Aan het begin van het script stellen we de `FirstWeekDay` variabele in op 3.

#### Load-script

```
// Example 2: Load Script setting the value of FirstWeekDay=3, i.e. Thursday
```

```
SET FirstWeekDay = 3;
```

```
Sales:
```

```
LOAD
```

```
    date,  
    sales,  
    week(date) as week,  
    weekday(date) as weekday
```

```
Inline [
```

```
date,sales  
01/01/2021,6000  
01/02/2021,3000  
01/03/2021,6000  
01/04/2021,8000  
01/05/2021,5000  
01/06/2020,7000  
01/07/2020,3000  
01/08/2020,5000
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
01/09/2020,9000
01/10/2020,5000
01/11/2020,7000
01/12/2020,7000
01/13/2020,7000
01/14/2020,7000
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- week
- weekday

Resultatentabel

Date	week	weekday
01/01/2021	52	Wo
01/02/2021	1	Do
01/03/2021	1	Vr
01/04/2021	1	Za
01/05/2021	1	Zo
01/06/2020	1	Ma
01/07/2020	1	Di
01/08/2020	1	Wo
01/09/2020	2	Do
01/10/2020	2	Vr
01/11/2020	2	Za
01/12/2020	2	Zo
01/13/2020	2	Ma
01/14/2020	2	Di

Omdat de `FirstWeekDay` systeemvariabele is ingesteld op 3, begint de eerste dag van de week op donderdag. In de resultatentabel is te zien dat elke nieuwe week begint op donderdag (2 en 9 januari).

### LongDayNames

De gedefinieerde notatie vervangt de lange naamgevingsconventie voor weekdays in de landinstellingen.

#### Syntaxis:

```
LongDayNames
```

---

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

Het volgende voorbeeld van de `LongDayNames`-functie definieert de dagnamen:

```
Set LongDayNames= 'Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday' ;
```

Bij het wijzigen van de variabele is een puntkomma ; vereist om de afzonderlijke waarden te scheiden.

De functie `LongDayNames` kan worden gebruikt in combinatie met de functie *Date (page 1224)* die `DayNames` als veldwaarden teruggeeft.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Standaardsysteemvariabele

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums, die wordt geladen in een tabel met de naam `Transactions`.
- Een veld `date`.
- De standaard `LongDayNames`-definitie.

#### Load-script

```
SET LongDayNames= 'Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday' ;
```

```
Transactions:
LOAD
date,
Date(date,'www') as dayname,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000
```



## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- dayname

Maak deze meting:

```
=sum(amount)
```

Resultatentabel		
date	dayname	=sum(amount)
01/01/2022	Zaterdag	1000
02/01/2022	Dinsdag	2123
03/01/2022	Dinsdag	4124
04/01/2022	Vrijdag	2431

In het load-script wordt om een veld genaamd dayname te maken de functie Date gebruikt met het veld date als het eerste argument. Het tweede argument in de functie is de opmaak www.

Door deze opmaak te gebruiken, worden de waarden van het eerste argument geconverteerd naar de overeenkomstige naam van de volledige dag die is ingesteld in de variabele LongDayNames. In de resultatentabel geven de veldwaarden van ons aangemaakte veld dayname dit weer.

### Voorbeeld 2 - Systeemvariabele wijzigen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Dezelfde gegevensset en het scenario uit het eerste voorbeeld worden gebruikt. Aan het begin van het script wordt de definitie LongDayNames echter gewijzigd om de dagen van de week in het Spaans te gebruiken.

#### Load-script

```
SET LongDayNames='Lunes;Martes;Miércoles;Jueves;Viernes;Sábado;Domingo';
```

```
Transactions:
```

```
LOAD
date,
Date(date,'www') as dayname,
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
id,  
amount  
INLINE  
[  
date, id, amount  
01/01/2022, 1, 1000  
02/01/2022, 2, 2123  
03/01/2022, 3, 4124  
04/01/2022, 4, 2431  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- dayname

Maak deze meting:

```
=sum(amount)
```

Resultatentabel		
date	dayname	=sum(amount)
01/01/2022	Sábado	1000
02/01/2022	Martes	2123
03/01/2022	Martes	4124
04/01/2022	Viernes	2431

In het load-script wordt de variabele `LongDayNames` gewijzigd om de dagen van de week in het Spaans weer te geven.

Vervolgens maakt u een veld genaamd `dayname`, dat de functie `date` is die wordt gebruikt met het veld `date` als het eerste argument.

Het tweede argument in de functie is de opmaak `www`. Met behulp van deze opmaak converteert Qlik Sense de waarden van het eerste argument in de overeenkomende volledige dagnaam die is ingesteld in de variabele `LongDayNames`.

In de resultatentabel geven de veldwaarden van ons gemaakte veld `dayname` de dagen van de week weer, geschreven in het Spaans en voluit.

### LongMonthNames

De gedefinieerde notatie vervangt de lange naamgevingsconventie voor maanden in de landinstellingen.

#### Syntaxis:

```
LongMonthNames
```

Bij het wijzigen van de variabele moet de `;` worden gebruikt om de afzonderlijke waarden te scheiden.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

Het volgende voorbeeld van de functie LongMonthNames definieert maandnamen voluit:

```
Set
```

```
LongMonthNames='January;February;March;April;May;June;July;August;September;October;November;December';
```

De functie LongMonthNames wordt vaak gebruikt in combinatie met de volgende functies:

Gerelateerde functies	
Functie	Interactie
<i>Date (page 1224)</i>	Scriptfunctie om DayNames als veldwaarden te retourneren.
<i>LongDayNames (page 223)</i>	Lange vormwaarden van DayNames.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht SET DateFormat in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Standaardstelselvariabelen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums die wordt geladen in een tabel genaamd Transactions.
- Een veld date.
- De standaard LongMonthNames-definitie.

#### Load-script

```
SET
```

```
LongMonthNames='January;February;March;April;May;June;July;August;September;October;November;December';
```

```
Transactions:
```

```
Load
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
date,  
Date(date,'MMMM') as monthname,  
id,  
amount  
Inline  
[  
date,id,amount  
01/01/2022,1,1000.45  
01/02/2022,2,2123.34  
01/03/2022,3,4124.35  
01/04/2022,4,2431.36  
01/05/2022,5,4787.78  
01/06/2022,6,2431.84  
01/07/2022,7,2854.83  
01/08/2022,8,3554.28  
01/09/2022,9,3756.17  
01/10/2022,10,3454.35  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- monthname

Maak deze meting:

```
=sum(amount)
```

Resultatentabel		
date	monthname	sum(amount)
01/01/2022	Januari	1000.45
01/02/2022	Januari	2123.34
01/03/2022	Januari	4124.35
01/04/2022	Januari	2431.36
01/05/2022	Januari	4787.78
01/06/2022	Januari	2431.84
01/07/2022	Januari	2854.83
01/08/2022	Januari	3554.28
01/09/2022	Januari	3756.17
01/10/2022	Januari	3454.35

De standaard LongMonthNames-definitie wordt gebruikt. In het load-script wordt om een veld genaamd month te maken de functie Date gebruikt met het veld date als het eerste argument. Het tweede argument in de functie is de opmaak .MMMM

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

Met behulp van deze opmaak converteert Qlik Sense de waarden van het eerste argument in de overeenkomende volledige maandnaam die is ingesteld in de variabele `LongMonthNames`. In de resultatentabel geven de veldwaarden van ons aangemaakte veld `month` dit weer.

### Voorbeeld 2 - Systeemvariabele wijzigen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums die wordt geladen in een tabel genaamd `Transactions`.
- Een veld `date`.
- De variabele `LongMonthNames` die is aangepast om de afgekorte dagen van de week in het Spaans te gebruiken.

#### Load-script

```
SET
LongMonthNames='Enero;Febrero;Marzo;Abril;Mayo;Junio;Julio;Agosto;Septiembre;OctubreNoviembre;
Diciembre';

Transactions:
LOAD
date,
Date(date,'MMMM') as monthname,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000
02/01/2022,2,2123
03/01/2022,3,4124
04/01/2022,4,2431
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg `sum(amount)` toe als een meting en deze velden als dimensies:

- `date`
- `monthname`

Maak deze meting:

```
=sum(amount)
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

	Resultatentabel	
date	monthname	sum(amount)
01/01/2022	Enero	1000.45
01/02/2022	Enero	2123.34
01/03/2022	Enero	4124.35
01/04/2022	Enero	2431.36
01/05/2022	Enero	4787.78
01/06/2022	Enero	2431.84
01/07/2022	Enero	2854.83
01/08/2022	Enero	3554.28
01/09/2022	Enero	3756.17
01/10/2022	Enero	3454.35

In het load-script wordt de variabele `LongMonthNames` gewijzigd om de maanden van het jaar in het Spaans weer te geven. Vervolgens, om een veld genaamd `monthname` te maken, wordt de functie `Date` gebruikt met het veld `date` als het eerste argument. Het tweede argument in de functie is de opmaak `MMMM`.

Met behulp van deze opmaak converteert Qlik Sense de waarden van het eerste argument in de overeenkomende volledige maandnaam die is ingesteld in de variabele `LongMonthNames`. In de resultatentabel geven de velwaarden van ons gemaakt veld `monthname` de maandnaam in het Spaans weer.

### MoneyDecimalSep

Het gedefinieerde scheidingsteken voor decimalen vervangt het decimaalteken voor valuta dat is ingesteld door uw landinstellingen.



*Standaard toont Qlik Sense getallen en tekst anders in tabeldiagrammen. Getallen worden rechts uitgelijnd en tekst wordt links uitgelijnd. Dit maakt het eenvoudiger om problemen bij de omzetting van tekst-naar-getallen te vinden. Tabellen op deze pagina die resultaten van Qlik Sense weergeven, gebruiken deze opmaak.*

#### Syntaxis:

##### **MoneyDecimalSep**

Qlik Sense-applicaties zullen tekstvelden die voldoen aan deze indeling interpreteren als geldwaarden. Het tekstveld moet het valutasympool bevatten dat is gedefinieerd in de systeemvariabele `MoneyFormat`. `MoneyDecimalSep` is vooral nuttig wanneer gegevensbronnen worden verwerkt die zijn ontvangen met verschillende regionale instellingen.

Het volgende voorbeeld geeft een mogelijk gebruik van de systeemvariabele `MoneyDecimalSep` weer:

```
set MoneyDecimalSep='.';
```

Deze functie wordt vaak gebruikt in combinatie met de volgende functies:

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Gerelateerde functies

Functie	Interactie
MoneyFormat	Bij de interpretatie van een tekstveld, wordt het symbool MoneyFormat gebruikt als onderdeel van de interpretatie. Voor getalnotatie wordt de notatie MoneyFormat door Qlik Sense gebruikt in diagramobjecten.
MoneyThousandSep	Bij interpretatie van een tekstveld, moet ook aan de functie MoneyThousandSep worden voldaan.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - MoneyDecimalSep puntnotatie (.)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in de tabel `Transactions`.
- Verstreckte gegevens waarbij het geldveld de tekstnotatie met een punt '.' heeft die als scheidingsteken voor decimalen wordt gebruikt. Elk record wordt bovendien voorafgegaan door het symbool \$, met uitzondering van het laatste record, dat vooraf wordt gegaan door het symbool £.

Vergeet niet dat de systeemvariabele `MoneyFormat` de dollar (\$) definieert als standaardvaluta.

#### Load-script

```
SET MoneyThousandSep=' ' ;
SET MoneyDecimalSep='.' ;
SET MoneyFormat='$###0.00;-###0.00' ;
```

`Transactions:`

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,'$14.41'
01/02/2022,2,'$2,814.32'
01/03/2022,3,'$249.36'
01/04/2022,4,'$24.37'
01/05/2022,5,'$7.54'
01/06/2022,6,'$243.63'
01/07/2022,7,'$545.36'
01/08/2022,8,'$3.55'
01/09/2022,9,'$3.436'
01/10/2022,10,'£345.66'
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:amount.

Voeg de volgende metingen toe:

- isNum(amount)
- sum(amount)

Controleer de onderstaande resultaten die de juiste interpretatie van alleen alle dollar-waarden (\$) laat zien.

amount	=isNum(amount)	=Sum(amount)
Totalen	0	\$3905.98
£345.66	0	\$0.00
\$3.436	-1	\$3.44
\$3.55	-1	\$3.55
\$7.54	-1	\$7.54
\$14.41	-1	\$14.41
\$24.37	-1	\$24.37
243.63	-1	\$243.63
\$249.36	-1	\$249.36
\$545.36	-1	\$545.36
\$2,814.32	-1	\$2814.32



## 2 Werken met variabelen in de editor voor het laden van gegevens

---

De bovenstaande resultatentabel laat zien hoe het veld `amount` correct is geïnterpreteerd voor alle waarden met het dollar-prefix (\$) en dat het `amount` met het pond-prefix (£) niet is omgezet in een geldelijke waarde.

### Voorbeeld 2 - MoneyDecimalSep komma-notatie (,)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in de tabel `Transactions`.
- Verstrekte gegevens waarbij het geldveld de tekstnotatie met een komma ',' heeft die als scheidingsteken voor decimalen wordt gebruikt. Elk record heeft ook het dollar-prefix (\$) met uitzondering van het laatste record, dat het onjuiste scheidingsteken voor decimalen gebruikt, de punt '.'.

Vergeet niet dat de systeemvariabele `MoneyFormat` de dollar (\$) definieert als standaardvaluta.

#### Load-script

```
SET MoneyThousandSep='.';
SET MoneyDecimalSep=',';
SET MoneyFormat='$###0.00;-###0.00';
```

Transactions:

```
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,'$14,41'
01/02/2022,2,'$2.814,32'
01/03/2022,3,'$249,36'
01/04/2022,4,'$24,37'
01/05/2022,5,'$7,54'
01/06/2022,6,'$243,63'
01/07/2022,7,'$545,36'
01/08/2022,8,'$3,55'
01/09/2022,9,'$3,436'
01/10/2022,10,'$345.66'
];
```

#### Resultaten

Alineatekst voor Resultaten.

## 2 Werken met variabelen in de editor voor het laden van gegevens

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als `dimensie:amount`.

Voeg de volgende metingen toe:

- `isNum(amount)`
- `sum(amount)`

Controleer de onderstaande resultaten voor een demonstratie van de juiste interpretatie van alle waarden met uitzondering van het bedrag waarbij de punt '.' als scheidingsteken voor decimalen wordt gebruikt. In dat geval had een punt moeten zijn gebruikt in plaats van een komma.

Resultatentabel

amount	=isNum(amount)	=Sum(amount)
Totalen	0	\$3905.98
\$345.66	0	\$0.00
\$3,436	-1	\$3.44
\$3,55	-1	\$3.55
\$7,54	-1	\$7.54
\$14,41	-1	\$14.41
\$24,37	-1	\$24.37
\$243,63	-1	\$243.63
\$249,36	-1	\$249.36
\$545,36	-1	\$545.36
\$2.814,32	-1	\$2814.32

### MoneyFormat

Deze systeemvariabele definieert het opmaakpatroon dat wordt gebruikt door Qlik voor automatische vertaling van tekst naar getal, waarbij het getal een valutasymbool als prefix krijgt. Het definieert tevens hoe metingen waarbij de opmaaknotatie-eigenschappen zijn ingesteld op Geld, in diagramobjecten worden weergegeven.

Het gedefinieerde symbool als onderdeel van het opmaakpatroon in de systeemvariabele `MoneyFormat` vervangt het valutasymbool dat is ingesteld door uw landinstellingen.



*Standaard toont Qlik Sense getallen en tekst anders in tabeldiagrammen. Getallen worden rechts uitgelijnd en tekst wordt links uitgelijnd. Dit maakt het eenvoudiger om problemen bij de omzetting van tekst-naar-getallen te vinden. Tabellen op deze pagina die resultaten van Qlik Sense weergeven, gebruiken deze opmaak.*

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Syntaxis:

#### MoneyFormat

```
Set MoneyFormat='$ #,##0.00; ($ #,##0.00)';
```

Deze opmaak wordt weergegeven in diagramobjecten wanneer de eigenschap `Number Formatting` van een numeriek veld is ingesteld op `Money`. Als het valutasympbool van het tekstveld overeenkomt met het symbool dat is ingesteld in de variabele `MoneyFormat` en numerieke tekstvelden worden geïnterpreteerd door Qlik Sense, interpreteert Qlik Sense dit veld als een geldelijke waarde.

Deze functie wordt vaak gebruikt in combinatie met de volgende functies:

#### Gerelateerde functies

Functie	Interactie
<i>MoneyDecimalSep</i> (page 230)	Voor getalnotatie wordt <code>MoneyDecimalSep</code> gebruikt in de veldindeling van objecten.
<i>MoneyThousandSep</i> (page 238)	Voor getalnotatie wordt <code>MoneyThousandSep</code> gebruikt in de veldindeling van objecten.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - MoneyFormat

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat een gegevensset die wordt geladen in een tabel met de naam `Transactions`. De standaarddefinitie voor de `MoneyFormat`-variabele wordt gebruikt.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Load-script

```
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat='$###0.00;-$$$0.00';
```

Transactions:

```
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,$10000000441
01/02/2022,2,$21237492432
01/03/2022,3,$249475336
01/04/2022,4,$24313369837
01/05/2022,5,$7873578754
01/06/2022,6,$24313884663
01/07/2022,7,$545883436
01/08/2022,8,$35545828255
01/09/2022,9,$37565817436
01/10/2022,10,$3454343566
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- amount

Voeg deze meting toe:

=Sum(amount)

Selecteer onder **Getalnotatie Geld** om Sum(amount) in te stellen als een geldelijke waarde.

Resultatentabel

date	Amount	=Sum(amount)
Totalen		\$165099674156.00
01/01/2022	\$10000000441	\$10000000441.00
01/02/2022	\$21237492432	\$21237492432.00
01/03/2022	\$249475336	\$249475336.00
01/04/2022	\$24313369837	\$24313369837.00
01/05/2022	\$7873578754	\$7873578754.00

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

date	Amount	=Sum(amount)
01/06/2022	\$24313884663	\$24313884663.00
01/07/2022	\$545883436	\$545883436.00
01/08/2022	\$35545828255	\$35545828255.00
01/09/2022	\$37565817436	\$37565817436.00
01/10/2022	\$3454343566	\$3454343566.00

De standaard `MoneyFormat`-definitie wordt gebruikt. Dit ziet er als volgt uit: `###0.00;-###0.00`. In de resultatentabel geeft de opmaak van het `amount`-veld het valutasymbool en de decimale punt weer en er zijn decimalen toegevoegd.

### Voorbeeld 2 - MoneyFormat met scheidingstekens voor duizendtallen en meerdere invoerindelingen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met gemengde-invoerindeling die is geladen in de tabel `Transactions` met scheidingstekens voor duizendtallen en scheidingstekens voor decimalen.
- De `MoneyFormat`-definitie wordt aangepast met een komma als scheidingstekens voor duizendtallen.
- Een van de rijen met gegevens heeft komma's als scheidingstekens voor duizendtallen op de verkeerde plek. Hier is te zien hoe dit bedrag tekst is en niet kan worden geïnterpreteerd als getal.

#### Load-script

```
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat = '$#,##0.00;-$,##0.00';
```

Transactions:

```
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,'$10,000,000,441.45'
01/02/2022,2,'$212,3749,24,32.23'
01/03/2022,3,$249475336.45
01/04/2022,4,$24,313,369,837
01/05/2022,5,$7873578754
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
01/06/2022,6,$24313884663
01/07/2022,7,$545883436
01/08/2022,8,$35545828255
01/09/2022,9,$37565817436
01/10/2022,10,$3454343566
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- amount

Voeg deze meting toe:

```
=Sum(amount)
```

Selecteer onder **Getalnotatie Geld** om `sum(amount)` in te stellen als een geldelijke waarde.

Resultatentabel

date	Amount	=Sum(amount)
Totalen		\$119,548,811,911.90
01/01/2022	\$10,000,000,441.45	\$10,000,000,441.45
01/02/2022	\$212,3749,24,32.23	\$0.00
01/03/2022	\$249475336.45	\$249,475,336.45
01/04/2022	\$24	\$24.00
01/05/2022	\$7873578754	\$7,873,578,754.00
01/06/2022	\$24313884663	\$24,313,884,663.00
01/07/2022	\$545883436	\$545,883,436.00
01/08/2022	\$35545828255	\$35,545,828,255.00
01/09/2022	\$37565817436	\$37,565,817,436.00
01/10/2022	\$3454343566	\$3,454,343,566.00

Aan het begin van het script kan de systeemvariabele `MoneyFormat` worden aangepast met een komma als scheidingsteken voor duizendtallen. In de Qlik Sense-tabel is te zien hoe de indeling dit scheidingsteken bevat. Daarnaast is de rij met het onjuiste scheidingsteken niet correct geïnterpreteerd en blijft de inhoud tekst. Daarom telt deze rij niet mee in het totaal.

### MoneyThousandSep

Het gedefinieerde scheidingsteken voor duizendtallen vervangt het cijfergroeperingssymbool voor valuta dat is ingesteld door uw landinstellingen.

## 2 Werken met variabelen in de editor voor het laden van gegevens



Standaard toont Qlik Sense getallen en tekst anders in tabeldiagrammen. Getallen worden rechts uitgelijnd en tekst wordt links uitgelijnd. Dit maakt het eenvoudiger om problemen bij de omzetting van tekst-naar-getallen te vinden. Tabellen op deze pagina die resultaten van Qlik Sense weergeven, gebruiken deze opmaak.

### Syntaxis:

#### MoneyThousandSep

Qlik Sense-applicaties zullen tekstvelden die voldoen aan deze indeling interpreteren als geldwaarden. Het tekstveld moet het valutasymbool bevatten dat is gedefinieerd in de systeemvariabele `MoneyFormat`. `MoneyThousandSep` is vooral nuttig wanneer gegevensbronnen worden verwerkt die zijn ontvangen met verschillende regionale instellingen.

Het volgende voorbeeld geeft een mogelijk gebruik van de systeemvariabele `MoneyThousandSep` weer:

```
set MoneyDecimalSep=',';
```

Deze functie wordt vaak gebruikt in combinatie met de volgende functies:

#### Gerelateerde functies

Functie	Interactie
<code>MoneyFormat</code>	Bij de interpretatie van een tekstveld, wordt het symbool <code>MoneyFormat</code> gebruikt als onderdeel van de interpretatie. Voor getalnotatie wordt de indeling <code>MoneyFormat</code> door Qlik Sense gebruikt in diagramobjecten.
<code>MoneyDecimalSep</code>	Bij interpretatie van een tekstveld, moet ook aan de functie <code>MoneyDecimalSep</code> worden voldaan.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Voorbeeld 1 - MoneyThousandSep komma-notatie (,)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in de tabel Transactions.
- Verstreckte gegevens waarbij het geldveld de tekstindeling met een komma heeft die als scheidingsteken voor duizendtallen wordt gebruikt. Elk record heeft ook de prefix \$.

Vergeet niet dat de systeemvariabele MoneyFormat de dollar (\$) definieert als standaardvaluta.

#### Load-script

```
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat='$###0.00;-$$$0.00';
```

Transactions:

```
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,'$10,000,000,441'
01/02/2022,2,'$21,237,492,432'
01/03/2022,3,'$249,475,336'
01/04/2022,4,'$24,313,369,837'
01/05/2022,5,'$7,873,578,754'
01/06/2022,6,'$24,313,884,663'
01/07/2022,7,'$545,883,436'
01/08/2022,8,'$35,545,828,255'
01/09/2022,9,'$37,565,817,436'
01/10/2022,10,'$3.454.343.566'
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:amount.

Voeg de volgende metingen toe:

- isNum(amount)
- sum(amount)



## 2 Werken met variabelen in de editor voor het laden van gegevens

---

Bekijk de onderstaande resultaten. De tabel laat de juiste interpretatie zien van alle waarden met een komma als scheidingsteken voor duizendtallen.

Het amount-veld is correct geïnterpreteerd voor alle waarden, met uitzondering van één waarde waarbij een punt als scheidingsteken voor duizendtallen is gebruikt.

amount	=isNum(amount)	=Sum(amount)
Totalen	0	\$161645330590.00
\$3.454.343.566	0	\$0.00
\$249,475,336	-1	\$249475336.00
\$545,883,436	-1	\$545883436.00
\$7,873,578,754	-1	\$7873578754.00
\$10,000,000,441	-1	\$10000000441.00
\$21,237,492,432	-1	\$21237492432.00
\$24,313,369,837	-1	\$24313369837.00
\$24,33,884,663	-1	\$24313884663.00
\$35,545,828,255	-1	\$35545828255.00
\$37,565,817,436	-1	\$37565817436.00

### Voorbeeld 2 - MoneyThousandSep punt-notatie (.)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in de tabel `Transactions`.
- Verstreckte gegevens waarbij het geldveld een tekstindeling met een punt heeft die als scheidingsteken voor duizendtallen wordt gebruikt. Elk record heeft ook de prefix \$.

Vergeet niet dat de systeemvariabele `MoneyFormat` de dollar (\$) definieert als standaardvaluta.

#### Load-script

```
SET MoneyThousandSep='.';
SET MoneyDecimalSep='';
SET MoneyFormat='$###0.00;-###0.00';
```

`Transactions:`

Load

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
date,  
id,  
amount  
Inline  
[  
date,id,amount  
01/01/2022,1,'$10.000.000.441'  
01/02/2022,2,'$21.237.492.432'  
01/03/2022,3,'$249.475.336'  
01/04/2022,4,'$24.313.369.837'  
01/05/2022,5,'$7.873.578.754'  
01/06/2022,6,'$24.313.884.663'  
01/07/2022,7,'$545.883.436'  
01/08/2022,8,'$35.545.828.255'  
01/09/2022,9,'$37.565.817.436'  
01/10/2022,10,'$3,454,343,566'  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:amount.

Voeg de volgende metingen toe:

- isNum(amount)
- sum(amount)

Bekijk de onderstaande resultaten voor de juiste interpretatie van alle waarden met een punt als scheidingsteken voor duizendtallen.

Het amount-veld is correct geïnterpreteerd voor alle waarden, met uitzondering van één waarde waarbij een komma als scheidingsteken voor duizendtallen is gebruikt.

Resultatentabel

amount	=isNum(amount)	=Sum(amount)
Totalen	0	\$161645330590.00
\$3,545,343,566	0	\$0.00
\$249.475.336	-1	\$249475336.00
\$545.883.436	-1	545883436.00
\$7.873.578.754	-1	\$7873578754.00
\$10.000.000.441	-1	\$1000000441.00
\$21.237.492.432	-1	\$21237492432.00
\$24.313.884.663	-1	\$24313884663.00
\$24.313.884.663	-1	\$24313884663.00

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

amount	=isNum(amount)	=Sum(amount)
\$35.545.828.255	-1	\$35545828255.00
\$37.565.817.436	-1	\$37565817436.00

### MonthNames

De gedefinieerde notatie vervangt de naamgevingsconventie voor maanden in de landinstellingen.

#### Syntaxis:

##### MonthNames

Bij het wijzigen van de variabele moet de ; worden gebruikt om de afzonderlijke waarden te scheiden.

#### Voorbeelden van functies

##### Voorbeeld

```
Set MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

Set

```
MonthNames='Enero;Feb;Marzo;Abr;Mayo;Jun;Jul;Agosto;Set;Oct;Nov;Dic';
```

##### Resultaten

Dit gebruik van de functie `MonthNames` definieert maandnamen in het Engels en hun afgekorte vorm.

Dit gebruik van de functie `MonthNames` definieert maandnamen in het Spaans en hun afgekorte vorm.

De functie `MonthNames` kan worden gebruikt in combinatie met de volgende functies:

#### Gerelateerde functies

Functie	Interactie
<i>month</i> (page 906)	Scriptfunctie om waarden gedefinieerd in <code>MonthNames</code> als veldwaarden te retourneren
<i>Date</i> (page 1224)	Scriptfunctie om waarden gedefinieerd in <code>MonthNames</code> als veldwaarden te retourneren op basis van een opgegeven opmaakargument
<i>LongMonthNames</i> (page 226)	Lange vormwaarden van <code>MonthNames</code>

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Standaardsysteemvariabelen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums die wordt geladen in een tabel genaamd `Transactions`.
- Een veld `date`.
- De standaard `MonthNames`-definitie.

#### Load-script

```
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

```
Transactions:
```

```
LOAD
```

```
date,
```

```
Month(date) as monthname,
```

```
id,
```

```
amount
```

```
INLINE
```

```
[
```

```
date,id,amount
```

```
01/01/2022,1,1000.45
```

```
01/02/2022,2,2123.34
```

```
01/03/2022,3,4124.35
```

```
01/04/2022,4,2431.36
```

```
01/05/2022,5,4787.78
```

```
01/06/2022,6,2431.84
```

```
01/07/2022,7,2854.83
```

```
01/08/2022,8,3554.28
```

```
01/09/2022,9,3756.17
```

```
01/10/2022,10,3454.35
```

```
];
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- monthname

Maak deze meting:

=sum(amount)

Resultatentabel		
date	monthname	sum(amount)
01/01/2022	Jan	1000.45
01/02/2022	Jan	2123.34
01/03/2022	Jan	4124.35
01/04/2022	Jan	2431.36
01/05/2022	Jan	4787.78
01/06/2022	Jan	2431.84
01/07/2022	Jan	2854.83
01/08/2022	Jan	3554.28
01/09/2022	Jan	3756.17
01/10/2022	Jan	3454.35

De standaard `MonthNames`-definitie wordt gebruikt. In het load-script wordt de functie `Month` gebruikt met het veld `date` als het opgegeven argument.

In de resultatentabel geeft de uitvoer van deze functie `Month` de maanden van het jaar weer in de notatie van de definitie `MonthNames`.

### Voorbeeld 2 - Systeemvariabele wijzigen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums die wordt geladen in een tabel genaamd `Transactions`.
- Een veld `date`.
- De variabele `MonthNames` die is aangepast om de afgekorte maanden in het Spaans te gebruiken.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Load-script

```
Set MonthNames='Enero;Feb;Marzo;Abr;Mayo;Jun;Jul;Agosto;Set;Oct;Nov;Dic';
```

```
Transactions:  
LOAD  
date,  
month(date) as month,  
id,  
amount  
INLINE  
[  
date,id,amount  
01/01/2022,1,1000  
02/01/2022,2,2123  
03/01/2022,3,4124  
04/01/2022,4,2431  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- monthname

Maak deze meting:

```
=sum(amount)
```

	Resultatentabel	
date	monthname	sum(amount)
01/01/2022	Enero	1000.45
01/02/2022	Enero	2123.34
01/03/2022	Enero	4124.35
01/04/2022	Enero	2431.36
01/05/2022	Enero	4787.78
01/06/2022	Enero	2431.84
01/07/2022	Enero	2854.83
01/08/2022	Enero	3554.28
01/09/2022	Enero	3756.17
01/10/2022	Enero	3454.35

In het load-script wordt eerst de variabele `MonthNames` gewijzigd om de maanden van het jaar afgekort in het Spaans weer te geven. De functie `month` wordt gebruikt met het veld `date` als het opgegeven argument.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

In de resultatentabel geeft de uitvoer van deze functie `Month` de maanden van het jaar weer in de notatie van de definitie `MonthNames`.

Het is belangrijk om te onthouden dat als de taal voor de variabele `MonthNames` wordt gewijzigd zoals in dit voorbeeld, de variabele `LongMonthNames` nog steeds de maanden van het jaar in het Engels bevat. De variabele `LongMonthNames` zou moeten worden gewijzigd als beide variabelen worden gebruikt in de toepassing.

### Voorbeeld 3 - Datumfunctie

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums die wordt geladen in een tabel genaamd `Transactions`.
- Een veld `date`.
- De standaard `MonthNames`-definitie.

#### Load-script

```
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

```
Transactions:
```

```
LOAD
date,
Month(date, 'MMM') as monthname,
id,
amount
INLINE
[
date,id,amount
01/01/2022,1,1000.45
01/02/2022,2,2123.34
01/03/2022,3,4124.35
01/04/2022,4,2431.36
01/05/2022,5,4787.78
01/06/2022,6,2431.84
01/07/2022,7,2854.83
01/08/2022,8,3554.28
01/09/2022,9,3756.17
01/10/2022,10,3454.35
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

- date
- monthname

Maak deze meting:

```
=sum(amount)
```

Resultatentabel		
date	monthname	sum(amount)
01/01/2022	Jan	1000.45
01/02/2022	Jan	2123.34
01/03/2022	Jan	4124.35
01/04/2022	Jan	2431.36
01/05/2022	Jan	4787.78
01/06/2022	Jan	2431.84
01/07/2022	Jan	2854.83
01/08/2022	Jan	3554.28
01/09/2022	Jan	3756.17
01/10/2022	Jan	3454.35

De standaard monthNames-definitie wordt gebruikt. In het load-script wordt de functie date gebruikt met het veld date als het eerste argument. Het tweede argument is MMM.

Met behulp van deze opmaak converteert Qlik Sense de waarden van het eerste argument in de overeenkomende maandnaam die is ingesteld in de variabele monthNames. In de resultatentabel geven de veldwaarden van ons aangemaakte veld month dit weer.

### NumericalAbbreviation

De numerieke afkorting bepaalt welke afkorting u moet gebruiken als het voorvoegsel dat de schaal van numerieke waarden aangeeft, bijvoorbeeld M voor mega of een miljoen ( $10^6$ ), en  $\mu$  voor micro ( $10^{-6}$ ).

#### Syntaxis:

##### **NumericalAbbreviation**

Stel de variabele NumericalAbbreviation in als tekenreeks met een lijst van afkortingen voor gedefinieerde paren, gescheiden door een puntkomma. Elke afkorting voor een gedefinieerd paar moet een schaal bevatten (de decimale exponent) en worden gescheiden met een dubbele punt, bijvoorbeeld 6:M voor een miljoen.

De standaardinstelling is '3:k;6:M;9:G;12:T;15:P;18:E;21:Z;24:Y;-3:m;-6:μ;-9:n;-12:p;-15:f;-18:a;-21:z;-24:y'.



## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Voorbeelden:

Deze instelling wijzigt het voorvoegsel voor duizend naar t en het voorvoegsel voor een miljard naar B. deze optie is handig voor financiële applicaties waar afkortingen worden verwacht zoals t\$, M\$ en B\$.

```
Set NumericalAbbreviation='3:t;6:M;9:B;12:T;15:P;18:E;21:Z;24:Y;-3:m;-6:μ;-9:n;-12:p;-15:f;-18:a;-21:z;-24:y';
```

### ReferenceDay

De instelling bepaalt welke dag in januari moet worden ingesteld als referentiedag om week 1 te definiëren. Met andere woorden, deze instelling schrijft voor hoeveel dagen in week 1 datums binnen januari moeten zijn.

### Syntaxis:

#### ReferenceDay

ReferenceDay stelt in hoeveel dagen er in de eerste week van het jaar zitten. ReferenceDay kan worden ingesteld op elke waarde tussen 1 en 7. Elke waarde buiten het bereik 1-7 wordt geïnterpreteerd als het middelpunt van de week (4), wat overeenkomt met ReferenceDay ingesteld op 4.

Als u geen waarde selecteert voor de instelling ReferenceDay, dan zal de standaardwaarde ReferenceDay=0 weergegeven, wat zal worden geïnterpreteerd als het middelpunt van de week (4), zoals te zien is in de waardentabel ReferenceDay hieronder.

De functie ReferenceDay wordt vaak gebruikt in combinatie met de volgende functies:

#### Gerelateerde functies

Variabele	Interactie
<i>BrokenWeeks</i> (page 205)	Als de Qlik Sense-app met ononderbroken weken werkt, wordt de variabeleinstelling ReferenceDay afgedwongen. Als er echter gebroken weken worden gebruikt, begint week 1 op 1 januari en eindigt deze samen met de variabeleinstelling FirstWeekDay en negeert de vlag ReferenceDay.
<i>FirstWeekDay</i> (page 219)	Geheel getal dat definieert welke dag moet worden gebruikt als eerste dag van de week.

Met Qlik Sense kunnen de volgende waarden worden ingesteld voor ReferenceDay:

#### ReferenceDay-waarden

Waarde	Referentiedag
0 (standaard)	4 januari
1	1 januari
2	januari 2
3	3 januari

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

Waarde	Referentiedag
4	4 januari
5	5 januari
6	6 januari
7	7 januari

In het volgende voorbeeld definieert `ReferenceDay = 3` 3 januari als de referentiedag:

```
SET ReferenceDay=3; //(set January 3 as the reference day)
```

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Load-script dat gebruikmaakt van de standaardwaarde; ReferenceDay=0

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- De variabele `ReferenceDay` die is ingesteld op 0.
- De variabele `BrokenWeeks` die is ingesteld op 0, waardoor de app ononderbroken weken moet gebruiken.
- Een gegevensset met datums van eind 2019 tot begin 2020.

#### Load-script

```
SET BrokenWeeks = 0;  
SET ReferenceDay = 0;
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
Sales:
LOAD
date,
sales,
week(date) as week,
weekday(date) as weekday
Inline [
date,sales
12/27/2019,5000
12/28/2019,6000
12/29/2019,7000
12/30/2019,4000
12/31/2019,3000
01/01/2020,6000
01/02/2020,3000
01/03/2020,6000
01/04/2020,8000
01/05/2020,5000
01/06/2020,7000
01/07/2020,3000
01/08/2020,5000
01/09/2020,9000
01/10/2020,5000
01/11/2020,7000
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- week
- weekday

date	week	weekday
12/27/2019	52	Vr
12/28/2019	52	Za
12/29/2019	1	Zo
12/30/2019	1	Ma
12/31/2019	1	Di
01/01/2020	1	Wo
01/02/2020	1	Do
01/03/2020	1	Vr
01/04/2020	1	Za
01/05/2020	2	Zo

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

date	week	weekday
01/06/2020	2	Ma
01/07/2020	2	Di
01/08/2020	2	Wo
01/09/2020	2	Do
01/10/2020	2	Vr
01/11/2020	2	Za

Week 52 eindigt op zaterdag 28 december. Omdat `ReferenceDay` vereist dat 4 januari in week 1 wordt opgenomen, begint week 1 daarom op 29 december en eindigt op zaterdag 4 januari.

### Voorbeeld - ReferenceDay-variabele ingesteld op 5

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- De variabele `ReferenceDay` die is ingesteld op 5.
- De variabele `BrokenWeeks` die is ingesteld op 0, waardoor de app ononderbroken weken moet gebruiken.
- Een gegevensset met datums van eind 2019 tot begin 2020.

#### Load-script

```
SET BrokenWeeks = 0;
SET ReferenceDay = 5;

Sales:
LOAD
date,
sales,
week(date) as week,
weekday(date) as weekday
Inline [
date,sales
12/27/2019,5000
12/28/2019,6000
12/29/2019,7000
12/30/2019,4000
12/31/2019,3000
01/01/2020,6000
01/02/2020,3000
01/03/2020,6000
01/04/2020,8000
01/05/2020,5000
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

```
01/06/2020,7000
01/07/2020,3000
01/08/2020,5000
01/09/2020,9000
01/10/2020,5000
01/11/2020,7000
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- week
- weekday

Resultatentabel

date	week	weekday
12/27/2019	52	Vr
12/28/2019	52	Za
12/29/2019	53	Zo
12/30/2019	53	Ma
12/31/2019	53	Di
01/01/2020	53	Wo
01/02/2020	53	Do
01/03/2020	53	Vr
01/04/2020	53	Za
01/05/2020	1	Zo
01/06/2020	1	Ma
01/07/2020	1	Di
01/08/2020	1	Wo
01/09/2020	1	Do
01/10/2020	1	Vr
01/11/2020	1	Za

Week 52 eindigt op zaterdag 28 december. De variabele `brokenweeks` dwingt de app om ononderbroken weken te gebruiken. De referentiedagwaarde van 5 vereist dat 5 januari wordt opgenomen in week 1.

Dit is echter acht dagen na afloop van week 52 van het voorgaande jaar. Daarom begint week 53 op 29 december en eindigt op 4 januari. Week 1 begint op zondag 5 januari.

## 2 Werken met variabelen in de editor voor het laden van gegevens

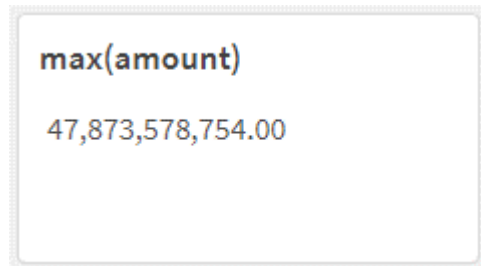
### ThousandSep

Het gedefinieerde scheidingsteken voor duizendtallen vervangt het cijfergroeperingssymbool van het besturingssysteem (landinstellingen).

#### Syntaxis:

##### ThousandSep

*Qlik Sense-object dat de variabele ThousandSep (met scheidingsteken voor duizendtallen) gebruikt*



Qlik Sense-apps zullen tekstvelden die voldoen aan deze indeling interpreteren als getallen. Deze opmaak wordt weergegeven in diagramobjecten wanneer de eigenschap van een **numeriek veld** is ingesteld op **Getal**.

Thousandsep is handig wanneer gegevensbronnen worden verwerkt die vanuit verschillende regionale instellingen zijn ontvangen.



*Als de ThousandSep-variabele wordt aangepast nadat objecten al zijn aangemaakt en zijn opgemaakt in de applicatie, moet de gebruiker ieder relevant veld opnieuw opmaken door de eigenschap **Getal** van **Getalnotatie** te deselecteren en vervolgens weer te selecteren.*

De volgende voorbeelden geven mogelijke toepassingen weer van de systeemvariabele ThousandSep:

```
set ThousandSep=','; //(for example, seven billion will be displayed as: 7,000,000,000)
```

```
set ThousandSep=' '; //(for example, seven billion will be displayed as: 7 000 000 000)
```

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

#### Gerelateerde onderwerpen

Onderwerp	Beschrijving
<i>DecimalSep</i> (page 217)	In gevallen van tekstveldinterpretatie, moeten de door deze functie verstrekte instellingen voor het scheidingsteken voor decimalen ook worden gerespecteerd. Voor getalopmaak wordt <b>DecimalSep</b> waar nodig door Qlik Sense gebruikt.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/YYYY. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Standaardsysteemvariabelen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in de tabel Transactions.
- Gebruik de standaarddefinitie voor de ThousandSep-variabele.

#### Load-script

```
Transactions:
Load
date,
id,
amount
Inline
[
date, id, amount
01/01/2022 , 1, 10000000441
01/02/2022 , 2, 21237492432
01/03/2022 , 3, 41249475336
01/04/2022 , 4, 24313369837
01/05/2022 , 5, 47873578754
01/06/2022 , 6, 24313884663
01/07/2022 , 7, 28545883436
01/08/2022 , 8, 35545828255
01/09/2022 , 9, 37565817436
01/10/2022 , 10, 3454343566
];
```

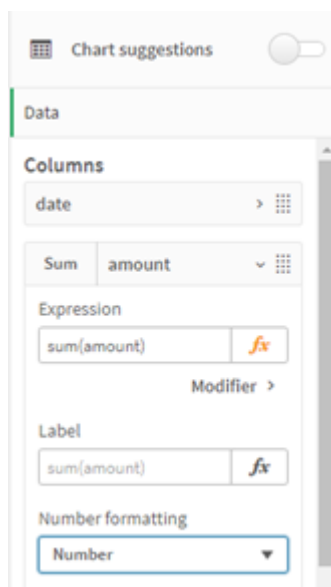
## 2 Werken met variabelen in de editor voor het laden van gegevens

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:date.
2. Voeg de volgende meting toe:  
=sum(amount)
3. Selecteer de meting in het eigenschappenvenster onder **Gegevens**.
4. Onder **Getalnotatie** selecteert u **Getal**.

*Getalnotatie aanpassen voor een diagrammeting*



Resultatentabel

date	=sum(amount)
01/01/2022	10,000,000,441.00
01/02/2022	21,237,492,432.00
01/03/2022	41,249,475,336.00
01/04/2022	24,313,369,837.00
01/05/2022	47,873,578,754.00
01/06/2022	24,313,884,663.00
01/07/2022	28,545,883,436.00
01/08/2022	35,545,828,255.00
01/09/2022	37,565,817,436.00
01/10/2022	3,454,343,566.00



## 2 Werken met variabelen in de editor voor het laden van gegevens

---

In dit voorbeeld wordt de standaarddefinitie `ThousandSep` gebruikt die is ingesteld op een komma als scheidingsteken. In de resultatentabel geeft de opmaak van het bedragveld een komma weer tussen duizendtallen.

### Voorbeeld 2 - Systeemvariabele wijzigen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als uit het eerste voorbeeld, geladen in de tabel `Transactions`.
- Aanpassing van de `ThousandSep`-definitie aan het begin van het script om een "\*" weer te geven als het scheidingsteken voor duizendtallen. Dit is een extreem voorbeeld en wordt uitsluitend gegeven om de functionaliteit van de variabele te laten zien.

De aanpassing die in dit voorbeeld wordt gebruikt, is extreem en komt zelden voor, maar wordt hier getoond om de functionaliteit van de variabele te demonstreren.

#### Load-script

```
SET ThousandSep='*';

Transactions:
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,1000000441
01/02/2022,2,21237492432
01/03/2022,3,41249475336
01/04/2022,4,24313369837
01/05/2022,5,47873578754
01/06/2022,6,24313884663
01/07/2022,7,28545883436
01/08/2022,8,35545828255
01/09/2022,9,37565817436
01/10/2022,10,3454343566
];
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.
2. Voeg de volgende meting toe:  
`=sum(amount)`
3. Selecteer de meting in het eigenschappenvenster onder **Gegevens**.
4. Onder **Getalnotatie** selecteert u **Aangepast**.

Resultatentabel

date	=sum(amount)
01/01/2022	10*000*000*441.00
01/02/2022	21*237*492*432.00
01/03/2022	41*249*475*336.00
01/04/2022	24*313*369*837.00
01/05/2022	47*873*578*754.00
01/06/2022	24*313*884*663.00
01/07/2022	28*545*883*436.00
01/08/2022	35*545*828*255.00
01/09/2022	37*565*817*436.00
01/10/2022	3*454*343*566.00

Aan het begin van het script kan de systeemvariabele `ThousandSep` worden gewijzigd in een `''`. In de resultatentabel geeft de opmaak van het bedragveld een `''` weer tussen duizendtallen.

### Voorbeeld 3 - Tekstinterpretatie

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in de tabel `Transactions`.
- Gegevens waarbij het numerieke veld een tekstindeling heeft met een komma die als scheidingsteken voor duizendtallen wordt gebruikt.
- Gebruik van de standaardstelsysteemvariabele `ThousandSep`.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Load-script

```
Transactions:
Load
date,
id,
amount
Inline
[
date,id,amount
01/01/2022,1,'10,000,000,441'
01/02/2022,2,'21,492,432'
01/03/2022,3,'4,249,475,336'
01/04/2022,4,'24,313,369,837'
01/05/2022,5,'4,873,578,754'
01/06/2022,6,'313,884,663'
01/07/2022,7,'2,545,883,436'
01/08/2022,8,'545,828,255'
01/09/2022,9,'37,565,817,436'
01/10/2022,10,'3,454,343,566'
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:date.
2. Voeg de volgende meting toe:  
=sum(amount)
3. Selecteer de meting in het eigenschappenvenster onder **Gegevens**.
4. Onder **Getalnotatie** selecteert u **Getal**.
5. Voeg de volgende meting toe om te bepalen of het bedragveld al dan niet een numerieke waarde is:  
=isnum(amount)

Resultatentabel

date	=sum(amount)	=isnum(amount)
01/01/2022	10,000,000,441.00	-1
01/02/2022	21,492,432.00	-1
01/03/2022	4,249,475,336.00	-1
01/04/2022	24,313,369,837.00	-1
01/05/2022	4,873,578,754.00	-1
01/06/2022	313,884,663.00	-1
01/07/2022	2,545,883,436.00	-1

## 2 Werken met variabelen in de editor voor het laden van gegevens

date	=sum(amount)	=isnum(amount)
01/08/2022	545,828,255.00	-1
01/09/2022	37,565,817,436.00	-1
01/10/2022	3*454*343*566.00	-1

Zodra de gegevens zijn geladen, zien we dat Qlik Sense het bedragveld als een numerieke waarde heeft geïnterpreteerd omdat de gegevens voldoen aan de `thousandsep`-variabele. Dit wordt gedemonstreerd door de `isnum()`-functie die iedere invoer evalueert tot -1 of `TRUE`.



In Qlik Sense wordt de booleaanse waarde *Waar* vertegenwoordigd door -1 en de waarde *Onwaar* door 0.

### TimeFormat

De gedefinieerde notatie vervangt de tijdnotatie van het besturingssysteem (landinstellingen).

#### Syntaxis:

```
TimeFormat
```

#### Voorbeeld:

```
set TimeFormat='hh:mm:ss';
```

### TimestampFormat

De gedefinieerde notatie vervangt de datum- en tijdnotatie van het besturingssysteem (landinstellingen).

#### Syntaxis:

```
TimestampFormat
```

#### Voorbeeld:

In de volgende voorbeelden wordt `1983-12-14T13:15:30Z` als invoer voor de tijdstempel gebruikt om de resultaten van verschillende **SET TimestampFormat**-opdrachten weer te geven. `YYYYMMDD` wordt gebruikt als datumnotatie en `h:mm:ss TT` als tijdnotatie. Bovenaan het load-script voor gegevens wordt de datumnotatie bepaald met de opdracht **SET DateFormat** en de tijdnotatie met de opdracht **SET TimeFormat**.

#### Resultaten

Voorbeeld	Resultaat
<code>SET TimestampFormat='YYYYMMDD';</code>	19831214
<code>SET TimestampFormat='M/D/YY hh:mm:ss[.fff]';</code>	12/14/83 13:15:30

## 2 Werken met variabelen in de editor voor het laden van gegevens

Voorbeeld	Resultaat
<code>SET TimestampFormat='DD/MM/YYYY hh:mm:ss[.fff]';</code>	14/12/1983 13:15:30
<code>SET TimestampFormat='DD/MM/YYYY hh:mm:ss[.fff] TT';</code>	14/12/1983 1:15:30 PM
<code>SET TimestampFormat='YYYY-MM-DD hh:mm:ss[.fff] TT';</code>	1983-12-14 01:15:30

### Voorbeelden: Load-script

Voorbeeld: Load-script

In het eerste load-script wordt `SET TimestampFormat='DD/MM/YYYY h:mm:ss[.fff] TT'` gebruikt. In het tweede load-script wordt de opmaak van de tijdstempel gewijzigd in `SET TimestampFormat='MM/DD/YYYY hh:mm:ss[.fff]'`. De verschillende resultaten geven weer hoe de opdracht **SET TimeFormat** werkt met verschillende indelingen voor tijdgegevens.

De onderstaande tabel geeft de gegevensverzameling weer die is gebruikt in de daaropvolgende load-scripts. De tweede kolom van de tabel geeft de opmaak van de verschillende tijdstempels in de gegevensverzameling weer. De eerste vijf tijdstempels volgen de ISO 8601-regels. De zesde niet.

### Gegevensverzameling

*Tabel met de tijdgegevens die zijn gebruikt en de opmaak voor elke tijdstempel in de gegevensverzameling.*

transaction_timestamp	time data format
2018-08-30	YYYY-MM-DD
20180830T193614.857	YYYYMMDDhhmmss.sss
20180830T193614.857+0200	YYYYMMDDhhmmss.sss±hhmm
2018-09-16T12:30-02:00	YYYY-MM-DDhh:mm±hh:mm
2018-09-16T13:15:30Z	YYYY-MM-DDhh:mmZ
9/30/18 19:36:14	M/D/YY hh:mm:ss

Maak een nieuwe sectie aan in de **Editor voor laden van gegevens**, voeg vervolgens het voorbeeldscript toe en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in uw app toe om het resultaat te bekijken.

### Load-script

```
SET FirstWeekDay=0; SET BrokenWeeks=1; SET ReferenceDay=0; SET
DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun'; SET
LongDayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday'; SET
DateFormat='YYYYMMDD'; SET TimestampFormat='DD/MM/YYYY h:mm:ss[.fff] TT'; Transactions: Load
*, Timestamp(transaction_timestamp, 'YYYY-MM-DD hh:mm:ss[.fff]') as LogTimestamp ; Load *
Inline [ transaction_id, transaction_timestamp, transaction_amount, transaction_quantity,
discount, customer_id, size, color_code 3750, 2018-08-30, 12423.56, 23, 0,2038593, L, Red
3751, 20180830T193614.857, 5356.31, 6, 0.1, 203521, m, orange 3752, 20180830T193614.857+0200,
15.75, 1, 0.22, 5646471, s, blue 3753, 2018-09-16T12:30-02:00, 1251, 7, 0, 3036491, l, black
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

3754, 2018-09-16T13:15:30Z, 21484.21, 1356, 75, 049681, xs, Red 3755, 9/30/18 19:36:14, - 59.18, 2, 0.3333333333333333, 2038593, M, Blue ];

### Resultaten

*Klik Sense-tabel met de resultaten van de variabele voor TimestampFormat-interpretatie die wordt gebruikt in het load-script. De laatste tijdstempel in de gegevensverzameling geeft geen correcte datum als resultaat.*

transaction_id	transaction_timestamp	LogTimeStamp
3750	2018-08-30	2018-08-30 00:00:00
3751	20180830T193614.857	2018-08-30 19:36:14
3752	20180830T193614.857+0200	2018-08-30 17:36:14
3753	2018-09-16T12:30-02:00	2018-09-16 14:30:00
3754	2018-09-16T13:15:30Z	2018-09-16 13:15:30
3755	9/30/18 19:36:14	-

Het volgende load-script gebruikt dezelfde gegevensverzameling. Het gebruikt echter *SET TimestampFormat='MM/DD/YYYY hh:mm:ss[.fff]'* voor overeenkomst met de zesde tijdstempel die niet voldoet aan ISO 8601.

Vervang het vorige voorbeeldscript in de **Editor voor laden van gegevens** met het onderstaande en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in uw app toe om het resultaat te bekijken.

### Load-script

```
SET FirstWeekDay=0; SET BrokenWeeks=1; SET ReferenceDay=0; SET
DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun'; SET
LongDayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday'; SET
DateFormat='YYYYMMDD'; SET TimestampFormat='MM/DD/YYYY hh:mm:ss[.fff]'; Transactions: Load
*, Timestamp(transaction_timestamp, 'YYYY-MM-DD hh:mm:ss[.fff]') as LogTimestamp ; Load *
Inline [ transaction_id, transaction_timestamp, transaction_amount, transaction_quantity,
discount, customer_id, size, color_code 3750, 2018-08-30, 12423.56, 23, 0,2038593, L, Red
3751, 20180830T193614.857, 5356.31, 6, 0.1, 203521, m, orange 3752, 20180830T193614.857+0200,
15.75, 1, 0.22, 5646471, s, blue 3753, 2018-09-16T12:30-02:00, 1251, 7, 0, 3036491, 1, Black
3754, 2018-09-16T13:15:30Z, 21484.21, 1356, 75, 049681, xs, Red 3755, 9/30/18 19:36:14, -
59.18, 2, 0.3333333333333333, 2038593, M, Blue ];
```

### Resultaten

*Klik Sense-tabel met de resultaten van de variabele voor TimestampFormat-interpretatie die wordt gebruikt in het load-script.*

transaction_id	transaction_timestamp	LogTimeStamp
3750	2018-08-30	2018-08-30 00:00:00

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

transaction_id	transaction_timestamp	LogTimeStamp
3751	20180830T193614.857	2018-08-30 19:36:14
3752	20180830T193614.857+0200	2018-08-30 17:36:14
3753	2018-09-16T12:30-02:00	2018-09-16 14:30:00
3754	2018-09-16T13:15:30Z	2018-09-16 13:15:30
3755	9/30/18 19:36:14	2018-09-16 19:36:14

### 2.15 Direct Discovery-variabelen

#### Direct Discovery-systeemvariabelen

##### DirectCacheSeconds

U kunt een cachelimit voor de Direct Discovery-queryresultaten instellen voor visualisaties. Wanneer deze tijdslimit is bereikt, maakt Qlik Sense de cache leeg als nieuwe Direct Discovery-query's worden gemaakt. Qlik Sense zoekt in de brongegevens naar de brongegevens voor de selecties en maakt de cache opnieuw voor de aangegeven tijdslimit. Het resultaat voor elke combinatie van selecties wordt apart in de cache opgenomen. Met andere woorden, de cache wordt voor elke selectie apart vernieuwd, zodat de ene selectie de cache alleen vernieuwt voor de geselecteerde velden en een tweede selectie de cache vernieuwt voor de relevante velden. Als de tweede selectie velden bevat die zijn vernieuwd in de eerste selectie, worden zij niet opnieuw bijgewerkt in de cache als de cachelimit nog niet is bereikt.

De Direct Discovery-cache is niet van toepassing op **tabel**visualisaties. Bij tabelselecties wordt telkens een query op de gegevensbron uitgevoerd.

De grenswaarde moet worden ingesteld in seconden. De standaard cachelimit is 1800 seconden (30 minuten).

De waarde die wordt gebruikt voor **DirectCacheSeconds** is de waarde die is ingesteld tijdens de uitvoering van de **DIRECT QUERY**-opdracht. De waarde kan niet worden gewijzigd tijdens de runtime.

##### Voorbeeld:

```
SET DirectCacheSeconds=1800;
```

##### DirectConnectionMax

U kunt asynchrone, parallelle aanroepen uitvoeren op de database door gebruik te maken van de voorziening voor verbindingsgroepering. De syntaxis van het laadscript voor het instellen van de groeperingsvoorziening is als volgt:

```
SET DirectConnectionMax=10;
```

De numerieke instelling geeft het maximaal aantal databaseverbindingen op dat door de Direct Discovery-code moet worden gebruikt tijdens het bijwerken van een werkblad. De standaardinstelling is 1.



*Deze variabele moet met zorg worden gebruikt. Bekend is dat bij instelling op een waarde groter dan 1 problemen ontstaan bij het maken van een verbinding met Microsoft SQL Server.*

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### DirectUnicodeStrings

Direct Discovery kan ondersteuning bieden voor de selectie van uitgebreide Unicode-gegevens door gebruik te maken van de SQL-standaardindeling voor uitgebreide tekenreeksliterals (N'<uitgebreide tekenreeks>') zoals vereist bij sommige databases (met name SQL Server). Het gebruik van deze syntaxis kan worden ingeschakeld voor Direct Discovery met de scriptvariabele **DirectUnicodeStrings**.

Als deze variabele wordt ingesteld op 'waar', wordt het gebruik van de standaard ANSI-markering voor brede tekens "N" vóór de tekenreeksliterals mogelijk gemaakt. Niet alle databases ondersteunen deze standaard. De standaardinstelling is 'false'.

### DirectDistinctSupport

Als een **DIMENSION**-veldwaarde wordt geselecteerd in een Qlik Sense-object, wordt een query gegenereerd voor de brondatabase. Wanneer de query groepering vereist, gebruikt Direct Discovery het trefwoord **DISTINCT** om alleen unieke waarden te selecteren. Sommige databases vereisen echter het trefwoord **GROUP BY**. Stel **DirectDistinctSupport** in op 'false' om **GROUP BY** te genereren in plaats van **DISTINCT** in query's voor unieke waarden.

```
SET DirectDistinctSupport='false';
```

If **DirectDistinctSupport** is ingesteld op true, wordt **DISTINCT** gebruikt. Als deze niet is ingesteld, wordt standaard gebruikgemaakt van **DISTINCT**.

### DirectEnableSubquery

In scenario's met meerdere tabellen met hoge cardinaliteit, is het mogelijk om subquery's te genereren in de SQL-query in plaats van een grote IN-clausule te genereren. Dit wordt geactiveerd door **DirectEnableSubquery** in te stellen op 'true'. De standaardwaarde is 'false'.



*Als **DirectEnableSubquery** is ingeschakeld, kunt u geen tabellen laden die zich niet in de Direct Discovery-modus bevinden.*

```
SET DirectEnableSubquery='true';
```

## Teradata-variabelen voor query banding

Teradata-query banding is een functie waarbij ondernemingstoepassingen in staat worden gesteld samen te werken met de onderliggende Teradata-database om betere boekhoudvoorzieningen, prioriteitsstelling en werklustbeheer te bieden. Via query banding kunt u metagegevens, zoals gebruikersreferenties, aan een query koppelen.

Er zijn twee variabelen beschikbaar. Beide zijn tekenreeksen die worden geëvalueerd en naar de database verzonden.

### SQLSessionPrefix

Deze tekenreeks wordt verzonden wanneer er een verbinding met de database wordt gemaakt.

```
SET SQLSessionPrefix = 'SET QUERY_BAND = ' & Chr(39) & 'who=' & OSUser() & ';' & Chr(39) & ' FOR SESSION;';
```

Als **OSUser()** bijvoorbeeld *WAlsb* retourneert, wordt deze geëvalueerd tot `SET QUERY_BAND = 'who=WAlsb;' FOR SESSION;`, die naar de database wordt verzonden bij het tot stand brengen van de verbinding.



## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### SQLQueryPrefix

Deze tekenreeks wordt verzonden voor elke afzonderlijke query.

```
SET SQLSessionPrefix = 'SET QUERY_BAND = ' & Chr(39) & 'who=' & OSuser() & ';' & Chr(39) & '
FOR TRANSACTION;';
```

### Direct Discovery-tekenvariabelen

#### DirectFieldColumnDelimiter

U kunt het teken dat wordt gebruikt als veldscheidingstekens instellen in **Direct Query**-opdrachten voor databases die een ander teken dan een komma vereisen als veldscheidingstekens. Het opgegeven teken moet worden omringd door enkele aanhalingstekens in de **SET**-instructie.

```
SET DirectFieldColumnDelimiter= '|'
```

#### DirectStringQuoteChar

U kunt een teken gebruiken om tekenreeksen van aanhalingstekens te voorzien in een gegenereerde query. De standaard is een enkel aanhalingsteken. Het opgegeven teken moet worden omringd door enkele aanhalingstekens in de **SET**-opdracht.

```
SET DirectStringQuoteChar= '''';
```

#### DirectIdentifierQuoteStyle

U kunt opgeven dat niet-ANSI-aanhalingstekens worden gebruikt voor id's in gegenereerde query's. Momenteel vormen GoogleBQ-aanhalingstekens de enige niet-ANSI-aanhalingstekens die beschikbaar zijn. De standaardinstelling is ANSI. Er kan gebruik worden gemaakt van hoofdletters, kleine letters en gemengde tekens (ANSI, ansi, Ansi).

```
SET DirectIdentifierQuoteStyle="GoogleBQ";
```

Zo wordt bijvoorbeeld gebruikgemaakt van ANSI-aanhalingstekens in de volgende **SELECT**-opdracht:

```
SELECT [Quarter] FROM [qvTest].[sales] GROUP BY [Quarter]
```

Als **DirectIdentifierQuoteStyle** is ingesteld op "GoogleBQ", wordt in de **SELECT**-opdracht als volgt gebruikgemaakt van aanhalingstekens:

```
SELECT [Quarter] FROM [qvTest.sales] GROUP BY [Quarter]
```

#### DirectIdentifierQuoteChar

U kunt een teken gebruiken het gebruik van aanhalingstekens voor id's in een gegenereerde query te besturen. Deze kan worden ingesteld op één teken (zoals een dubbel aanhalingsteken) of twee tekens (zoals een stel vierkante haken). De standaard is een dubbel aanhalingsteken.

```
SET DirectIdentifierQuoteChar='[]';
SET DirectIdentifierQuoteChar='``';
SET DirectIdentifierQuoteChar=' ' ;
SET DirectIdentifierQuoteChar='''''';
```

#### DirectTableBoxListThreshold

Als Direct Discovery-velden worden gebruikt in een **Table**visualisatie, wordt een drempelwaarde ingesteld om het aantal weergegeven rijen te beperken. De standaard drempelwaarde is 1000. De instelling voor de standaard drempelwaarde kan worden gewijzigd door de variabele **DirectTableBoxListThreshold** in te

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

stellen in het load-script. Bijvoorbeeld:

```
SET DirectTableBoxListThreshold=5000;
```

De ingestelde drempelwaarde geldt alleen voor **Tabel**visualisaties die Direct Discovery-velden bevatten.

**Tabel**visualisaties die alleen velden in het geheugen bevatten worden niet beperkt door de instelling **DirectTableBoxListThreshold**.

Er worden geen velden weergegeven in de **Tabel**visualisatie totdat de selectie minder records heeft dat de drempelwaarde.

### Direct Discovery-variabelen voor getalinterpretatie

#### **DirectMoneyDecimalSep**

Het gedefinieerde decimaalscheidingsteken vervangt het decimaalteken voor valuta in de SQL-opdracht die wordt gegenereerd voor het laden van gegevens met Direct Discovery. Dit teken moet overeenkomen met het teken dat wordt gebruikt in **DirectMoneyFormat**.

Standaardwaarde is `'.'`

#### **Voorbeeld:**

```
Set DirectMoneyDecimalSep='.';
```

#### **DirectMoneyFormat**

Het gedefinieerde teken vervangt de valutannotatie in de SQL-opdracht die wordt gegenereerd voor het laden van gegevens met Direct Discovery. Het valutateken voor het scheidingsteken voor duizendtallen moet niet worden opgenomen.

Standaardwaarde is `'#.0000'`

#### **Voorbeeld:**

```
Set DirectMoneyFormat='#.0000';
```

#### **DirectTimeFormat**

De gedefinieerde tijdnotatie vervangt de tijdnotatie in de SQL-opdracht die wordt gegenereerd voor het laden van gegevens met Direct Discovery.

#### **Voorbeeld:**

```
Set DirectTimeFormat='hh:mm:ss';
```

#### **DirectDateFormat**

De gedefinieerde datumnotatie vervangt de datumnotatie in de SQL-opdracht die wordt gegenereerd voor het laden van gegevens met Direct Discovery.

#### **Voorbeeld:**

```
Set DirectDateFormat='MM/DD/YYYY';
```

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### DirectTimeStampFormat

De gedefinieerde indeling vervangt de datum- en tijdnnotatie in het SQL-opdracht die is gegenereerd in het SQL-opdracht voor het laden van gegevens laden met behulp van Direct Discovery.

#### Voorbeeld:

```
set DirectTimeStampFormat='M/D/YY hh:mm:ss[.fff]';
```

## 2.16 Foutvariabelen

Na uitvoering van het script zijn alle foutvariabelen voorzien van een waarde. De eerste variabele, `ErrorMode`, wordt ingevoerd door de gebruiker en de laatste drie door Qlik Sense met informatie over fouten in het script.

### Overzicht van foutvariabelen

Elke variabele wordt na het overzicht verder omschreven. U kunt ook in de syntaxis op de variabelenaam klikken om meteen naar de details van die specifieke variabele te gaan.

Raadpleeg de Qlik Sense online Help voor nadere details over de variabelen.

#### ErrorMode

Deze foutvariabele bepaalt welke actie in Qlik Sense wordt uitgevoerd als bij uitvoering van het script een fout optreedt.

**ErrorMode**

#### ScriptError

Deze foutvariabele retourneert de foutcode van de laatst uitgevoerde scriptopdracht.

**ScriptError**

#### ScriptErrorCount

De foutvariabele retourneert het totale aantal opdrachten die fouten hebben veroorzaakt bij de uitvoering van het huidige script. Deze variabele wordt voorafgaand aan de uitvoering van het script altijd op de beginwaarde 0 ingesteld.

**ScriptErrorCount**

#### ScriptErrorList

Deze foutvariabele bevat een samengestelde lijst van alle scriptfouten die tijdens de laatste uitvoering van het script zijn opgetreden. Elke fout staat op een nieuwe regel.

**ScriptErrorList**

## ErrorMode

Deze foutvariabele bepaalt welke actie in Qlik Sense wordt uitgevoerd als bij uitvoering van het script een fout optreedt.

#### Syntaxis:

**ErrorMode**

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

### Argumenten:

Argumenten

Argument	Beschrijving
<b>ErrorMode=1</b>	De standaardinstelling. De uitvoering van het script wordt gestopt en de gebruiker wordt gevraagd om actie te ondernemen (niet-batchmodus).
<b>ErrorMode =0</b>	De fout wordt simpelweg genegeerd in Qlik Sense en het script wordt verder uitgevoerd bij de volgende scriptopdracht.
<b>ErrorMode =2</b>	In Qlik Sense wordt een foutbericht "Uitvoering van script is mislukt..." gegenereerd zodra de fout optreedt, zonder dat de gebruiker eerst wordt gevraagd om actie te ondernemen.

### Voorbeeld:

```
set ErrorMode=0;
```

## ScriptError

Deze foutvariabele retourneert de foutcode van de laatst uitgevoerde scriptopdracht.

### Syntaxis:

**ScriptError**

Elke keer dat de scriptopdracht correct is uitgevoerd, wordt deze variabele teruggezet op 0. Als een fout optreedt, wordt een interne Qlik Sense-foutcode ingesteld. Foutcodes zijn dubbele waarden met een numerieke component en een tekstcomponent. De volgende foutcodes komen voor:

Foutcodes van scripts

Foutcode	Beschrijving
0	Geen fout. Dubbele waardetekst is leeg.
1	Algemene fout.
2	Syntaxisfout.
3	Algemene ODBC-fout.
4	Algemene OLE DB-fout.
5	Algemene aangepaste databasefout.
6	Algemene XML-fout.
7	Algemene HTML-fout.

## 2 Werken met variabelen in de editor voor het laden van gegevens

---

Foutcode	Beschrijving
8	Bestand niet gevonden.
9	Database niet gevonden.
10	Tabel niet gevonden.
11	Veld niet gevonden.
12	Bestand heeft verkeerde indeling.
16	Semantische fout.

### Voorbeeld:

```
set ErrorMode=0;

LOAD * from abc.qvf;

if ScriptError=8 then

exit script;

//no file;

end if
```

### ScriptErrorCount

De foutvariabele retourneert het totale aantal opdrachten die fouten hebben veroorzaakt bij de uitvoering van het huidige script. Deze variabele wordt voorafgaand aan de uitvoering van het script altijd op de beginwaarde 0 ingesteld.

#### Syntaxis:

```
ScriptErrorCount
```

### ScriptErrorList

Deze foutvariabele bevat een samengestelde lijst van alle scriptfouten die tijdens de laatste uitvoering van het script zijn opgetreden. Elke fout staat op een nieuwe regel.

#### Syntaxis:

```
ScriptErrorList
```

## 2 Scriptuitdrukkingen

Uitdrukkingen kunnen zowel in **LOAD**- als in **SELECT**-opdrachten worden gebruikt. De hier beschreven syntaxis en functies zijn van toepassing op de **LOAD**-opdracht en niet op de **SELECT**-opdracht, omdat deze laatste wordt geïnterpreteerd door het ODBC-stuurprogramma en niet door Qlik Sense. De meeste ODBC-stuurprogramma's zijn echter in staat een aantal van de onderstaande functies te interpreteren.

Uitdrukkingen bestaan uit functies, velden en operatoren die worden gecombineerd volgens een speciale syntaxis.

Alle uitdrukkingen in een Qlik Sense-script retourneren, afhankelijk van wat van toepassing is, een getal en/of een tekenreeks. Logische functies en operators leveren 0 op voor False en -1 voor True. Conversies van getal naar tekenreeks en vice versa zijn daarbij geïmpliceerd. Logische operatoren en functies interpreteren 0 als False en al het andere als True.

De algemene syntaxis voor een uitdrukking is:

Algemene syntaxis

Uitdrukking	Velden	Operator
expression ::= (constant	constant	
expression ::= (constant	fieldref	
expression ::= (constant	operator1 expression	
expression ::= (constant	expression operator2 expression	
expression ::= (constant	function	
expression ::= (constant	( expression )	)

waarbij:

- **constant** een tekenreeks (tekst, datum of tijd) is die tussen enkele rechte aanhalingstekens staat, of een getal. Constanten worden geschreven zonder scheidingstekens voor duizendtallen en met een punt als scheidingstekens voor decimalen.
- **fieldref** is een veldnaam van de geladen tabel.
- **operator1** is een unaire operator (werkt op één uitdrukking, de uitdrukking rechts van de operator).
- **operator2** is een binaire operator (werkt op twee uitdrukkingen, een aan elke zijde).
- **function ::= functionname( parameters)**
- **parameters ::= expression { , expression }**

Het aantal en de typen parameters zijn niet willekeurig. Deze zijn afhankelijk van de gebruikte functie.

Uitdrukkingen en functies kunnen naar believen worden genest, en zolang de uitdrukking een interpreteerbare waarde retourneert, geeft Qlik Sense geen foutberichten weer.

### 3 Diagramuitdrukkingen

Een uitdrukking (visualisatie) is een combinatie van functies, velden en rekenkundige operatoren (+ \* / =) en andere metingen. Uitdrukkingen worden gebruikt voor het verwerken van gegevens in de app om een resultaat op te leveren dat kan worden bekeken in een visualisatie. Hun gebruik beperkt zich niet tot metingen. U kunt visualisaties bouwen die dynamischer en krachtiger zijn, met uitdrukkingen voor titels, subtitels, voetnoten en zelfs dimensies.

Dit betekent bijvoorbeeld dat de titel van een visualisatie geen statische tekst hoeft te zijn, maar kan worden gemaakt op basis van een uitdrukking waarvan het resultaat verandert afhankelijk van de uitgevoerde selecties.



*Voor gedetailleerde naslaginformatie over scriptfuncties en diagramfuncties, raadpleegt u de [Scriptsyntaxis en diagramfuncties](#).*

#### 3.1 Het aggregatiebereik definiëren

Er zijn gewoonlijk twee factoren die samen bepalen welke records worden gebruikt voor het definiëren van de waarde van aggregatie in een uitdrukking. Bij het werken in visualisaties zijn deze factoren:

- Dimensionale waarde (bij aggregatie in een diagramuitdrukking)
- Selecties

Samen definiëren deze factoren het bereik van de aggregatie. U komt mogelijk situaties tegen waarbij u de selectie, de dimensie of beide buiten beschouwing wilt laten bij de berekening. In diagramfuncties kunt u dit bereiken door de kwalificatie TOTAL, set-analyse of een combinatie van beide te gebruiken.

#### Aggregatie: methode en beschrijving

Method	Beschrijving
Kwalificatie TOTAL	<p>Bij gebruik van de kwalificatie TOTAL binnen uw aggregatiefunctie wordt de dimensionale waarde buiten beschouwing gelaten.</p> <p>De aggregatie wordt uitgevoerd op alle mogelijke veldwaarden.</p> <p>Na de kwalificatie <b>TOTAL</b> kan een lijst met een of meer veldnamen tussen punthaken worden opgegeven. Deze veldnamen moeten een subset zijn van de dimensievariabelen van het diagram. In dit geval wordt bij de berekening geen rekening gehouden met de dimensievariabelen van het diagram met uitzondering van de aangegeven dimensievariabelen. Er wordt één waarde geretourneerd voor elke combinatie van veldwaarden in de opgegeven dimensievelden. Ook velden die momenteel niet een dimensie zijn in een diagram, kunnen worden opgenomen in de lijst. Dit kan nuttig zijn bij groepsdimensies waarbij de dimensievelden niet vast zijn. Als alle variabelen in de groep worden aangegeven, werkt de functie als het drill-downniveau verandert.</p>
Set-analyse	Bij gebruik van set-analyse binnen uw aggregatie wordt de selectie buiten beschouwing gelaten. De aggregatie wordt uitgevoerd op alle waarden verdeeld over de dimensies.
Kwalificatie TOTAL en set-analyse	Bij gebruik van de kwalificatie <b>TOTAL</b> en set-analyse binnen uw aggregatie worden de selectie en de dimensies buiten beschouwing gelaten.
Kwalificatie ALL	<p>Bij gebruik van de kwalificatie <b>ALL</b> binnen uw aggregatie worden de selectie en de dimensies buiten beschouwing gelaten. Het equivalent kan worden bereikt met de instructie {1} voor set-analyse en de kwalificatie <b>TOTAL</b> :</p> <p>=sum(All sales)</p> <p>=sum({1} Total sales)</p>

#### Voorbeeld: Kwalificatie TOTAL

In het volgende voorbeeld wordt getoond hoe TOTAL kan worden gebruikt voor het berekenen van een relatief aandeel. Ervan uitgaande dat Q2 is geselecteerd, wordt met TOTAL de som van alle waarden berekend, waarbij de dimensies buiten beschouwing worden gelaten.

#### Voorbeeld: Kwalificatie total

Year	Quarter	Sum(Amount)	Sum(TOTAL Amount)	Sum(Amount)/Sum(TOTAL Amount)
		3000	3000	100%
2012	Q2	1700	3000	56,7%
2013	Q2	1300	3000	43,3%





Als u de getallen wilt weergeven als een percentage, in het eigenschappenvenster, selecteert u voor de meting die u wilt weergeven als een percentagewaarde, onder **Number formatting**, de optie **Number** en kiest u onder **Formatting** de optie **Simple** en een van de %-indelingen.

#### Voorbeeld: Set-analyse

In het volgende voorbeeld wordt getoond hoe set-analyse kan worden gebruikt om een vergelijking tussen gegevensverzamelingen te maken voordat er iets is geselecteerd. Ervan uitgaande dat Q2 is geselecteerd, wordt via set-analyse met de set-definitie {1} de som van alle waarden berekend, waarbij selecties buiten beschouwing worden gelaten maar de som wordt verdeeld over de dimensies.

Voorbeeld: Set-analyse

Year	Quarter	Sum(Amount)	Sum({1} Amount)	Sum(Amount)/Sum({1} Amount)
		3000	10800	27,8%
2012	Q1	0	1100	0%
2012	Q3	0	1400	0%
2012	Q4	0	1800	0%
2012	Q2	1700	1700	100%
2013	Q1	0	1000	0%
2013	Q3	0	1100	0%
2013	Q4	0	1400	0%
2013	Q2	1300	1300	100%

#### Voorbeeld: Kwalificatie TOTAL en set-analyse

In het volgende voorbeeld wordt getoond hoe set-analyse en de kwalificatie TOTAL kunnen worden gecombineerd om een vergelijking tussen gegevensverzamelingen te maken voordat er iets is geselecteerd en over alle dimensies heen. Ervan uitgaande dat Q2 is geselecteerd, wordt via set-analyse met de set-definitie {1} en de kwalificatie TOTAL de som van alle waarden berekend waarbij eventuele selecties en de dimensies buiten beschouwing worden gelaten.

Voorbeeld: Kwalificatie TOTAL en set-analyse

Year	Quarter	Sum (Amount)	Sum({1} TOTAL Amount)	Sum(Amount)/Sum({1} TOTAL Amount)
		3000	10800	27,8%
2012	Q2	1700	10800	15,7%
2013	Q2	1300	10800	12%

Gegevens die worden gebruikt in voorbeelden:

```
AggregationScope: LOAD * inline [ Year Quarter Amount 2012 Q1 1100 2012 Q2 1700 2012 Q3 1400  
2012 Q4 1800 2013 Q1 1000 2013 Q2 1300 2013 Q3 1100 2013 Q4 1400] (delimiter is ' ');
```

### 3.2 Set-analyse

Als u een selectie maakt in een app, definieert u een subset van records in de gegevens. Aggregatiefuncties zoals `sum()`, `Max()`, `Min()`, `Avg()` en `count()` worden berekend op basis van deze subset.

Dit betekent dat uw selectie het bereik van de aggregatie definieert; deze definieert de set met records op basis waarvan berekeningen worden uitgevoerd.

Set-analyse biedt een manier om een bereik te definiëren dat verschilt van de set met records die wordt gedefinieerd door de huidige selectie. Dit nieuwe bereik kan ook als een alternatieve selectie worden beschouwd.

Dit kan nuttig zijn als u de huidige selectie wilt vergelijken met een specifieke waarde, bijvoorbeeld een waarde van vorig jaar of het wereldwijde marktaandeel.

### Ingestelde uitdrukkingen

Set-uitdrukkingen kunnen binnen en buiten aggregatiefuncties worden gebruikt en worden tussen accolades geplaatst.

#### Voorbeeld: Binnenste set-uitdrukking

```
sum( {<Year={2021}>} Sales )
```

#### Voorbeeld: Buitenste set-uitdrukking

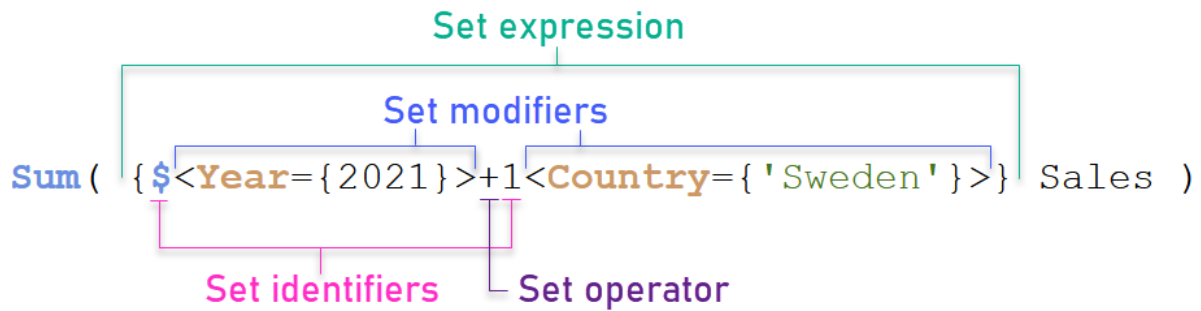
```
{<Year={2021}>} Sum(Sales) / Count(distinct Customer)
```

Een set-uitdrukking bestaat uit een combinatie van de volgende elementen:

- **Identifiers.** Een set-id vertegenwoordigt een selectie die ergens anders is gedefinieerd. Deze vertegenwoordigt ook een specifieke set met records in de gegevens. Dit kan de huidige selectie, een selectie van een bladwijzer of een selectie van een alternatieve status zijn. Een simpele set-uitdrukking bestaat uit een enkele id, zoals een dollarteken, `{ $ }`, oftewel alle records in de huidige selectie.  
Voorbeelden: `$`, `1`, `BookMark1`, `state2`
- **Operators.** Een set-operator kan worden gebruikt om samenvoegingen, verschillen of snijpunten tussen verschillende set-id's te maken. Op deze manier kunt u een subset of een superset maken van de selecties die zijn gedefinieerd door de set-id's.  
Voorbeelden: `+`, `-`, `*`, `/`
- **Modifiers.** Een set-modificatie kan worden toegevoegd aan de set-id om de selectie te wijzigen. Een modificatie kan ook afzonderlijk worden gebruikt om de standaard-id te wijzigen. Een modificatie moet tussen punthaken `<...>` worden geplaatst.  
Voorbeelden: `<Year={2020}>`, `<Supplier={ACME}>`

De elementen worden gecombineerd tot set-uitdrukkingen.

### Elementen in een set-uitdrukking



De bovenstaande set-uitdrukking is bijvoorbeeld gebaseerd op de aggregatie `sum(Sales)`.

De eerste operand retourneert verkopen voor het jaar 2021 van de huidige selectie. Dit wordt aangegeven door de set-id `$` en de modificatie die een selectie van het jaar 2021 bevat. De tweede operand retourneert `sales` voor Sweden en negeert de huidige selectie, die wordt aangegeven door de set-id `1`.

Tot slot retourneert de uitdrukking een set met records die toebehoort aan een van de twee set-operanden, zoals wordt aangegeven door de set-operator `+`.

## Voorbeelden

Voorbeelden waarin de bovenstaande elementen van de set-uitdrukking worden gecombineerd, zijn beschikbaar in de volgende onderwerpen:

## Natuurlijke sets

Een set-uitdrukking vertegenwoordigt over het algemeen zowel een set met records in het gegevensmodel als een selectie die deze subset van gegevens definieert. In dit geval wordt de set een natuurlijke set genoemd.

Set-id's, met of zonder set-modificaties, vertegenwoordigen altijd natuurlijke sets.

Een set-uitdrukking die gebruikmaakt van set-operators vertegenwoordigt ook een subset van de records, maar deze kan over het algemeen nog steeds niet worden omschreven met behulp van selecties met veldwaarden. Een dergelijke uitdrukking wordt een niet-natuurlijke set genoemd.

De set die wordt verstrekt door `{1-$}` kan niet altijd worden gedefinieerd door een selectie. Dit is daarom geen natuurlijke set. Dit kunt u zien door de volgende gegevens te laden, deze aan een tabel toe te voegen en dan selecties te maken met behulp van filtervakken.

```
Load * Inline
[Dim1, Dim2, Number
A, X, 1
A, Y, 1
B, X, 1
B, Y, 1];
```

Als u selecties maakt voor `Dim1` en `Dim2` ziet u de weergave zoals in de onderstaande tabel.

Tabel met natuurlijke en niet-natuurlijke sets

Dim1	Dim2	Sum({\$} Number)	Sum({1-\$} Number)
<b>Totals</b>		<b>1</b>	<b>3</b>
A	X	1	0
A	Y	0	1
B	X	0	1
B	Y	0	1

De set-uitdrukking in de eerste meting gebruikt een natuurlijke set: deze komt overeen met de selectie die is gemaakt {\$}.

De tweede meting is anders. Deze gebruikt {1-\$}. Het is niet mogelijk om een selectie te maken die overeenkomt met deze set, daarom is dit een niet-natuurlijke set.

Dit onderscheid heeft een aantal gevolgen:

- Set-modificaties kunnen alleen worden toegepast op set-id's. Ze kunnen niet worden toegepast op een willekeurige set-uitdrukking. Het is bijvoorbeeld niet mogelijk om de volgende set-uitdrukking te gebruiken:  
`{ (BM01 * BM02) <Field={x,y}> }`  
 Hier geven de normale (ronde) haken aan dat de snijpunten tussen BM01 en BM02 geëvalueerd moeten worden voordat de set-modificatie wordt toegepast. De reden is dat er geen elementset aanwezig is die kan worden gewijzigd.
- U kunt geen niet-natuurlijke set gebruiken in de elementfuncties van  $P()$  en  $\in()$ . Deze functies retourneren een elementset, maar het is niet mogelijk om de elementset af te leiden van een niet-natuurlijke set.
- Aan een meting die een niet-natuurlijke set gebruikt kan niet altijd de juiste dimensiewaarde worden toegewezen als het gegevensmodel veel tabellen bevat. In het volgende diagram worden enkele uitgezonderde verkoopcijfers bijvoorbeeld toegewezen aan de correcte Country, terwijl anderen NULL en Country bevatten.

Diagram met een niet-natuurlijke set

ProductCategory	Country	Values
		Sum({\$} Sales)   Sum({1-\$} Sales)
Baby Clothes		127791.28   0
Children's Clothes		0   81681.54
Men's Clothes		0   140987.45
Men's Footwear		0   232747.44
Sportswear		0   270272.76
Swimwear		0   29548.6
Women's Clothes		0   649348.5
Women's Footwear		0   140654.44
-		0   131935.86
Belgium		0   1005.02
Germany		0   773.3
Portugal		0   1279.74

Of de toewijzing correct wordt uitgevoerd is afhankelijk van het gegevensmodel. In dit geval kan het aantal niet worden toegewezen als het betrekking heeft op een land dat niet in de selectie is opgenomen.

Id	Beschrijving
1	Vertegenwoordigt de volledige set van alle records in de toepassing, ongeacht eventueel uitgevoerde selecties.
\$	Vertegenwoordigt de records van de huidige selectie. De set-uitdrukking <b>{}</b> staat dus gelijk aan het niet aangeven van een set-uitdrukking.
\$1	Vertegenwoordigt de vorige selectie. \$2 vertegenwoordigt de vorige selectie op één na, enzovoort.
\$_1	Vertegenwoordigt de volgende (voorwaartse) selectie. \$_2 vertegenwoordigt de volgende selectie min één, enzovoort.
BM01	U kunt elke gewenste bladwijzer-id of bladwijzer-naam gebruiken.
MyAltState	U kunt verwijzen naar de selecties die zijn uitgevoerd in een alternatieve state met behulp van de naam van de state.

Voorbeeld	Resultaat
sum ({1} Sales)	Retourneert de totale verkopen voor de app, waarbij selecties wel wordt genegeerd maar de dimensie niet.
sum ({\$} Sales)	Retourneert de verkopen voor de huidige selectie, oftewel hetzelfde als sum(Sales).

Voorbeeld	Resultaat
sum({\$1} Sales)	Retourneert de verkopen voor de vorige selectie.
sum({BM01} Sales)	Retourneert de verkopen voor de bladwijzer met de naam <i>BM01</i> .

Voorbeeld	Resultaat
sum({\$<OrderDate = DeliveryDate>} Sales)	Retourneert de verkopen voor de huidige selectie waarbij OrderDate = DeliveryDate.
sum({1<Region = {US}>} Sales)	Retourneert de verkopen voor de regio US, waarbij de huidige selectie buiten beschouwing wordt gelaten.
sum({\$<Region = >} Sales)	Retourneert de verkopen voor de selectie, maar met de selectie in <i>Region</i> verwijderd.
sum({<Region = >} Sales)	Retourneert hetzelfde als in het voorbeeld hierboven. Wanneer de te modificeren set wordt weggelaten, wordt \$ verondersteld.
sum({\$<Year={2000}, Region="{U*}">} Sales)	Retourneert de verkopen voor de huidige selectie, maar met nieuwe selecties in zowel <i>Year</i> als <i>Region</i> .

## Set-Id's

Een set-id vertegenwoordigt een set met records in de gegevens. Dit kunnen alle gegevens zijn of een subset van de gegevens. Het is een set met records die door een selectie wordt gedefinieerd. Dit kan de huidige selectie, alle gegevens (geen selectie), een selectie van een bladwijzer of een selectie van een alternatieve status zijn.

In het voorbeeld `sum( {$<Year = {2009}>} sales )` is de id het dollarteken: \$. Deze vertegenwoordigt de huidige selectie. Maar ook alle mogelijke records. Deze set kan vervolgens worden gewijzigd door het modificatiegedeelte van de set-uitdrukking: de selectie 2009 in *Year* wordt toegevoegd.

In een complexere set-uitdrukking kunnen twee id's samen met een operator worden gebruikt om een samenvoeging, een verschil of een snijpunt van twee recordsets te vormen.

De volgende tabel bevat een aantal algemene id's.

Voorbeelden met algemene id's

Id	Beschrijving
1	Vertegenwoordigt de volledige set van alle records in de toepassing, ongeacht eventueel uitgevoerde selecties.
\$	Vertegenwoordigt de records van de huidige selectie in de standaard-state. De set-uitdrukking {\$} staat dus meestal gelijk aan het niet aangeven van een set-uitdrukking.

Id	Beschrijving
\$1	Vertegenwoordigt de vorige selectie in de standaard-state. \$2 vertegenwoordigt de vorige selectie min één, enzovoort.
\$_1	Vertegenwoordigt de volgende (vooruit) selectie. \$_2 vertegenwoordigt de volgende selectie min één, enzovoort.
BM01	U kunt elke gewenste bladwijzer-id of bladwijzernaam gebruiken.
AltState	U kunt naar een alternatieve state verwijzen door middel van de naam van de state.
AltState::BM01	Een bladwijzer bevat de selecties van alle states en u kunt naar een specifieke bladwijzer verwijzen door de bladwijzernaam in aanmerking te nemen.

De volgende tabel bevat voorbeelden met verschillende id's.

Voorbeelden met verschillende id's

Voorbeeld	Resultaat
sum ({1} sales)	Retourneert de totale verkopen voor de app, waarbij selecties wel wordt genegeerd maar de dimensie niet.
sum ({\$} sales)	Retourneert de verkopen voor de huidige selectie, oftewel hetzelfde als sum(sales).
sum ({\$1} sales)	Retourneert de verkopen voor de vorige selectie.
sum ({BM01} sales)	Retourneert de verkopen voor de bladwijzer met de naam BM01.

## Set-operatoren

Set-operatoren worden gebruikt voor het opnemen, uitsluiten of het doorkruisen van gegevensverzamelingen. Alle operatoren gebruiken sets als operanden en leveren een set op als resultaat.

U kunt set-operatoren in twee verschillende situaties gebruiken:

- Om de bewerking Set uit te voeren voor set-id's die een set met records in de gegevens vertegenwoordigen.
- Om de bewerking Set uit te voeren voor elementsets, voor de veldwaarden of in een set-modificatie.

De volgende tabel bevat de operatoren die kunnen worden gebruikt in set-uitdrukkingen.

Operatoren

Operator	Beschrijving
+	Samenvoegen. Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij een van de twee set-operanden.

Operator	Beschrijving
-	Uitsluiten. Deze binaire bewerking retourneert een set records of elementen die horen bij de eerste maar niet bij de andere van de twee set-operanden. Daarnaast levert het de complementerende set op als het wordt gebruikt als unaire operator.
*	Doorsnede. Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij beide set-operanden.
/	Symmetrisch verschil (XOR). Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij een van de twee set-operanden, maar niet bij beide set-operanden.

De volgende tabel bevat voorbeelden met operatoren.

### Voorbeelden met operatoren

Voorbeeld	Resultaat
<code>Sum ({1-\$} sales)</code>	Retourneert de verkoop voor alles behalve de huidige selectie.
<code>Sum ({\$*BM01} sales)</code>	Retourneert de verkoop voor de gemeenschappelijke elementen voor de selectie en bladwijzer BM01.
<code>Sum ({\$-(\$+BM01)} sales)</code>	Retourneert de verkoop voor alles behalve de selectie en bladwijzer BM01.
<code>Sum ({\$&lt;Year={2009}&gt;+1&lt;Country={'Sweden'}&gt;} sales)</code>	Retourneert de verkopen voor het jaar 2009 die zijn gekoppeld aan de huidige selecties en voegt de volledige verzameling gegevens die zijn gekoppeld aan het land Sweden voor alle jaren toe.
<code>Sum ({\$&lt;Country={'S*'}+{'*land'}&gt;} sales)</code>	Retourneert de verkopen voor landen die beginnen met s of eindigen met land.

## Set-modificaties

Set-uitdrukkingen worden gebruikt om het bereik van een berekening te definiëren. Het belangrijkste onderdeel van de set-uitdrukking is de set-modificator die een selectie definieert. Deze wordt gebruikt om de gebruikersselectie of de selectie in de set-id te wijzigen. Het resultaat definieert een nieuw bereik voor de berekening.

De set-modificatie bestaat uit een of meer veldnamen, elk gevolgd door een selectie die op het veld wordt toegepast. De modificatie is omsloten door punthaken: < >

Bijvoorbeeld:

- `Sum ( {$<Year = {2015}>} sales )`
- `Count ( {1<Country = {Germany}>} distinct OrderID )`
- `Sum ( {$<Year = {2015}, Country = {Germany}>} sales )`

## Elementsets

Een elementset kan als volgt worden gedefinieerd:



- Een lijst met waarden
- Een zoekopdracht
- Een verwijzing naar een ander veld
- Een setfunctie

Als de definitie van een elementset wordt verwijderd, zorgt de set-modificatie dat alle selecties in dit veld worden gewist. Bijvoorbeeld:

```
Sum( {$<Year = >} Sales )
```

### Voorbeelden: Diagramuitdrukkingen voor set-modificaties op basis van elementsets

Voorbeelden - Diagramuitdrukkingen

#### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
MyTable:
Load * Inline [
Country, Year, Sales
Argentina, 2014, 66295.03
Argentina, 2015, 140037.89
Austria, 2014, 54166.09
Austria, 2015, 182739.87
Belgium, 2014, 182766.87
Belgium, 2015, 178042.33
Brazil, 2014, 174492.67
Brazil, 2015, 2104.22
Canada, 2014, 101801.33
Canada, 2015, 40288.25
Denmark, 2014, 45273.25
Denmark, 2015, 106938.41
Finland, 2014, 107565.55
Finland, 2015, 30583.44
France, 2014, 115644.26
France, 2015, 30696.98
Germany, 2014, 8775.18
Germany, 2015, 77185.68
];
```

#### Diagramuitdrukkingen

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Tabel - Set-modificaties op basis van elementsets

Land	Sum (Sales)	Sum({1<Country={Belgium}>}Sales)	Sum({1<Country={**A**}>}Sales)	Sum({1<Country={A*}>}Sales)	Sum({1<Year={\$(=Max(Year))}>}Sales)
Totalen	1645397.3	360809.2	1284588.1	443238.88	788617.07
Argentinië	206332.92	0	206332.92	206332.92	140037.89
Oostenrijk	236905.96	0	236905.96	236905.96	182739.87
België	360809.2	360809.2	0	0	178042.33
Brazilië	176596.89	0	176596.89	0	2104.22
Canada	142089.58	0	142089.58	0	40288.25
Denemarken	152211.66	0	152211.66	0	106938.41
Finland	138148.99	0	138148.99	0	30583.44
France	146341.24	0	146341.24	0	30696.98
Germany	85960.86	0	85960.86	0	77185.68

### Uitleg

- Dimensies:
  - Country
- Metingen:
  - Sum(Sales)  
Sum sales zonder set-uitdrukking.
  - Sum({1<Country={Belgium}>}Sales)  
Selecteer Belgium en vervolgens de som die overeenkomt met sales.
  - Sum({1<Country={\*\*A\*\*}>}Sales)  
Selecteer alle landen die een A bevatten en vervolgens de som die overeenkomt met sales.
  - Sum({1<Country={A\*}>}Sales)  
Selecteer alle landen die beginnen met een A en vervolgens de som die overeenkomt met sales.
  - Sum({1<Year={\$(=Max(Year))}>}Sales)  
Bereken de Max(Year), dit is 2015, en vervolgens de som die overeenkomt met sales.

### Set-modificaties op basis van elementsets

My new sheet

Country	Sum (Sales)	Sum( {1<Country = {Belgium}>} Sales )	Sum( {1<Country = {"*A*"}>} Sales )	Sum( {1<Country = {"A*"}>} Sales )	Sum( {1<Year = {\$(=Max(Year))}>} Sales )
<b>Totals</b>	<b>1645397.3</b>	<b>360809.2</b>	<b>1284588.1</b>	<b>443238.88</b>	<b>788617.07</b>
Argentina	206332.92	0	206332.92	206332.92	140037.89
Austria	236905.96	0	236905.96	236905.96	182739.87
Belgium	360809.2	360809.2	0	0	178042.33
Brazil	176596.89	0	176596.89	0	2104.22
Canada	142089.58	0	142089.58	0	40288.25
Denmark	152211.66	0	152211.66	0	106938.41
Finland	138148.99	0	138148.99	0	30583.44
France	146341.24	0	146341.24	0	30696.98
Germany	85960.86	0	85960.86	0	77185.68

### Weergegeven waarden

Het meest voorkomende voorbeeld van een elementset is een elementset die is gebaseerd op een lijst van veldwaarden die tussen accolades zijn geplaatst. Bijvoorbeeld:

- {<Country = {Canada, Germany, Singapore}>}
- {<Year = {2015, 2016}>}

De binnenste accolades definiëren de elementset. De individuele waarden worden gescheiden door komma's.

### Aanhalingstekens en hoofdlettergevoeligheid

Als de waarden lege waarden of speciale tekens bevatten, moeten de waarden tussen aanhalingstekens worden geplaatst. Bij het gebruik van enkele aanhalingstekens is er sprake van een hoofdlettergevoelige overeenkomst met de waarde van één veld. Bij het gebruik van dubbele aanhalingstekens is er sprake van een niet-hoofdlettergevoelige overeenkomst met één of meerdere veldwaarden. Bijvoorbeeld:

- <Country = {'New Zealand'}>  
Komt alleen overeen met New Zealand.
- <Country = {"New Zealand"}>  
Komt overeen met New Zealand, NEW ZEALAND en new zealand.

Datums moeten tussen aanhalingstekens worden geplaatst. U moet de datumindeling gebruiken van het desbetreffende veld. Bijvoorbeeld:

- <ISO\_Date = {'2021-12-31'}>
- <US\_Date = {'12/31/2021'}>
- <UK\_Date = {'31/12/2021'}>

Dubbele aanhalingstekens kunnen worden vervangen door vierkante haken of accent graves.

### Zoekopdrachten

U kunt elementsets ook maken met behulp van zoekopdrachten. Bijvoorbeeld:

- `<Country = {"C*"}>`
- `<Ingredient = {"*garlic*"}>`
- `<Year = {">2015"}>`
- `<Date = {">12/31/2015"}>`

Jokertekens kunnen worden gebruikt voor zoekopdrachten in tekst: Een asterisk (\*) vertegenwoordigt een willekeurig aantal tekens en een vraagteken (?) vertegenwoordigt één teken. Relationale operatoren kunnen worden gebruikt om numerieke zoekopdrachten te definiëren.

U moet bij zoekopdrachten altijd dubbele aanhalingstekens gebruiken. Zoekopdrachten zijn niet hoofdlettergevoelig.

### Uitbreidingen met dollartekens

Uitbreidingen met dollartekens zijn nodig als u een berekening binnen uw elementenset wilt gebruiken. Als u bijvoorbeeld alleen naar het laatst mogelijke jaar wilt kijken, gebruikt u:

```
<Year = {$(=Max(Year))}>
```

### Geselecteerde waarden in andere velden

Modificaties kunnen worden gebaseerd op de geselecteerde waarden van een ander veld. Bijvoorbeeld:

```
<OrderDate = DeliveryDate>
```

Deze modificatie neemt de geselecteerde waarden uit `DeliveryDate` en past deze toe als selectie op `OrderDate`. Als er veel afzonderlijke waarden zijn - meer dan een paar honderd - vormt deze bewerking een te intensieve belasting van de processor en u dient deze dan te vermijden.

### Functies elementset

De elementset kan ook worden gebaseerd op de set-functies `P()` (mogelijke waarden) en `E()` (uitgesloten waarden).

Als u bijvoorbeeld de landen wilt selecteren waarin het product `cap` is verkocht, kunt u het volgende gebruiken:

```
<Country = P({1<Product={Cap}>} Country)>
```

Als u de landen wilt bekijken waar het product `cap` niet is verkocht, kunt u het volgende gebruiken:

```
<Country = E({1<Product={Cap}>} Country)>
```

### Set-modificaties met zoekopdrachten

U kunt elementsets maken door middel van zoekopdrachten met set-modificaties.

Bijvoorbeeld:

- `<Country = {"C*"}>`
- `<Year = {">2015"}>`
- `<Ingredient = {"*garlic*"}>`

Zoekopdrachten moeten altijd tussen dubbele aanhalingstekens, vierkante haakjes of accent graves worden geplaatst. U kunt een lijst gebruiken met een combinatie van letterlijke tekenreeksen (enkele aanhalingstekens) en zoekopdrachten (dubbele aanhalingstekens). Bijvoorbeeld:

```
<Product = {'Nut', "*Bolt", washer}>
```

### Tekstzoekopdrachten

Jokertekens kunnen worden gebruikt voor tekstzoekopdrachten:

- Een asterisk (\*) vertegenwoordigt een willekeurig aantal tekens.
- Een vraagteken (?) vertegenwoordigt één teken.
- Een accent circonflexe (^) vertegenwoordigt het begin van een woord.

Bijvoorbeeld:

- `<Country = {"C*", "*land"}>`  
Moet overeenkomen met alle landen die beginnen met een c of eindigen op land.
- `<Country = {"*^z*"}>`  
Hiermee worden alle landen aangegeven die overeenkomen met een woord dat begint met z, zoals New Zealand.

### Numerieke zoekopdrachten

U kunt numerieke zoekopdrachten uitvoeren met behulp van de volgende relationele operatoren: >, >=, <, <=

Een numerieke zoekopdracht begint altijd met een van deze operatoren. Bijvoorbeeld:

- `<Year = {">2015"}>`  
Moet overeenkomen met 2016 en opeenvolgende jaren.
- `<Date = {">=1/1/2015<1/1/2016"}>`  
Moet overeenkomen met alle datums in 2015. Houd rekening met de syntaxis voor het beschrijven van een periode tussen twee datums. De datumindeling moet overeenkomen met de datumindeling van het desbetreffende veld.

### Zoekopdrachten voor uitdrukkingen

U kunt zoekopdrachten voor uitdrukkingen gebruiken om geavanceerdere zoekopdrachten uit te voeren. Een aggregatie wordt vervolgens geëvalueerd voor elke veldwaarde in het zoekveld. Alle waarden waarvoor de zoekuitdrukking een waarde oplevert, worden geselecteerd.

Een zoekopdracht voor een uitdrukking begint altijd met een isgelijktteken: =

Bijvoorbeeld:

```
<Customer = {"=Sum(Sales)>1000"}>
```

Hiermee worden alle klanten geretourneerd met verkopen die hoger zijn dan 1000. `sum(Sales)` wordt berekend op basis van de huidige selectie. Dit betekent dat als u een selectie in een ander veld hebt, zoals het veld `Product`, u de klanten krijgt die alleen voldoen aan de verkoopvoorwaarde voor de geselecteerde producten.

Als u wilt dat de voorwaarde onafhankelijk is van de selectie, moet u de set-analyse in de zoekreeks gebruiken. Bijvoorbeeld:

```
<Customer = {"=Sum({1} Sales)>1000"}>
```

De uitdrukkingen na het isgelijktteken worden geïnterpreteerd als een booleaanse waarde. Dit betekent dat als het iets anders evalueert, elk getal dat niet gelijk is aan nul als waar wordt geïnterpreteerd en nul en tekenreeksen als onwaar.

### Quotes

Gebruik aanhalingstekens als de zoekreeksen lege of speciale karakters bevatten. Bij het gebruik van enkele aanhalingstekens is er sprake van een hoofdlettergevoelige overeenkomst met de waarde van één veld. Bij het gebruik van dubbele aanhalingstekens is er sprake van een niet-hoofdlettergevoelige zoekopdracht die mogelijk overeenkomt met meerdere veldwaarden.

Bijvoorbeeld:

- `<Country = {'New Zealand'}>`  
Moet alleen overeenkomen met `New Zealand`.
- `<Country = {"New Zealand"}>`  
Moet overeenkomen met `New Zealand`, `NEW ZEALAND` en `new zealand`

Dubbele aanhalingstekens kunnen worden vervangen door vierkante haken of accent graves.



*In vorige versies van Qlik Sense was er geen onderscheid tussen enkele en dubbele aanhalingstekens en alle aanhalingstekens werden als zoekopdrachten behandeld. Om achterwaartse compatibiliteit te behouden, blijven apps die gemaakt zijn met oudere versies van Qlik Sense werken zoals ze in eerdere versies deden. Apps gemaakt met Qlik Sense van november 2017 of later maken onderscheid tussen de twee soorten aanhalingstekens.*

Voorbeelden: Diagramuitdrukkingen voor set-modificaties met zoekopdrachten

Voorbeelden - Diagramuitdrukkingen

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
MyTable:
```

```
Load
```

```
Year(Date) as Year,
```

```
Date#(Date, 'YYYY-MM-DD') as ISO_Date,
```

```
Date(Date#(Date,'YYYY-MM-DD'),'M/D/YYYY') as US_Date,
Country, Product, Amount
Inline
[Date, Country, Product, Amount
2018-02-20, Canada, washer, 6
2018-07-08, Germany, Anchor bolt, 10
2018-07-14, Germany, Anchor bolt, 3
2018-08-31, France, Nut, 2
2018-09-02, Czech Republic, Bolt, 1
2019-02-11, Czech Republic, Bolt, 3
2019-07-31, Czech Republic, washer, 6
2020-03-13, France, Anchor bolt, 1
2020-07-12, Canada, Anchor bolt, 8
2020-09-16, France, washer, 1];
```

### Voorbeeld 1: Diagramuitdrukkingen voor set-modificaties met tekstzoekopdrachten

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Tabel - Set-modificaties met tekstzoekopdrachten

Land	Sum (Amount)	Sum({<Country= "C*">} Amount)	Sum({<Country= "*^R*">} Amount)	Sum({<Product= "*bolt*">} Amount)
<b>Totalen</b>	<b>41</b>	<b>24</b>	<b>10</b>	<b>26</b>
Canada	14	14	0	8
Czech Republic	10	10	10	4
France	4	0	0	1
Germany	13	0	0	13

### Uitleg

- Dimensies:
  - Country
- Metingen:
  - Sum(Amount)  
Sum Amount zonder set-uitdrukking.
  - Sum({<Country={"C\*">} Amount)  
Sum Amount voor alle landen die beginnen met c, zoals Canada en Czech Republic.
  - Sum({<Country={"\*^R\*">} Amount)  
Sum Amount voor alle landen die een woord bevatten dat begint met R, zoals Czech Republic.
  - Sum({<Product={"\*bolt\*">} Amount)  
Sum Amount voor alle producten die de tekenreeks bolt bevatten, zoals Bolt en Anchor bolt.

## Set-modificaties met tekstzoekopdrachten

My new sheet

Country	Sum (Amount)	Sum({<Country={"C*"}>} Amount)	Sum({<Country={"**R*"}>} Amount)	Sum({<Product={"*bolt*"}>} Amount)
<b>Totals</b>	<b>41</b>	<b>24</b>	<b>10</b>	<b>26</b>
Canada	14	14	0	8
Czech Republic	10	10	10	4
France	4	0	0	1
Germany	13	0	0	13

## Voorbeeld 2: Diagramuitdrukkingen met numerieke zoekopdrachten

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Tabel - Set-modificaties met numerieke zoekopdrachten

Land	Sum (Amount)	Sum({<Year={"}>2019"}>} Amount)	Sum({<ISO_Date={">=2019-07-01"}>} Amount)	Sum({<US_Date={">=4/1/2018<=12/31/2018"}>} Amount)
<b>Totalen</b>	<b>41</b>	<b>10</b>	<b>16</b>	<b>16</b>
Canada	14	8	8	0
Czech Republic	10	0	6	1
France	4	2	2	2
Germany	13	0	0	13

## Uitleg

- Dimensies:
  - Country
- Metingen:
  - Sum(Amount)  
Sum Amount zonder set-uitdrukking.
  - Sum({<Year={">2019"}>}Amount)  
Sum Amount voor alle jaren na 2019.
  - Sum({<ISO\_Date={">=2019-07-01"}>}Amount)  
Sum Amount voor alle datums op of na 2019-07-01. De indeling van de datum in de zoekopdracht moet overeenkomen met de indeling van het veld.
  - Sum({<US\_Date={">=4/1/2018<=12/31/2018"}>}Amount)



Sum Amount voor alle datums van 4/1/2018 tot 12/31/2018, inclusief de begin- en einddatum. De indeling van de datums in de zoekopdracht moet overeenkomen met de indeling van het veld.

*Set-modificaties met numerieke zoekopdrachten*

My new sheet

Country	Q	Sum (Amount)	Sum({<Year={">2019"}>} Amount)	Sum({<ISO_Date={">=2019-07-01"}>} Amount)	Sum({<US_Date={">=4/1/2018<=12/31/2018"}>} Amount)
Totals		41	10	16	16
Canada		14	8	8	0
Czech Republic		10	0	6	1
France		4	2	2	2
Germany		13	0	0	13

#### Voorbeeld 3: Diagramuitdrukkingen met zoekopdrachten voor uitdrukkingen

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Table - Set modifiers with expression searches

Country	Sum (Amount)	Sum({<Country={"=Sum (Amount)>10"}>} Amount)	Sum({<Country={"=Count(distinct Product)=1"}>} Amount)	Sum({<Product={"=Count (Amount)>3"}>} Amount)
Totals	41	27	13	22
Canada	14	14	0	8
Czech Republic	10	0	0	0
France	4	0	0	1
Germany	13	13	13	13

#### Uitleg

- Dimensies:
  - Country
- Metingen:
  - Sum(Amount)  
Sum Amount zonder set-uitdrukking.
  - Sum({<Country={"=Sum(Amount)>10"}>} Amount)  
Sum Amount voor alle landen die een aggregeerde som hebben van Amount die groter is dan 10.
  - Sum({<Country={"=Count(distinct Product)=1"}>} Amount)

### 3 Diagramuitdrukkingen

Sum Amount voor alle landen die zijn gekoppeld met precies één afzonderlijk product.

- Sum({<Product={"}=Count(Amount)>3"}>}Amount)

Sum Amount voor alle landen die meer dan drie transacties in de gegevens hebben.

#### Set-modificaties met zoekopdrachten voor uitdrukkingen

My new sheet

Country	Q	Sum (Amount)	Sum({<Country={"}=Sum(Amount)>10"}>} Amount)	Sum({<Country={"}=Count(distinct Product)=1"}>} Amount)	Sum({<Product={"}=Count(Amount)>3"}>} Amount)
Totals		41	27	13	22
Canada		14	14	0	8
Czech Republic		10	0	0	0
France		4	0	0	1
Germany		13	13	13	13

Voorbeelden	Resultaten
sum( {\$-1<Product = {"*Internal*", "*Domestic*"}>} Sales )	Retourneert de verkoop voor de huidige selectie, exclusief transacties met betrekking tot producten met de reeks 'Internal' of 'Domestic' in de productnaam.
sum( {\$<Customer = {"=Sum ({1<Year = {2007}>} Sales ) > 1000000"}>} Sales )	Retourneert de verkoop voor de huidige selectie, maar met een nieuwe selectie in het veld 'Customer': alleen klanten die in 2007 een totale verkoop van meer dan 1000000 hadden.

#### Set-modificaties met uitbreiding via dollartekens


Uitdrukkingen met dollarteken zijn constructies die worden berekend voordat de uitdrukking wordt geparseerd en beoordeeld. Het resultaat wordt vervolgens in de uitdrukking ingevoegd in plaats van s(...). De berekening van de uitdrukking wordt vervolgens uitgevoerd met behulp van het resultaat van de uitdrukking met dollarteken.

De uitdrukkingeditor toont een voorbeeld van de uitdrukking met dollarteken zodat u kunt controleren waarin de uitdrukking resulteert.

Voorbeeld van de uitdrukking met dollartekens in de uitdrukkingeditor

Edit expression

```
1 Sum({<US_Date={ ">=$ (=AddYears (Max (US_Date) , -1) ) " }>}Amount)
```

 OK  
Sum({<US\_Date={ ">=9/16/2019" }>}Amount)

Gebruik uitbreidingen met dollartekens als u een berekening binnen uw elementset wilt gebruiken.

Als u bijvoorbeeld alleen wilt kijken naar het laatst mogelijke jaar, kunt de volgende constructie gebruiken:

```
<Year = {$(=Max(Year))}>
```

Max(Year) wordt eerst berekend en het resultaat wordt ingevoegd in de uitdrukking in plaats van \$(...).

Het resultaat na de uitdrukking met dollartekens is de volgende uitdrukking:

```
<Year = {2021}>
```

De uitdrukking in de uitdrukking met dollartekens wordt berekend op basis van de huidige selectie. Als u een selectie in een ander veld hebt, beïnvloedt dit het resultaat van de uitdrukking.

Als u wilt dat de berekening onafhankelijk is van de selectie, moet u de set-analyse in de uitdrukking met dollartekens gebruiken. Bijvoorbeeld:

```
<Year = {$(=Max({1} Year))}>
```

### Tekenreeksen

Als u wilt dat de uitdrukking met dollartekens resulteert in een tekenreeks, gelden de normale regels voor aanhalingstekens. Bijvoorbeeld:

```
<Country = {'$(=FirstSortedValue(Country,Date))'}>
```

Het resultaat na de uitdrukking met dollartekens is de volgende uitdrukking:

```
<Country = {'New Zealand'}>
```

Er ontstaat een fout in de syntaxis als u geen aanhalingstekens gebruikt.

### Getallen

Als u wilt dat de uitdrukking met dollartekens resulteert in een getal, moet u zorgen dat de uitdrukking dezelfde opmaak krijgt als het veld. U moet de uitdrukking daarom soms in een opmaakfunctie verpakken.

Bijvoorbeeld:

```
<Amount = {$(=Num(Max(Amount), '###0.00'))}>
```

Het resultaat na de uitdrukking met dollarteken is de volgende uitdrukking:

```
<Amount = {12362.00}>
```

Gebruik een hash om af te dwingen dat een uitdrukking altijd een decimale komma gebruikt en geen scheidingstekens voor duizendtallen. Bijvoorbeeld:

```
<Amount = {$(#=Max(Amount))}>
```

### Datums

Als u wilt dat de uitdrukking met dollarteken resulteert in een datum, moet u zorgen dat de uitdrukking de juiste opmaak heeft. U moet de uitdrukking daarom soms in een opmaakfunctie verpakken.

Bijvoorbeeld:

```
<Date = {'$(=Date(Max(Date)))'}>
```

Het resultaat na de uitdrukking met dollarteken is de volgende uitdrukking:

```
<Date = {'12/31/2015'}>
```

Net als bij tekenreeksen moet u de juiste aanhalingstekens gebruiken.

Een veelvoorkomende gebruikscase is om uw berekening te beperken tot de laatste maand (of jaar). In dat geval kunt u een numerieke zoekopdracht gebruiken in combinatie met de functie `AddMonths()`.

Bijvoorbeeld:

```
<Date = {">=$(=AddMonths(Today(), -1))"}>
```

Het resultaat na de uitdrukking met dollarteken is de volgende uitdrukking:

```
<Date = {">=9/31/2021"}>
```

Hiermee worden alle evenementen geselecteerd die de afgelopen maand hebben plaatsgevonden.

Voorbeeld: Diagramuitdrukkingen voor set-modificaties met uitdrukkingen met dollarteken

Voorbeeld - diagramuitdrukkingen

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
Let vToday = Today();
MyTable:
Load
Year(Date) as Year,
Date#(Date, 'YYYY-MM-DD') as ISO_Date,
Date(Date#(Date, 'YYYY-MM-DD'), 'M/D/YYYY') as US_Date,
Country, Product, Amount
Inline
[Date, Country, Product, Amount
2018-02-20, Canada, washer, 6
```

2018-07-08, Germany, Anchor bolt, 10  
 2018-07-14, Germany, Anchor bolt, 3  
 2018-08-31, France, Nut, 2  
 2018-09-02, Czech Republic, Bolt, 1  
 2019-02-11, Czech Republic, Bolt, 3  
 2019-07-31, Czech Republic, Washer, 6  
 2020-03-13, France, Anchor bolt, 1  
 2020-07-12, Canada, Anchor bolt, 8  
 2021-10-15, France, washer, 1];

### Diagramuitdrukkingen met uitdrukkingen met dollartekens

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Tabel - Set-modificaties met uitbreiding via dollartekens

Land	Sum (Amount)	Sum({<US_ Date= {'\$(vToday)'}>} Amount)	Sum({<ISO_Date= {"\$(=Date(Min(ISO_ Date),'YYYY-MM- DD'))"}>} Amount)	Sum({<US_Date= {">=\$(=AddYears(Max (US_Date),-1))"}>} Amount)
<b>Totalen</b>	<b>41</b>	<b>1</b>	<b>6</b>	<b>1</b>
Canada	14	0	6	0
Czech Republic	10	0	0	0
France	4	1	0	1
Germany	13	0	0	0

### Uitleg

- Dimensies:
  - Country
- Metingen:
  - Sum(Amount)  
Som Amount zonder set-uitdrukking.
  - Sum({<US\_Date={'\$(vToday)'}>}Amount)  
Sum Amount voor alle records waarbij US\_Date gelijk is aan de variabele vToday.
  - Sum({<ISO\_Date={"\$(=Date(Min(ISO\_Date), 'YYYY-MM-DD'))"}>}Amount)  
Sum Amount voor alle records waarbij ISO\_Date gelijk is aan de eerst (laagst) mogelijke ISO\_Date. De functie date() is nodig om te garanderen dat de indeling van de datum overeenkomt met de indeling van het veld.
  - Sum({<US\_Date={">=\$(=AddYears(Max(US\_Date), -1))"}>}Amount)  
Sum Amount voor alle records met een US\_Date na of op de datum, één jaar door de laatste (hoogst) mogelijke US\_Date. De functie AddYears() retourneert een datum in de indeling die is opgegeven door de variabele DateFormat en deze moet overeenkomen met de indeling van het veld US\_Date.

## Set-modificaties met uitbreiding via dollartekens

My new sheet

Country	Sum (Amount)	Sum( {<US_Date=[S(vToday)]>} Amount )	Sum( {<ISO_Date=[S(=Date(Min(ISO_Date),YYYY-MM-DD)]>} Amount )	Sum( {<US_Date=[S(=AddYears(Max(US_Date),-1)]>} Amount )
Totals	41	1	6	1
Canada	14	0	6	0
Czech Republic	10	0	0	0
France	4	1	0	1
Germany	13	0	0	0

Voorbeelden	Resultaten
sum( {<Year = {\$(#vLastYear)}>} Sales )	Retourneert de verkoop van het vorige jaar in relatie tot de huidige selectie. Hier wordt de variabele vLastYear die het relevante jaar bevat, gebruikt in een uitbreiding via een dollarteken.
sum( {<Year = {\$(#=Only(Year)-1)}>} Sales )	Retourneert de verkoop van het vorige jaar in relatie tot de huidige selectie. Een uitbreiding met een dollarteken wordt gebruikt om het vorige jaar te berekenen.

## Set-modificaties met set-operatoren

Set-operatoren worden gebruikt voor het opnemen, uitsluiten of het doorkruisen van verschillende elementsets. Ze combineren de verschillende methoden voor het definiëren van elementsets.

De operatoren zijn gelijk aan de operatoren die voor set-id's worden gebruikt.

## Operatoren

Operator	Beschrijving
+	Samenvoegen. Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij een van de twee set-operanden.
-	Uitsluiten. Deze binaire bewerking retourneert een set records of elementen die horen bij de eerste maar niet bij de andere van de twee set-operanden. Daarnaast levert het de complementerende set op als het wordt gebruikt als unaire operator.
*	Doorsnede. Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij beide set-operanden.
/	Symmetrisch verschil (XOR). Deze binaire bewerking retourneert een set die bestaat uit de records of elementen die horen bij een van de twee set-operanden, maar niet bij beide set-operanden.

De volgende twee modificaties definiëren bijvoorbeeld dezelfde set met veldwaarden:

- <Year = {1997, "20\*"}>
- <Year = {1997} + {"20\*"}>

Beide uitdrukkingen selecteren 1997 en de jaren die beginnen met 20. In andere woorden: dit is de samenvoeging van de twee voorwaarden.

Bij set-operatoren kunnen complexere definities worden gebruikt. Bijvoorbeeld:

```
<Year = {1997, "20*"} - {2000}>
```

Met deze uitdrukking worden dezelfde jaren als hierboven geselecteerd, maar wordt het jaar 2000 uitgesloten.

### Voorbeelden: Diagramuitdrukkingen voor set-modificaties met set-operatoren

#### Voorbeelden - Diagramuitdrukkingen

#### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
MyTable:
Load
Year(Date) as Year,
Date#(Date,'YYYY-MM-DD') as ISO_Date,
Date(Date#(Date,'YYYY-MM-DD'),'M/D/YYYY') as US_Date,
Country, Product, Amount
Inline
[Date, Country, Product, Amount
2018-02-20, Canada, washer, 6
2018-07-08, Germany, Anchor bolt, 10
2018-07-14, Germany, Anchor bolt, 3
2018-08-31, France, Nut, 2
2018-09-02, Czech Republic, Bolt, 1
2019-02-11, Czech Republic, Bolt, 3
2019-07-31, Czech Republic, washer, 6
2020-03-13, France, Anchor bolt, 1
2020-07-12, Canada, Anchor bolt, 8
2020-09-16, France, washer, 1];
```

#### Diagramuitdrukkingen

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Tabel - Set-modificaties met set-operatoren

Land	Sum (Amount)	Sum({<Year= ">2018"- {2020}>} Amount)	Sum ({<Country=- {Germany}>} Amount)	Sum({<Country= {Germany}+P({<Product= {Nut}>} Country)>} Amount)
<b>Totalen</b>	<b>41</b>	<b>9</b>	<b>28</b>	<b>17</b>
Canada	14	0	14	0
Czech Republic	10	9	10	0
France	4	0	4	4
Germany	13	0	0	13

#### Uitleg

- Dimensies:
  - Country
- Metingen:
  - Sum(Amount)  
Sum Amount zonder set-uitdrukking.
  - Sum({<Year="{>2018"}-{2020}>}Amount)  
Sum Amount voor alle jaren na 2018, behalve 2020.
  - Sum({<Country=- {Germany}>}Amount)  
Sum Amount voor alle landen, behalve Germany. Let op de unaire operator voor uitsluitingen.
  - Sum({<Country={Germany}+P({<Product={Nut}>}Country)>}Amount)  
Sum Amount voor Germany en alle landen die zijn gekoppeld aan het product Nut.

#### Set-modificaties met set-operatoren

My new sheet

Country	Sum (Amount)	Sum({<Year="{>2018"}-{2020}>} Amount)	Sum({<Country=- {Germany}>} Amount)	Sum({<Country={Germany}+P({<Product= {Nut}>} Country)>} Amount)
<b>Totals</b>	<b>41</b>	<b>9</b>	<b>28</b>	<b>17</b>
Canada	14	0	14	0
Czech Republic	10	9	10	0
France	4	0	4	4
Germany	13	0	0	13



Voorbeelden	Resultaten
<code>sum( {\$&lt;Product = Product + {OurProduct1} - {OurProduct2} &gt;} Sales )</code>	Retourneert de verkoop voor de huidige selectie, maar met het product "OurProduct1" toegevoegd aan de lijst van geselecteerde producten en "OurProduct2" hieruit verwijderd.
<code>sum( {\$&lt;Year = Year + {"20*",1997} - {2000} &gt;} Sales )</code>	Retourneert de verkoop voor de huidige selectie, maar met extra selecties in het veld "Year": 1997 en alle jaren die beginnen met "20", maar niet 2000.  Opmerking: als 2000 in de huidige selectie is opgenomen, blijft dit jaar nog opgenomen na de modificatie.
<code>sum( {\$&lt;Year = (Year + {"20*",1997}) - {2000} &gt;} Sales )</code>	Retourneert bijna hetzelfde als hierboven, maar hier wordt het jaar 2000 uitgesloten, ook als het in de huidige selectie is opgenomen. Het voorbeeld laat zien dat het gebruik van haakjes om een volgorde aan te geven van groot belang kan zijn.
<code>sum( {\$&lt;Year = {"*"} - {2000}, Product = {"*bearing*"} &gt;} Sales )</code>	Retourneert de verkoop voor de huidige selectie, maar met een nieuwe selectie in "Year": alle jaren behalve 2000 en alleen voor producten die de tekenreeks 'lager' bevatten.

### Set-modificaties met impliciete set-operatoren

De standaard manier voor het schrijven van selecties in een set-modificatie is het gebruik van een isgelijktteken. Bijvoorbeeld:

```
Year = {">2015"}
```

De uitdrukking rechts van het isgelijktteken in de set-modificatie wordt een elementset genoemd. Deze definieert een set met afzonderlijke veldwaarden, oftewel een selectie.

Met deze notatie wordt een nieuwe selectie gedefinieerd en wordt de huidige selectie in het veld genegeerd. Dus als de set-id een selectie in dit veld bevat, wordt de oude selectie vervangen door de selectie in de elementset.

Als u uw selectie wilt baseren op de huidige selectie in het veld, moet u een andere uitdrukking gebruiken.

Als u bijvoorbeeld de oude selectie in acht wilt nemen en de vereiste wilt toevoegen dat het jaar na 2015 moet worden meegenomen, kunt u het volgende schrijven:

```
Year = Year * {">2015"}
```

De asterisk is een set-operator die een snijpunt definieert. U krijgt hierdoor het snijpunt tussen de huidige selectie in `Year` en de aanvullende vereiste dat het jaar na 2015 moet worden meegenomen. U kunt dit ook op de volgende manier schrijven:

```
Year *= {">2015"}
```

De toewijzingsoperator (`*=`) definieert impliciet een snijpunt.

Zo kunnen impliciete samenvoegingen, uitsluitingen en symmetrische verschillen met behulp van de volgende tekens worden gedefinieerd: +=, -=, /=

Voorbeelden: Diagramuitdrukkingen voor set-modificaties met impliciete set-operatoren

Voorbeelden - Diagramuitdrukkingen

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
MyTable:
Load
Year(Date) as Year,
Date#(Date,'YYYY-MM-DD') as ISO_Date,
Date(Date#(Date,'YYYY-MM-DD'),'M/D/YYYY') as US_Date,
Country, Product, Amount
Inline
[Date, Country, Product, Amount
2018-02-20, Canada, Washer, 6
2018-07-08, Germany, Anchor bolt, 10
2018-07-14, Germany, Anchor bolt, 3
2018-08-31, France, Nut, 2
2018-09-02, Czech Republic, Bolt, 1
2019-02-11, Czech Republic, Bolt, 3
2019-07-31, Czech Republic, Washer, 6
2020-03-13, France, Anchor bolt, 1
2020-07-12, Canada, Anchor bolt, 8
2020-09-16, France, Washer, 1];
```

### Diagramuitdrukkingen met impliciete set-operatoren

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Selecteer Canada en Czech Republic uit een lijst met landen.

Tabel - Diagramuitdrukkingen met impliciete set-operatoren

Land	Sum (Amount)	Sum({<Country*= {Canada}>} Amount)	Sum({<Country-= {Canada}>} Amount)	Sum({<Country+= {France}>} Amount)
<b>Totalen</b>	<b>24</b>	<b>14</b>	<b>10</b>	<b>28</b>
Canada	14	14	0	14
Czech Republic	10	0	10	10
France	0	0	0	4

#### Uitleg

- Dimensies:
  - Country
- Metingen:
  - Sum(Amount)
 

Sum Amount voor de huidige selectie. Houd er rekening mee dat alleen Canada en Czech Republic niet-nulwaarden bevatten.
  - Sum({<Country\*={Canada}>}Amount)
 

Sum Amount voor de huidige selectie, samen met de vereiste dat country Canada moet zijn. Als Canada geen onderdeel uitmaakt van de gebruikersselectie, retourneert de set-uitdrukking een lege set en bevatten alle rijen in de kolom de waarde 0.
  - Sum({<Country-={Canada}>}Amount)
 

Sum Amount voor de huidige selectie, maar eerst wordt Canada uitgesloten van de selectie country. Als Canada geen onderdeel uitmaakt van de gebruikersselectie, worden in de set-uitdrukking geen getallen gewijzigd.
  - Sum({<Country+={France}>}Amount)
 

Sum Amount voor de huidige selectie, maar eerst wordt France toegevoegd aan de selectie country. Als France al onderdeel uitmaakt van de gebruikersselectie, worden in de set-uitdrukking geen getallen gewijzigd.

Set-modificaties met impliciete set-operatoren.

Country	Sum (Amount)	Sum({<Country*={Canada}>} Amount)	Sum({<Country-={Canada}>} Amount)	Sum({<Country+={France}>} Amount)
<b>Totals</b>	<b>24</b>	<b>14</b>	<b>10</b>	<b>28</b>
Canada	14	14	0	14
Czech Republic	10	0	10	10
France	0	0	0	4

Voorbeelden	Resultaten
<pre>sum( {&lt;Product += {OurProduct1, OurProduct2} &gt;} Sales )</pre>	Retourneert de verkoop voor de huidige selectie, maar gebruikt een impliciete binding om de producten 'OurProduct1' en 'OurProduct2' toe te voegen aan de lijst met geselecteerde producten.

Voorbeelden	Resultaten
<pre>sum( {\$&lt;Year += {"20*", 1997} - {2000} &gt;} Sales )</pre>	<p>Retourneert de verkoop voor de huidige selectie, maar gebruikt een impliciete binding om een aantal jaren aan de selectie toe te voegen: 1997 en alle jaren die beginnen met "20", maar niet 2000.</p> <p>Opmerking: als 2000 in de huidige selectie is opgenomen, blijft dit jaar nog opgenomen na de modificatie. Hetzelfde als <code>&lt;Year=Year + ({"20*", 1997} - {2000})&gt;</code>.</p>
<pre>sum( {\$&lt;Product *= {OurProduct1} &gt;} Sales )</pre>	<p>Retourneert de verkoop voor de huidige selectie, maar alleen voor de gezamenlijk geselecteerde huidige producten en het product 'OurProduct1'.</p>

### Set-modificaties die gebruikmaken van set-functies

Soms moet u een set met veldwaarden definiëren met behulp van een geneste set-definitie. U wilt mogelijk alle klanten selecteren die een specifiek product hebben geselecteerd, zonder het product te selecteren.

In dergelijke gevallen gebruikt u de elementsetfuncties `P()` en `E()`. Deze retourneren de elementsets van mogelijke waarden en uitgesloten waarden van een veld. Binnen de haakjes kunt u het desbetreffende veld opgeven en een set-uitdrukking die het bereik definieert. Bijvoorbeeld:

```
P({1<Year = {2021}>} Customer)
```

Deze retourneert een set van klanten met transacties in 2021. U kunt deze vervolgens gebruiken in een set-modificatie. Bijvoorbeeld:

```
Sum({<Customer = P({1<Year = {2021}>} Customer)>} Amount)
```

Deze set-uitdrukking selecteert vervolgens deze klanten, maar beperkt de selectie niet tot 2021.

Deze functies kunnen niet worden gebruikt in andere uitdrukkingen.

Bovendien kunnen alleen natuurlijke sets worden gebruikt in de elementsetfuncties. Met andere woorden een set records die met een eenvoudige selectiebewerking kan worden gedefinieerd.

Bijvoorbeeld, de set van `{1-$}` kan niet altijd door middel van een selectie worden gedefinieerd en is dan ook geen natuurlijke set. Als deze functies op niet-natuurlijke sets worden gebruikt, worden er onverwachte resultaten geretourneerd.

Voorbeelden: Diagramuitdrukkingen voor set-modificaties die gebruikmaken van setfuncties

Voorbeelden - Diagramuitdrukkingen

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```

MyTable:
Load
Year(Date) as Year,
Date#(Date,'YYYY-MM-DD') as ISO_Date,
Date(Date#(Date,'YYYY-MM-DD'),'M/D/YYYY') as US_Date,
Country, Product, Amount
Inline
[Date, Country, Product, Amount
2018-02-20, Canada, washer, 6
2018-07-08, Germany, Anchor bolt, 10
2018-07-14, Germany, Anchor bolt, 3
2018-08-31, France, Nut, 2
2018-09-02, Czech Republic, Bolt, 1
2019-02-11, Czech Republic, Bolt, 3
2019-07-31, Czech Republic, Washer, 6
2020-03-13, France, Anchor bolt, 1
2020-07-12, Canada, Anchor bolt, 8
2020-09-16, France, washer, 1];

```

### Diagramuitdrukkingen

Maak een tabel in een Qlik Sense-werkblad met de volgende diagramuitdrukkingen.

Tabel - Set-modificaties die gebruikmaken van setfuncties

Land	Sum (Amount)	Sum({<Country=P {<Year={2019}>} Country>} Amount)	Sum({<Product=P {<Year={2019}>} Product>} Amount)	Sum({<Country=E {<Product= {Washer}>} Country>} Amount)
<b>Totalen</b>	<b>41</b>	<b>10</b>	<b>17</b>	<b>13</b>
Canada	14	0	6	0
Czech Republic	10	10	10	0
France	4	0	1	0
Germany	13	0	0	13

### Uitleg

- Dimensies:
  - Country
- Metingen:
  - Sum(Amount)  
Sum Amount zonder set-uitdrukking.
  - Sum({<Country=P({<Year={2019}>} Country)>} Amount)  
Sum Amount voor de landen die zijn gekoppeld aan het jaar 2019. Hiermee wordt de berekening voor 2019 echter niet beperkt.

### 3 Diagramuitdrukkingen

- `Sum({<Product=P({<Year={2019}>} Product)>} Amount)`  
Sum Amount voor de producten die zijn gekoppeld aan het jaar 2019. Hiermee wordt de berekening voor 2019 echter niet beperkt.
- `Sum({<Country=E({<Product={Washer}>} Country)>} Amount)`  
Sum Amount voor de landen die niet zijn gekoppeld aan het product washer.

Set-modificaties die gebruikmaken van set-functies

My new sheet

Country	Sum (Amount)	Sum({<Country=P({<Year={2019}>} Country)>} Amount)	Sum({<Product=P({<Year={2019}>} Product)>} Amount)	Sum({<Country=E({<Product={Washer}>} Country)>} Amount)
<b>Totals</b>	<b>41</b>	<b>10</b>	<b>17</b>	<b>13</b>
Canada	14	0	6	0
Czech Republic	10	10	10	0
France	4	0	1	0
Germany	13	0	0	13

Voorbeelden	Resultaten
<code>sum({&lt;Customer = P({1&lt;Product= {'Shoe'}&gt;} Customer)&gt;} Sales )</code>	Retourneert de verkoop voor de huidige selectie, maar alleen de klanten die ooit het product 'Shoe' gekocht hebben. De elementfunctie P ( ) retourneert hier een lijst met mogelijke klanten; de klanten die worden geïmpliceerd door de selectie 'Shoe' in het veld Product.
<code>sum({&lt;Customer = P({1&lt;Product= {'Shoe'}&gt;})&gt;} Sales )</code>	Hetzelfde als hierboven. Als het veld in de elementfunctie wordt weggelaten, retourneert de functie de mogelijke waarden van het veld dat is opgegeven in de outer-toewijzing.
<code>sum({&lt;Customer = P({1&lt;Product= {'Shoe'}&gt;} Supplier)&gt;} Sales )</code>	Retourneert de verkoop voor de huidige selectie, maar alleen de klanten die ooit het product 'Shoe' geleverd hebben. De klant is dus ook een leverancier. De elementfunctie P ( ) retourneert hier een lijst met mogelijke leveranciers; de leveranciers die worden geïmpliceerd door de selectie 'Shoe' in het veld Product. De lijst met leveranciers wordt vervolgens gebruikt als selectie in het veld Customer.
<code>sum({&lt;Customer = E({1&lt;Product= {'Shoe'}&gt;})&gt;} Sales )</code>	Retourneert de verkoop voor de huidige selectie, maar alleen de klanten die nooit het product 'Shoe' hebben gekocht. De elementfunctie E ( ) retourneert hier een lijst met uitgesloten klanten; de klanten die worden uitgesloten door de selectie 'Shoe' in het veld Product.

### Binnenste en buitenste set-uitdrukkingen

Set-uitdrukkingen kunnen binnen en buiten aggregatiefuncties worden gebruikt en worden tussen accolades geplaatst.

Als u een set-uitdrukking binnen een aggregatiefunctie gebruikt, kan deze er zo uit komen te zien:

#### Voorbeeld: Binnenste set-uitdrukking

```
Sum( {<Year={2021}>} Sales )
```

Gebruik een set-uitdrukking buiten de aggregatiefunctie als u uitdrukkingen met meerdere aggregaties hebt en niet dezelfde set-uitdrukking in elke aggregatiefunctie wilt schrijven.

Als u een buitenste set-uitdrukking gebruikt, moet deze aan het begin van het bereik worden geplaatst.

#### Voorbeeld: Buitenste set-uitdrukking

```
{<Year={2021}>} Sum(Sales) / Count(distinct Customer)
```

Als u een set-uitdrukking buiten de aggregatiefunctie gebruikt, kunt u deze ook toepassen op bestaande mastermetingen.

#### Voorbeeld: Buitenste set-uitdrukking toegepast op mastermeting

```
{<Year={2021}>} [Master Measure]
```

Een set-uitdrukking die buiten aggregatiefuncties wordt gebruikt, heeft invloed op de volledige uitdrukking, tenzij deze tussen haakjes is geplaatst. In dat geval definiëren de haakjes het bereik. In het onderstaande voorbeeld van een statische bereik wordt de set-uitdrukking alleen toegepast op de aggregatie binnen de haakjes.

#### Voorbeeld: Statisch bereik toepassen

```
( {<Year={2021}>} Sum(Amount) / Count(distinct Customer) ) - Avg(CustomerSales)
```

### Regels

#### Statisch bereik

De set-uitdrukking is van invloed op de volledige uitdrukking, tenzij deze tussen haakjes is geplaatst. Indien dit het geval is, definiëren de haakjes het statische bereik.

#### Positie

De set-uitdrukking moet aan het begin van het statische bereik worden geplaatst.

#### Context

De context is de selectie die relevant is voor de uitdrukking. Normaal gesproken is de context altijd de standaardstate van de huidige selectie. Maar als een object op een alternatieve state is ingesteld, is de context de alternatieve state van de huidige selectie.

U kunt ook een context definiëren in de vorm van een buitenste set-uitdrukking.

### Overname

Binnenste set-uitdrukkingen hebben voorrang op buitenste set-uitdrukkingen. Als de binnenste set-uitdrukking een set-id bevat, vervangt deze de context. Ander worden de context en set-uitdrukking samengevoegd.

- $\{\$ \langle \text{SetExpression} \rangle\}$  - overschrijft de buitenste set-uitdrukking
- $\{\langle \text{SetExpression} \rangle\}$  - wordt samengevoegd met de buitenste set-uitdrukking

### Toewijzing van elementset

De toewijzing van de elementset bepaalt hoe de twee selecties worden samengevoegd. Als een normaal gelijkteken wordt gebruikt, heeft de selectie in de binnenste set-uitdrukking voorrang. Anders wordt de impliciete set-operator gebruikt.

- $\{\langle \text{Field} = \{ \text{value} \} \rangle\}$  - deze binnenste selectie vervangt elke buitenste selectie in "Field".
- $\{\langle \text{Field} += \{ \text{value} \} \rangle\}$  - deze binnenste selectie wordt samengevoegd met de buitenste selectie in "Field", met behulp van de verenigingsoperator.
- $\{\langle \text{Field} * = \{ \text{value} \} \rangle\}$  - deze binnenste selectie wordt samengevoegd met de buitenste selectie in "Field", met behulp van de doorsnede-operator.

### Overname in meerdere stappen

Overname kan in meerdere stappen worden uitgevoerd. Voorbeelden:

- Huidige selectie  $\rightarrow \text{Sum}(\text{Amount})$   
De aggregatiefunctie gebruikt de context, dat is hier de huidige selectie.
- Huidige selectie  $\rightarrow \{\langle \text{Set1} \rangle\} \text{Sum}(\text{Amount})$   
set1 neemt de huidige selectie over en het resultaat is de context voor de aggregatiefunctie.
- Huidige selectie  $\rightarrow \{\langle \text{Set1} \rangle\} (\{\langle \text{Set2} \rangle\} \text{Sum}(\text{Amount}))$   
set2 neemt set1 over, die vervolgens de huidige selectie overneemt en het resultaat is de context voor de aggregatiefunctie.

### De functie Aggr()

De functie `Aggr()` maakt een geneste aggregatie die twee afzonderlijke aggregaties bevat. In het onderstaande voorbeeld wordt een `count()` berekend voor elke waarde van `Dim`, en de resulterende matrix is geaggregeerd met behulp van de functie `sum()`.

#### Voorbeeld:

`Sum(Aggr(Count(X), Dim))`

`count()` is de binnenste aggregatie en `sum()` is de buitenste aggregatie.

- De binnenste aggregatie neemt geen context over van de buitenste aggregatie.
- De binnenste aggregatie neemt de context van de functie `Aggr()` over, die mogelijk een set-uitdrukking bevat.
- Zowel de functie `Aggr()` als de buitenste aggregatiefunctie nemen de context van een buitenste set-uitdrukking over.



### Zelfstudie - Een set-uitdrukking maken

U kunt set-uitdrukkingen maken om gegevensanalyse te ondersteunen. In deze context wordt de analyse vaak set-analyse genoemd. Set-analyse biedt een manier om een bereik te definiëren dat verschilt van de set met records die wordt gedefinieerd door de huidige selectie in een app.

#### Wat u gaat leren

Deze zelfstudie bevat de gegevens en diagramuitdrukkingen om set-uitdrukkingen te maken met set-Id's, set-operatoren en set-modificaties.

#### Wie moet deze zelfstudie volgen

Deze zelfstudie is bedoeld voor app-ontwikkelaars die vertrouwd zijn met het werken met de scripteditor en diagramuitdrukkingen.

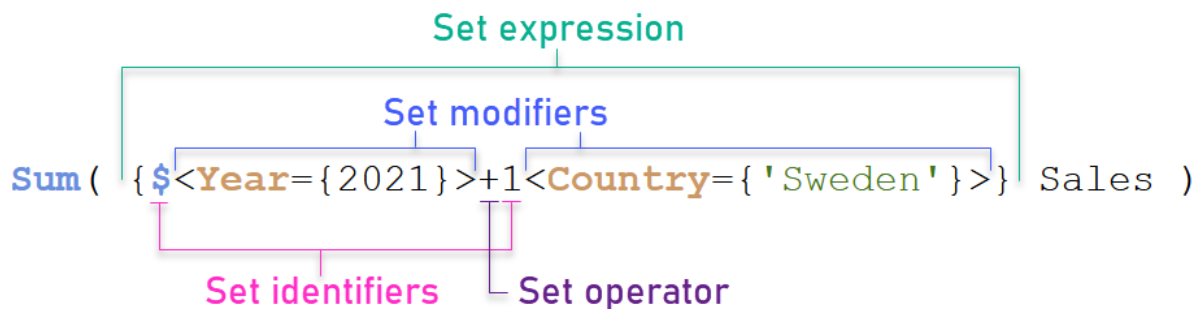
#### Wat u eerst moet doen

Een toewijzing voor Professional-toegang tot Qlik Sense Enterprise, waarmee u gegevens kunt laden en apps kunt maken.

#### Elementen in een set-uitdrukking

Set-uitdrukkingen zijn ingesloten in een aggregatiefunctie, zoals `sum()`, `max()`, `min()`, `avg()` of `count()`. Set-uitdrukkingen worden gemaakt van bouwblokken die we elementen noemen. Deze elementen zijn set-id's, set-operatoren en set-modificaties.

*Elementen in een set-uitdrukking*



De bovenstaande set-uitdrukking is bijvoorbeeld gebaseerd op de aggregatie `sum(Sales)`. De set-uitdrukking is ingesloten tussen de buitenste accolades: `{ }`

De eerste operand in de uitdrukking is: `$<Year={2021}>`

Deze operand retourneert verkopen voor het jaar 2021 voor de huidige selectie. De modificatie, `<Year={2021}>`, bevat de selectie van het jaar 2021. De set-id `$` geeft aan dat de set-uitdrukking is gebaseerd op de huidige selectie.

De tweede operand in de uitdrukking is: `1<Country={'Sweden'}>`

Deze operand retourneert Sales voor Sweden. De modificatie, `<Country={'Sweden'}>`, bevat de selectie van het land Sweden. De set-id 1 geeft aan dat de selecties die in de app zijn gemaakt, worden genegeerd.

Tot slot geeft de set-operator + aan dat de uitdrukking een set retourneert die bestaat uit de records die kunnen toebehoren aan elk van de twee set-operanden.

### Zelfstudie - Een set-uitdrukking maken

Voer de volgende procedures uit om de set-uitdrukkingen te maken die in deze zelfstudie worden getoond.

#### Een nieuwe app maken en gegevens laden

##### Doe het volgende:

1. Maak een nieuwe app.
2. Klik op **Scripteditor**. Als alternatief kunt u **Voorbereiden > Editor voor het laden van gegevens** kiezen in de navigatiebalk.
3. Maak een nieuwe sectie in de **Editor voor het laden van gegevens**.
4. Kopieer de volgende gegevens en plak deze in de nieuwe sectie: *Gegevens voor zelfstudie van set-uitdrukkingen (page 313)*
5. Klik op **Gegevens laden**. De gegevens worden geladen als een inline load.

#### Set-uitdrukkingen maken met modificaties

De set-modificatie bestaat uit een of meer veldnamen, elk gevolgd door een selectie die op het veld wordt toegepast. De modificatie is omsloten door punthaken. Bijvoorbeeld in deze set-uitdrukking:

```
sum ( {<Year = {2015}>} Sales )
```

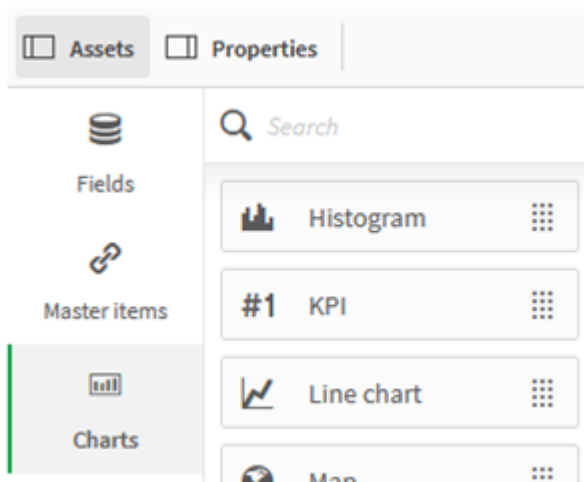
De modificatie is:

```
<Year = {2015}>
```

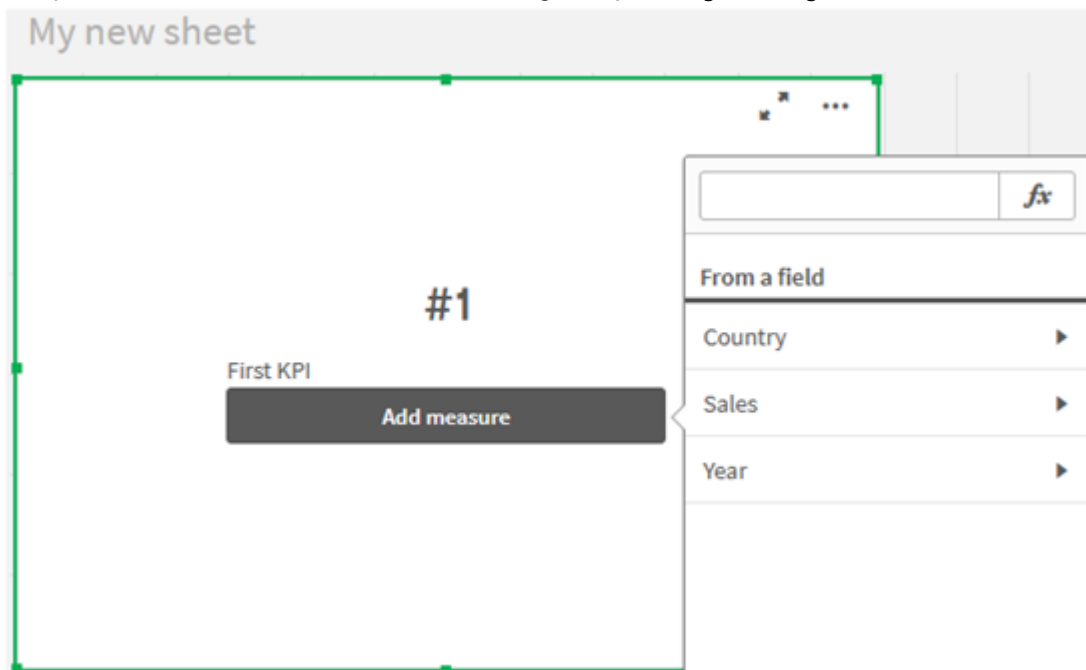
Deze modificatie geeft aan dat gegevens uit het jaar 2015 worden geselecteerd. De accolades die de modificatie omsluiten, duiden op een set-uitdrukking.

##### Doe het volgende:

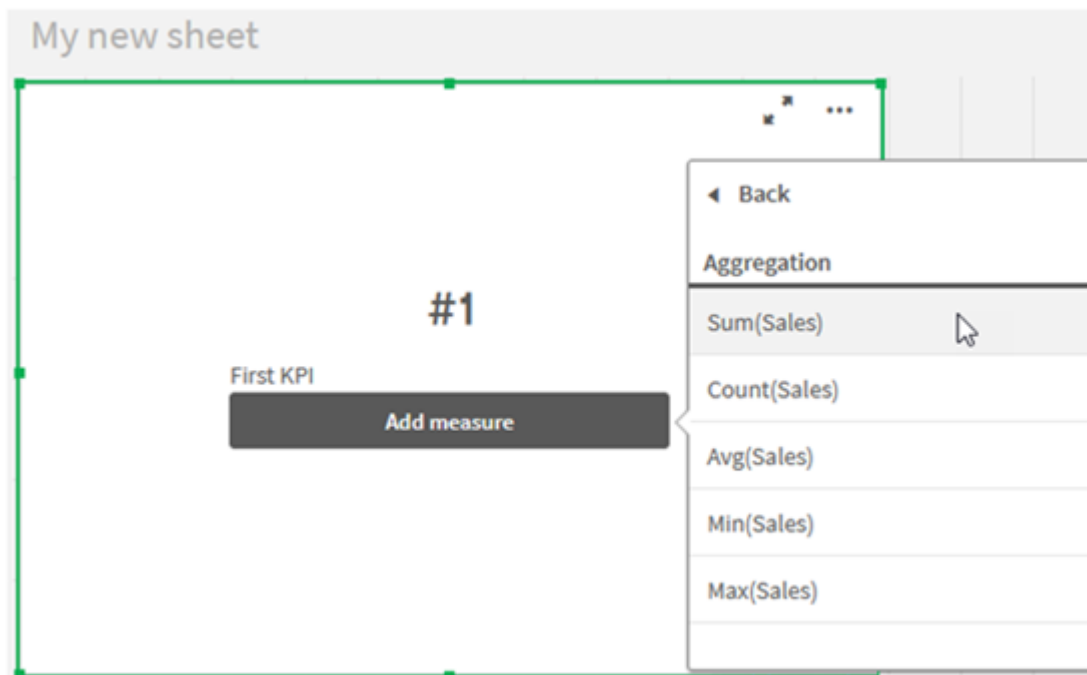
1. Open in een werkblad het venster **Bedrijfsmiddelen** in de navigatiebalk en klik vervolgens op **Diagrammen**.



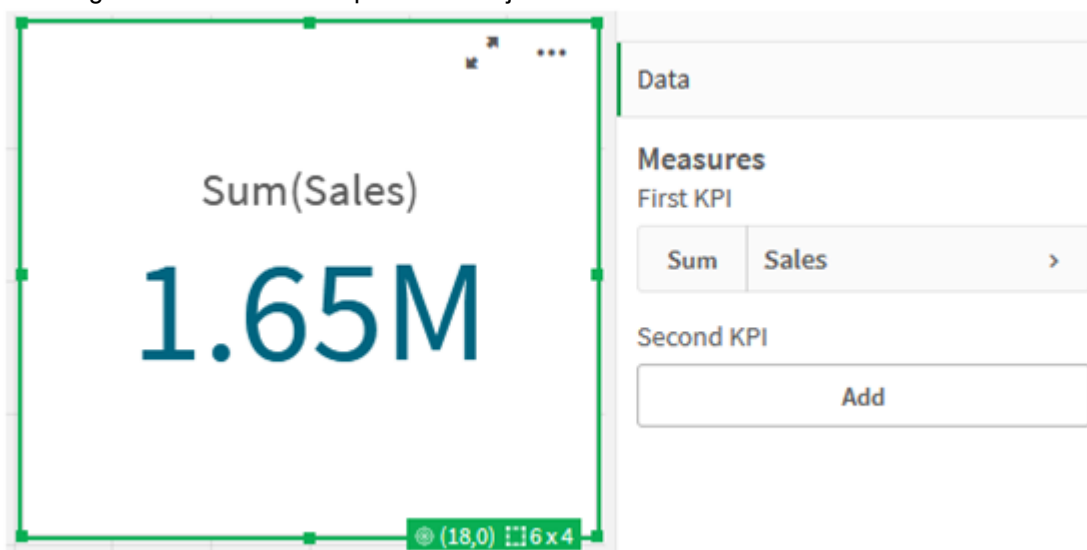
2. Sleep een KPI naar een werkblad en klik vervolgens op **Meting toevoegen**.



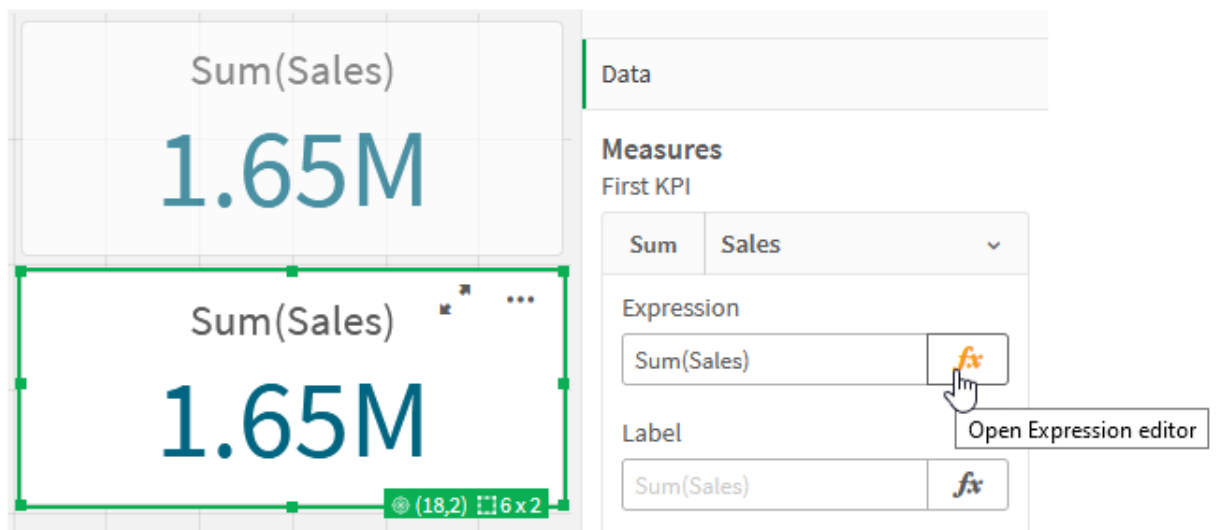
3. Klik op sales en selecteer vervolgens `sum(Sales)` voor de aggregatie.



De KPI geeft de som van verkopen voor alle jaren weer.



4. Kopieer en plak de KPI om een nieuwe KPI te maken.
5. Klik op de nieuwe KPI, klik op **Verkoop** onder **Metingen** en klik vervolgens op **Uitdrukkingeditor** openen.



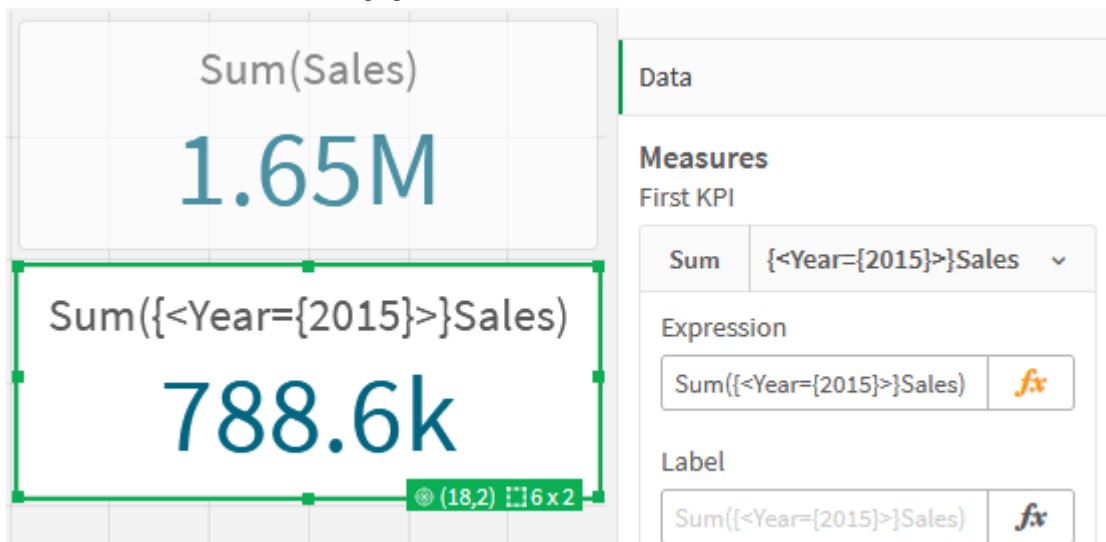
De uitdrukkingeditor wordt geopend met de aggregatie `sum(Sales)`.



6. Maak in de uitdrukkingeditor een uitdrukking voor de som van Sales alleen voor 2015:
  - i. Voeg accolades toe om een set-uitdrukking aan te duiden: `sum({}Sales)`
  - i. Voeg punthaken toe om een set-modificatie aan te duiden: `sum({<>}Sales)`
  - ii. Voeg tussen punthaken het selectieveld toe, in dit geval is het veld `year`, gevolgd door een gelijkteken. Sluit vervolgens 2015 in tussen nog een paar accolades. De resulterende set-modificatie is: `{<Year={2015}>}`.  
De volledige uitdrukking is:  
`sum({<Year={2015}>}Sales)`



- iii. Klik op **Toepassen** om de uitdrukking op te slaan en de uitdrukkingseeditor te sluiten. De som van Sales voor 2015 wordt weergegeven in de KPI.



- 7. Maak nog twee KPI's met de volgende uitdrukkingen:

`sum({<Year={2015,2016}>}Sales)`

De modificatie in het bovenstaande is `<Year={2015,2016}>`. De uitdrukking retourneert de som van Sales voor 2015 en 2016

`sum({<Year={2015},Country={'Germany'}>} Sales)`

De modificatie in het bovenstaande is `<Year={2015}, Country={'Germany'}>`. De uitdrukking retourneert de som van Sales voor 2015, waarbij 2015 kruist met Germany.

KPI's die gebruikmaken van set-modificaties

### Set-id's toevoegen

De bovenstaande set-uitdrukkingen gebruiken huidige selecties als basis, omdat geen id werd gebruikt. Voeg vervolgens id's toe om het gedrag op te geven wanneer u selecties maakt.

### Doe het volgende:

Maak of kopieer op uw werkblad de volgende set-uitdrukkingen:

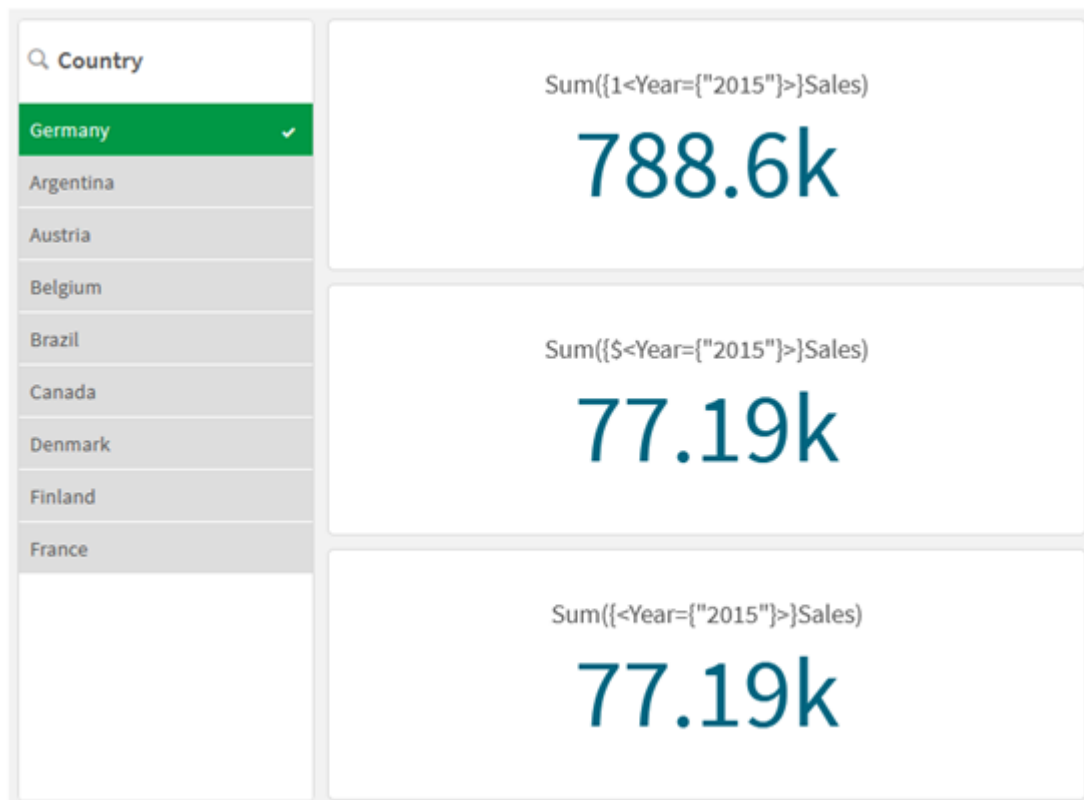
```
sum({$<Year={"2015"}>}Sales)
```

De id \$ baseert de set-uitdrukking op de huidige selecties die in de gegevens zijn gemaakt. Dit is ook het standaardgedrag wanneer een id niet wordt gebruikt.

```
sum({1<Year={"2015"}>}Sales)
```

Met de id 1 houdt de aggregatie van `sum(sales)` op 2015 geen rekening met de huidige selectie. De waarde van de aggregatie wijzigt niet wanneer de gebruiker andere selecties maakt. Als hieronder bijvoorbeeld Germany is geselecteerd, wordt de waarde voor het aggregatietotaal van 2015 niet gewijzigd.

*KPI's die gebruikmaken van set-modificaties en id's*



### Operatoren toevoegen

Set-operatoren worden gebruikt voor het opnemen, uitsluiten of het doorkruisen van gegevensverzamelingen. Alle operatoren gebruiken sets als operanden en leveren een set op als resultaat.

U kunt set-operatoren in twee verschillende situaties gebruiken:

- Om de bewerking Set uit te voeren voor set-id's die een set met records in de gegevens vertegenwoordigen.
- Om de bewerking Set uit te voeren voor elementsets, voor de veldwaarden of in een set-modificatie.

### Doe het volgende:

Maak of kopieer op uw werkblad de volgende set-uitdrukking:

```
sum({$<Year={2015}>+1<Country={'Germany'}>}Sales)
```



### 3 Diagramuitdrukkingen

Hier bewerkstelt de plustekenoperator (+) een samenvoeging (union) van de gegevensverzamelingen voor 2015 en Germany. Zoals hierboven werd aangegeven met betrekking tot set-id's, betekent de dollarteken-id (\$) dat de huidige selecties voor de eerste operand, <Year={2015}> worden gebruikt. De id 1 betekent dat de selectie wordt genegeerd voor de tweede operand, <Country={'Germany'}>.

*KPI die gebruikmaakt van de operator plusteken (+)*

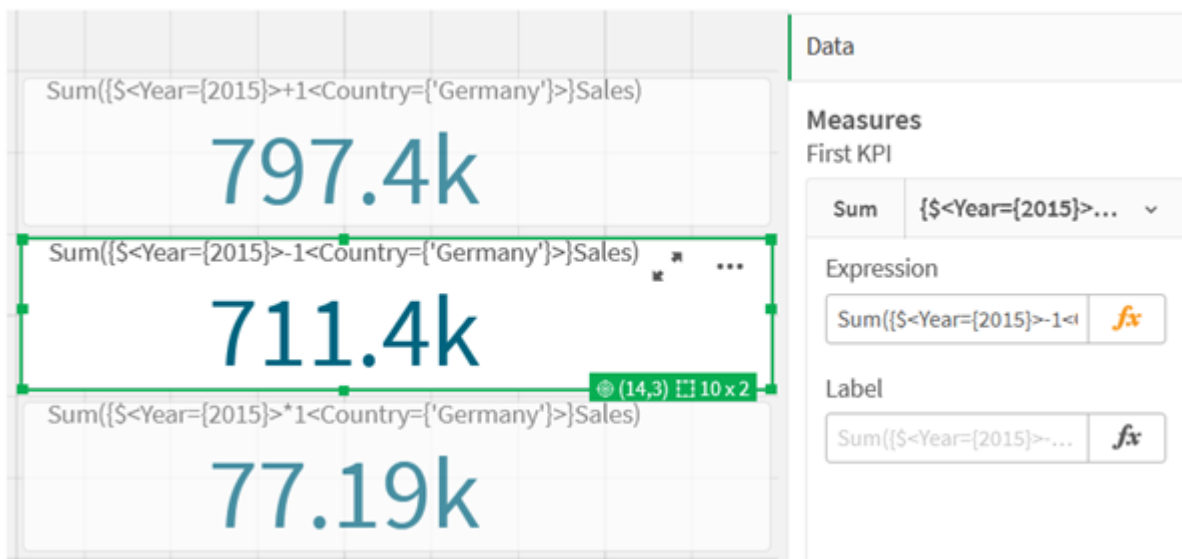


Gebruik anders een minteken (-) om een gegevensverzameling te retourneren die bestaat uit de records die behoren bij 2015 maar niet bij Germany. Of gebruik een sterretje (\*) om een verzameling te retourneren van de records die behoren bij beide verzamelingen.

`Sum({$<Year={2015}>-1<Country={'Germany'}>}Sales)`

`Sum({$<Year={2015}>*1<Country={'Germany'}>}Sales)`

*KPI's die operators gebruiken*



#### Gegevens voor zelfstudie van set-uitdrukkingen

Load-script

Laad de volgende gegevens als een inline-load en maak vervolgens de diagramuitdrukkingen in de zelfstudie.

```
//Create table SalesByCountry
SalesByCountry:
Load * Inline [
Country, Year, Sales
Argentina, 2016, 66295.03
Argentina, 2015, 140037.89
Austria, 2016, 54166.09
Austria, 2015, 182739.87
Belgium, 2016, 182766.87
Belgium, 2015, 178042.33
Brazil, 2016, 174492.67
Brazil, 2015, 2104.22
Canada, 2016, 101801.33
Canada, 2015, 40288.25
Denmark, 2016, 45273.25
Denmark, 2015, 106938.41
Finland, 2016, 107565.55
Finland, 2015, 30583.44
France, 2016, 115644.26
France, 2015, 30696.98
Germany, 2016, 8775.18
Germany, 2015, 77185.68
];
```

### Syntaxis voor set-uitdrukkingen

De volledige syntaxis (het optionele gebruik van standaard haakjes voor de definitie van prioriteit is niet inbegrepen) wordt beschreven met Backus-Naur-formalisme:

```
set_expression ::= { set_entity { set_operator set_entity } }
set_entity ::= set_identifier [ set_modifier ] | set_modifier
set_identifier ::= 1 | $ | $N | $_N | bookmark_id | bookmark_name
set_operator ::= + | - | * | /
set_modifier ::= < field_selection {, field_selection } >
field_selection ::= field_name [ = | += | -= | *= | /= ] element_set_
expression
element_set_expression ::= [ - ] element_set { set_operator element_set }
element_set ::= [ field_name ] | { element_list } | element_function
element_list ::= element { , element }
element_function ::= ( P | E ) ( [set_expression] [field_name] )
element ::= field_value | " search_mask "
```

### 3.3 Algemene syntaxis voor diagramuitdrukkingen

De volgende algemene syntaxisstructuur kan worden gebruikt voor diagramuitdrukkingen, met veel optionele parameters:

```
expression ::= ( constant | expressionname | operator1 expression | expression operator2
expression | function | aggregation function | (expression) )
```

waarbij:

**constant** een tekenreeks (tekst, datum of tijd) is die tussen enkele rechte aanhalingstekens staat, of een getal. Constanten worden geschreven zonder scheidingstekens voor duizendtallen en met een punt als scheidingstekens voor decimalen.

**expressionname** is de naam (het label) is van een andere uitdrukking in hetzelfde diagram.

**operator1** is een unaire operator (werkt op één uitdrukking, de uitdrukking rechts van de operator).

**operator2** is een binaire operator (werkt op twee uitdrukkingen, een aan elke zijde).

```
function ::= functionname ( parameters )  
parameters ::= expression { , expression }
```

Het aantal en de typen parameters zijn niet willekeurig. Deze zijn afhankelijk van de gebruikte functie.

```
aggregationfunction ::= aggregationfunctionname ( parameters2 )  
parameters2 ::= aggregationexpression { , aggregationexpression }
```

Het aantal en de typen parameters zijn niet willekeurig. Deze zijn afhankelijk van de gebruikte functie.

### 3.4 Algemene syntaxis voor aggregaties

De volgende algemene syntaxisstructuur kan worden gebruikt voor aggregaties, met veel optionele parameters:

```
aggregationexpression ::= ( fieldref | operator1 aggregationexpression | aggregationexpression operator2  
aggregationexpression | functioninaggr | ( aggregationexpression ) )
```

**fieldref** is een veldnaam.

```
functionaggr ::= functionname ( parameters2 )
```

Uitdrukkingen en functies kunnen zo naar wens worden genest. Zolang een **fieldref** is omsloten door precies één aggregatiefunctie en de uitdrukking een interpreteerbare waarde oplevert, verschijnt er geen foutmelding in Qlik Sense.

## 4 Operatoren

In dit hoofdstuk komen de operatoren die u in Qlik Sense kunt gebruiken aan de orde. Er zijn twee soorten operatoren:

- Unaire operatoren (hebben slechts één operand)
- Binaire operatoren (hebben twee operanden)

De meeste operatoren zijn binair.

De volgende operatoren kunnen worden gedefinieerd:

- Bit-operatoren
- Logische operatoren
- Numerieke operatoren
- Relatieve operatoren
- Tekenreeksoperatoren

### 4.1 Bit-operatoren

Alle bit-operatoren zetten de operanden om in (of kappen ze af tot) ondertekende gehele getallen (32-bits) en retourneren het resultaat op dezelfde wijze. Alle bewerkingen worden bit voor bit uitgevoerd. Als een operand niet kan worden geïnterpreteerd als een getal, zal de bewerking NULL retourneren.

Bit-operatoren

Operator	Volledige naam	Beschrijving
bitnot	Inverse op bitniveau.	Unaire operator. De bewerking retourneert de logische inverse van de operand die bit voor bit wordt uitgevoerd.  <b>Voorbeeld:</b>  bitnot 17 retourneert -18
bitand	AND op bitniveau.	De bewerking levert de logische AND op van de operanden die bit voor bit worden uitgevoerd.  <b>Voorbeeld:</b>  17 bitand 7 retourneert 1

Operator	Volledige naam	Beschrijving
bitor	OR op bitniveau.	De bewerking levert de logische OR op van de operanden die bit voor bit worden uitgevoerd.  <b>Voorbeeld:</b>  17 bitor 7 retourneert 23
bitxor	XOR op bitniveau.	De bewerking retourneert de logische exclusieve OR van de operanden die bit voor bit worden uitgevoerd.  <b>Voorbeeld:</b>  17 bitxor 7 retourneert 22
>>	Bit naar rechts verschuiven.	De bewerking retourneert de eerste operand die naar rechts is verschoven. Het aantal stappen wordt gedefinieerd in de tweede operand.  <b>Voorbeeld:</b>  8 >> 2 retourneert 2
<<	Bit naar links verschuiven.	De bewerking retourneert de eerste operand die naar links is verschoven. Het aantal stappen wordt gedefinieerd in de tweede operand.  <b>Voorbeeld:</b>  8 << 2 retourneert 32

## 4.2 Logische operatoren

Alle logische operatoren interpreteren de operanden logisch en geven True (-1) of False (0) als resultaat.

Logische operatoren

Operator	Beschrijving
not	Logische inverse. Een van de weinige unaire operatoren. De bewerking retourneert de logische inverse van de operand.
and	Logische EN. De bewerking retourneert de logische EN van de operanden.
or	Logische OF. De bewerking retourneert de logische OF van de operanden.
Xor	Logische uitsluitende OF. De bewerking retourneert de logische OF van de operanden. Met andere woorden, net als de logische OF maar met dat verschil dat het resultaat False is als beide operanden True zijn.

### 4.3 Numerieke operatoren

Alle numerieke operatoren gebruiken de numerieke waarden van de operanden en geven een numerieke waarde als resultaat.

Numerieke operatoren

Operator	Beschrijving
+	Teken voor positief getal (unaire operator) of rekenkundige optelling. De binaire bewerking retourneert de som van de twee operanden.
-	Teken voor negatief getal (unaire operator) of rekenkundige aftrekking. De unaire bewerking retourneert de operand vermenigvuldigd met -1. De binaire bewerking retourneert het verschil van de twee operanden.
*	Rekenkundige vermenigvuldiging. De bewerking retourneert het product van de twee operanden.
/	Rekenkundige deling. De bewerking retourneert de verhouding van de twee operanden.

### 4.4 Relationale operatoren

Alle relationele operatoren vergelijken de waarden van operanden en retourneren als resultaat True (-1) of False (0). Alle relationele operatoren zijn binair.

Relationele operatoren

Operator	Beschrijving
<	Kleiner dan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking.
<=	Kleiner dan of gelijk aan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking.
>	Groter dan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking.
>=	Groter dan of gelijk aan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking.

Operator	Beschrijving
=	Gelijk aan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking.
<>	Niet gelijk aan. Er wordt een numerieke vergelijking gemaakt als beide operanden numeriek geïnterpreteerd kunnen worden. De operator retourneert de logische waarde van het resultaat van de vergelijking.
<b>precedes</b>	<p>In tegenstelling tot de operator &lt; wordt er geen poging gedaan de argumentwaarden numeriek te interpreteren voordat er wordt vergeleken. De bewerking retourneert Waar als de waarde links van de operator een tekstrepresentatie heeft die in een ASCII-vergelijking vóór de tekstrepresentatie van de waarde aan de rechterkant komt.</p> <p><b>Voorbeeld:</b></p> <p>'1 ' precedes ' 2' retourneert FALSE</p> <p>' 1' precedes ' 2' retourneert TRUE</p> <p>aangezien de ASCII-waarde van een spatie (' ') van minder waarde is dan de ASCII-waarde van een getal.</p> <p>Vergelijk dit met:</p> <p>'1 ' &lt; ' 2' retourneert TRUE</p> <p>' 1' &lt; ' 2' retourneert TRUE</p>

Operator	Beschrijving
<b>follows</b>	<p>In tegenstelling tot de operator <code>&lt;</code> wordt er geen poging gedaan de argumentwaarden numeriek te interpreteren voordat er wordt vergeleken. De bewerking retourneert Waar als de waarde links van de operator een tekstrepresentatie heeft die in een ASCII-vergelijking vóór de tekstrepresentatie van de waarde aan de rechterkant komt.</p> <p><b>Voorbeeld:</b></p> <p><code>' 2' follows '1'</code> retourneert FALSE</p> <p><code>'2' follows ' 1'</code> retourneert TRUE</p> <p>aangezien de ASCII-waarde van een spatie (' ') van minder waarde is dan de ASCII-waarde van een getal.</p> <p>Vergelijk dit met:</p> <p><code>' 2' &gt; ' 1'</code> retourneert TRUE</p> <p><code>' 2' &gt; '1 '</code> retourneert TRUE</p>

## 4.5 Tekenreeksoperatoren

Er zijn twee tekenreeksoperatoren. De eerste maakt gebruik van de tekenreekswaarden van de operanden en retourneert een tekenreeks. De andere vergelijkt de operanden en retourneert een booleaanse waarde waarmee de overeenkomst wordt aangegeven.

### &

Tekenreeksverbinding. De bewerking retourneert een tekenreeks, die bestaat uit de achter elkaar geplaatste tekenreeksen van de twee operanden.

#### Voorbeeld:

`'abc' & 'xyz'` retourneert `'abcxyz'`

### like

Tekenreeksvergelijking met jokertekens. De bewerking retourneert de booleaanse waarde True (-1) als de tekenreeks vóór de operator overeenkomt met de tekenreeks achter de operator. De tweede tekenreeks kan de jokertekens \* (elk aantal willekeurige tekens) of ? (een enkel willekeurig karakter) bevatten.

#### Voorbeeld:

`'abc' like 'a*'` retourneert True (-1)



'abcd' like 'a?c\*' retourneert True (-1)

'abc' like 'a??bc' retourneert False (0)

# 5 Script- en diagramfuncties

Transformeer en aggregeer gegevens met functies in load-scripts voor gegevens en diagramuitdrukkingen.

Veel functies kunnen op dezelfde wijze worden gebruikt in zowel scripts voor het laden van gegevens als diagramuitdrukkingen, maar er is een aantal uitzonderingen:

- Sommige functies kunnen alleen worden gebruikt in scripts voor het laden van gegevens, aangeduid met - scriptfunctie.
- Sommige functies kunnen alleen worden gebruikt in diagramuitdrukkingen, aangeduid met - diagramfunctie.
- Sommige functies kunnen worden gebruikt in zowel scripts voor het laden van gegevens als diagramuitdrukkingen, maar met verschillen in parameters en toepassing. Deze worden beschreven in aparte onderwerpen aangeduid met - scriptfunctie of - diagramfunctie.

## 5.1 Analytische verbindingen voor uitbreidingen aan de serverzijde (SSE)

Functies die zijn ingeschakeld door analytische verbindingen zijn alleen zichtbaar als u de analytische verbindingen hebt geconfigureerd en Qlik Sense is gestart. U kunt analytische verbindingen configureren in de QMC. Zie [Analytische verbindingen maken](#).

U kunt de analytische verbindingen configureren in de . Zie in de gids het onderwerp "Analytische verbindingen maken".QMCQlik Sense-sites beheren

)

In Qlik Sense Desktop configureert u de analytische verbindingen door het bestand *Settings.ini* te bewerken, zie het onderwerp "Configuring analytic connections in Qlik Sense Desktop" ("Analytische verbindingen configureren in ") in de handleiding Qlik Sense Desktop.

## 5.2 Aggregatiefuncties

De functiegroep met zogeheten aggregatiefuncties bestaat uit functies die meerdere veldwaarden als invoer aannemen en een enkel resultaat per groep retourneren, waarbij de groepering wordt gedefinieerd door een diagramdimensie of een **group by**-clausule in de scriptopdracht.

Aggregatiefuncties zijn onder meer **Sum()**, **Count()**, **Min()**, **Max()** en nog veel meer.

De meeste aggregatiefuncties kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen, maar de syntaxis verschilt.

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Bij het toewijzen van een naam aan een entiteit moet u niet dezelfde naam geven aan meer dan één veld, variabele of meting. Er geldt een strikte volgorde van prioriteit voor het oplossen van conflicten tussen entiteiten met identieke namen. Deze volgorde wordt weerspiegeld in objecten of contexten waarin deze entiteiten worden gebruikt. Deze volgorde van prioriteit is als volgt:

- Binnen een aggregatiefunctie krijgt een veld voorrang op een variabele. Metinglabels zijn niet relevant in aggregaties en krijgen geen prioriteit.
- Buiten een aggregatiefunctie krijgt een metinglabel voorrang op een variabele, die weer voorrang krijgt op een veld.
- Daarnaast kan een meting buiten een aggregatie opnieuw gebruikt worden door te verwijzen naar het label, tenzij het label is berekend. In die situatie krijgt de meting minder prioriteit om het risico te voorkomen dat de uitdrukking naar zichzelf verwijst, en in dit geval wordt de naam altijd eerst geïnterpreteerd als een metinglabel, vervolgens als een veldnaam en daarna als een variabelenaam.

### Aggregatiefuncties gebruiken in het script voor het laden van gegevens

Aggregatiefuncties kunnen alleen worden gebruikt binnen de opdrachten **LOAD** en **SELECT**.

### Aggregatiefuncties gebruiken in diagramuitdrukkingen

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Een aggregatiefunctie aggregeren over de set mogelijke records die wordt gedefinieerd door de selectie. Maar er kan een alternatieve set records worden gedefinieerd door gebruik te maken van een set-uitdrukking bij set-analyse.

### Hoe aggregaties worden berekend

Een aggregatie doorloopt de records van een specifieke tabel en aggregaat de records in die tabel. Zo wordt bijvoorbeeld met **Count(<Field>)** het aantal records geteld in de tabel waar <Field> voorkomt. Als u alleen de unieke veldwaarden wilt aggregeren, moet u de **distinct**-clausule gebruiken, zoals **Count(distinct <Field>)**.

Als de aggregatiefunctie velden uit verschillende tabellen bevat, doorloopt de aggregatiefunctie de records van het vectorproduct van de velden. Dit zorgt voor prestatieverlies, en om die reden moet u zulke aggregaties vermijden, met name als u grote aantallen gegevens gebruikt.

### Aggregatie van sleutelvelden

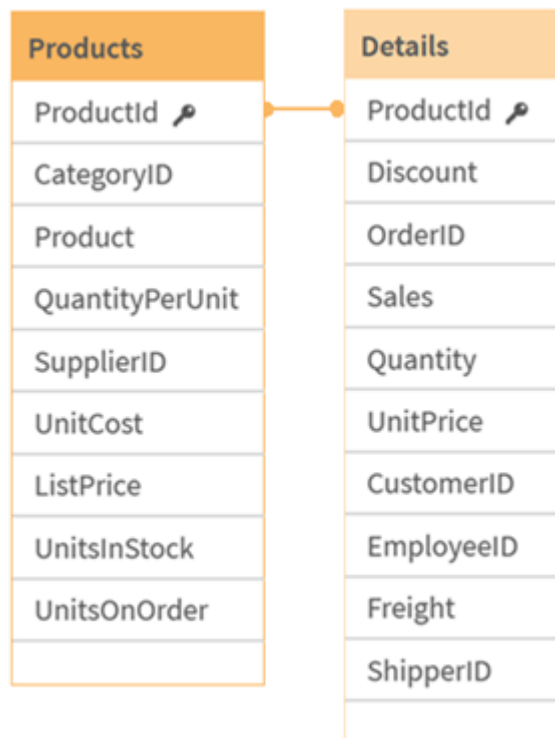
Door de manier waarop aggregaties worden berekend, kunt u geen sleutelvelden aggregeren omdat daarbij niet duidelijk is welke tabel voor de aggregatie moet worden gebruikt. Als bijvoorbeeld het veld <Key> twee tabellen aan elkaar koppelt, is niet duidelijk of **Count(<Key>)** het aantal records van de eerste of tweede tabel moet retourneren.

Maar als u het component **distinct** gebruikt, is de aggregatie op de juiste manier gedefinieerd en kan deze vervolgens worden berekend.

Dus als u een sleutelveld in een aggregatiefunctie gebruikt zonder het component **distinct** retourneert Qlik Sense een aantal waar u mogelijk niets aan hebt. De oplossing is in dit geval om het component **distinct** of een kopie van de sleutel te gebruiken - een kopie die alleen in één tabel voorkomt.

In de volgende tabellen is ProductID bijvoorbeeld de sleutel tussen de tabellen.

*De sleutel ProductID tussen tabellen van producten en details*



Count(ProductID) kan zowel geteld worden in de tabel Products (die maar één record per product heeft - ProductID is de primaire sleutel) of het kan in de Details-tabel geteld worden (die waarschijnlijk meerdere records per product heeft). Als u het aantal verschillende producten wilt tellen, moet u Count(distinct ProductID) gebruiken. Als u het aantal rijen in een specifieke tabel wilt tellen, moet u de sleutel niet gebruiken.

### Basisaggregatiefuncties

#### Overzicht basisaggregatiefuncties

De basisaggregatiefuncties zijn een groep van de meest gebruikte aggregatiefuncties.

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Basisaggregatiefuncties gebruiken in het script voor het laden van gegevens

##### FirstSortedValue

**FirstSortedValue()** retourneert de waarde van de uitdrukking die is opgegeven in **value** die overeenkomt met het resultaat van het sorteren van **sort\_weight**-argument, bijvoorbeeld de naam van het product met de laagste prijs per eenheid. U kunt de **n**-de waarde in de sorteervolgorde opgeven in **rank**. Als meer dan één uitkomst dezelfde **sort\_weight** voor de opgegeven **rank** deelt, retourneert de functie NULL. De gesorteerde waarden worden herhaald over een aantal records, zoals is vastgelegd door een **group by**-clausule, of geaggregeerd over een volledige gegevensverzameling als er geen **group by**-clausule is gedefinieerd.

```
FirstSortedValue ([ distinct ] expression, sort_weight [, rank ])
```

##### Max

**Max()** retourneert de hoogste numerieke waarde van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. Door een **rank** **n** op te geven, wordt de **n**-de hoogste waarde gezocht.

```
Max ( expression[, rank])
```

##### Min

**Min()** retourneert de laagste numerieke waarde van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. Door een **rank** **n** op te geven, wordt de **n**-de laagste waarde gezocht.

```
Min ( expression[, rank])
```

##### Mode

**Mode()** retourneert de vaakst voorkomende waarde, de moduswaarde, van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. De functie **Mode()** kan zowel numerieke waarden als tekstwaarden retourneren.

```
Mode (expression )
```

##### Only

**Only()** retourneert een waarde als er één, en niet meer dan één, mogelijk resultaat is voor de geaggregeerde gegevens. Als records slechts één waarde bevatten, wordt die waarde geretourneerd, anders wordt NULL geretourneerd. Gebruik de **group by**-clausule om meerdere records te evalueren. De functie **Only()** kan numerieke waarden en tekstwaarden retourneren.

**Only** (expression )

### Sum

**Sum()** berekent het totaal van de waarden die zijn geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

**Sum** ([distinct] expression)

## Basisaggregatiefuncties gebruiken in diagramuitdrukkingen

Aggregatiefuncties voor diagrammen kunnen alleen worden gebruikt op velden in diagramuitdrukkingen. De argumentuitdrukking van een aggregatiefunctie mag niet nog een andere aggregatiefunctie bevatten.

### FirstSortedValue

**FirstSortedValue()** retourneert de waarde van de uitdrukking die is opgegeven in **value** die overeenkomt met het resultaat van het sorteren van **sort\_weight**-argument, bijvoorbeeld de naam van het product met de laagste prijs per eenheid. U kunt de n-de waarde in de sorteervolgorde opgeven in **rank**. Als meer dan één uitkomst dezelfde **sort\_weight** voor de opgegeven **rank** deelt, retourneert de functie NULL.

```
FirstSortedValue - diagramfunctie([SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] value, sort_weight [,rank])
```

### Max

**Max()** retourneert de hoogste waarde van de geaggregeerde gegevens. Door een **rank** n op te geven, wordt de n-de hoogste waarde gezocht.

```
Max - diagramfunctieMax() retourneert de hoogste waarde van de geaggregeerde gegevens. Door een rank n op te geven, wordt de n-de hoogste waarde gezocht. Bekijk eventueel ook FirstSortedValue en rangemax, die een functionaliteit hebben die vergelijkbaar is met die van de functie Max. Max([SetExpression] [TOTAL [<fld {,fld}>]] expr [,rank])
```

numeriek ArgumentenArgumentBeschrijvingexprDe uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.rankDe standaardwaarde van rank is 1. Dit komt overeen met de hoogste waarde. Als u bij rank 2 opgeeft, wordt de op één na hoogste waarde geretourneerd. Als u bij rank 3 opgeeft, wordt de op twee na hoogste waarde geretourneerd, enzovoort.SetExpressionDe aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. TOTALAls het woord TOTAL voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd. Wanneer u TOTAL [<fld {.fld}>] gebruikt en de TOTAL-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. GegevensCustomerProductUnitSalesUnitPriceAstridaAA416AstridaAA1015AstridaBB99BetacabBB510BetacabCC220BetacabDD-25CanutilityAA815CanutilityCC-19Voorbeelden en resultatenVoorbeeldenResultatenMax(UnitSales)10, omdat dit de hoogste waarde in UnitSales is.De waarde van een bestelling wordt berekend op basis van het

aantal verkochte eenheden in (UnitSales) vermenigvuldigd met de prijs per eenheid. `Max(UnitSales*UnitPrice)` 150, omdat dit de hoogste waarde is van het resultaat van de berekening van alle mogelijke waarden van `(UnitSales)*(UnitPrice)`. `Max(UnitSales, 2)` 9, want dit is de op één hoogste waarde. `Max(TOTAL UnitSales)` 10, omdat de kwalificatie TOTAL betekent dat de hoogst mogelijke waarde wordt gevonden, waarbij de diagramdimensies buiten beschouwing worden gelaten. Voor een diagram met Customer als dimensie, zorgt de kwalificatie TOTAL ervoor dat de maximale waarde in de volledige gegevensverzameling wordt geretourneerd, in plaats van de maximale UnitSales voor elke klant. Voer de selectie Customer B uit. `Max({1} TOTAL UnitSales)` 15, ongeacht de uitgevoerde selectie, omdat de Set Analysis-uitdrukking {1} de set records definieert die moeten worden geëvalueerd als ALL, ongeacht welke selectie wordt gemaakt. Gegevens die worden gebruikt in voorbeelden: `ProductData:LOAD * inline [Customer|Product|UnitSales|UnitPriceAstrida|AA|4|16Astrida|AA|10|15Astrida|BB|9|9Betacab|BB|5|10Betacab|CC|2|20Betacab|DD||25Canutility|AA|8|15Canutility|CC||19] (delimiter is '|'); FirstSortedValue RangeMax ([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr [,rank])`

### Min

**Min()** retourneert de laagste waarde van de geaggregeerde gegevens. Door een **rank** n op te geven, wordt de n-de laagste waarde gezocht.

```
Min - diagramfunctie ([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr [,rank])
```

### Mode

**Mode()** retourneert de vaakst voorkomende waarde, de modus, in de geaggregeerde gegevens. De functie **Mode()** kan zowel tekstwaarden als numerieke waarden verwerken.

```
Mode - diagramfunctie ({[SetExpression] [TOTAL [<fld {,fld}>]]} expr)
```

### Only

**Only()** retourneert een waarde als er één, en niet meer dan één, mogelijk resultaat is voor de geaggregeerde gegevens. Als u bijvoorbeeld zoekt naar het enige product waarvan de prijs per eenheid =9 en er zijn meerdere producten met een prijs per eenheid van 9, dan wordt het resultaat NULL geretourneerd.

```
Only - diagramfunctie ([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr)
```

### Sum

**Sum()** berekent het totaal van de waarden die worden geleverd door de uitdrukking of het veld voor de geaggregeerde gegevens.

```
Sum - diagramfunctie ([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr)
```

## FirstSortedValue

**FirstSortedValue()** retourneert de waarde van de uitdrukking die is opgegeven in **value** die overeenkomt met het resultaat van het sorteren van **sort\_weight**-argument, bijvoorbeeld de naam van het product met de laagste prijs per eenheid. U kunt de *n*-de waarde in de sorteervolgorde opgeven in **rank**. Als meer dan één uitkomst dezelfde **sort\_weight** voor de opgegeven **rank** deelt, retourneert de functie NULL. De gesorteerde waarden worden herhaald over een aantal records, zoals is vastgelegd door een **group by**-clausule, of geaggregeerd over een volledige gegevensverzameling als er geen **group by**-clausule is gedefinieerd.

### Syntaxis:

```
FirstSortedValue ([ distinct ] value, sort-weight [, rank ])
```

Retourgegevenstypen: dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
value Expression	De functie vindt de waarde van de uitdrukking <b>value</b> die overeenkomt met het resultaat van de sortering <b>sort_weight</b> .
sort-weight Expression	De uitdrukking die de gegevens bevat die moeten worden gesorteerd. De eerste (laagste) waarde van <b>sort_weight</b> wordt gevonden, op basis waarvan de overeenkomstige waarde van de uitdrukking <b>value</b> wordt bepaald. Als u een minteken vóór <b>sort_weight</b> plaatst, retourneert de functie de laatste (hoogste) gesorteerde waarde.
rank Expression	Door een <b>rank</b> "n" op te geven die groter is dan 1, krijgt u de n-de gesorteerde waarde.
distinct	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.



### Scriptvoorbeelden

Voorbeeld	Resultaat
<pre>Temp: LOAD * inline [ Customer Product OrderNumber UnitSales CustomerID Astrida AA 1 10 1 Astrida AA 7 18 1 Astrida BB 4 9 1 Astrida CC 6 2 1 Betacab AA 5 4 2 Betacab BB 2 5 2 Betacab DD 12 25 2 Canutility AA 3 8 3 Canutility CC 13 19 3 Divadip AA 9 16 4 Divadip AA 10 16 4 Divadip DD 11 10 4 ] (delimiter is ' ');  FirstSortedValue: LOAD Customer,FirstSortedValue(Product, UnitSales) as MyProductWithSmallestOrderByCustomer Resident Temp Group By Customer;</pre>	<p>Customer MyProductWithSmallestOrderByCustomer Astrida CC Betacab AA Canutility AA Divadip DD</p> <p>De functie sorteert UnitSales van kleinst naar grootst en zoekt daarbij de waarde van Customer met de kleinste waarde van UnitSales, de kleinste bestelling.</p> <p>Omdat CC overeenkomt met de kleinste bestelling (waarde van UnitSales=2) voor klant Astrida. AA komt overeen met de kleinste bestelling (4) voor klant Betacab, AA komt overeen met de grootste bestelling (8) voor klant Canutility en DD komt overeen met de grootste bestelling (10) voor klant Divadip..</p>
<p>Stel dat de tabel <b>Temp</b> is geladen zoals in het vorige voorbeeld:</p> <pre>LOAD Customer,FirstSortedValue(Product, -UnitSales) as MyProductWithLargestOrderByCustomer Resident Temp Group By Customer;</pre>	<p>Customer MyProductWithLargestOrderByCustomer Astrida AA Betacab DD Canutility CC Divadip -</p> <p>Er staat een minteken voor het argument sort_weight, dus de functie sorteert de grootste eerst.</p> <p>Omdat AA overeenkomt met de grootste bestelling (waarde van UnitSales:18) voor klant Astrida, DD overeenkomt met de grootste bestelling (12) voor klant Betacab en CC overeenkomt met de grootste bestelling (13) voor klant Canutility. Er zijn twee identieke waarden voor de grootste bestelling (16) voor klant Divadip, dus dit resulteert in een null-waarde.</p>

Voorbeeld	Resultaat
<p>Stel dat de tabel <b>Temp</b> is geladen zoals in het vorige voorbeeld:</p> <pre>LOAD Customer,FirstSortedValue(distinct Product, - Unitsales) as MyProductWithSmallestOrderByCustomer Resident Temp Group By Customer;</pre>	<pre>Customer MyProductWithLargestOrderByCustomer Astrida AA Betacab DD Canutility CC Divadip AA</pre> <p>Dit is hetzelfde als het vorige voorbeeld, behalve dat de kwalificatie <b>distinct</b> is gebruikt. Dit veroorzaakt een dubbel resultaat voor Divadip dat kan worden genegeerd, wat een null-waarde retourneert.</p>

### FirstSortedValue - diagramfunctie

**FirstSortedValue()** retourneert de waarde van de uitdrukking die is opgegeven in **value** die overeenkomt met het resultaat van het sorteren van **sort\_weight**-argument, bijvoorbeeld de naam van het product met de laagste prijs per eenheid. U kunt de **n**-de waarde in de sorteervolgorde opgeven in **rank**. Als meer dan één uitkomst dezelfde **sort\_weight** voor de opgegeven **rank** deelt, retourneert de functie NULL.

#### Syntaxis:

```
FirstSortedValue([SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] value,
sort_weight [,rank])
```

**Retourgegevenstypen:** dual

#### Argumenten:

##### Argumenten

Argument	Beschrijving
value	Uitvoerveld. De functie vindt de waarde van de uitdrukking <b>value</b> die overeenkomt met het resultaat van de sortering <b>sort_weight</b> .
sort_weight	Invoerveld. De uitdrukking die de gegevens bevat die moeten worden gesorteerd. De eerste (laagste) waarde van <b>sort_weight</b> wordt gevonden, op basis waarvan de overeenkomstige waarde van de uitdrukking <b>value</b> wordt bepaald. Als u een minteken vóór <b>sort_weight</b> plaatst, retourneert de functie de laatste (hoogste) gesorteerde waarde.
rank	Door een <b>rank</b> "n" op te geven die groter is dan 1, krijgt u de n-de gesorteerde waarde.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.

Argument	Beschrijving
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.

#### Voorbeelden en resultaten:

Gegevens			
Customer	Product	UnitSales	UnitPrice
Astrida	AA	4	16
Astrida	AA	10	15
Astrida	BB	9	9
Betacab	BB	5	10
Betacab	CC	2	20
Betacab	DD	-	25
Canutility	AA	8	15
Canutility	CC	-	19

#### Voorbeelden en resultaten

Voorbeeld	Resultaat
firstsortedvalue (Product, UnitPrice)	BB, oftewel het product met de laagste unitPrice(9).
firstsortedvalue (Product, UnitPrice, 2)	BB, oftewel het product met de op één na laagste unitPrice(10).
firstsortedvalue (Customer, -UnitPrice, 2)	Betacab, oftewel de Customer met het product met de op één na hoogste unitPrice(20).

Voorbeeld	Resultaat
<code>firstsortedvalue (Customer, UnitPrice, 3)</code>	<p>NULL, omdat er twee waarden van <code>customer</code> (Astrida en Canutility) zijn met dezelfde rank (op twee na laagste) <code>unitPrice(15)</code>.</p> <p>U kunt met de kwalificatie <code>distinct</code> ervoor zorgen dat er geen onverwachte null-resultaten optreden.</p>
<code>firstsortedvalue (Customer, - UnitPrice*UnitsSales, 2)</code>	<p>Canutility, oftewel de <code>customer</code> met de op één na hoogste <code>unitPrice</code> voor verkooporderwaarde vermenigvuldigd met de <code>unitsales</code> (120).</p>

Gegevens die worden gebruikt in voorbeelden:

```
ProductData:
LOAD * inline [
Customer|Product|UnitsSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD||25
Canutility|AA|8|15
Canutility|CC||19
] (delimiter is '|');
```

### Max

**Max()** retourneert de hoogste numerieke waarde van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. Door een **rank** n op te geven, wordt de n-de hoogste waarde gezocht.

#### Syntaxis:

```
Max ( expr [, rank] )
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
<code>expr</code> Expression	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
<code>rank</code> Expression	De standaardwaarde van <b>rank</b> is 1. Dit komt overeen met de hoogste waarde. Als u bij <b>rank</b> 2 opgeeft, wordt de op één na hoogste waarde geretourneerd. Als u bij <b>rank</b> 3 opgeeft, wordt de op twee na hoogste waarde geretourneerd, enzovoort.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

### Voorbeeld:

Temp:

```
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|CustomerID
Astrida|AA|1|10|1
Astrida|AA|7|18|1
Astrida|BB|4|9|1
Astrida|CC|6|2|1
Betacab|AA|5|4|2
Betacab|BB|2|5|2
Betacab|DD
Canutility|DD|3|8
Canutility|CC
] (delimiter is '|');
```

Max:

```
LOAD Customer, Max(UnitSales) as MyMax Resident Temp Group By Customer;
```

Resultaattabel

Customer	MyMax
Astrida	18
Betacab	5
Canutility	8

### Voorbeeld:

Stel dat de tabel **Temp** is geladen zoals in het vorige voorbeeld:

```
LOAD Customer, Max(UnitSales,2) as MyMaxRank2 Resident Temp Group By Customer;
```

Resultaattabel

Customer	MyMaxRank2
Astrida	10
Betacab	4
Canutility	-

## Max - diagramfunctie

**Max()** retourneert de hoogste waarde van de geaggregeerde gegevens. Door een **rank** n op te geven,

wordt de n-de hoogste waarde gezocht.



Bekijk eventueel ook **FirstSortedValue** en **rangemax**, die een functionaliteit hebben die vergelijkbaar is met die van de functie **Max**.

### Syntaxis:

```
Max ([{SetExpression}] [TOTAL [<fld {,fld}>]] expr [,rank])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
rank	De standaardwaarde van <b>rank</b> is 1. Dit komt overeen met de hoogste waarde. Als u bij <b>rank</b> 2 opgeeft, wordt de op één na hoogste waarde geretourneerd. Als u bij <b>rank</b> 3 opgeeft, wordt de op twee na hoogste waarde geretourneerd, enzovoort.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {,fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.

### Voorbeelden en resultaten:

#### Gegevens

Customer	Product	UnitSales	UnitPrice
Astrida	AA	4	16
Astrida	AA	10	15
Astrida	BB	9	9
Betacab	BB	5	10
Betacab	CC	2	20

Customer	Product	UnitSales	UnitPrice
Betacab	DD	-	25
Canutility	AA	8	15
Canutility	CC	-	19

## Voorbeelden en resultaten

Voorbeelden	Resultaten
Max(UnitSales)	10, omdat dit de hoogste waarde in UnitSales is.
De waarde van een bestelling wordt berekend op basis van het aantal verkochte eenheden in (UnitSales) vermenigvuldigd met de prijs per eenheid. Max(UnitSales*UnitPrice)	150, omdat dit de hoogste waarde is van het resultaat van de berekening van alle mogelijke waarden van (UnitSales)*(UnitPrice).
Max(UnitSales, 2)	9, want dit is de op één hoogste waarde.
Max(TOTAL UnitSales)	10, omdat de kwalificatie TOTAL betekent dat de hoogst mogelijke waarde wordt gevonden, waarbij de diagramdimensies buiten beschouwing worden gelaten. Voor een diagram met Customer als dimensie, zorgt de kwalificatie TOTAL ervoor dat de maximale waarde in de volledige gegevensverzameling wordt geretourneerd, in plaats van de maximale UnitSales voor elke klant.
Voer de selectie Customer B uit. Max({1} TOTAL UnitSales)	15, ongeacht de uitgevoerde selectie, omdat de Set Analysis-uitdrukking {1} de set records definieert die moeten worden geëvalueerd als ALL, ongeacht welke selectie wordt gemaakt.

Gegevens die worden gebruikt in voorbeelden:

```
ProductData:
LOAD * inline [
Customer|Product|UnitSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD||25
Canutility|AA|8|15
Canutility|CC||19
] (delimiter is '|');
```

### Zie ook:

p *FirstSortedValue* - diagramfunctie (page 330)  
p *RangeMax* (page 1338)

### Min

**Min()** retourneert de laagste numerieke waarde van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. Door een **rank** n op te geven, wordt de n-de laagste waarde gezocht.

### Syntaxis:

```
Min ( expr [, rank] )
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr Expression	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
rank Expression	De standaardwaarde van <b>rank</b> is 1. Dit komt overeen met de laagste waarde. Als u bij <b>rank 2</b> opgeeft, wordt de op één na laagste waarde geretourneerd. Als u bij <b>rank 3</b> opgeeft, wordt de op twee na laagste waarde geretourneerd, enzovoort.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

### Voorbeeld:

```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|CustomerID
Astrida|AA|1|10|1
Astrida|AA|7|18|1
Astrida|BB|4|9|1
Astrida|CC|6|2|1
Betacab|AA|5|4|2
Betacab|BB|2|5|2
Betacab|DD
Canutility|DD|3|8
Canutility|CC
```



```
] (delimiter is '|');
Min:
LOAD Customer, Min(UnitSales) as MyMin Resident Temp Group By Customer;
```

Resultaattabel

Customer	MyMin
Astrida	2
Betacab	4
Canutility	8

### Voorbeeld:

Stel dat de tabel **Temp** is geladen zoals in het vorige voorbeeld:

```
LOAD Customer, Min(UnitSales,2) as MyMinRank2 Resident Temp Group By Customer;
```

Resultaattabel

Customer	MyMinRank2
Astrida	9
Betacab	5
Canutility	-

### Min - diagramfunctie

**Min()** retourneert de laagste waarde van de geaggregeerde gegevens. Door een **rank** n op te geven, wordt de n-de laagste waarde gezocht.



Bekijk eventueel ook **FirstSortedValue** en **rangemin**, die een functionaliteit hebben die vergelijkbaar is met die van de functie **Min**.

### Syntaxis:

```
Min({[SetExpression] [TOTAL [<fld {,fld}>]]} expr [,rank])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
rank	De standaardwaarde van <b>rank</b> is 1. Dit komt overeen met de laagste waarde. Als u bij <b>rank 2</b> opgeeft, wordt de op één na laagste waarde geretourneerd. Als u bij <b>rank 3</b> opgeeft, wordt de op twee na laagste waarde geretourneerd, enzovoort.

Argument	Beschrijving
SetExpression	De aggregatiefunctie aggregereert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Voorbeelden en resultaten:

Gegevens			
Customer	Product	UnitSales	UnitPrice
Astrida	AA	4	16
Astrida	AA	10	15
Astrida	BB	9	9
Betacab	BB	5	10
Betacab	CC	2	20
Betacab	DD	-	25
Canutility	AA	8	15
Canutility	CC	-	19



*De functie Min() moet een niet-NULL-waarde retourneren uit de reeks waarden die worden gegeven door de uitdrukking, mits die er is. In de voorbeelden retourneert de functie dus, omdat er NULL-waarden in de gegevens voorkomen, de eerste niet-NULL-waarde die wordt geëvalueerd op basis van de uitdrukking.*

### Voorbeelden en resultaten

Voorbeelden	Resultaten
Min(UnitSales)	2, omdat dit de laagste niet-NULL-waarde in unitSales is.

Voorbeelden	Resultaten
De waarde van een bestelling wordt berekend op basis van het aantal verkochte eenheden in (UnitSales) vermenigvuldigd met de prijs per eenheid.  Min(UnitSales*UnitPrice)	40, omdat dit de laagste niet-NULL-waarde is in het resultaat van de berekening van alle mogelijke waarden van (UnitSales)*(UnitPrice).
Min(UnitSales, 2)	4, omdat dit de op twee na laagste waarde is (na de NULL-waarden).
Min(TOTAL UnitSales)	2, omdat de kwalificatie TOTAL betekent dat de laagst mogelijke waarde wordt gevonden, waarbij de diagramdimensies buiten beschouwing worden gelaten. Voor een diagram met Customer als dimensie zorgt de kwalificatie TOTAL ervoor dat de minimale waarde in de volledige gegevensverzameling wordt geretourneerd, in plaats van de minimale UnitSales voor elke klant.
Voer de selectie Customer B uit.  Min({1} TOTAL UnitSales)	2, onafhankelijk van de selectie van Customer B.  De Set Analysis-uitdrukking {1} definieert de set records die moeten worden geëvalueerd als ALL, ongeacht welke selectie wordt gemaakt.

Gegevens die worden gebruikt in voorbeelden:

```
ProductData:
LOAD * inline [
Customer|Product|UnitSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD||25
Canutility|AA|8|15
Canutility|CC||19
] (delimiter is '|');
```

### Zie ook:

p [FirstSortedValue - diagramfunctie \(page 330\)](#)  
p [RangeMin \(page 1342\)](#)

## Mode

**Mode()** retourneert de vaakst voorkomende waarde, de moduswaarde, van de geaggregeerde gegevens in de uitdrukking, zoals gedefinieerd door een **group by**-clausule. De functie **Mode()** kan zowel numerieke waarden als tekstwaarden retourneren.

### Syntaxis:

**Mode** ( *expr* )

**Retourgegevenstypen:** dual

#### Argumenten

Argument	Beschrijving
expr Expression	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

### Beperkingen:

Als meerdere waarden even vaak voorkomen, is het resultaat NULL.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

#### Scriptvoorbeelden

Voorbeeld	Resultaat
<pre>Temp: LOAD * inline [ Customer Product OrderNumber UnitsSales CustomerID Astrida AA 1 10 1 Astrida AA 7 18 1 Astrida BB 4 9 1 Astrida CC 6 2 1 Betacab AA 5 4 2 Betacab BB 2 5 2 Betacab DD Canutility DD 3 8 Canutility CC ] (delimiter is ' ');  Mode: LOAD Customer, Mode(Product) as MyMostOftenSoldProduct Resident Temp Group By Customer;</pre>	<p>MyMostOftenSoldProduct</p> <p>AA</p> <p>omdat AA het enige product is dat vaker dan één keer is verkocht.</p>

### Mode - diagramfunctie

**Mode()** retourneert de vaakst voorkomende waarde, de modus, in de geaggregeerde gegevens. De functie **Mode()** kan zowel tekstwaarden als numerieke waarden verwerken.

### Syntaxis:

```
Mode ([SetExpression] [TOTAL [<fld {,fld}>]]) expr)
```

Retourgegevenstypen: dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregereert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {,fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Voorbeelden en resultaten:

#### Gegevens

Customer	Product	UnitSales	UnitPrice
Astrida	AA	4	16
Astrida	AA	10	15
Astrida	BB	9	9
Betacab	BB	5	10
Betacab	CC	2	20
Betacab	DD	-	25
Canutility	AA	8	15
Canutility	CC	-	19

### Voorbeelden en resultaten

Voorbeelden	Resultaten
Mode(UnitPrice) Voer de selectie Customer A uit.	15, omdat dit de meestvoorkomende waarde in unitSales is.  Retourneert NULL (-). Geen enkele waarde komt vaker voor dan de andere.
Mode(Product) Voer de selectie Customer A uit.	AA, omdat dit de meest voorkomende waarde in Product is.  Retourneert NULL (-). Geen enkele waarde komt vaker voor dan de andere.
Mode (TOTAL UnitPrice)	15, omdat de kwalificatie TOTAL betekent dat de meest voorkomende waarde nog steeds 15 is, ook als de diagramdimensies buiten beschouwing worden gelaten.
Voer de selectie Customer B uit.  Mode({1} TOTAL UnitPrice)	15, ongeacht de uitgevoerde selectie, omdat de Set Analysis-uitdrukking {1} de set records definieert die moeten worden geëvalueerd als ALL, ongeacht welke selectie wordt gemaakt.

Gegevens die worden gebruikt in voorbeelden:

```

ProductData:
LOAD * inline [
Customer|Product|UnitsSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD||25
Canutility|AA|8|15
Canutility|CC||19
] (delimiter is '|');
  
```

### Zie ook:

[p Avg - diagramfunctie \(page 396\)](#)  
[p Median - diagramfunctie \(page 436\)](#)

### Only

**Only()** retourneert een waarde als er één, en niet meer dan één, mogelijk resultaat is voor de geaggregeerde gegevens. Als records slechts één waarde bevatten, wordt die waarde geretourneerd, anders wordt NULL geretourneerd. Gebruik de **group by**-clausule om meerdere records te evalueren. De functie **Only()** kan numerieke waarden en tekstwaarden retourneren.

### Syntaxis:

```
Only ( expr )
```

**Retourgegevenstypen:** dual

### Argumenten

Argument	Beschrijving
expr Expression	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

Temp:

```
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|CustomerID
Astrida|AA|1|10|1
Astrida|AA|7|18|1
Astrida|BB|4|9|1
Astrida|CC|6|2|1
Betacab|AA|5|4|2
Betacab|BB|2|5|2
Betacab|DD
Canutility|DD|3|8
Canutility|CC
] (delimiter is '|');
only:
LOAD Customer, Only(CustomerID) as MyUniqIDCheck Resident Temp Group By Customer;
```

### Resultaattabel

Customer	MyUniqIDCheck
Astrida	1 omdat alleen klant Astrida volledige records heeft die CustomerID bevatten.

### Only - diagramfunctie

**Only()** retourneert een waarde als er één, en niet meer dan één, mogelijk resultaat is voor de geaggregeerde gegevens. Als u bijvoorbeeld zoekt naar het enige product waarvan de prijs per eenheid =9 en er zijn meerdere producten met een prijs per eenheid van 9, dan wordt het resultaat NULL geretourneerd.

**Syntaxis:**

```
Only ([{SetExpression}] [TOTAL [<fld {,fld}>]] expr)
```

**Retourgegevenstypen:** dual

**Argumenten:**

### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {.fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>



*Gebruik Only() als u een NULL-resultaat wilt als er meerdere mogelijke waarden zijn in de steekproefgegevens.*

**Voorbeelden en resultaten:**

### Gegevens

Customer	Product	UnitSales	UnitPrice
Astrida	AA	4	16
Astrida	AA	10	15
Astrida	BB	9	9
Betacab	BB	5	10
Betacab	CC	2	20
Betacab	DD	-	25
Canutility	AA	8	15
Canutility	CC	-	19



### Voorbeelden en resultaten

Voorbeelden	Resultaten
<code>only({&lt;UnitPrice={9}&gt;} Product)</code>	BB, omdat dit het enige Product is dat een unitPrice van '9' heeft.
<code>only({&lt;Product={DD}&gt;} Customer)</code>	Betacab, omdat dit de enige customer is die een Product verkoopt met de naam 'DD'.
<code>only({&lt;UnitPrice={20}&gt;} UnitsSales)</code>	Het aantal unitsales waarbij unitPrice gelijk is aan 20 bedraagt 2, omdat er slechts één waarde van unitsales is waarbij de unitPrice gelijk is aan 20.
<code>only({&lt;UnitPrice={15}&gt;} UnitsSales)</code>	NULL, omdat er twee waarden van unitsales zijn waarbij de unitPrice gelijk is aan 15.

Gegevens die worden gebruikt in voorbeelden:

```
ProductData:
LOAD * inline [
Customer|Product|UnitsSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD||25
Canutility|AA|8|15
Canutility|CC||19
] (delimiter is '|');
```

### Sum

**Sum()** berekent het totaal van de waarden die zijn geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

#### Syntaxis:

```
sum ( [ distinct] expr)
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

#### Argumenten

Argument	Beschrijving
distinct	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.
expr Expression	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

#### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|CustomerID
Astrida|AA|1|10|1
Astrida|AA|7|18|1
Astrida|BB|4|9|1
Astrida|CC|6|2|1
Betacab|AA|5|4|2
Betacab|BB|2|5|2
Betacab|DD
Canutility|DD|3|8
Canutility|CC
] (delimiter is '|');
Sum:
LOAD Customer, Sum(UnitSales) as MySum Resident Temp Group By Customer;
```

Resultaattabel

Customer	MySum
Astrida	39
Betacab	9
Canutility	8

### Sum - diagramfunctie

**Sum()** berekent het totaal van de waarden die worden geleverd door de uitdrukking of het veld voor de geaggregeerde gegevens.

#### Syntaxis:


```
Sum ([{SetExpression}] [DISTINCT] [TOTAL [<fld {,fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.

Argument	Beschrijving
DISTINCT	<p>Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> <i>Hoewel de kwalificatie <b>DISTINCT</b> wordt ondersteund, moet u heel zorgvuldig zijn bij het gebruiken hiervan omdat de lezer ten onrechte het idee kan krijgen dat een totale waarde wordt weergegeven als sommige gegevens zijn weggelaten.</i></p> </div>
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

**Voorbeelden en resultaten:**

## Gegevens

Customer	Product	UnitSales	UnitPrice
Astrida	AA	4	16
Astrida	AA	10	15
Astrida	BB	9	9
Betacab	BB	5	10
Betacab	CC	2	20
Betacab	DD	-	25
Canutility	AA	8	15
Canutility	CC	-	19

## Voorbeelden en resultaten

Voorbeelden	Resultaten
Sum(UnitSales)	38. Het totaal van de waarden in unitSales.
Sum(UnitSales*UnitPrice)	505. Het totaal van unitPrice vermenigvuldigd met unitSales geaggregeerd.

Voorbeelden	Resultaten
Sum (TOTAL UnitsSales*UnitPrice)	505 voor alle rijen in de tabel alsmede het totaal, omdat de kwalificatie TOTAL betekent dat de som nog steeds 505 is, ongeacht de diagramdimensies.
Voer de selectie customer B uit.  Sum({1} TOTAL UnitsSales*UnitPrice)	15, ongeacht de uitgevoerde selectie, omdat de Set Analysis-uitdrukking {1} de set records definieert die moeten worden geëvalueerd als ALL, ongeacht welke selectie wordt gemaakt.

Gegevens die worden gebruikt in voorbeelden:

```
ProductData:
LOAD * inline [
Customer|Product|UnitsSales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD||25
Canutility|AA|8|15
Canutility|CC||19
] (delimiter is '|');
```

### Teller-aggregatiefuncties

Teller-aggregatiefuncties retourneren verschillende typen tellers van een uitdrukking via een aantal records in een script voor het laden van gegevens of een aantal waarden in een diagramdimensie.

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Teller-aggregatiefuncties in het script voor het laden van gegevens

##### Count

**Count()** retourneert het aantal waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

```
Count ([distinct ] expression | * )
```

##### MissingCount

**MissingCount()** retourneert het aantal ontbrekende waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

```
MissingCount ([ distinct ] expression)
```

##### NullCount

**NullCount()** retourneert het aantal NULL-waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

```
NullCount ([ distinct ] expression)
```

### NumericCount

**NumericCount()** retourneert het aantal numerieke waarden dat is gevonden in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

```
NumericCount ([ distinct ] expression)
```

### TextCount

**TextCount()** retourneert het aantal niet-numerieke veldwaarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

```
TextCount ([ distinct ] expression)
```

## Teller-aggregatiefuncties in diagramuitdrukkingen

De volgende teller-aggregatiefuncties kunnen worden gebruikt in diagrammen:

### Count

**Count()** wordt gebruikt om het aantal waarden, tekst en numeriek, in elke diagramdimensie te aggregeren.

```
Count - diagramfunctie ({ [SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] }  
expr)
```

### MissingCount

**MissingCount()** wordt gebruikt om het aantal ontbrekende waarde in elke diagramdimensie te aggregeren. Ontbrekende waarden zijn alle niet-numerieke waarden.

```
MissingCount - diagramfunctie ({ [SetExpression] [DISTINCT] [TOTAL [<fld  
{,fld}>]] } expr)
```

### NullCount

**NullCount()** wordt gebruikt om het aantal NULL-waarden in elke diagramdimensie te aggregeren.

```
NullCount - diagramfunctie ({ [SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] }  
expr)
```

### NumericCount

**NumericCount()** wordt gebruikt om het aantal numerieke waarden in elke diagramdimensie te aggregeren.

```
NumericCount - diagramfunctie ({ [SetExpression] [DISTINCT] [TOTAL [<fld  
{,fld}>]] } expr)
```

### TextCount

**TextCount()** wordt gebruikt om het aantal veldwaarden die niet-numeriek zijn in elke diagramdimensie te aggregeren.

```
TextCount - diagramfunctie ({ [SetExpression] [DISTINCT] [TOTAL [<fld  
{,fld}>]] } expr)
```

### Count

**Count()** retourneert het aantal waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

### Syntaxis:

```
Count( [distinct ] expr)
```

**Retourgegevenstypen:** geheel getal

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
distinct	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

#### Scriptvoorbeelden

Voorbeeld	Resultaat
<pre>Temp: LOAD * inline [ Customer Product OrderNumber UnitSales UnitPrice Astrida AA 1 4 16 Astrida AA 7 10 15 Astrida BB 4 9 9 Betacab CC 6 5 10 Betacab AA 5 2 20 Betacab BB 1 25  25 Canutility AA 3 8 15 Canutility CC   19 Divadip CC 2 4 16 Divadip DD 3 1 25 ] (delimiter is ' ');  Count1: LOAD Customer,Count(OrderNumber) as OrdersByCustomer Resident Temp Group By Customer;</pre>	<pre>Customer OrdersByCustomer Astrida 3 Betacab 3 Canutility 2 Divadip 2 Zolang de dimensie Customer wordt opgenomen in de tabel op het werkblad, anders is 3, 2 het resultaat voor OrdersByCustomer.</pre>
<p>Stel dat de tabel <b>Temp</b> is geladen zoals in het vorige voorbeeld:</p> <pre>LOAD Count(OrderNumber) as TotalOrderNumber Resident Temp;</pre>	<pre>TotalOrderNumber 10</pre>

Voorbeeld	Resultaat
<p>Stel dat de tabel <b>Temp</b> is geladen zoals in het eerste voorbeeld:</p> <pre>LOAD Count(distinct OrderNumber) as TotalOrderNumber Resident Temp;</pre>	<p>TotalOrderNumber 8</p> <p>Omdat er twee waarden van OrderNumber zijn die dezelfde waarde, 1, hebben en één null-waarde.</p>

### Count - diagramfunctie

**Count()** wordt gebruikt om het aantal waarden, tekst en numeriek, in elke diagramdimensie te aggregeren.

#### Syntaxis:

```
Count ([SetExpression] [DISTINCT] [TOTAL [<fld {, fld}>]]) expr)
```

**Retourgegevenstypen:** geheel getal

#### Argumenten:

##### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregereert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {, fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

#### Voorbeelden en resultaten:


##### Data

Customer	Product	OrderNumber	UnitSales	Unit Price
Astrida	AA	1	4	16
Astrida	AA	7	10	15

Customer	Product	OrderNumber	UnitSales	Unit Price
Astrida	BB	4	9	9
Betacab	BB	6	5	10
Betacab	CC	5	2	20
Betacab	DD	1	25	25
Canutility	AA	3	8	15
Canutility	CC			19
Divadip	AA	2	4	16
Divadip	DD	3		25

In de volgende voorbeelden wordt ervan uitgegaan dat alle klanten worden geselecteerd, behalve waar aangegeven.

### Voorbeelden en resultaten

Voorbeeld	Resultaat
Count(OrderNumber)	10, omdat er 10 velden zijn die een waarde voor OrderNumber zouden kunnen hebben en alle records, zelfs lege, worden geteld.  <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  "0" telt als waarde en niet als lege cel. Alleen als een meting aggregaert tot 0 voor een dimensie, wordt die dimensie niet opgenomen in diagrammen. </div>
Count(Customer)	10, omdat met Count het aantal exemplaren in alle velden wordt geëvalueerd.
Count(DISTINCT [Customer])	4, omdat bij gebruik van de kwalificatie Distinct alleen unieke exemplaren worden geëvalueerd met Count.
Stel dat klant Canutility is geselecteerd.  Count (OrderNumber)/Count({1} TOTAL OrderNumber)	0.2, omdat de uitdrukking het aantal orders van de geselecteerde klant retourneert als percentage van de orders van alle klanten. In dit geval 2/10.
Stel dat klanten Astrida en Canutility zijn geselecteerd  Count(TOTAL <Product> OrderNumber)	5, omdat dat het aantal orders is dat is geplaatst voor producten voor alleen de geselecteerde klanten en omdat lege cellen worden meegeteld.

Gegevens die worden gebruikt in voorbeelden:



```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB|1|25| 25
Canutility|AA|3|8|15
Canutility|CC|||19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
] (delimiter is '|');
```

### MissingCount

**MissingCount()** retourneert het aantal ontbrekende waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

#### Syntaxis:

```
MissingCount ( [ distinct ] expr)
```

**Retourgegevenstypen:** geheel getal

#### Argumenten:

##### Argumenten

Argument	Beschrijving
expr Expression	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
distinct	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.

#### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

### Scriptvoorbeelden

Voorbeeld	Resultaat
<pre>Temp: LOAD * inline [ Customer Product OrderNumber UnitSales UnitPrice Astrida AA 1 4 16 Astrida AA 7 10 15 Astrida BB 4 9 9 Betacab CC 6 5 10 Betacab AA 5 2 20 Betacab BB   25 Canutility AA   15 Canutility CC    19 Divadip CC 2 4 16 Divadip DD 3 1 25 ] (delimiter is ' '); MissCount1: LOAD Customer,MissingCount(OrderNumber) as MissingOrdersByCustomer Resident Temp Group By Customer;  Load MissingCount(OrderNumber) as TotalMissingCount Resident Temp;</pre>	<p>Customer MissingOrdersByCustomer Astrida 0 Betacab 1 Canutility 2 Divadip 0</p> <p>De tweede opdracht geeft als resultaat:</p> <p>TotalMissingCount 3 in een tabel met die dimensie.</p>
<p>Stel dat de tabel <b>Temp</b> is geladen zoals in het vorige voorbeeld:</p> <pre>LOAD MissingCount(distinct OrderNumber) as TotalMissingCountDistinct Resident Temp;</pre>	<p>TotalMissingCountDistinct 1 omdat er slechts één OrderNumber is, ontbreekt er één waarde.</p>

### MissingCount - diagramfunctie

**MissingCount()** wordt gebruikt om het aantal ontbrekende waarde in elke diagramdimensie te aggregeren. Ontbrekende waarden zijn alle niet-numerieke waarden.

#### Syntaxis:

```
MissingCount ({ [SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] } expr)
```

**Retourgegevenstypen:** geheel getal

#### Argumenten:

#### Argumenten


Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.

Argument	Beschrijving
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Voorbeelden en resultaten:

Data				
Customer	Product	OrderNumber	UnitSales	Unit Price
Astrida	AA	1	4	16
Astrida	AA	7	10	15
Astrida	BB	4	9	9
Betacab	BB	6	5	10
Betacab	CC	5	2	20
Betacab	DD			25
Canutility	AA			15
Canutility	CC			19
Divadip	AA	2	4	16
Divadip	DD	3		25

### Voorbeelden en resultaten

Voorbeeld	Resultaat
MissingCount([OrderNumber])	<p>3, omdat 3 van de 10 velden OrderNumber leeg zijn</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> "0" telt als waarde en niet als lege cel. Alleen als een meting aggregaert tot 0 voor een dimensie, wordt die dimensie niet opgenomen in diagrammen.</p> </div>

Voorbeeld	Resultaat
<pre>MissingCount ([OrderNumber])/MissingCount ({1} Total [OrderNumber])</pre>	<p>De uitdrukking retourneert het aantal onvolledige orders van de geselecteerde klant retourneert als fractie van de onvolledige orders van alle klanten. Er zijn in totaal 3 ontbrekende waarden voor OrderNumber voor alle klanten. Dus voor elke Customer die een ontbrekende waarde heeft voor Product bedraagt het resultaat 1/3.</p>

Gebruikte gegevens in voorbeeld:

```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB||| 25
Canutility|AA|||15
Canutility|CC| |19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
] (delimiter is '|');
```

### NullCount

**NullCount()** retourneert het aantal NULL-waarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een **group by**-clause.

**Syntaxis:**

```
NullCount ( [ distinct ] expr)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

#### Argumenten

Argument	Beschrijving
expr Expression	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
distinct	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

Scriptvoorbeelden

Voorbeeld	Resultaat
<pre>Set NULLINTERPRET = NULL; Temp: LOAD * inline [ Customer Product OrderNumber UnitSales CustomerID Astrida AA 1 10 1 Astrida AA 7 18 1 Astrida BB 4 9 1 Astrida CC 6 2 1 Betacab AA 5 4 2 Betacab BB 2 5 2 Betacab DD    Canutility AA 3 8  Canutility CC NULL   ] (delimiter is ' '); Set NULLINTERPRET=; NullCount1: LOAD Customer,NullCount(OrderNumber) as NullOrdersByCustomer Resident Temp Group By Customer;  LOAD NullCount(OrderNumber) as TotalNullCount Resident Temp;</pre>	<p>Customer NullOrdersByCustomer Astrida 0 Betacab 0 Canutility 1</p> <p>De tweede opdracht geeft als resultaat:</p> <p>TotalNullCount 1</p> <p>in een tabel met die dimensie, omdat slechts één record een null-waarde bevat.</p>

### NullCount - diagramfunctie

**NullCount()** wordt gebruikt om het aantal NULL-waarden in elke diagramdimensie te aggregeren.

#### Syntaxis:

```
NullCount ( { [SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] } expr )
```

**Retourgegevenstypen:** geheel getal

#### Argumenten:

Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
set_ expression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.

Argument	Beschrijving
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.

### Voorbeelden en resultaten:

#### Voorbeelden en resultaten

Voorbeeld	Resultaat
NullCount ([OrderNumber])	1, omdat we een null-waarde hebben geïntroduceerd met NullInterpret in de inline <b>LOAD</b> -opdracht.

### Gebruikte gegevens in voorbeeld:

```
Set NULLINTERPRET = NULL;
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|CustomerID
Astrida|AA|1|10|1
Astrida|AA|7|18|1
Astrida|BB|4|9|1
Astrida|CC|6|2|1
Betacab|AA|5|4|2
Betacab|BB|2|5|2
Betacab|DD|||
Canutility|AA|3|8|
Canutility|CC|NULL||
] (delimiter is '|');
Set NULLINTERPRET=;
```

### NumericCount

**NumericCount()** retourneert het aantal numerieke waarden dat is gevonden in de uitdrukking, zoals gedefinieerd door een **group by**-clausule.

#### Syntaxis:

```
NumericCount ( [ distinct ] expr)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

Argument	Beschrijving
expr Expression	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
distinct	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

Scriptvoorbeeld

Voorbeeld	Resultaat
<pre>LOAD NumericCount(OrderNumber) as TotalNumericCount Resident Temp;</pre>	<p>De tweede opdracht geeft als resultaat:</p> <p>TotalNumericCount</p> <p>7</p> <p>in een tabel met die dimensie.</p>
<p>Stel dat de tabel <b>Temp</b> is geladen zoals in het vorige voorbeeld:</p> <pre>LOAD NumericCount(distinct OrderNumber) as TotalNumericCountDistinct Resident Temp;</pre>	<p>TotalNumericCountDistinct</p> <p>6</p> <p>6omdat er één is die een andere dupliceert, dus is het resultaat 6 niet-duplicaten.OrderNumber</p>

**Voorbeeld:**

```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB||| 25
Canutility|AA|||15
Canutility|CC| |19
Divadip|CC|2|4|16
Divadip|DD|7|1|25
] (delimiter is '|');
NumCount1:
```

```
LOAD Customer, NumericCount(OrderNumber) as NumericCountByCustomer Resident Temp Group By Customer;
```

Resultaattabel

Customer	NumericCountByCustomer
Astrida	3
Betacab	2
Canutility	0
Divadip:	2

### NumericCount - diagramfunctie

**NumericCount()** wordt gebruikt om het aantal numerieke waarden in elke diagramdimensie te aggregeren.

#### Syntaxis:

```
NumericCount ({ [SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]] } expr)
```

**Retourgegevenstypen:** geheel getal

#### Argumenten:

Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
set_ expression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {,fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.




### Voorbeelden en resultaten:

Data				
Customer	Product	OrderNumber	UnitSales	Unit Price
Astrida	AA	1	4	16
Astrida	AA	7	10	15
Astrida	BB	4	9	1
Betacab	BB	6	5	10
Betacab	CC	5	2	20
Betacab	DD			25
Canutility	AA			15
Canutility	CC			19
Divadip	AA	2	4	16
Divadip	DD	3		25

In de volgende voorbeelden wordt ervan uitgegaan dat alle klanten worden geselecteerd, behalve waar aangegeven.

### Voorbeelden en resultaten

Voorbeeld	Resultaat
NumericCount ([OrderNumber])	7, omdat drie van de 10 velden in OrderNumber leeg zijn.  <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>"0" telt als waarde en niet als lege cel. Alleen als een meting aggregaat tot 0 voor een dimensie, wordt die dimensie niet opgenomen in diagrammen.</i> </div>
NumericCount ([Product])	0, omdat alle productnamen in tekstvorm zijn. Normaal gesproken zou u hiermee kunnen controleren of er geen tekstvelden zijn met numerieke inhoud.
NumericCount (DISTINCT [OrderNumber])/Count (DISTINCT [OrderNumber])	Hiermee wordt het aantal verschillende numerieke bestelnummers geteld en gedeeld door het aantal numerieke en niet-numerieke bestelnummers. Dit wordt 1 als alle veldwaarden numeriek zijn. Normaal gesproken zou u hiermee kunnen controleren of alle veldwaarden numeriek zijn. In het voorbeeld zijn er 7 distinctieve numerieke waarden voor OrderNumber van 8 distinctieve numerieke en niet-numerieke waarden, zodat de uitdrukking 0.875 retourneert.

Gebruikte gegevens in voorbeeld:

```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB||| 25
Canutility|AA|||15
Canutility|CC| |19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
] (delimiter is '|');
```

### TextCount

**TextCount()** retourneert het aantal niet-numerieke veldwaarden dat is geaggregeerd in de uitdrukking, zoals gedefinieerd door een group by-clausule.group by

**Syntaxis:**

```
TextCount ( [ distinct ] expr)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

Argument	Beschrijving
expr Expression	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
distinct	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

**Voorbeeld:**

```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
```

```

Betacab|AA|5|2|20
Betacab|BB||| 25
Canutility|AA|||15
Canutility|CC| |19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
] (delimiter is '|');
TextCount1:
LOAD Customer,TextCount(Product) as ProductTextCount Resident Temp Group By Customer;

```

Resultaattabel	
Customer	ProductTextCount
Astrida	3
Betacab	3
Canutility	2
Divadip:	2

### Voorbeeld:

```

LOAD Customer,TextCount(OrderNumber) as OrderNumberTextCount Resident Temp Group By Customer;

```

Resultaattabel	
Customer	OrderNumberTextCount
Astrida	0
Betacab	1
Canutility	2
Divadip:	0

### TextCount - diagramfunctie

**TextCount()** wordt gebruikt om het aantal veldwaarden die niet-numeriek zijn in elke diagramdimensie te aggregeren.

#### Syntaxis:

```
TextCount ([SetExpression] [DISTINCT] [TOTAL [<fld {,fld}>]]) expr
```

**Retourgegevenstypen:** geheel getal

#### Argumenten:

Argumenten	
Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

Argument	Beschrijving
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Voorbeelden en resultaten:

Data				
Customer	Product	OrderNumber	UnitSales	Unit Price
Astrida	AA	1	4	16
Astrida	AA	7	10	15
Astrida	BB	4	9	1
Betacab	BB	6	5	10
Betacab	CC	5	2	20
Betacab	DD			25
Canutility	AA			15
Canutility	CC			19
Divadip	AA	2	4	16
Divadip	DD	3		25

### Voorbeelden en resultaten

Voorbeeld	Resultaat
TextCount ([Product])	10, omdat alle 10 velden in Product tekst bevatten.  <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>Lege cellen worden geëvalueerd als niet-tekst en worden niet geteld door ."0" telt als waarde en niet als lege cel. Alleen als een meting aggregereert tot 0 voor een dimensie, wordt die dimensie niet opgenomen in diagrammen. TextCount</i> </div>
TextCount ([OrderNumber])	3, omdat lege cellen worden meegeteld. Normaal gesproken zou u hiermee controleren of er geen numerieke velden zijn die tekst bevatten of niet gelijk zijn aan nul.
TextCount (DISTINCT [Product])/Count ([Product])	Hiermee wordt het aantal distinctieve tekstwaarden van Product (4) opgeteld en gedeeld door het totale aantal waarden in Product (10). Het resultaat is 0,4.

Gebruikte gegevens in voorbeeld:

```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitsSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|1|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB||| 25
Canutility|AA|||15
Canutility|CC|||19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
] (delimiter is '|');
```

## Financiële aggregatiefuncties

In dit hoofdstuk worden aggregatiefuncties voor financiële activiteiten beschreven met betrekking tot betalingen en cashflow.

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Financiële aggregatiefuncties in het script voor het laden van gegevens

#### IRR

**IRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een reeks cashflows die wordt vertegenwoordigd door de getallen in een uitdrukking, herhaald voor een aantal records dat is vastgelegd door een group by-clausule.

```
IRR (expression)
```

#### XIRR

**XIRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een schema van cashflows (niet noodzakelijk periodiek) die worden gerepresenteerd door getallenparen in **pmt** en **date**, die worden herhaald over een aantal records zoals vastgelegd door een group by-clausule. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

```
XIRR (valueexpression, dateexpression )
```

#### NPV

De scriptfunctie **NPV()** neemt een kortingstarief en meerdere waarden op volgorde van periode. Ontvangsten, (inkomsten) zijn positief en uitgaven (toekomstige betalingen) worden voor deze berekeningen als negatieve waarden beschouwd. Deze vinden plaats aan het einde van elke periode.

```
NPV (rate, expression)
```

#### XNPV

De scriptfunctie **XNPV()** neemt specifieke datums die overeenkomen met elke kasstroom waarbij een andere korting dan het kortingstarief wordt toegepast. Deze verschilt van de functie **NPV()** aangezien **NPV()** alle tijdsperiodes als gelijkwaardig beschouwt. Om deze reden is **XNPV()** preciezer dan **NPV()**.

```
XNPV (rate, valueexpression, dateexpression)
```

### Financiële aggregatiefuncties in diagramuitdrukkingen

Deze financiële aggregatiefuncties kunnen worden gebruikt in diagrammen.

#### IRR

**IRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een reeks cashflows die wordt vertegenwoordigd door de getallen in een uitdrukking die wordt geleverd door **value**, herhaald voor de diagramdimensies.

```
IRR - diagramfunctie [TOTAL [<fld {,fld}>]] value)
```

#### NPV

**NPV()** retourneert de geaggregeerde huidige nettowaarde van een investering gebaseerd op een **discount\_rate** per periode en een reeks toekomstige betalingen (negatieve waarden) en inkomsten (positieve waarden) vertegenwoordigd door de getallen in **value**, herhaald op de dimensies van het diagram. We gaan er vanuit dat de betalingen en inkomsten telkens aan het eind van elke periode worden verrekend.

```
NPV - diagramfunctie ([TOTAL [<fld {,fld}>]] discount_rate, value)
```

### XIRR

**XIRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een schema van cashflows (niet noodzakelijk periodiek) dat wordt vertegenwoordigd door getallenparen in de uitdrukkingen **pmt** en **date**, herhaald op de dimensies van het diagram. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

```
XIRR - diagramfunctie (page 379) ([TOTAL [<fld {,fld}>]] pmt, date)
```

### XNPV

**XNPV()** retourneert de geaggregeerde interne opbrengstwaarde voor een schema van cashflows (niet noodzakelijk periodiek) dat wordt vertegenwoordigd door getallenparen in de uitdrukkingen **pmt** en **date**, herhaald op de dimensies van het diagram. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

```
XNPV - diagramfunctie ([TOTAL [<fld {,fld}>]] discount_rate, pmt, date)
```

### IRR

**IRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een reeks cashflows die wordt vertegenwoordigd door de getallen in een uitdrukking, herhaald voor een aantal records dat is vastgelegd door een group by-clausule.

Deze cashflows hoeven niet even groot te zijn zoals voor een annuïteit. De cashflows moeten echter met regelmatige intervallen plaatsvinden, zoals maandelijks of jaarlijks. Het interne opbrengstpercentage is het ontvangen rentepercentage voor een investering die bestaat uit betalingen (negatieve waarden) en inkomsten (positieve waarden) die periodiek plaatsvinden. Voor de functie is minimaal één positieve en één negatieve waarde vereist voor de berekening.

#### Syntaxis:

```
IRR (value)
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
value	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

#### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

#### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

### Voorbeelden en resultaten:

Voorbeelden en resultaten

Voorbeeld	Jaar	IRR2013
<pre>Cashflow: LOAD 2013 as Year, * inline [ Date Discount Payments 2013-01-01 0.1 -10000 2013-03-01 0.1 3000 2013-10-30 0.1 4200 2014-02-01 0.2 6800 ] (delimiter is ' ');  Cashflow1: LOAD Year,IRR(Payments) as IRR2013 Resident Cashflow Group By Year;</pre>	2013	0.1634

### IRR - diagramfunctie

**IRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een reeks cashflows die wordt vertegenwoordigd door de getallen in een uitdrukking die wordt geleverd door **value**, herhaald voor de diagramdimensies.

Deze cashflows hoeven niet even groot te zijn zoals voor een annuïteit. De cashflows moeten echter met regelmatige intervallen plaatsvinden, zoals maandelijks of jaarlijks. De interne rentabiliteit is het rentetarief dat wordt ontvangen voor een investering die bestaat uit betalingen (negatieve waarden) en inkomsten (positieve waarden) die zich in regelmatige perioden voordoen. De functie heeft ten minste een positieve en een negatieve waarde nodig om te worden berekend.

#### Syntaxis:

```
IRR ([TOTAL [<fld {,fld}>]] value)
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

Argumenten

Argument	Beschrijving
value	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {,fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>




### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

### Voorbeelden en resultaten:

#### Voorbeelden en resultaten

Voorbeeld	Resultaat
IRR (Payments)	0.1634  De betalingen worden geacht periodiek van aard te zijn, bijvoorbeeld maandelijks.  <div style="border: 1px solid gray; padding: 5px;"> <i>Het veld Datum wordt gebruikt in het XIRR-voorbeeld waarbij betalingen niet-periodiek kunnen zijn, mits u de datums opgeeft waarop de betalingen werden uitgevoerd.</i></div>

Gegevens die worden gebruikt in voorbeelden:

Cashflow:

```
LOAD 2013 as Year, * inline [
Date|Discount|Payments
2013-01-01|0.1|-10000
2013-03-01|0.1|3000
2013-10-30|0.1|4200
2014-02-01|0.2|6800
] (delimiter is '|');
```

### Zie ook:

p *XIRR - diagramfunctie (page 379)*

p *Aggr - diagramfunctie (page 543)*

### NPV

De scriptfunctie **NPV()** neemt een kortingstarief en meerdere waarden op volgorde van periode. Ontvangsten, (inkomsten) zijn positief en uitgaven (toekomstige betalingen) worden voor deze berekeningen als negatieve waarden beschouwd. Deze vinden plaats aan het einde van elke periode.

De huidige nettowaarde of Net Present Value/ NPV, wordt gebruikt om de huidige totale waarde van een toekomstige stream van cashflows te berekenen. Om de NPV te berekenen, moeten we de toekomstige cashflows per periode berekenen en het correcte kortingstarief bepalen. De scriptfunctie **NPV()** neemt een

kortingstarief en meerdere waarden op volgorde van periode. Ontvangsten, (inkomsten) zijn positief en uitgaven (toekomstige betalingen) worden voor deze berekeningen als negatieve waarden beschouwd. Deze vinden plaats aan het einde van elke periode.

### Syntaxis:

```
NPV(discount_rate, value)
```

**Retourgegevenstypen:** numeriek. Het resultaat wordt standaard geformatteerd als een valuta.

De formule om de huidige nettowaarde te berekenen is:

$$NPV = \sum_{t=1}^n \frac{R_t}{(1+i)^t}$$

waarbij:

- $R_t$  = Netto-ontvangsten/-uitgaven in cash gedurende één periode  $t$
- $i$  = Kortingspercentage dat of opbrengst die zou kunnen worden verdiend uit alternatieve investeringen
- $t$  = Aantal timerperioden

### Argumenten

Argument	Beschrijving
discount_rate	<b>discount_rate</b> is het percentagetarief van de toegepaste korting. Een waarde van 0,1 zou een kortingspercentage van 10% aangeven.
value	Dit veld bevat waarden voor meerdere perioden op volgorde van periode. De eerste waarde wordt aangenomen als de cashflow aan het eind van periode 1, enzovoort.

### Beperkingen:

De NPV() -functie heeft de volgende beperkingen:

- Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.
- Cashflowwaarden moeten op volgorde van oplopende periode staan.

### Wanneer gebruiken

NPV() is een financiële functie die wordt gebruikt om de winstgevendheid van een project te controleren en andere metingen af te leiden. Deze functie is handig wanneer cashflows beschikbaar zijn in de vorm van onbewerkte gegevens.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht SET DateFormat in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege

landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Enkele betaling (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset van één project en de daarbij behorende cashflow voor één periode die in de tabel `CashFlow` wordt geladen.
- Een resident-load uit de tabel `CashFlow` die wordt gebruikt om het NPV-veld te berekenen voor het project in de tabel `NPV`.
- Een vastgesteld kortingspercentage van 10%, dat wordt gebruikt in de berekening van de NPV.
- De opdracht `Group By` die wordt gebruikt om alle betalingen voor het project te groeperen.

#### Load-script

```
CashFlow:
Load
*
Inline
[
PrjId,PeriodId,Values
1,1,1000
];

NPV:
Load
    PrjId,
    NPV(0.1,Values) as NPV //Discount Rate of 10%
Resident CashFlow
Group By PrjId;
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- NPV

Resultatentabel

PrjId	NPV
1	\$909.09

Om aan het eind van één periode een enkele betaling van \$1000 te kunnen ontvangen met een kortingspercentage van 10% per periode, is de NPV gelijk aan \$1000 gedeeld door (1 + kortingspercentage). De effectieve NPV is gelijk aan \$909,09

### Voorbeeld 2 - Meerdere betalingen (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset van één project en de daarbij behorende cashflow voor meerdere perioden die in de tabel `cashFlow` wordt geladen.
- Een resident-load uit de tabel `cashFlow` die wordt gebruikt om het NPV-veld te berekenen voor het project in de tabel `NPV`.
- Een vastgesteld kortingspercentage van 10% (0,1), dat wordt gebruikt in de berekening van de NPV.
- De opdracht `Group By` die wordt gebruikt om alle betalingen voor het project te groeperen.

#### Load-script

CashFlow:

Load

\*

Inline

[

PrjId,PeriodId,Values

1,1,1000

1,2,1000

];

NPV:

Load

PrjId,

NPV(0.1,Values) as NPV //Discount Rate of 10%

Resident CashFlow

Group By PrjId;

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- NPV

Resultatentabel

PrjId	NPV
1	\$1735.54

Om aan het eind van twee perioden betalingen van \$1000 te kunnen ontvangen met een kortingspercentage van 10% per periode, is de effectieve NPV gelijk aan \$1735,54.

### Voorbeeld 3 - Meerdere betalingen (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Kortingsstarieven voor twee projecten die in de tabel `Project` worden geladen.
- Cashflows voor meerdere perioden per project op project-id en periode-id. Deze periode-id kan worden gebruikt om de records te sorteren in het geval dat de gegevens niet zijn gesorteerd.
- De combinatie van `Noconcatenate`, `resident-loads` en de `Left Join`-functie om een tijdelijke tabel aan te maken, `tmpNPV`. De tabel combineert de records uit de tabellen `Project` en `CashFlow` in één tabel. In deze tabel worden de kortingsstarieven voor iedere periode herhaald.
- Een `resident-load` uit de tabel `tmpNPV` die wordt gebruikt om het NPV-veld te berekenen voor elk project in de tabel `NPV`.
- Het kortingspercentage voor één waarde dat aan elk project is gekoppeld. Dit wordt opgehaald met de `only()`-functie en wordt gebruikt in de NPV-berekening per project.
- De `Group By`-opdracht, die wordt gebruikt om alle betalingen per project te groeperen op project-id.

Om te vermijden dat synthetische of overbodige gegevens in het gegevensmodel worden geladen, wordt de tabel `tmpNPV` aan het einde van het script verwijderd.

#### Load-script

```
Project:
Load * inline [
PrjId,Discount_Rate
1,0.1
2,0.15
];
```

CashFlow:

Load

\*

Inline

[

PrjId,PeriodId,Values

1,1,1000

1,2,1000

1,3,1000

2,1,500

2,2,500

2,3,1000

2,4,1000

];

tmpNPV:

NoConcatenate Load \*

Resident Project;

Left Join

Load \*

Resident CashFlow;

NPV:

Load

PrjId,

NPV(Only(Discount\_Rate),Values) as NPV //Discount Rate will be 10% for Project 1 and 15% for

Project 2

Resident tmpNPV

Group By PrjId;

Drop table tmpNPV;

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- NPV

Resultatentabel

PrjId	NPV
1	\$2486.85
2	\$2042.12

Project-id 1 verwacht dat betalingen van \$1000 aan het einde van drie perioden worden ontvangen met een kortingspercentage van 10% per periode. Daarom is de effectieve NPV gelijk aan \$2486,85.

Project-id 2 verwacht twee betalingen van \$500 en nog eens betalingen van \$1000 in vier perioden met een kortingspercentage van 15%. Daarom is de effectieve NPV gelijk aan \$2042.12.

### Voorbeeld 4 - Voorbeeld van winstgevendheid van project (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Kortingstarieven en aanvankelijke investeringen (periode 0) voor twee projecten die in de tabel `Project` worden geladen.
- Cashflows voor meerdere perioden per project op project-id en periode-id. Deze periode-id kan worden gebruikt om de records te sorteren in het geval dat de gegevens niet zijn gesorteerd.
- De combinatie van `Noconcatenate`, `resident-loads` en de `Left Join`-functie om een tijdelijke tabel aan te maken, `tmpNPV`. De tabel combineert de records uit de tabellen `Project` en `CashFlow` in één tabel. In deze tabel worden de kortingstarieven voor iedere periode herhaald.
- Het kortingspercentage voor de enkele waarden dat bij elk project hoort, dat wordt opgehaald met de `only()`-functie en wordt gebruikt in de NPV-berekening per project.
- Een `resident-load` uit de tabel `tmpNPV` wordt gebruikt om het NPV-veld te berekenen voor elk project in de tabel `NPV`.
- Een extra veld dat de NPV verdeelt op basis van de aanvankelijke investering van elk project, wordt gemaakt om de winstgevendheidsindex van het project te berekenen.
- Een `Group By`-opdracht, die op project-id groepeerd, wordt gebruikt om alle betalingen voor elk project te groeperen.

Om te vermijden dat synthetische of overbodige gegevens in het gegevensmodel worden geladen, wordt de tabel `tmpNPV` aan het einde van het script verwijderd.

#### Load-script

```
Project:
Load * inline [
PrjId,Discount_Rate, Initial_Investment
1,0.1,100000
2,0.15,100000
];
```

```
CashFlow:
Load
*
Inline
[
PrjId,PeriodId,values,
1,1,35000
1,2,35000
1,3,35000
2,1,30000
2,2,40000
2,3,50000
```

```
2,4,60000
];

tmpNPV:
NoConcatenate Load *
Resident Project;
Left Join
Load *
Resident CashFlow;

NPV:
Load
    PrjId,
    NPV(Only(Discount_Rate),Values) as NPV, //Discount Rate will be 10% for Project 1 and
15% for Project 2
    NPV(Only(Discount_Rate),Values)/ Only(Initial_Investment) as Profitability_Index
Resident tmpNPV
Group By PrjId;

Drop table tmpNPV;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- NPV

Maak de volgende meting:

=only(Profitability\_Index)

Resultatentabel

PrjId	NPV	=only(Profitability_Index)
1	\$87039.82	0.87
2	\$123513.71	1.24

Project-id 1 heeft een effectieve NPV van \$87039,82 en een aanvankelijke investering van \$100.000. Daarom is de winstgevenheidsindex gelijk aan 0,87. Omdat het minder is dan 1, is het project niet winstgevend.

Project-id 2 heeft een effectieve NPV van \$123513,71 en een aanvankelijke investering van \$100.000. Daarom is de winstgevenheidsindex gelijk aan 1,24. Omdat het groter is dan 1, is het project winstgevend.

### NPV - diagramfunctie

**NPV()** retourneert de geaggregeerde huidige nettowaarde van een investering gebaseerd op een **discount\_rate** per periode en een reeks toekomstige betalingen (negatieve waarden) en inkomsten (positieve waarden) vertegenwoordigd door de getallen in **value**, herhaald op de dimensies van het diagram. We gaan er vanuit dat de betalingen en inkomsten telkens aan het eind van elke periode worden verrekend.



### Syntaxis:

```
NPV([TOTAL [<fld {,fld}>]] discount_rate, value)
```

**Retourgegevenstypen:** numeriek Het resultaat wordt standaard geformatteerd als een valuta.

### Argumenten:

#### Argumenten

Argument	Beschrijving
discount_rate	<b>discount_rate</b> is the rate of discount over the length of the period. <b>discount_rate</b> is het percentagetarief van de toegepaste korting.
value	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {,fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p> <p>Na de kwalificatie <b>TOTAL</b> kan een lijst met een of meer veldnamen tussen punthaken worden opgegeven. Deze veldnamen moeten een subset zijn van de dimensievariabelen van het diagram. In dit geval wordt bij de berekening geen rekening gehouden met de dimensievariabelen van het diagram met uitzondering van de aangegeven dimensievariabelen. Er wordt één waarde geretourneerd voor elke combinatie van veldwaarden in de opgegeven dimensievelden. Ook velden die momenteel niet een dimensie zijn in een diagram, kunnen worden opgenomen in de lijst. Dit kan nuttig zijn bij groepsdimensies waarbij de dimensievelden niet vast zijn. Als alle variabelen in de groep worden aangegeven, werkt de functie als het drill-downniveau verandert.</p>

### Beperkingen:

**discount\_rate** en **value** mogen geen aggregatiefuncties bevatten, tenzij deze interne aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

### Voorbeelden en resultaten:

#### Voorbeelden en resultaten

Voorbeeld	Resultaat
NPV(Discount, Payments)	-\$540.12

Gegevens die worden gebruikt in voorbeelden:

```
Cashflow:
LOAD 2013 as Year, * inline [
Date|Discount|Payments
2013-01-01|0.1|-10000
2013-03-01|0.1|3000
2013-10-30|0.1|4200
2014-02-01|0.2|6800
] (delimiter is '|');
```

**Zie ook:**

p *XNPV - diagramfunctie (page 386)*

p *Aggr - diagramfunctie (page 543)*

### XIRR

**XIRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een schema van cashflows (niet noodzakelijk periodiek) die worden gerepresenteerd door getallenparen in **pmt** en **date**, die worden herhaald over een aantal records zoals vastgelegd door een group by-clausule. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

**Syntaxis:**

```
XIRR (pmt, date )
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
pmt	Betalingen. De uitdrukking of het veld met de cashflows die horen bij het betalingenschema dat is gegeven in <b>date</b> .
date	De uitdrukking of het veld met het datumschema dat hoort bij de cashflowbetalingen die zijn gegeven in <b>pmt</b> .

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden NULL-waarden of ontbrekende waarden bevat, wordt het hele gegevenspaar genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

### Voorbeelden en resultaten

Voorbeeld	Jaar	XIRR2013
<pre>Cashflow: LOAD 2013 as Year, * inline [ Date Discount Payments 2013-01-01 0.1 -10000 2013-03-01 0.1 3000 2013-10-30 0.1 4200 2014-02-01 0.2 6800 ] (delimiter is ' ');  Cashflow1: LOAD Year,XIRR(Payments, Date) as XIRR2013 Resident Cashflow Group By Year;</pre>	2013	0.5385

### XIRR - diagramfunctie

**XIRR()** retourneert het geaggregeerde interne opbrengstpercentage voor een schema van cashflows (niet noodzakelijk periodiek) dat wordt vertegenwoordigd door getallenparen in de uitdrukkingen **pmt** en **date**, herhaald op de dimensies van het diagram. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

#### Syntaxis:

```
XIRR([TOTAL [<fld {,fld}>]] pmt, date)
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

#### Argumenten

Argument	Beschrijving
pmt	Betalingen. De uitdrukking of het veld met de cashflows die horen bij het betalingschema dat is gegeven in <b>date</b> .
date	De uitdrukking of het veld met het datumschema dat hoort bij de cashflowbetalingen die zijn gegeven in <b>pmt</b> .
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {,fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Beperkingen:

**pmt** en **date** mogen geen aggregatiefuncties bevatten, tenzij deze interne aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Voorbeelden en resultaten:

Voorbeelden en resultaten

Voorbeeld	Resultaat
XIRR(Payments, Date)	0.5385

Gegevens die worden gebruikt in voorbeelden:

```
Cashflow:  
LOAD 2013 as Year, * inline [  
Date|Discount|Payments  
2013-01-01|0.1|-10000  
2013-03-01|0.1|3000  
2013-10-30|0.1|4200  
2014-02-01|0.2|6800  
] (delimiter is '|');
```

---

### Zie ook:

p *IRR - diagramfunctie (page 368)*

p *Aggr - diagramfunctie (page 543)*

### XNPV

De scriptfunctie **XNPV()** neemt specifieke datums die overeenkomen met elke kasstroom waarbij een andere korting dan het kortingstarief wordt toegepast. Deze verschilt van de functie **NPV()** aangezien **NPV()** alle tijdsperioden als gelijkwaardig beschouwt. Om deze reden is **XNPV()** preciezer dan **NPV()**.

### Syntaxis:

```
XNPV(discount_rate, pmt, date)
```

**Retourgegevenstypen:** numeriek. Het resultaat wordt standaard geformatteerd als een valuta.

De formule om de XNPV te berekenen is:

$$XNPV = \sum_{i=1}^n \frac{P_i}{(1+rate)^{(d_i-d_1)/365}}$$


waarbij:

- $P_i$  = Netto-ontvangsten/-uitgaven in cash gedurende één periode  $i$
- $d_1$  = de eerste betaaldatum
- $d_i$  = de  $i^e$  betaaldatum
- $rate$  = kortingspercentage

De huidige nettowaarde of Net Present Value/ NPV, wordt gebruikt om de huidige totale waarde van een toekomstige stream van cashflows te berekenen. Om de NPV te berekenen, moeten we de toekomstige cashflows per periode berekenen en het correcte kortingstatief bepalen.

`XNPV()` neemt een kortingspercentage en meerdere waarden op volgorde van periode. Ontvangsten (inkomsten) zijn positief en uitgaven (toekomstige betalingen) worden als negatieve waarden beschouwd. Deze vinden plaats aan het einde van elke periode.

#### Argumenten

Argument	Beschrijving
<code>discount_rate</code>	<b>discount_rate</b> is het percentagetarief van de toegepaste korting.  Een waarde van 0,1 zou een kortingspercentage van 10% aangeven.
<code>value</code>	Dit veld bevat cashflowwaarden. De eerste waarde wordt vanaf het begin als de cashflow beschouwd en de bijbehorende datum wordt als referentie gebruikt voor het berekenen van de huidige waarde voor alle toekomstige cashflows.  <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <b><i>XNPV()</i></b> past geen korting toe op de aanvankelijke cashflow. De kortingen voor daaropvolgende betalingen zijn gebaseerd op een jaar van 365 dagen. Dit verschilt van de <b>NPV()</b>, waarbij korting op iedere betaling wordt toegepast. </div>
<code>date</code>	Dit veld bevat de datum waarop de cashflow ( <b>value</b> , de tweede parameter) heeft plaatsgevonden. De eerste waarde wordt gebruikt als begindatum voor het berekenen van de offsets voor toekomstige cashflows.

#### Beperkingen:

Als er tekstwaarden zijn, NULL-waarden en ontbrekende waarden in een of beide delen van een gegevenspaar, wordt het gegevenspaar genegeerd.

### Wanneer gebruiken

- `XNPV()` wordt gebruikt in financiële modellen voor het berekenen van de huidige nettowaarde (NPV) van een investeringskans.
- Vanwege de grotere nauwkeurigheid wordt voor alle typen financiële modellen de voorkeur gegeven aan `XNPV` boven `NPV`.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Enkele betaling (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset van één project en de daarbij behorende cashflow voor één jaar in de tabel `CashFlow`. De begindatum voor de berekening is ingesteld op 1 juli 2022, met een nettocashflow van 0. Na één jaar, treedt een cashflow van \$1000 op.
- Een resident-load uit de tabel `CashFlow` die wordt gebruikt om het `XNPV`-veld te berekenen voor het project in de tabel `XNPV`.
- Een vastgesteld kortingspercentage van 10% (0,1) wordt gebruikt in de berekening van de `XNPV`.
- De opdracht `Group By` wordt gebruikt om alle betalingen voor het project te groeperen.

#### Load-script

```
CashFlow:
Load
*
Inline
[
PrjId,Dates,Values
1, '07/01/2022',0
```

```
1, '07/01/2023', 1000  
];
```

XNPV:

Load

```
    PrjId,  
    XNPV(0.1, Values, Dates) as XNPV //Discount Rate of 10%  
Resident CashFlow  
Group By PrjId;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- XNPV

Resultatentabel

PrjId	XNPV
1	\$909.09

Volgens formule, is de XNPV-waarde voor de eerste record 0 en voor de tweede is de XNPV-waarde \$909,09. Daarmee is de totale XNPV \$909,09.

### Voorbeeld 2 - Meerdere betalingen (script)

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset van één project en de daarbij behorende cashflow voor één jaar in de tabel CashFlow.
- Een resident-load uit de tabel CashFlow die wordt gebruikt om het XNPV-veld te berekenen voor het project in de tabel XNPV.
- Een vastgesteld kortingspercentage van 10% (0,1) wordt gebruikt in de berekening van de XNPV.
- De opdracht Group By wordt gebruikt om alle betalingen voor het project te groeperen.

### Load-script

CashFlow:

Load

\*

Inline

[

PrjId, Dates, Values

1, '07/01/2022', 0

```
1, '07/01/2024', 500
1, '07/01/2023', 1000
];

XNPV:
Load
    PrjId,
    XNPV(0.1, Values, Dates) as XNPV //Discount Rate of 10%
Resident CashFlow
Group By PrjId;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- XNPV

Resultatentabel

PrjId	XNPV
1	\$1322.21

In dit voorbeeld wordt er aan het eind van het eerste jaar een betaling van \$1000 ontvangen en een betaling van \$500 aan het eind van het tweede jaar. Met een kortingspercentage van 10% per periode is de effectieve XNPV gelijk aan \$1322,21.

Let op: alleen de eerste rij gegevens mag verwijzen naar de basisgegevens voor berekeningen. Voor de rest van de rijen is de volgorde niet belangrijk omdat de datumparameter wordt gebruikt om de verlopen periode te berekenen.

### Voorbeeld 3 - Meerdere betalingen en onregelmatige cashflows (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Kortingsstarieven voor twee projecten in de tabel `Project`.
- Cashflows voor meerdere perioden per project op project-id en datums. Het `Dates`-veld wordt gebruikt om de duur te berekenen dat het kortingspercentage wordt toegepast op de cashflow. Met uitzondering van de eerste record (aanvankelijke cashflow en datum), is de volgorde van de records niet belangrijk en deze veranderen zou geen invloed moeten hebben op de berekeningen.
- Door een combinatie van `NoConcatenate`, `resident-loads` en de functie `Left Join` te gebruiken, wordt een tijdelijke tabel, `tmpNPV`, aangemaakt die de records van de tabellen `Project` en `CashFlow` in één tabel combineert. In deze tabel worden de kortingsstarieven voor iedere cashflow herhaald.



- Een resident-load uit de tabel `tmpNPV` die wordt gebruikt om het `XNPV`-veld te berekenen voor elk project in de tabel `XNPV`.
- Het kortingspercentage voor de enkele waarde dat bij elk project hoort, wordt opgehaald met de `only()`-functie en wordt gebruikt in de `XNPV`-berekening per project.
- Een `Group By`-opdracht, gegroepeerd op project-id, wordt gebruikt om alle betalingen en bijbehorende datums voor elk project te groeperen.
- Om te vermijden dat synthetische of overbodige gegevens in het gegevensmodel worden geladen, wordt de tabel `tmpXNPV` aan het einde van het script verwijderd.

### Load-script

Project:

```
Load * inline [  
PrjId,Discount_Rate  
1,0.1  
2,0.15  
];
```

CashFlow:

```
Load  
*  
Inline  
[  
PrjId,Dates,Values  
1,'07/01/2021',0  
1,'07/01/2022',1000  
1,'07/01/2023',1000  
2,'07/01/2020',0  
2,'07/01/2023',500  
2,'07/01/2024',1000  
2,'07/01/2022',500  
];
```

tmpXNPV:

```
NoConcatenate Load *  
Resident Project;  
Left Join  
Load *  
Resident CashFlow;
```

XNPV:

```
Load  
PrjId,  
XNPV(Only(Discount_Rate),Values,Dates) as XNPV //Discount Rate will be 10% for Project 1 and  
15% for Project 2  
Resident tmpXNPV  
Group By PrjId;
```

```
Drop table tmpXNPV;
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- PrjId
- XNPV

Resultatentabel

PrjId	XNPV
1	\$1735.54
2	\$278.36

Project-id 1 heeft een aanvankelijke cashflow van \$0 op 1 juli 2021. Er zijn twee betalingen van \$1000 die aan het eind van twee achtereenvolgende jaren moeten worden ontvangen met een kortingspercentage van 10% per periode. Daarom is de effectieve NPV gelijk aan \$1735,54.

Project-id 2 heeft een aanvankelijke uitgave van \$1000 (vandaar het negatieve teken) op 1 juli 2020. Na twee jaar wordt een betaling van \$500 verwacht. Na drie jaar wordt nog een betaling van \$500 verwacht. Als laatste wordt op 1 juli 2024 een betaling van \$1000 verwacht. Met het kortingspercentage van 15% is de effectieve XNPV gelijk aan \$278,36.

### XNPV - diagramfunctie

**XNPV()** retourneert de geaggregeerde interne opbrengstwaarde voor een schema van cashflows (niet noodzakelijk periodiek) dat wordt vertegenwoordigd door getallenparen in de uitdrukkingen **pmt** en **date**, herhaald op de dimensies van het diagram. De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

#### Syntaxis:

```
XNPV ([TOTAL [<fld{,fld}>]] discount_rate, pmt, date)
```

**Retourgegevenstypen:** numeriek Het resultaat wordt standaard geformatteerd als een valuta.

#### Argumenten:

Argumenten

Argument	Beschrijving
discount_rate	<b>discount_rate</b> is the rate of discount over the length of the period. <b>discount_rate</b> is het percentagetarief van de toegepaste korting.
pmt	Betalingen. De uitdrukking of het veld met de cashflows die horen bij het betalingenschema dat is gegeven in <b>date</b> .
date	De uitdrukking of het veld met het datumschema dat hoort bij de cashflowbetalingen die zijn gegeven in <b>pmt</b> .

Argument	Beschrijving
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Beperkingen:

**discount\_rate**, **pmt** en **date** mogen geen aggregatiefuncties bevatten, tenzij deze interne aggregaties de kwalificaties **TOTAL** of **ALL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Voorbeelden en resultaten:

#### Voorbeelden en resultaten

Voorbeeld	Resultaat
XNPV(Discount, Payments, Date)	-\$3164.35

Gegevens die worden gebruikt in voorbeelden:

```
Cashflow:
LOAD 2013 as Year, * inline [
Date|Discount|Payments
2013-01-01|0.1|-10000
2013-03-01|0.1|3000
2013-10-30|0.1|4200
2014-02-01|0.2|6800
] (delimiter is '|');
```

### Zie ook:

p [NPV - diagramfunctie \(page 376\)](#)  
p [Aggr - diagramfunctie \(page 543\)](#)

## Statistische aggregatiefuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Statistische aggregatiefuncties in het script voor het laden van gegevens

De volgende statistische aggregatiefuncties kunnen worden gebruikt in scripts.

#### Avg

**Avg()** retourneert de gemiddelde waarde van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
Avg ([distinct] expression)
```

#### Correl

**Correl()** retourneert de geaggregeerde correlatiecoëfficiënt voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

```
Correl (x-expression, y-expression)
```

#### Fractile

**Fractile()** retourneert de waarde die overeenkomt met de inclusieve fractiel (kwantiel) van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
Fractile (expression, fractile)
```

#### FractileExc

**FractileExc()** retourneert de waarde die overeenkomt met de exclusieve fractiel (kwantiel) van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
FractileExc (expression, fractile)
```

#### Kurtosis

**Kurtosis()** retourneert de kurtosis van de gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
Kurtosis ([distinct ] expression )
```

#### LINEST\_B

**LINEST\_B()** retourneert geaggregeerde b-waarde (y-intercept) van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

```
LINEST_B (y-expression, x-expression [, y0 [, x0 ]])
```

#### LINEST\_df

**LINEST\_DF()** retourneert geaggregeerde vrijheidsgraden van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

**LINEST\_DF** (y-expression, x-expression [, y0 [, x0 ]])

### LINEST\_f

Deze scriptfunctie retourneert de geaggregeerde F-statistiek ( $r^2/(1-r^2)$ ) van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression die worden herhaald over een aantal records zoals vastgelegd door een **group by**-clause.

**LINEST\_F** (y-expression, x-expression [, y0 [, x0 ]])

### LINEST\_m

**LINEST\_M()** retourneert geaggregeerde m-waarde (helling) van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clause.

**LINEST\_M** (y-expression, x-expression [, y0 [, x0 ]])

### LINEST\_r2

retourneert de geaggregeerde -waarde (determinatiecoëfficiënt) van een lineaire regressie, die wordt gedefinieerd door de vergelijking voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in en , die worden herhaald over aantal records zoals vastgelegd door een - clause. $r^2y=mx+bx-expressiony-expressiongroup by$

**LINEST\_R2** (y-expression, x-expression [, y0 [, x0 ]])

### LINEST\_seb

**LINEST\_SEB()** retourneert de geaggregeerde standaardfout van de b-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clause.

**LINEST\_SEB** (y-expression, x-expression [, y0 [, x0 ]])

### LINEST\_sem

**LINEST\_SEM()** retourneert de geaggregeerde standaardfout van de m-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clause.

**LINEST\_SEM** (y-expression, x-expression [, y0 [, x0 ]])

### LINEST\_sey

**LINEST\_SEY()** retourneert de geaggregeerde standaardfout van de geschatte y-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clause.

**LINEST\_SEY** (y-expression, x-expression [, y0 [, x0 ]])

### **LINEST\_ssreg**

retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie, die wordt gedefinieerd door de vergelijking voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in en , herhaald over een aantal records zoals vastgelegd door een -clause.y=mx+bx-expressiony-expressiongroup by

```
LINEST_SSREG (y-expression, x-expression [, y0 [, x0 ]])
```

### **Linest\_ssresid**

**LINEST\_SSRESID()** retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clause.

```
LINEST_SSRESID (y-expression, x-expression [, y0 [, x0 ]])
```

### **Median**

**Median()** retourneert de geaggregeerde mediaan van de gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clause.

```
Median (expression)
```

### **Skew**

**Skew()** retourneert de scheefheid van de uitdrukking voor een aantal records zoals gedefinieerd door een **group by**-clause.

```
Skew ([ distinct] expression)
```

### **Stdev**

**Stdev()** retourneert de standaarddeviatie van de waarden in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clause.

```
Stdev ([distinct] expression)
```

### **Sterr**

**Sterr()** retourneert de geaggregeerde standaardfout (stdev/sqrt(n)) voor een reeks waarden die wordt vertegenwoordigd door de uitdrukking, herhaald over een aantal records zoals gedefinieerd door een **group by**-clause.

```
Sterr ([distinct] expression)
```

### **STEYX**

**STEYX()** retourneert de geaggregeerde standaardfout van de voorspelde y-waarde voor elke x-waarde in de regressie voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals gedefinieerd door een **group by**-clause.

```
STEYX (y-expression, x-expression)
```

## Statistische aggregatiefuncties in diagramuitdrukkingen

De volgende statistische aggregatiefuncties kunnen worden gebruikt in diagrammen.

### Avg

**Avg()** retourneert het geaggregeerde gemiddelde van uitdrukking of veld geïtereerd op de diagramdimensies.

```
Avg - diagramfunctie ([[SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]])  
expr)
```

### Correl

**Correl()** retourneert de geaggregeerde correlatiecoëfficiënt voor twee gegevensverzamelingen. De correlatiefunctie is een maat voor de relatie tussen de gegevensverzamelingen en is geaggregeerd voor (x,y)-waardeparen, herhaald op de dimensies van het diagram.

```
Correl - diagramfunctie ([[SetExpression] [TOTAL [<fld {, fld}>]]) value1,  
value2 )
```

### Fractile

**Fractile()** retourneert de waarde die correspondeert met de inclusieve fractiel (kwantiel) van de geaggregeerde gegevens in het bereik van de uitdrukking, geïtereerd op de diagramdimensies.

```
Fractile - diagramfunctie ([[SetExpression] [TOTAL [<fld {, fld}>]]) expr,  
fraction)
```

### FractileExc

**FractileExc()** retourneert de waarde die correspondeert met de exclusieve fractiel (kwantiel) van de geaggregeerde gegevens in het bereik van de uitdrukking, geïtereerd op de diagramdimensies.

```
FractileExc - diagramfunctie ([[SetExpression] [TOTAL [<fld {, fld}>]]) expr,  
fraction)
```

### Kurtosis

**Kurtosis()** retourneert de piekvorm van het bereik van geaggregeerde waarden in de uitdrukking of het veld, geïtereerd op de diagramdimensies.

```
Kurtosis - diagramfunctie ([[SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]])  
expr)
```

### LINEST\_b

**LINEST\_B()** retourneert de geaggregeerde b-waarde (y-intercept) van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

```
LINEST_R2 - diagramfunctie ([[SetExpression] [TOTAL [<fld{ ,fld}>]]) y_value,  
x_value[, y0_const[, x0_const]])
```

### LINEST\_df

**LINEST\_DF()** retourneert de geaggregeerde vrijheidsgraden van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

```
LINEST_DF - diagramfunctie ({[SetExpression] [TOTAL [<fld{, fld}>]]} y_value, x_value [, y0_const [, x0_const]])
```

LINEST\_f

**LINEST\_F()** retourneert de geaggregeerde F-statistiek ( $r^2/(1-r^2)$ ) van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

```
LINEST_F - diagramfunctie ({[SetExpression] [TOTAL [<fld{, fld}>]]} y_value, x_value [, y0_const [, x0_const]])
```

LINEST\_m

**LINEST\_M()** retourneert de geaggregeerde m-waarde (helling) van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

```
LINEST_M - diagramfunctie ({[SetExpression] [TOTAL [<fld{, fld}>]]} y_value, x_value [, y0_const [, x0_const]])
```

LINEST\_r2

**LINEST\_R2()** retourneert de geaggregeerde r<sup>2</sup>-waarde (determinatiecoëfficiënt) van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

```
LINEST_R2 - diagramfunctie ({[SetExpression] [TOTAL [<fld{, fld}>]] }y_value, x_value[, y0_const[, x0_const]])
```

LINEST\_seb

**LINEST\_SEB()** retourneert de geaggregeerde standaardfout van de b-waarde van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

```
LINEST_SEB - diagramfunctie ({[SetExpression] [TOTAL [<fld{, fld}>]] }y_value, x_value[, y0_const[, x0_const]])
```

LINEST\_sem

**LINEST\_SEM()** retourneert de geaggregeerde standaardfout van de m-waarde van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

```
LINEST_SEM - diagramfunctie ([{set_expression}][ distinct ] [total [<fld{, fld}>] ] y-expression, x-expression [, y0 [, x0 ] ] )
```

LINEST\_sey

**LINEST\_SEY()** retourneert de geaggregeerde standaardfout van de geschatte y-waarde van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.



```
LINEST_SEY - diagramfunctie({[SetExpression] [TOTAL [<fld{ ,fld}>]] }y_value,  
x_value[, y0_const[, x0_const]])
```

LINEST\_ssreg

**LINEST\_SSREG()** retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

```
LINEST_SSREG - diagramfunctie({[SetExpression] [TOTAL [<fld{ ,fld}>]] }y_  
value, x_value[, y0_const[, x0_const]])
```

LINEST\_ssresid

**LINEST\_SSRESID()** retourneert de geaggregeerde residu-som van het kwadraat van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

```
LINEST_SSRESID - diagramfunctieLINEST_SSRESID() retourneert de geaggregeerde  
residu-som van het kwadraat van een lineaire regressie, gedefinieerd door de  
vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd  
door getallenparen in de uitdrukkingen x_value en y_value, geïtereerd op de  
diagramdimensies. LINEST_SSRESID({[SetExpression]} [DISTINCT] [TOTAL [<fld{,  
fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

numeriek ArgumentenArgumentBeschrijvingy\_valueDe uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.x\_valueDe uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.y0, x0Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat. Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende. SetExpressionDe aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren. DISTINCTAls het woord DISTINCT voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd. TOTALAls het woord TOTAL voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd. Wanneer u TOTAL [<fld{ .fld}>] gebruikt en de TOTAL-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden. Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de

regressielijn door een vaste coördinaat gaat. De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie. Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd. An example of how to use **linest** functionsavg (`{[SetExpression] [TOTAL [<fld{ ,fld}>]] }y_value, x_value[, y0_const[, x0_const]]`)

Median

**Median()** retourneert de mediaan van het bereik van waarden dat is geaggregeerd in de uitdrukking, geïtereerd op de diagramdimensies.

```
Median - diagramfunctie ({[SetExpression] [TOTAL [<fld{ , fld}>]]} expr)
```

MutualInfo

**MutualInfo** berekent de wederzijdse informatie tussen twee velden of tussen geaggregeerde waarden in **Aggr()**.

```
MutualInfo - diagramfunctie (page 438) {[SetExpression] [DISTINCT] [TOTAL  
target, driver [, datatype [, breakdownbyvalue [, samplesize ]]]}
```

Skew

**Skew()** retourneert het geaggregeerde scheefheid van de uitdrukking of het veld geïtereerd op de diagramdimensies.

```
Skew - diagramfunctie {[SetExpression] [DISTINCT] [TOTAL [<fld{ , fld}>]]  
expr)
```

Stdev

**Stdev()** retourneert de standaarddeviatie van het bereik van geaggregeerde waarden in de uitdrukking of het veld, geïtereerd op de diagramdimensies.

```
Stdev - diagramfunctie ({[SetExpression] [DISTINCT] [TOTAL [<fld{ , fld}>]]}  
expr)
```

Sterr

**Sterr()** retourneert de waarde van de standaardfout van het gemiddelde, ( $stdev/\sqrt{n}$ ), voor de serie geaggregeerde waarden in de uitdrukking, geïtereerd op de diagramdimensies.

```
Sterr - diagramfunctie ({[SetExpression] [DISTINCT] [TOTAL [<fld{ , fld}>]]}  
expr)
```

STEYX

**STEYX()** retourneert de geaggregeerde standaardfout bij het voorspellen van y-waarden voor elke x-waarde in een lineaire regressie die wordt gedefinieerd door een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **y\_value** en **x\_value**.

**STEYX** - **diagramfunctie** { [SetExpression] [TOTAL [<fld{, fld}>]] } y\_value, x\_value)

### Avg

**Avg()** retourneert de gemiddelde waarde van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

#### Syntaxis:

**Avg** ([DISTINCT] expr)

**Retourgegevenstypen:** numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
DISTINCT	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.

#### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

##### Resulterende gegevens

Voorbeeld	Resultaat
<pre>Temp: crosstable (Month, Sales) load * inline [ Customer Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Astrida 46 60 70 13 78 20 45 65 78 12 78 22 Betacab 65 56 22 79 12 56 45 24 32 78 55 15 Canutility 77 68 34 91 24 68 57 36 44 90 67 27 Divadip 36 44 90 67 27 57 68 47 90 80 94 ] (delimiter is ' ');  Avg1: LOAD Customer, Avg(Sales) as MyAverageSalesByCustomer Resident Temp Group By Customer;</pre>	<pre>Customer MyAverageSalesByCustomer Astrida 48.916667 Betacab 44.916667 Canutility 56.916667 Divadip 63.083333</pre> <p>Dit kan worden gecontroleerd in het werkblad door een tabel te maken die de volgende meting bevat:</p> <pre>Sum(Sales)/12</pre>

Voorbeeld	Resultaat
<p>Stel dat de tabel <b>Temp</b> is geladen zoals in het vorige voorbeeld:</p> <pre>LOAD Customer,Avg(DISTINCT Sales) as MyAvgSalesDistinct Resident Temp Group By Customer;</pre>	<pre>Customer MyAverageSalesByCustomer Astrida 43.1 Betacab 43.909091 Canutility 55.909091 Divadip 61</pre> <p>Alleen de distinctieve waarden worden geteld. Deel het totaal door het aantal niet-dubbele waarden.</p>

### Avg - diagramfunctie

**Avg()** retourneert het geaggregeerde gemiddelde van uitdrukking of veld geïtereerd op de diagramdimensies.

#### Syntaxis:

```
Avg ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {,fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

### Voorbeelden en resultaten:

Example table

Customer	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Astrida	46	60	70	13	78	20	45	65	78	12	78	22
Betacab	65	56	22	79	12	56	45	24	32	78	55	15
Canutility	77	68	34	91	24	68	57	36	44	90	67	27
Divadip	57	36	44	90	67	27	57	68	47	90	80	94

Voorbeelden van functies

Voorbeeld	Resultaat
Avg(Sales)	Voor een tabel met de dimensie Customer en de meting Avg([Sales]), is 2566 het resultaat als <b>Totale</b> n worden weergegeven.
Avg([TOTAL (Sales)])	53,458333 voor alle waarden van Customer, omdat de kwalificatie TOTAL betekent dat dimensies buiten beschouwing worden gelaten.
Avg (DISTINCT (Sales))	51,862069 voor het totaal, omdat het gebruik van de kwalificatie Distinct betekent dat alleen unieke waarden in sales worden geëvalueerd voor elke customer.

Gegevens die worden gebruikt in voorbeelden:

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

sales2013:

```
Crosstable (MonthText, sales) LOAD * inline [
```

```
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

### Zie ook:

[p Aggr - diagramfunctie \(page 543\)](#)

### Correl

**Correl()** retourneert de geaggregeerde correlatiecoëfficiënt voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

### Syntaxis:

```
Correl(value1, value2)
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value1, value2	De uitdrukkingen of velden die de twee voorbeeldsets bevatten waarvoor de correlatiecoëfficiënt moet worden gemeten.

### Beperkingen:

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

### Resulterende gegevens

Voorbeeld	Resultaat
<pre> Salary: Load *, 1 as Grp; LOAD * inline [ "Employee name" Gender Age Salary Aiden Charles Male 20 25000 Brenda Davies Male 25 32000 Charlotte Edberg Female 45 56000 Daroush Ferrara Male 31 29000 Eunice Goldblum Female 31 32000 Freddy Halvorsen Male 25 26000 Gauri Indu Female 36 46000 Harry Jones Male 38 40000 Ian Underwood Male 40 45000 Jackie Kingsley Female 23 28000 ] (delimiter is ' ');  Correl1: LOAD Grp, Correl(Age,Salary) as Correl_Salary Resident Salary Group By Grp; </pre>	<p>In een tabel met de dimensie <code>correl_salary</code>, wordt het resultaat van de berekening <code>Correl()</code> in het script voor het laden van gegevens weergegeven: 0,9270611</p>

### Correl - diagramfunctie

**Correl()** retourneert de geaggregeerde correlatiecoëfficiënt voor twee gegevensverzamelingen. De correlatiefunctie is een maat voor de relatie tussen de gegevensverzamelingen en is geaggregeerd voor (x,y)-waardeparen, herhaald op de dimensies van het diagram.

#### Syntaxis:

```
Correl( [{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] value1, value2 )
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

#### Argumenten

Argument	Beschrijving
value1, value2	De uitdrukkingen of velden die de twee voorbeeldsets bevatten waarvoor de correlatiecoëfficiënt moet worden gemeten.

Argument	Beschrijving
SetExpression	De aggregatiefunctie aggregaat standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Voorbeelden en resultaten:

#### Voorbeelden van functies

Voorbeeld	Resultaat
correl(Age, salary)	Voor een tabel met de dimensie Employee name en de meting correl(Age, salary), is 0,9270611 het resultaat. Het resultaat wordt alleen weergegeven voor de cel met totalen.
correl (TOTAL Age, salary))	0.927. Dit en de volgende resultaten worden weergegeven met drie cijfers achter het decimaalteken omwille van de leesbaarheid.  Als u een filtervak maakt met de dimensie Gender en van hieruit selecties uitvoert, ziet u het resultaat 0.951 als Female is geselecteerd en 0.939 als Male is geselecteerd. Dit komt doordat bij de selectie alle resultaten die niet tot de andere waarde van Gender behoren worden uitgesloten.
correl({1} TOTAL Age, salary))	0.927. Onafhankelijk van selecties. Dit komt doordat in de set-uitdrukking {1} alle selecties en dimensies buiten beschouwing worden gelaten.{1}



Voorbeeld	Resultaat
<pre>Correl (TOTAL &lt;Gender&gt; Age, Salary))</pre>	<p>0,927 in de cel met totalen, 0,939 voor alle waarden van Male en 0,951 voor alle waarden van Female. Dit komt overeen met de resultaten van het uitvoeren van de selecties in een filtervak op basis van Gender.</p>

Gegevens die worden gebruikt in voorbeelden:

```
Salary:
LOAD * inline [
"Employee name"|Gender|Age|Salary
Aiden Charles|Male|20|25000
Brenda Davies|Male|25|32000
Charlotte Edberg|Female|45|56000
Daroush Ferrara|Male|31|29000
Eunice Goldblum|Female|31|32000
Freddy Halvorsen|Male|25|26000
Gauri Indu|Female|36|46000
Harry Jones|Male|38|40000
Ian Underwood|Male|40|45000
Jackie Kingsley|Female|23|28000
] (delimiter is '|');
```

**Zie ook:**

[p Aggr - diagramfunctie \(page 543\)](#)

[p Avg - diagramfunctie \(page 396\)](#)

[p RangeCorrel \(page 1329\)](#)

### Fractile

**Fractile()** retourneert de waarde die overeenkomt met de inclusieve fractiel (kwantiel) van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.



*U kunt **FractileExc** (page 405) gebruiken om de exclusieve fractiel te berekenen.*

**Syntaxis:**

```
Fractile(expr, fraction)
```

**Retourgegevenstypen:** numeriek

De functie retourneert de waarde die overeenkomt met de rangorde zoals gedefinieerd door rangorde = fractie \* (N+1) waarbij N het aantal waarden is in expr. Als rangorde een niet-geheel getal is, wordt een interpolatie uitgevoerd tussen de twee dichtstbijzijnde waarden.

### Argumenten:

Argumenten

Argument	Beschrijving
expr	De uitdrukking of het veld met de te gebruiken gegevens bij het berekenen van de fractiel.
fraction	Een getal tussen 0 en 1 die correspondeert met de fractiel (kwantiel uitgedrukt als een fractie) die moet worden berekend.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Resulterende gegevens

Voorbeeld	Resultaat
<pre>Table1: crosstable LOAD recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  Fractile1: LOAD Type, Fractile(value,0.75) as MyFractile Resident Table1 Group By Type;</pre>	<p>In een tabel met de dimensies Type en MyFractile zijn de resultaten van de berekeningen Fractile() in het script voor het laden van gegevens:</p> <pre>Type MyFractile Comparison 27.5 Observation 36</pre>

### Fractile - diagramfunctie

**Fractile()** retourneert de waarde die correspondeert met de inclusieve fractiel (kwantiel) van de geaggregeerde gegevens in het bereik van de uitdrukking, geïtereerd op de diagramdimensies.



*U kunt **FractileExc** - diagramfunctie (page 406) gebruiken om de exclusieve fractiel te berekenen.*

#### Syntaxis:

```
Fractile([SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]] expr, fraction)
```

**Retourgegevenstypen:** numeriek

De functie retourneert de waarde die overeenkomt met de rangorde zoals gedefinieerd door  $\text{rangorde} = \text{fractie} * (N+1)$  waarbij  $N$  het aantal waarden is in **expr**. Als rangorde een niet-geheel getal is, wordt een interpolatie uitgevoerd tussen de twee dichtstbijzijnde waarden.

#### Argumenten:

##### Argumenten

Argument	Beschrijving
<b>expr</b>	De uitdrukking of het veld met de te gebruiken gegevens bij het berekenen van de fractiel.
<b>fraction</b>	Een getal tussen 0 en 1 die correspondeert met de fractiel (kwantiel uitgedrukt als een fractie) die moet worden berekend.
<b>SetExpression</b>	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking <b>Set-analyse</b> kunt u een alternatieve set records definiëren.
<b>DISTINCT</b>	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
<b>TOTAL</b>	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {,fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

### Voorbeelden en resultaten:

Example table

Customer	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Astrida	46	60	70	13	78	20	45	65	78	12	78	22
Betacab	65	56	22	79	12	56	45	24	32	78	55	15
Canutility	77	68	34	91	24	68	57	36	44	90	67	27
Divadip	57	36	44	90	67	27	57	68	47	90	80	94

Voorbeelden van functies

Voorbeeld	Resultaat
Fractile (Sales, 0.75)	Voor een tabel met de dimensie customer en de meting Fractile([Sales]), is 71,75 het resultaat als <b>Totale</b> n worden weergegeven. Dit is het punt in de verdeling van waarden van sales waar 75% van de waarden onder vallen.
Fractile (TOTAL Sales, 0.75))	71.75 voor alle waarden van customer, omdat de kwalificatie TOTAL betekent dat dimensies buiten beschouwing worden gelaten.
Fractile (DISTINCT Sales, 0.75)	70 voor het totaal, omdat het gebruik van de kwalificatie DISTINCT betekent dat alleen unieke waarden in sales worden geëvalueerd voor elke customer.

Gegevens die worden gebruikt in voorbeelden:

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

```
Sales2013:
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

### Zie ook:

[p Aggr - diagramfunctie \(page 543\)](#)

## FractileExc

**FractileExc()** retourneert de waarde die overeenkomt met de exclusieve fractiel (kwantiel) van de geaggregeerde gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.



*U kunt [Fractile \(page 401\)](#) gebruiken om de inclusieve fractiel te berekenen.*

### Syntaxis:

```
FractileExc (expr, fractie)
```

### Retourgegevenstypen: numeriek

De functie retourneert de waarde die overeenkomt met de rangorde zoals gedefinieerd door  $\text{rangorde} = \text{fractie} * (N+1)$  waarbij  $N$  het aantal waarden is in `expr`. Als rangorde een niet-geheel getal is, wordt een interpolatie uitgevoerd tussen de twee dichtstbijzijnde waarden.

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking of het veld met de te gebruiken gegevens bij het berekenen van de fractiel.
fraction	Een getal tussen 0 en 1 die correspondeert met de fractiel (kwantiel uitgedrukt als een fractie) die moet worden berekend.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

### Resulterende gegevens

Voorbeeld	Resultaat
<pre>Table1: crosstable LOAD recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  Fractile1: LOAD Type, FractileExc(Value,0.75) as MyFractile Resident Table1 Group By Type;</pre>	<p>In een tabel met de dimensies Type en MyFractile zijn de resultaten van de berekeningen FractileExc() in het script voor het laden van gegevens:</p> <pre>Type MyFractile Comparison 28.5 Observation 38</pre>

### FractileExc - diagramfunctie

**FractileExc()** retourneert de waarde die correspondeert met de exclusieve fractiel (kwantiel) van de geaggregeerde gegevens in het bereik van de uitdrukking, geïtereerd op de diagramdimensies.



*U kunt Fractile - diagramfunctie (page 403) gebruiken om de inclusieve fractiel te berekenen.*

#### Syntaxis:

```
FractileExc ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr,
fraction)
```

#### Retourgegevenstypen: numeriek

De functie retourneert de waarde die overeenkomt met de rangorde zoals gedefinieerd door  $\text{rangorde} = \text{fractie} * (N+1)$  waarbij  $N$  het aantal waarden is in `expr`. Als `rangorde` een niet-geheel getal is, wordt een interpolatie uitgevoerd tussen de twee dichtstbijzijnde waarden.

### Argumenten:

Argumenten

Argument	Beschrijving
expr	De uitdrukking of het veld met de te gebruiken gegevens bij het berekenen van de fractiel.
fraction	Een getal tussen 0 en 1 die correspondeert met de fractiel (kwantiel uitgedrukt als een fractie) die moet worden berekend.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

### Voorbeelden en resultaten:

Example table

Customer	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Astrida	46	60	70	13	78	20	45	65	78	12	78	22
Betacab	65	56	22	79	12	56	45	24	32	78	55	15
Canutility	77	68	34	91	24	68	57	36	44	90	67	27
Divadip	57	36	44	90	67	27	57	68	47	90	80	94

### Voorbeelden van functies

Voorbeeld	Resultaat
FractileExc (Sales, 0.75)	Voor een tabel met de dimensie customer en de meting FractileExc([Sales]), is 75,25 het resultaat als <b>Totale</b> n worden weergegeven. Dit is het punt in de verdeling van waarden van sales waar 75% van de waarden onder vallen.
FractileExc (TOTAL Sales, 0.75))	75,25 voor alle waarden van customer, omdat de kwalificatie TOTAL betekent dat dimensies buiten beschouwing worden gelaten.
FractileExc (DISTINCT Sales, 0.75)	73,50 voor het totaal, omdat het gebruik van de kwalificatie DISTINCT betekent dat alleen unieke waarden in sales worden geëvalueerd voor elke customer.

Gegevens die worden gebruikt in voorbeelden:

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

Sales2013:

```
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

**Zie ook:**

[p Aggr - diagramfunctie \(page 543\)](#)

### Kurtosis

**Kurtosis()** retourneert de kurtosis van de gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.



### Syntaxis:

```
Kurtosis([distinct ] expr )
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
distinct	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

### Resulterende gegevens

Voorbeeld	Resultaat
<pre>Table1: crosstable LOAD recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  Kurtosis1: LOAD Type, Kurtosis(value) as MyKurtosis1, Kurtosis(DISTINCT value) as MyKurtosis2 Resident Table1 Group By Type;</pre>	<p>In een tabel met de dimensies Type, MyKurtosis1 en MyKurtosis2 zijn de resultaten van de berekeningen Kurtosis() in het script voor het laden van gegevens:</p> <pre>Type MyKurtosis1 MyKurtosis2 Comparison -1.1612957 -1.4982366 observation -1.1148768 -0.93540144</pre>

### Kurtosis - diagramfunctie

**Kurtosis()** retourneert de piekvorm van het bereik van geaggregeerde waarden in de uitdrukking of het veld, geïtereerd op de diagramdimensies.

#### Syntaxis:

```
Kurtosis ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

**Voorbeelden en resultaten:**

Example table

Type	Value																		
Comparison	27	28	31	31	19	14	34	34	12	32	12	12	13	21	39	37	2		
Observation	350	42	15	15	21	44	16	28	48	16	33	38	41	22	12	39	19	25	

### Voorbeelden van functies

Voorbeeld	Resultaat
kurtosis (value)	Voor een tabel met de dimensie type en de meting kurtosis(value), is 1.252 het resultaat als <b>Totale</b> n worden weergegeven voor de tabel en de cijfernotatie is ingesteld op 3 significante posities. Voor comparison is het 1.161 en voor observation is het 1.115.
kurtosis (TOTAL value)	1.252 voor alle waarden van type, omdat de kwalificatie TOTAL betekent dat dimensies buiten beschouwing worden gelaten.

Gegevens die worden gebruikt in voorbeelden:

Table1:

```
crosstable LOAD recno() as ID, * inline [
observation|comparison
35|2
40|27
12|38
15|31
21|1
14|19
46|1
10|34
28|3
48|1
16|2
30|3
32|2
48|1
31|2
22|1
12|3
39|29
19|37
25|2 ] (delimiter is '|');
```

**Zie ook:**

[p Avg - diagramfunctie \(page 396\)](#)

### LINEST\_B

**LINEST\_B()** retourneert geaggregeerde b-waarde (y-intercept) van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

**Syntaxis:**

```
LINEST_B (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y(0), x(0)	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 456)*

### LINEST\_B - diagramfunctie

**LINEST\_B()** retourneert de geaggregeerde b-waarde (y-intercept) van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.


**Syntaxis:**

```
LINEST_B ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value  
[, y0_const [ , x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y0_const, x0_const	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;">  <i>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.</i> </div>
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*  
p *Avg - diagramfunctie (page 396)*

### LINEST\_DF

**LINEST\_DF()** retourneert geaggregeerde vrijheidsgraden van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

### Syntaxis:

```
LINEST_DF (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y(0), x(0)	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.

### Beperkingen:

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*

## LINEST\_DF - diagramfunctie

**LINEST\_DF()** retourneert de geaggregeerde vrijheidsgraden van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.


**Syntaxis:**

```
LINEST_DF ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value [, y0_const [, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

## Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y0, x0	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;">  <i>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.</i> </div>
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {, fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.



### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

---

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*

p *Avg - diagramfunctie (page 396)*

### LINEST\_F

Deze scriptfunctie retourneert de geaggregeerde F-statistiek ( $r^2/(1-r^2)$ ) van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression die worden herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

### Syntaxis:

```
LINEST_F (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y(0), x(0)	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.

### Beperkingen:

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*

### LINEST\_F - diagramfunctie

**LINEST\_F()** retourneert de geaggregeerde F-statistiek ( $r^2/(1-r^2)$ ) van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

### Syntaxis:

```
LINEST_F ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value  
[, y0_const [, x0_const]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y0, x0	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.</i> </div>
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.

Argument	Beschrijving
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*

p *Avg - diagramfunctie (page 396)*

## LINEST\_M

**LINEST\_M()** retourneert geaggregeerde m-waarde (helling) van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

### Syntaxis:

```
LINEST_M (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.

Argument	Beschrijving
y(0), x(0)	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.

### Beperkingen:

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*

### LINEST\_M - diagramfunctie

**LINEST\_M()** retourneert de geaggregeerde m-waarde (helling) van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

### Syntaxis:


```
LINEST_M([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value  
[, y0_const [, x0_const]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.

Argument	Beschrijving
y0, x0	<p>Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> <i>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.</i></p> </div>
SetExpression	De aggregatiefunctie aggregereert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 456)*

p *Avg - diagramfunctie (page 396)*

**LINEST\_R2**

retourneert de geaggregeerde -waarde (determinatiecoëfficiënt) van een lineaire regressie, die wordt gedefinieerd door de vergelijking voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in en , die worden herhaald over aantal records zoals vastgelegd door een -clause.  $r^2 = \frac{y}{mx+bx-expression} - y-expression$  group by

**Syntaxis:**

```
LINEST_R2 (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek**Argumenten:**

## Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y(0), x(0)	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.

**Beperkingen:**

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

*p Voorbeelden van het gebruik van linest-functies (page 456)*

### LINEST\_R2 - diagramfunctie

**LINEST\_R2()** retourneert de geaggregeerde r<sup>2</sup>-waarde (determinatiecoëfficiënt) van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_R2 ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y0, x0	<p>Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.</i> </div>
SetExpression	De aggregatiefunctie aggregereert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*  
p *Avg - diagramfunctie (page 396)*

### LINEST\_SEB

**LINEST\_SEB()** retourneert de geaggregeerde standaardfout van de b-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

### Syntaxis:

```
LINEST_SEB (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y(0), x(0)	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.

### Beperkingen:

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*



### LINEST\_SEB - diagramfunctie

**LINEST\_SEB()** retourneert de geaggregeerde standaardfout van de b-waarde van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

#### Syntaxis:

```
LINEST_SEB ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y0, x0	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;">  <i>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.</i> </div>
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {, fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

---

### Zie ook:

p *Voorbeelden van het gebruik van lineest-functies (page 456)*

p *Avg - diagramfunctie (page 396)*

## LINEST\_SEM

**LINEST\_SEM()** retourneert de geaggregeerde standaardfout van de m-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

### Syntaxis:

```
LINEST_SEM (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y(0), x(0)	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.

### Beperkingen:

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Zie ook:**

p *Voorbeelden van het gebruik van linest-functies (page 456)*

### LINEST\_SEM - diagramfunctie

**LINEST\_SEM()** retourneert de geaggregeerde standaardfout van de m-waarde van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

**Syntaxis:**

```
LINEST_SEM([SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y0, x0	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.</i> </div>
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.

Argument	Beschrijving
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

p *Voorbeelden van het gebruik van lineest-functies (page 456)*

p *Avg - diagramfunctie (page 396)*

## LINEST\_SEY

**LINEST\_SEY()** retourneert de geaggregeerde standaardfout van de geschatte y-waarde van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

### Syntaxis:

```
LINEST_SEY (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.

Argument	Beschrijving
y(0), x(0)	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.

### Beperkingen:

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*

### LINEST\_SEY - diagramfunctie

**LINEST\_SEY()** retourneert de geaggregeerde standaardfout van de geschatte y-waarde van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

### Syntaxis:


```
LINEST_SEY ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.

Argument	Beschrijving
y0, x0	<p>Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> <i>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.</i></p> </div>
SetExpression	De aggregatiefunctie aggregereert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*

p *Avg - diagramfunctie (page 396)*

### LINEST\_SSREG

retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie, die wordt gedefinieerd door de vergelijking voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in en , herhaald over een aantal records zoals vastgelegd door een -clausule.y=mx+bx-expressiony-expressiongroup by

### Syntaxis:

```
LINEST_SSREG (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y(0), x(0)	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.

### Beperkingen:

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*

## LINEST\_SSREG - diagramfunctie

**LINEST\_SSREG()** retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.

### Syntaxis:

```
LINEST_SSREG ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value[, y0_const[, x0_const]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y0, x0	<p>Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.</i> </div>
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.



### Zie ook:

- p *Voorbeelden van het gebruik van linest-functies (page 456)*
- p *Avg - diagramfunctie (page 396)*

### LINEST\_SSRESID

**LINEST\_SSRESID()** retourneert de geaggregeerde regressie voor de som van de kwadraten van een lineaire regressie, die wordt gedefinieerd door de vergelijking  $y=mx+b$  voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals vastgelegd door een **group by**-clausule.

### Syntaxis:

```
LINEST_SSRESID (y_value, x_value[, y0 [, x0 ]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y(0), x(0)	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.

### Beperkingen:

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Zie ook:

- p *Voorbeelden van het gebruik van linest-functies (page 456)*

## LINEST\_SSRESID - diagramfunctie

**LINEST\_SSRESID()** retourneert de geaggregeerde residu-som van het kwadraat van een lineaire regressie, gedefinieerd door de vergelijking  $y=mx+b$  voor een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **x\_value** en **y\_value**, geïtereerd op de diagramdimensies.


### Syntaxis:

```
LINEST_SSRESID ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value,
x_value[, y0_const[, x0_const]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.
y0, x0	Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste coördinaat gaat.  <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;">  <i>Tenzij zowel y0 als x0 zijn opgegeven, zijn voor deze berekening minimaal twee geldige gegevensparen nodig. Als y0 en x0 zijn opgegeven, is een enkel gegevenspaar voldoende.</i> </div>
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {,fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.

Met de optionele waarde y0 kan worden afgedwongen dat de regressielijn de y-as op een bepaald punt snijdt. Als zowel y0 als x0 wordt opgegeven, wordt afgedwongen dat de regressielijn door een vaste

coördinaat gaat.

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

---

### Zie ook:

p *Voorbeelden van het gebruik van linest-functies (page 456)*

p *Avg - diagramfunctie (page 396)*

## Median

**Median()** retourneert de geaggregeerde mediaan van de gegevens in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

### Syntaxis:

**Median** (*expr*)

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

Voorbeeld: Scriptuitdrukking die gebruikmaakt van de mediaan

Voorbeeld - scriptuitdrukking

### Load-script

Laad de volgende inline-gegevens- en scriptuitdrukking in de editor voor het laden van gegevens voor dit voorbeeld.

```
Table 1: Load RecNo() as RowNo, Letter, Number Inline [Letter, Number A,1 A,3 A,4 A,9 B,2 B,8 B,9];
```

```
Median: LOAD Letter, Median(Number) as MyMedian Resident Table1 Group
```

### Een visualisatie maken

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Letter** en **MyMedia** als dimensies.

### Resultaat

Letter	MyMedian
A	3.5
B	8

### Uitleg

De mediaan wordt beschouwd als het "middelste" getal als de getallen gesorteerd zijn in volgorde van kleinste naar grootste. Als de gegevensverzameling een even aantal waarden heeft, retourneert de functie het gemiddelde van de twee middelste waarden. In dit voorbeeld wordt de mediaan berekend voor elke set van waarden van **A** en **B**, die respectievelijk 3.5 en 8 zijn.

### Median - diagramfunctie

**Median()** retourneert de mediaan van het bereik van waarden dat is geaggregeerd in de uitdrukking, geïtereerd op de diagramdimensies.

### Syntaxis:

```
Median ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {, fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Voorbeeld: Diagramuitdrukking die gebruikmaakt van de mediaan

Voorbeeld - diagramuitdrukking

### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
Load RecNo() as RowNo, Letter, Number Inline [Letter, Number A,1 A,3 A,4 A,9 B,2 B,8 B,9];
```

### Een visualisatie maken

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Letter** als dimensie.

### Diagramuitdrukking

Voeg de volgende uitdrukking als meting toe aan de tabel:

```
Median(Number)
```

### Resultaat

Letter	Median(Number)
Totals	4
A	3.5
B	8

### Uitleg

De mediaan wordt beschouwd als het "middelste" getal als de getallen gesorteerd zijn in volgorde van kleinste naar grootste. Als de gegevensverzameling een even aantal waarden heeft, retourneert de functie het gemiddelde van de twee middelste waarden. In dit voorbeeld wordt de mediaan berekend voor elke set van waarden van **A** en **B**, die respectievelijk 3.5 en 8 zijn.

De mediaan voor **Totalen** wordt berekend vanaf alle waarden en is gelijk aan 4.

---

### Zie ook:

[p Avg - diagramfunctie \(page 396\)](#)

### MutualInfo - diagramfunctie

**MutualInfo** berekent de wederzijdse informatie tussen twee velden of tussen geaggregeerde waarden in **Aggr()**.

**MutualInfo** retourneert de geaggregeerde wederzijdse informatie voor twee gegevensverzamelingen. Hierdoor is een analyse van de sleutelfactos tussen een veld een mogelijke driver mogelijk. Wederzijdse informatie is een maat voor de relatie tussen de gegevensverzamelingen en is geaggregeerd voor (x,y)-waardeparen, herhaald op de dimensies van het diagram. Wederzijdse informatie wordt gemeten tussen 0 en 1 en kan worden weergegeven als een percentielwaarde. **MutualInfo** wordt gedefinieerd door selecties of door een set-uitdrukking.

**MutualInfo** maakt verschillende soorten van MI-analyse mogelijk:

- Paarsgewijze MI: Bereken de MI tussen een driverveld en een doelveld.
- Driveranalyse op waarde: De MI wordt berekend tussen individuele veldwaarden in de driver- en veldwaarden.
- Functie selecteren: Gebruik **MutualInfo** in een rasterdiagram zodat een matrix ontstaat waarin alle velden met elkaar worden vergeleken op basis van de MI.

**MutualInfo** geeft niet noodzakelijk een causaliteit aan tussen velden die wederzijdse informatie bevatten. Twee velden kunnen wederzijdse informatie bevatten, maar zijn mogelijk geen gelijkwaardige drivers. Bij het vergelijken van de verkoop van ijs en de buitentemperatuur toont **MutualInfo** bijvoorbeeld dat er wederzijdse informatie aanwezig is. Het geeft niet aan of de buitentemperatuur invloed heeft op de verkoop van ijs, wat aannemelijk is, of dat de verkoop van ijs invloed heeft op de buitentemperatuur, wat onwaarschijnlijk is.

Bij het berekenen van wederzijdse informatie hebben koppelingen invloed op de overeenkomst tussen de frequentie van waarden van velden die afkomstig zijn uit verschillende tabellen.

De geretourneerde waarden voor dezelfde velden of selecties kunnen licht van elkaar afwijken. Dit komt doordat de oproep van **MutualInfo** wordt uitgevoerd voor willekeurig geselecteerde voorbeeldgegevens en de inherente willekeur van het **MutualInfo**-algoritme.

**MutualInfo** kan worden toegepast op de functie **Aggr()**.

#### Syntaxis:

```
MutualInfo ({SetExpression}) [DISTINCT] [TOTAL] field1, field2 , datatype [,  
breakdownbyvalue [, samplesize ]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
field1, field2	De uitdrukkingen of velden die de twee voorbeeldsets bevatten waarvoor de wederzijdse informatie moet worden gemeten.
datatype	De gegevenstypen in het doel en de driver, 1 of 'dd' voor discreet:discreet 2 of 'cc' voor doorlopend:doorlopend 3 of 'cd' voor doorlopend:discreet 4 of 'dc' voor discreet:doorlopend Gegevenstypen zijn niet hoofdlettergevoelig.
breakdownbyvalue	Een statische waarde die overeenkomt met de waarde in de driver. Indien opgegeven, wordt met de berekening de MI-bijdrage voor die waarde berekend. U kunt <b>ValueList()</b> of <b>ValueLoop()</b> gebruiken. Als <b>Null()</b> is toegevoegd, wordt met de berekening de totale MI berekend voor alle waarden in de driver.  Uitsplitsing naar waarde vereist dat de driver discrete gegevens bevat.
samplesize	Het aantal waarden van het doel en de driver in de steekproef. De steekproef vindt willekeurig plaats. <b>MutualInfo</b> vereist een minimale steekproefgrootte van 80. Standaard bemonstert <b>MutualInfo</b> tot 10.000 gegevensparen, aangezien <b>MutualInfo</b> zwaar en tijdrovend kan zijn. U kunt een groter aantal gegevensparen in de steekproef specificeren. Als er bij <b>MutualInfo</b> een time-out optreedt, maak dan de grootte van de steekproef kleiner.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.

Argument	Beschrijving
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Beperkingen:

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

#### Voorbeelden van functies

Voorbeeld	Resultaat
mutualinfo (Age, salary, 1)	Voor een tabel die de dimensie Employee name en de meting mutualinfo(Age, salary, 1) bevat, is 0.99820986 het resultaat. Het resultaat wordt alleen weergegeven voor de cel met totalen.
mutualinfo (TOTAL Age, salary, 1, null(), 81)	Als u een filtervak maakt met de dimensie Gender en van hieruit selecties uitvoert, ziet u het resultaat 0.99805677 als Female is geselecteerd en 0.99847373 als Male is geselecteerd. Dit komt doordat bij de selectie alle resultaten die niet tot de andere waarde van Gender behoren worden uitgesloten.
mutualinfo (TOTAL Age, Gender, 1, ValueLoop (25,35))	0.68196996. Als u een willekeurige waarde in Gender kiest, verandert deze in 0.
mutualinfo ({1} TOTAL Age, salary, 1, null())	0.99820986. Dit vindt onafhankelijk van selecties plaats. De set-uitdrukking {1} negeert alle selecties en dimensies.

Gegevens die worden gebruikt in voorbeelden:

salary:

```
LOAD * inline [
```



"Employee name"|Age|Gender|Salary

Aiden Charles|20|Male|25000

Ann Lindquist|69|Female|58000

Anna Johansen|37|Female|36000

Anna Karlsson|42|Female|23000

Antonio Garcia|20|Male|61000

Benjamin Smith|42|Male|27000

Bill Yang|49|Male|50000

Binh Protzmann|69|Male|21000

Bob Park|51|Male|54000

Brenda Davies|25|Male|32000

Celine Gagnon|48|Female|38000

Cezar Sandu|50|Male|46000

Charles Ingvar Jönsson|27|Male|58000

Charlotte Edberg|45|Female|56000

Cindy Lynn|69|Female|28000

Clark Wayne|63|Male|31000

Daroush Ferrara|31|Male|29000

David Cooper|37|Male|64000

David Leg|58|Male|57000

Eunice Goldblum|31|Female|32000

Freddy Halvorsen|25|Male|26000

Gauri Indu|36|Female|46000

George van Zaant|59|Male|47000

Glenn Brown|58|Male|40000

Harry Jones|38|Male|40000

Helen Brolin|52|Female|66000

Hiroshi Ito|24|Male|42000

Ian Underwood|40|Male|45000

Ingrid Hendrix|63|Female|27000

Ira Baume|39|Female|39000

Jackie Kingsley|23|Female|28000

Jennica Williams|36|Female|48000

Jerry Tessel|31|Male|57000

Jim Bond|50|Male|58000

Joan Callins|60|Female|65000

Joan Cleaves|25|Female|61000

Joe Cheng|61|Male|41000

John Doe|36|Male|59000

John Lemon|43|Male|21000

Karen Helmkey|54|Female|25000

Karl Berger|38|Male|68000

Karl Straubbaum|30|Male|40000

Kaya Alpan|32|Female|60000

Kenneth Finley|21|Male|25000

Leif Shine|63|Male|70000

Lennart Skoglund|63|Male|24000

Leona Korhonen|46|Female|50000

Lina André|50|Female|65000

Louis Presley|29|Male|36000

Luke Langston|50|Male|63000

Marcus Salvatori|31|Male|46000

Marie Simon|57|Female|23000

Mario Rossi|39|Male|62000

Markus Danzig|26|Male|48000  
Michael Carlen|21|Male|45000  
Michelle Tyson|44|Female|69000  
Mike Ashkenaz|45|Male|68000  
Miro Ito|40|Male|39000  
Nina Mihn|62|Female|57000  
Olivia Nguyen|35|Female|51000  
Olivier Simenon|44|Male|31000  
Östen Ärlig|68|Male|57000  
Pamala Garcia|69|Female|29000  
Paolo Romano|34|Male|45000  
Pat Taylor|67|Female|69000  
Paul Dupont|34|Male|38000  
Peter Smith|56|Male|53000  
Pierre Clouseau|21|Male|37000  
Preben Jørgensen|35|Male|38000  
Rey Jones|65|Female|20000  
Ricardo Gucci|55|Male|65000  
Richard Ranieri|30|Male|64000  
Rob Carsson|46|Male|54000  
Rolf Wesselund|25|Male|51000  
Ronaldo Costa|64|Male|39000  
Sabrina Richards|57|Female|40000  
Sato Hiromu|35|Male|21000  
Sehoon Daw|57|Male|24000  
Stefan Lind|67|Male|35000  
Steve Cioazzi|58|Male|23000

```
Sunil Gupta|45|Male|40000
Sven Svensson|45|Male|55000
Tom Lindwall|46|Male|24000
Tomas Nilsson|27|Male|22000
Trinity Rizzo|52|Female|48000
Vanessa Lambert|54|Female|27000
] (delimiter is '|');
```

### Skew

**Skew()** retourneert de scheefheid van de uitdrukking voor een aantal records zoals gedefinieerd door een **group by**-clausule.

#### Syntaxis:

```
Skew([ distinct] expr)
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
DISTINCT	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.

#### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `type` en `MySkew` als dimensies.

### Resulterende gegevens

Voorbeeld	Resultaat
<pre>Table1: crosstable LOAD recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  Skew1: LOAD Type, Skew(Value) as MySkew Resident Table1 Group By Type;</pre>	<p>De resultaten van de berekening Skew() zijn:</p> <ul style="list-style-type: none"> <li>• Type is MySkew</li> <li>• comparison is 0.86414768</li> <li>• observation is 0.32625351</li> </ul>

### Skew - diagramfunctie

**Skew()** retourneert het geaggregeerde scheefheid van de uitdrukking of het veld geïtereerd op de diagramdimensies.

#### Syntaxis:

```
Skew ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.

Argument	Beschrijving
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Beperkingen:

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `type` als dimensie en `skew(value)` als meting.

`totals` moet zijn ingeschakeld in de eigenschappen van de tabel.

Voorbeeld	Resultaat
<pre>Table1: crosstable LOAD recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');</pre>	<p>De resultaten van de berekening Skew(Value) zijn:</p> <ul style="list-style-type: none"> <li>• Totalis 0.23522195</li> <li>• Comparison is 0.86414768</li> <li>• observation is 0.32625351</li> </ul>

### Zie ook:

[p Avg - diagramfunctie \(page 396\)](#)

### Stdev

**Stdev()** retourneert de standaarddeviatie van de waarden in de uitdrukking voor een aantal records, zoals gedefinieerd door een **group by**-clausule.

### Syntaxis:

```
Stdev ([distinct] expr)
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
distinct	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `Type` en `MyStdev` als dimensies.

#### Resulterende gegevens

Voorbeeld	Resultaat
<pre>Table1: crosstable LOAD recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  stdev1: LOAD Type, stdev(Value) as MyStdev Resident Table1 Group By Type;</pre>	<p>De resultaten van de berekening <code>Stdev()</code> zijn:</p> <ul style="list-style-type: none"> <li>• <code>Type</code> is <code>MyStdev</code></li> <li>• <code>Comparison</code> is 14.61245</li> <li>• <code>Observation</code> is 12.507997</li> </ul>

### Stdev - diagramfunctie

**Stdev()** retourneert de standaarddeviatie van het bereik van geaggregeerde waarden in de uitdrukking of het veld, geïtereerd op de diagramdimensies.

#### Syntaxis:

```
Stdev ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```



**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregereert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `Type` als dimensie en `stdev(value)` als meting.

`Total`s moet zijn ingeschakeld in de eigenschappen van de tabel.

Voorbeeld	Resultaat
<pre> stdev(value) Table1: crosstable LOAD recno() as ID, * inline [ observation comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' '); </pre>	<p>De resultaten van de berekening Stdev(Value) zijn:</p> <ul style="list-style-type: none"> <li>• Totalis 15.47529</li> <li>• Comparison is 14.61245</li> <li>• Observation is 12.507997</li> </ul>

### Zie ook:

p *Avg - diagramfunctie (page 396)*

p *STEXY - diagramfunctie (page 454)*

### Sterr

**Sterr()** retourneert de geaggregeerde standaardfout (stdev/sqrt(n)) voor een reeks waarden die wordt vertegenwoordigd door de uitdrukking, herhaald over een aantal records zoals gedefinieerd door een **group by**-clausule.

### Syntaxis:

```
Sterr ([distinct] expr)
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
distinct	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

#### Resulterende gegevens

Voorbeeld	Resultaat
<pre>Table1: crosstable LOAD recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');  Sterr1: LOAD Type, Sterr(Value) as MySterr Resident Table1 Group By Type;</pre>	<p>In een tabel met de dimensies Type en MySterr zijn de resultaten van de berekening Sterr() in het script voor het laden van gegevens:</p> <pre>Type MySterr Comparison 3.2674431 Observation 2.7968733</pre>

### Sterr - diagramfunctie

**Sterr()** retourneert de waarde van de standaardfout van het gemiddelde, ( $stdev/\sqrt{n}$ ), voor de serie geaggregeerde waarden in de uitdrukking, geïtereerd op de diagramdimensies.

#### Syntaxis:

```
Sterr ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] expr)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregereert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `type` als dimensie en `sterr(value)` als meting.

`total`s moet zijn ingeschakeld in de eigenschappen van de tabel.

Voorbeeld	Resultaat
<pre>Table1: crosstable LOAD recno() as ID, * inline [ Observation Comparison 35 2 40 27 12 38 15 31 21 1 14 19 46 1 10 34 28 3 48 1 16 2 30 3 32 2 48 1 31 2 22 1 12 3 39 29 19 37 25 2 ] (delimiter is ' ');</pre>	<p>De resultaten van de berekening Sterr(Value) zijn:</p> <ul style="list-style-type: none"> <li>• Totalis 2.4468583</li> <li>• Comparison is 3.2674431</li> <li>• Observation is 2.7968733</li> </ul>

### Zie ook:

p *Avg - diagramfunctie (page 396)*

p *STEYX - diagramfunctie (page 454)*

## STEYX

**STEYX()** retourneert de geaggregeerde standaardfout van de voorspelde y-waarde voor elke x-waarde in de regressie voor een reeks coördinaten die wordt vertegenwoordigd door getallenparen in x-expression en y-expression, herhaald over een aantal records zoals gedefinieerd door een **group by**-clausule.

### Syntaxis:

```
STEYX (y_value, x_value)
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat het bereik van bekende y-waarden bevat die moeten worden gemeten.

Argument	Beschrijving
x_value	De uitdrukking die of het veld dat het bereik van bekende x-waarden bevat die moeten worden gemeten.

### Beperkingen:

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

#### Resulterende gegevens

Voorbeeld	Resultaat
<pre>Trend: Load *, 1 as Grp; LOAD * inline [ Month KnownY KnownX Jan 2 6 Feb 3 5 Mar 9 11 Apr 6 7 May 8 5 Jun 7 4 Jul 5 5 Aug 10 8 Sep 9 10 Oct 12 14 Nov 15 17 Dec 14 16 ] (delimiter is ' ');  STEYX1: LOAD Grp, STEYX(KnownY, KnownX) as MySTEYX Resident Trend Group By Grp;</pre>	<p>In een tabel met de dimensie <code>MySTEYX</code> is het resultaat van de berekening <code>STEYX()</code> in het load-script voor gegevens 2.0714764.</p>

### STEYX - diagramfunctie

**STEYX()** retourneert de geaggregeerde standaardfout bij het voorspellen van y-waarden voor elke x-waarde in een lineaire regressie die wordt gedefinieerd door een serie coördinaten die wordt vertegenwoordigd door getallenparen in de uitdrukkingen **y\_value** en **x\_value**.

### Syntaxis:

```
STEYX ([{SetExpression}] [DISTINCT] [TOTAL [<fld{, fld}>]] y_value, x_value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
y_value	De uitdrukking die of het veld dat de reeks van bekende y-waarden bevat die moet worden gemeten.
x_value	De uitdrukking die of het veld dat de reeks van bekende x-waarden bevat die moet worden gemeten.
SetExpression	De aggregatiefunctie aggregereert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.
TOTAL	Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.  Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b> -kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.

**Beperkingen:**

De parameter van de aggregatiefunctie mag geen andere aggregatiefuncties bevatten, tenzij deze ingesloten aggregaties de kwalificatie **TOTAL** bevatten. Gebruik voor meer geavanceerde geneste aggregaties de geavanceerde functie **Aggr**, in combinatie met een opgegeven dimensie.

Als een of beide delen van een gegevenspaar tekstwaarden, NULL-waarden of ontbrekende waarden bevatten, wordt het hele gegevenspaar genegeerd.

**Voorbeelden en resultaten:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Bouw vervolgens een strakke tabel met `knownY` en `knownX` als dimensies en `SteyX(knownY, knownX)` als meting.

`Totals` moet zijn ingeschakeld in de eigenschappen van de tabel.

Voorbeeld	Resultaat
<pre>Trend: LOAD * inline [ Month KnownY KnownX Jan 2 6 Feb 3 5 Mar 9 11 Apr 6 7 May 8 5 Jun 7 4 Jul 5 5 Aug 10 8 Sep 9 10 Oct 12 14 Nov 15 17 Dec 14 16 ] (delimiter is ' ');</pre>	<p>Het resultaat van de berekening STEYX(KnownY,KnownX) is 2.071 (indien cijfernotatie is ingesteld op 3 cijfers achter het decimaalteken).</p>

### Zie ook:

p Avg - diagramfunctie (page 396)

p Sterr - diagramfunctie (page 451)

### Voorbeelden van het gebruik van linest-functies

De linest-functies worden gebruikt om te zoeken naar waarden die verband houden met lineaire regressieanalyse. In dit gedeelte wordt beschreven hoe visualisaties kunnen worden gebouwd aan de hand van de voorbeeldgegevens, om de waarden te vinden van de linest-functies die beschikbaar zijn in Qlik Sense. De functies linest kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

Raadpleeg de individuele onderwerpen over de linest-diagram- en scriptfunctie voor een beschrijving van syntaxis en argumenten.

### Gegevens- en scriptuitdrukkingen die in de voorbeelden worden gebruikt

Laad de volgende inline-gegevens- en scriptuitdrukkingen in de editor voor het laden van gegevens voor de onderstaande linest()-voorbeelden.

```
T1: LOAD *, 1 as Grp; LOAD * inline [ X|Y 1|0 2|1 3|3 4|8 5|14 6|20 7|0 8|50 9|25 10|60 11|38
12|19 13|26 14|143 15|98 16|27 17|59 18|78 19|158 20|279 ] (delimiter is '|');
Grp, linest_B(Y,X) as Linest_B, linest_DF(Y,X) as Linest_DF, linest_F(Y,X) as Linest_F,
linest_M(Y,X) as Linest_M, linest_R2(Y,X) as Linest_R2, linest_SEB(Y,X,1,1) as Linest_SEB,
linest_SEM(Y,X) as Linest_SEM, linest_SEY(Y,X) as Linest_SEY, linest_SSREG(Y,X) as Linest_
SSREG, linest_SSRESID(Y,X) as Linest_SSRESID resident T1 group by Grp;
```

R1: LOAD

### Voorbeeld 1: Scriptuitdrukkingen met linest

Voorbeeld: Scriptuitdrukkingen

### Een visualisatie maken vanaf berekeningen van het script voor het laden van gegevens

Maak een tabelvisualisatie in een Qlik Sense-werkblad met de volgende velden als kolommen:



- Linest\_B
- Linest\_DF
- Linest\_F
- Linest\_M
- Linest\_R2
- Linest\_SEB
- Linest\_SEM
- Linest\_SEY
- Linest\_SSREG
- Linest\_SSRESID

### Resultaat

De tabel die de resultaten van de linest-berekeningen bevat die zijn uitgevoerd in het script voor het laden van gegevens zouden er als volgt moeten uitzien:

Resultatentabel

Linest_B	Linest_DF	Linest_F	Linest_M	Linest_R2	Linest_SEB
-35.047	18	20.788	8.605	0.536	22.607

Resultatentabel

Linest_SEM	Linest_SEY	Linest_SSREG	Linest_SSRESID
1.887	48.666	49235.014	42631.186

### Voorbeeld 2: Diagramuitdrukkingen met linest

Voorbeeld: Diagramuitdrukkingen

Maak een tabelvisualisatie in een Qlik Sense-werkblad met de volgende velden als dimensies:

```
valueList('Linest_b', 'Linest_df', 'Linest_f', 'Linest_m', 'Linest_r2', 'Linest_SEB', 'Linest_SEM', 'Linest_SEY', 'Linest_SSREG', 'Linest_SSRESID')
```

Deze uitdrukking gebruikt de functie voor synthetische dimensies om labels voor de dimensies te maken met de namen van de linest-functies. U kunt het label wijzigen in **Linest functions** om ruimte te besparen.

Voeg de volgende uitdrukking als meting toe aan de tabel:

```
Pick(Match(ValueList('Linest_b', 'Linest_df', 'Linest_f', 'Linest_m', 'Linest_r2', 'Linest_SEB', 'Linest_SEM', 'Linest_SEY', 'Linest_SSREG', 'Linest_SSRESID'), 'Linest_b', 'Linest_df', 'Linest_f', 'Linest_m', 'Linest_r2', 'Linest_SEB', 'Linest_SEM', 'Linest_SEY', 'Linest_SSREG', 'Linest_SSRESID'), Linest_b(Y,X), Linest_df(Y,X), Linest_f(Y,X), Linest_m(Y,X), Linest_r2(Y,X), Linest_SEB(Y,X,1,1), Linest_SEM(Y,X), Linest_SEY(Y,X), Linest_SSREG(Y,X), Linest_SSRESID(Y,X) )
```

Deze uitdrukking geeft de waarde van het resultaat van iedere linest-functie weer naast de overeenkomstige naam in de synthetische dimensie. Het resultaat van `Linest_b(Y,X)` wordt weergegeven naast **linest\_b**, enzovoort.

## Resultaat

Resultatentabel

Linest functions	Linest function results
Linest_b	-35.047
Linest_df	18
Linest_f	20.788
Linest_m	8.605
Linest_r2	0.536
Linest_SEB	22.607
Linest_SEM	1.887
Linest_SEY	48.666
Linest_SSREG	49235.014
Linest_SSRESID	42631.186

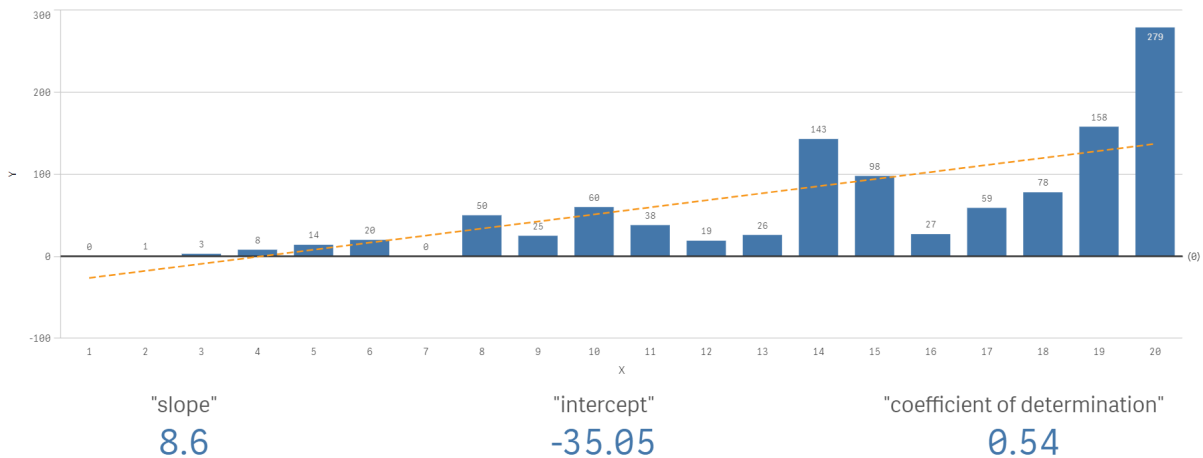
## Voorbeeld 3: Diagramuitdrukkingen met linest

## Voorbeeld: Diagramuitdrukkingen

1. Maak een visualisatie van een staafdiagram in een Qlik Sense-werkblad met **X** als een dimensie en **Y** als een meting.
2. Voeg een lineaire trendlijn toe aan de Y-meting.
3. Voeg een KPI-visualisatie toe aan het werkblad.
  1. Voeg *helling* toe als een label voor de KPI.
  2. Voeg `sum(Linest_M)` toe als een uitdrukking voor de KPI.
4. Voeg een tweede KPI-visualisatie toe aan het werkblad.
  1. Voeg *snijpunt* toe als een label voor de KPI.
  2. Voeg `sum(Linest_B)` toe als een uitdrukking voor de KPI.
5. Voeg een derde KPI-visualisatie toe aan het werkblad.
  1. Voeg *Determinatiecoëfficiënt* toe als een label voor de KPI.
  2. Voeg `sum(Linest_R2)` toe als een uitdrukking voor de KPI.

### Resultaat

LinestFuncInGraph



### Uitleg

De staafdiagram geeft het plotten van de X- en Y-gegevens weer. Relevante linest()-functies bieden waarden voor de vergelijking van de lineaire regressie waarop de trendlijn is gebaseerd, namelijk  $y = m * x + b$ . De vergelijking gebruikt de methode van de "minste vierkanten" om een rechte lijn (trendlijn) te berekenen door een matrix te retourneren die een lijn beschrijft die het beste overeenkomt met de gegevens.

De KPI's geven de resultaten weer van de linest()-functies **sum(Linest\_M)** voor helling en **sum(Linest\_B)** voor het Y-snijpunt, die variabelen zijn in de vergelijking van de lineaire regressie en de overeenkomstige geaggregeerde R2-waarde voor de determinatiecoëfficiënt.

## Statistische testfuncties

Statistische testfuncties kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen, maar de syntaxis verschilt.

### Chi-2 testfuncties

Worden veel gebruikt in de studie van kwalitatieve variabelen. U kunt geobserveerde frequenties vergelijken in een eenzijdige frequentietabel met verwachte frequenties, of de verbinding tussen twee variabelen bestuderen in een contingentietabel.

### T-testfuncties

T-testfuncties worden gebruikt voor statistisch onderzoek van twee steekproefgemiddelden. Een t-test met twee steekproeven waarbij wordt onderzocht of twee steekproeven verschillend zijn, wordt veel gebruikt als twee normale verdelingen onbekende afwijkingen hebben en de steekproef voor het experiment klein is.

### Z-testfuncties

Een statistisch onderzoek van twee steekproefgemiddelden. Een z-test met twee steekproeven onderzoekt of twee steekproeven van elkaar verschillen. Dit wordt over het algemeen gebruikt als twee normale verdelingen onbekende varianties hebben en als een experiment een grote steekproefgrootte heeft.

### Chi2-testfuncties

Worden veel gebruikt in de studie van kwalitatieve variabelen. U kunt geobserveerde frequenties vergelijken in een eenzijdige frequentietabel met verwachte frequenties, of de verbinding tussen twee variabelen bestuderen in een contingentietabel. Chi-squared test functions are used to determine whether there is a statistically significant difference between the expected frequencies and the observed frequencies in one or more groups. Often a histogram is used, and the different bins are compared to an expected distribution.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

#### Chi2Test\_chi2

retourneert de geaggregeerde chi-test voor een of twee series waarden. Chi2Test\_chi2()

```
retourneert de geaggregeerde chi-test voor een of twee series waarden. Chi2Test_chi2() (col, row, actual_value[, expected_value])
```

#### Chi2Test\_df

Chi2Test\_df() retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de chi<sup>2</sup>-test voor een of twee series waarden.

```
Chi2Test_df() retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de chi2-test voor een of twee series waarden. (col, row, actual_value[, expected_value])
```

#### Chi2Test\_p

Chi2Test\_p() retourneert de geaggregeerde p-waarde (significantie) van de chi<sup>2</sup>-test voor een of twee series waarden.

```
Chi2Test_p - diagramfunctie (col, row, actual_value[, expected_value])
```

---

#### Zie ook:

p *T-testfuncties* (page 463)

p *Z-testfuncties* (page 500)

### Chi2Test\_chi2

retourneert de geaggregeerde chi-test voor een of twee series waarden. Chi2Test\_chi2()

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.



Alle Qlik Sense  $\chi^2$ -testfuncties hebben dezelfde argumenten.

#### Syntaxis:

```
Chi2Test_chi2(col, row, actual_value[, expected_value])
```

Retourgegevenstypen: numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
col, row	De opgegeven kolom en rij in de matrix met waarden die worden getest.
actual_value	De geobserveerde waarde van de gegevens in de opgegeven <b>col</b> en <b>row</b> .
expected_value	De verwachte waarde voor de verdeling in de opgegeven <b>col</b> en <b>row</b> .

#### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

#### Voorbeelden:

```
Chi2Test_chi2( Grp, Grade, Count )
```

```
Chi2Test_chi2( Gender, Description, Observed, Expected )
```

---

#### Zie ook:

[p Voorbeelden van het gebruik van  \$\chi^2\$ -test-functies in grafieken \(page 517\)](#)

[p Voorbeelden van het gebruik van  \$\chi^2\$ -test-functies in het load-script voor gegevens \(page 520\)](#)

### Chi2Test\_df

**Chi2Test\_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de  $\chi^2$ -test voor een of twee series waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.



*Alle Qlik Sense  $\chi^2$ -testfuncties hebben dezelfde argumenten.*

### Syntaxis:

```
Chi2Test_df(col, row, actual_value[, expected_value])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
col, row	De opgegeven kolom en rij in de matrix met waarden die worden getest.
actual_value	De geobserveerde waarde van de gegevens in de opgegeven <b>col</b> en <b>row</b> .
expected_value	De verwachte waarde voor de verdeling in de opgegeven <b>col</b> en <b>row</b> .

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
Chi2Test_df( Grp, Grade, Count )  
Chi2Test_df( Gender, Description, Observed, Expected )
```

### Zie ook:

*p Voorbeelden van het gebruik van  $\chi^2$ -test-functies in grafieken (page 517)*

*p Voorbeelden van het gebruik van  $\chi^2$ -test-functies in het load-script voor gegevens (page 520)*

### Chi2Test\_p - diagramfunctie

De test kan worden uitgevoerd op de waarden in **Chi2Test\_p()** retourneert de geaggregeerde p-waarde (significantie) van de  $\chi^2$ -test voor een of twee series waarden., waarbij wordt getest op variaties binnen de opgegeven - en -matrix, of de waarden in worden vergeleken met de corresponderende waarden in , indien deze zijn opgegeven.actual\_valuecolrowactual\_valueexpected\_value

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.



*Alle Qlik Sense  $\chi^2$ -testfuncties hebben dezelfde argumenten.*

### Syntaxis:

```
Chi2Test_p(col, row, actual_value[, expected_value])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
col, row	De opgegeven kolom en rij in de matrix met waarden die worden getest.
actual_value	De geobserveerde waarde van de gegevens in de opgegeven <b>col</b> en <b>row</b> .
expected_value	De verwachte waarde voor de verdeling in de opgegeven <b>col</b> en <b>row</b> .

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
Chi2Test_p( Grp, Grade, Count )  
Chi2Test_p( Gender, Description, Observed, Expected )
```

### Zie ook:

[p Voorbeelden van het gebruik van  \$\chi^2\$ -test-functies in grafieken \(page 517\)](#)

[p Voorbeelden van het gebruik van  \$\chi^2\$ -test-functies in het load-script voor gegevens \(page 520\)](#)

## T-testfuncties

T-testfuncties worden gebruikt voor statistisch onderzoek van twee steekproefgemiddelden. Een t-test met twee steekproeven waarbij wordt onderzocht of twee steekproeven verschillend zijn, wordt veel gebruikt als twee normale verdelingen onbekende afwijkingen hebben en de steekproef voor het experiment klein is.

In de volgende gedeelten worden de statistische t-testfuncties gegroepeerd op basis van de steekproeftest voor cursisten die geldt voor elk type functie.

*Een typisch t-test-rapport opstellen (page 521)*

### T-testen voor twee onafhankelijke steekproeven

De volgende functies zijn van toepassing op t-toetsen voor cursisten voor twee onafhankelijke steekproeven.

`ttest_conf`

**TTest\_conf** retourneert het geaggregeerde vertrouwensinterval voor de t-testwaarde voor twee onafhankelijke steekproeven.

```
TTest_conf retourneert het geaggregeerde vertrouwensinterval voor de t-testwaarde voor twee onafhankelijke steekproeven. ( grp, value [, sig[, eq_var]])
```

`ttest_df`

**TTest\_df()** retourneert de geaggregeerde waarde (vrijheidsgraden) van de t-toets van Student voor twee onafhankelijke series waarden.

```
TTest_df() retourneert de geaggregeerde waarde (vrijheidsgraden) van de t-toets van Student voor twee onafhankelijke series waarden. (grp, value [, eq_var])
```

`ttest_dif`

**TTest\_dif()** is een numerieke functie die het geaggregeerde gemiddelde verschil van Student t-test voor twee onafhankelijke series waarden retourneert.

```
TTest_dif() is een numerieke functie die het geaggregeerde gemiddelde verschil van Student t-test voor twee onafhankelijke series waarden retourneert. (grp, value)
```

`ttest_lower`

**TTest\_lower()** retourneert de geaggregeerde ondergrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

```
TTest_lower() retourneert de geaggregeerde ondergrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden. (grp, value [, sig[, eq_var]])
```

`ttest_sig`

**TTest\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden.

```
TTest_sig() retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden. (grp, value [, eq_var])
```

`ttest_sterr`

**TTest\_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden.



**TTest\_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden. (grp, value [, eq\_var])

ttest\_t

**TTest\_t()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

**TTest\_t()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden. (grp, value [, eq\_var])

ttest\_upper

**TTest\_upper()** retourneert de geaggregeerde bovengrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

**TTest\_upper()** retourneert de geaggregeerde bovengrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden. (grp, value [, sig [, eq\_var]])

### T-testen voor twee onafhankelijke gewogen steekproeven

De volgende functies zijn van toepassing op t-testen voor cursisten voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

ttestw\_conf

**TTestw\_conf()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

**TTestw\_conf()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden. (weight, grp, value [, sig[, eq\_var]])

ttestw\_df

**TTestw\_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van twee onafhankelijke series waarden.

**TTestw\_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van twee onafhankelijke series waarden. (weight, grp, value [, eq\_var])

ttestw\_dif

**TTestw\_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van twee onafhankelijke series waarden.

**TTestw\_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van twee onafhankelijke series waarden. ( weight, grp, value)

ttestw\_lower

**TTestw\_lower()** retourneert de geaggregeerde ondergrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

**TTestw\_lower()** retourneert de geaggregeerde ondergrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden. (weight, grp, value [, sig[, eq\_var]])

ttestw\_sig

**TTestw\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden.

```
TTestw_sig() retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden. ( weight, grp, value [, eq_var])
```

ttestw\_sterr

**TTestw\_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden.

```
TTestw_sterr() retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden. (weight, grp, value [, eq_var])
```

ttestw\_t

**TTestw\_t()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

```
TTestw_t() retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden. (weight, grp, value [, eq_var])
```

ttestw\_upper

**TTestw\_upper()** retourneert de geaggregeerde bovengrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

```
TTestw_upper() retourneert de geaggregeerde bovengrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden. (weight, grp, value [, sig [, eq_var]])
```

### T-testen voor één steekproef

De volgende functies zijn van toepassing op t-testen voor cursisten voor één steekproef.

ttest1\_conf

**TTest1\_conf()** retourneert de geaggregeerde waarde van het vertrouwensinterval voor een serie waarden.

```
TTest1_conf() retourneert de geaggregeerde waarde van het vertrouwensinterval voor een serie waarden. (value [, sig])
```

ttest1\_df

**TTest1\_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden.

```
TTest1_df() retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden. (value)
```

ttest1\_dif

**TTest1\_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden.

**TTest1\_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden. (value)

ttest1\_lower

**TTest1\_lower()** retourneert de geaggregeerde ondergrenswaarde van het vertrouwensinterval voor een serie waarden.

**TTest1\_lower()** retourneert de geaggregeerde ondergrenswaarde van het vertrouwensinterval voor een serie waarden. (value [, sig])

ttest1\_sig

**TTest1\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor een serie waarden.

**TTest1\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor een serie waarden. (value)

ttest1\_sterr

**TTest1\_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor een serie waarden.

**TTest1\_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor een serie waarden. (value)

ttest1\_t

**TTest1\_t()** retourneert de geaggregeerde t-waarde voor een serie waarden.

**TTest1\_t()** retourneert de geaggregeerde t-waarde voor een serie waarden. (value)

ttest1\_upper

**TTest1\_upper()** retourneert de geaggregeerde bovengrenswaarde van het vertrouwensinterval voor een serie waarden.

**TTest1\_upper()** retourneert de geaggregeerde bovengrenswaarde van het vertrouwensinterval voor een serie waarden. (value [, sig])

### T-testen voor één gewogen steekproef

De volgende functies zijn van toepassing op t-testen voor cursisten voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

ttest1w\_conf

**TTest1w\_conf()** is een numerieke functie die de geaggregeerde waarde van het vertrouwensinterval voor een serie waarden retourneert.

**TTest1w\_conf()** is een numerieke functie die de geaggregeerde waarde van het vertrouwensinterval voor een serie waarden retourneert. (weight, value [, sig])

ttest1w\_df

**TTest1w\_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden.

```
TTest1w_df() retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden. (weight, value)
```

ttest1w\_dif

**TTest1w\_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden.

```
TTest1w_dif() retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden. (weight, value)
```

ttest1w\_lower

**TTest1w\_lower()** retourneert de geaggregeerde ondergrenswaarde van het-vertrouwensinterval voor een serie waarden.

```
TTest1w_lower() retourneert de geaggregeerde ondergrenswaarde van het-vertrouwensinterval voor een serie waarden. (weight, value [, sig])
```

ttest1w\_sig

**TTest1w\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor een serie waarden.

```
TTest1w_sig() retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor een serie waarden. (weight, value)
```

ttest1w\_sterr

**TTest1w\_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor een serie waarden.

```
TTest1w_sterr() retourneert de geaggregeerde standaardfout van de t-toets van Student voor een serie waarden. (weight, value)
```

ttest1w\_t

**TTest1w\_t()** retourneert de geaggregeerde t-waarde voor een serie waarden.

```
TTest1w_t() retourneert de geaggregeerde t-waarde voor een serie waarden. (weight, value)
```

ttest1w\_upper

**TTest1w\_upper()** retourneert de geaggregeerde bovengrenswaarde van het-vertrouwensinterval voor een serie waarden.

```
TTest1w_upper() retourneert de geaggregeerde bovengrenswaarde van het-vertrouwensinterval voor een serie waarden. (weight, value [, sig])
```

### TTest\_conf

**TTest\_conf** retourneert het geaggregeerde vertrouwensinterval voor de t-testwaarde voor twee onafhankelijke steekproeven.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

#### Syntaxis:

```
TTest_conf ( grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

#### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

#### Voorbeelden:

```
TTest_conf( Group, value )  
TTest_conf( Group, value, sig, false )
```

### Zie ook:

p *Een typisch t-test-rapport opstellen (page 521)*

### TTest\_df

**TTest\_df()** retourneert de geaggregeerde waarde (vrijheidsgraden) van de t-toets van Student voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest_df (grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTest_df( Group, value )  
TTest_df( Group, value, false )
```

### Zie ook:

p *Een typisch t-test-rapport opstellen (page 521)*

### TTest\_dif

**TTest\_dif()** is een numerieke functie die het geaggregeerde gemiddelde verschil van Student t-test voor twee onafhankelijke series waarden retourneert.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest_dif (grp, value [, eq_var] )
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTest_dif( Group, value )  
TTest_dif( Group, value, false )
```

### Zie ook:

p *Een typisch t-test-rapport opstellen (page 521)*

### TTest\_lower

**TTest\_lower()** retourneert de geaggregeerde ondergrenswaarde van het-vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest_lower (grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.



### Voorbeelden:

```
TTest_lower( Group, Value )  
TTest_lower( Group, Value, Sig, false )
```

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest\_sig

**TTest\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest_sig (grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTest_sig( Group, Value )  
TTest_sig( Group, Value, false )
```

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest\_sterr

**TTest\_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest_sterr (grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTest_sterr( Group, Value )  
TTest_sterr( Group, Value, false )
```

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest\_t

**TTest\_t()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest_t(grp, value[, eq_var])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeeld:

```
TTest_t( Group, Value, false )
```

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest\_upper

**TTest\_upper()** retourneert de geaggregeerde bovengrenswaarde van het-vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest_upper (grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTest_upper( Group, Value )  
TTest_upper( Group, Value, sig, false )
```

---

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTestw\_conf

**TTestw\_conf()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTestw_conf (weight, grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .

Argument	Beschrijving
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingwaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTestw_conf( weight, Group, value )  
TTestw_conf( weight, Group, value, sig, false )
```

---

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTestw\_df

**TTestw\_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTestw_df (weight, grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTestw_df( weight, Group, value )  
TTestw_df( weight, Group, value, false )
```

---

**Zie ook:**

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

TTestw\_dif

**TTestw\_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTestw_dif (weight, grp, value)
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukking leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTestw_dif( weight, Group, value )  
TTestw_dif( weight, Group, value, false )
```

---

### Zie ook:

*p Een typisch t-test-rapport opstellen (page 521)*

### TTestw\_lower

**TTestw\_lower()** retourneert de geaggregeerde ondergrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.



### Syntaxis:

```
TTestw_lower (weight, grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTestw_lower( weight, group, value )  
TTestw_lower( weight, group, value, sig, false )
```

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTestw\_sig

**TTestw\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTestw_sig ( weight, grp, value [, eq_var] )
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTestw_sig( weight, Group, Value )  
TTestw_sig( weight, Group, Value, false )
```

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTestw\_sterr

**TTestw\_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

#### Syntaxis:

```
TTestw_sterr (weight, grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

#### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

#### Voorbeelden:

```
TTestw_sterr( weight, Group, value )  
TTestw_sterr( weight, Group, Value, false )
```

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTestw\_t

**TTestw\_t()** retourneert de geaggregeerde t-waarde voor twee onafhankelijke waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
ttestw_t (weight, grp, value [, eq_var])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTestw_t( weight, Group, value )
```

```
TTestw_t( weight, Group, value, false )
```

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

TTestw\_upper

**TTestw\_upper()** retourneert de geaggregeerde bovengrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor twee onafhankelijke steekproeven waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTestw_upper (weight, grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTestw_upper( weight, Group, value )  
TTestw_upper( weight, Group, value, sig, false )
```

---

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest1\_conf

**TTest1\_conf()** retourneert de geaggregeerde waarde van het vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest1_conf (value [, sig ])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTest1_conf( value )  
TTest1_conf( value, 0.005 )
```

---

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest1\_df

**TTest1\_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest1_df (value)
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeeld:

```
TTest1_df( value )
```

---

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

---

### TTest1\_dif

**TTest1\_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

#### Syntaxis:

```
TTest1_dif (value)
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .

#### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

#### Voorbeeld:

```
TTest1_dif( value )
```

---

#### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest1\_lower

**TTest1\_lower()** retourneert de geaggregeerde ondergrenswaarde van het-vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.



Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest1_lower (value [, sig])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTest1_lower( value )  
TTest1_lower( value, 0.005 )
```

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest1\_sig

**TTest1\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest1_sig (value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1_sig( value )
```

---

**Zie ook:**

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

**TTest1\_sterr**

**TTest1\_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1_sterr (value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
TTest1_sterr( value )
```

---

**Zie ook:**

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

**TTest1\_t**

**TTest1\_t()** retourneert de geaggregeerde t-waarde voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
TTest1_t (value)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

### Voorbeeld:

```
TTest1_t( value )
```

---

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest1\_upper

**TTest1\_upper()** retourneert de geaggregeerde bovengrenswaarde van het vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest1_upper (value [, sig])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTest1_upper( value )  
TTest1_upper( value, 0.005 )
```

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest1w\_conf

**TTest1w\_conf()** is een **numerieke** functie die de geaggregeerde waarde van het vertrouwensinterval voor een serie waarden retourneert.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest1w_conf (weight, value [, sig ])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
TTest1w_conf( weight, value )  
TTest1w_conf( weight, value, 0.005 )
```

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest1w\_df

**TTest1w\_df()** retourneert de geaggregeerde df-waarde (vrijheidsgraden) van de t-toets van Student van een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest1w_df (weight, value)
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeeld:

```
TTest1w_df( weight, value )
```

### Zie ook:

p *Een typisch t-test-rapport opstellen (page 521)*

### TTest1w\_dif

**TTest1w\_dif()** retourneert het geaggregeerde gemiddelde verschil van de t-toets van Student van een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest1w_dif (weight, value)
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeeld:

```
TTest1w_dif( weight, value )
```

### Zie ook:

p *Een typisch t-test-rapport opstellen (page 521)*

### TTest1w\_lower

**TTest1w\_lower()** retourneert de geaggregeerde ondergrenswaarde van het vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

#### Syntaxis:

```
TTest1w_lower (weight, value [, sig ])
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.

#### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

#### Voorbeelden:

```
TTest1w_lower( weight, value )  
TTest1w_lower( weight, value, 0.005 )
```

---

#### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)



### TTest1w\_sig

**TTest1w\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van Students t-toets op voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

#### Syntaxis:

```
TTest1w_sig (weight, value)
```

**Retourgegevenstypen:** numeriek

#### Argumenten:

##### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .

#### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

#### Voorbeeld:

```
TTest1w_sig( weight, value )
```

---

#### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest1w\_sterr

**TTest1w\_sterr()** retourneert de geaggregeerde standaardfout van de t-toets van Student voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest1w_sterr (weight, value)
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeeld:

```
TTest1w_sterr( weight, value )
```

---

### Zie ook:

p *Een typisch t-test-rapport opstellen (page 521)*

### TTest1w\_t

**TTest1w\_t()** retourneert de geaggregeerde t-waarde voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest1w_t ( weight, value)
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

### Voorbeeld:

```
TTest1w_t( weight, value )
```

---

### Zie ook:

[p Een typisch t-test-rapport opstellen \(page 521\)](#)

### TTest1w\_upper

**TTest1w\_upper()** retourneert de geaggregeerde bovengrenswaarde van het-vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op student t-testen voor één steekproef waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
TTest1w_upper (weight, value [, sig])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
value	De steekproeven die worden geëvalueerd. Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
weight	Elke waarde in <b>value</b> kan een of meer keer worden geteld, afhankelijk van de corresponderende wegingsfactor in <b>weight</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
TTest1w_upper( weight, value )  
TTest1w_upper( weight, value, 0.005 )
```

**Zie ook:**

*p Een typisch t-test-rapport opstellen (page 521)*

## Z-testfuncties

Een statistisch onderzoek van twee steekproefgemiddelden. Een z-test met twee steekproeven onderzoekt of twee steekproeven van elkaar verschillen. Dit wordt over het algemeen gebruikt als twee normale verdelingen onbekende varianties hebben en als een experiment een grote steekproefgrootte heeft.

De volgende statistische z-testfuncties zijn gegroepeerd volgens het type invoergegevensreeksen dat van toepassing is op de functie.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

*Voorbeelden van het gebruik van z-test-functies (page 525)*

### Functies voor notatie met één kolom

De volgende functies zijn van toepassing op z-tests met eenvoudige invoergegevensreeksen.

ztest\_conf

**ZTest\_conf()** retourneert de geaggregeerde z-waarde voor een serie waarden.

```
ZTest_conf() retourneert de geaggregeerde z-waarde voor een serie waarden.  
(value [, sigma [, sig ]])
```

ztest\_dif

**ZTest\_dif()** retourneert het geaggregeerde gemiddelde verschil van de z-toets van een serie waarden.

```
ZTest_dif() retourneert het geaggregeerde gemiddelde verschil van de z-toets  
van een serie waarden. (value [, sigma])
```

ztest\_sig

**ZTest\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van z-toets op voor een serie waarden.

```
ZTest_sig() retourneert het geaggregeerde tweezijdige niveau van  
significantie van z-toets op voor een serie waarden. (value [, sigma])
```

ztest\_sterr

**ZTest\_sterr()** retourneert de geaggregeerde standaardfout van de z-toets voor een serie waarden.

```
ZTest_sterr() retourneert de geaggregeerde standaardfout van de z-toets voor  
een serie waarden. (value [, sigma])
```

ztest\_z

**ZTest\_z()** retourneert de geaggregeerde z-waarde voor een serie waarden.

```
ZTest_z() retourneert de geaggregeerde z-waarde voor een serie waarden.  
(value [, sigma])
```

ztest\_lower

**ZTest\_lower()** retourneert de geaggregeerde ondergrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

```
ZTest_lower() retourneert de geaggregeerde ondergrenswaarde van  
het vertrouwensinterval voor twee onafhankelijke series waarden. (grp, value  
[, sig [, eq_var]])
```

ztest\_upper

**ZTest\_upper()** retourneert de geaggregeerde bovengrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

```
ZTest_upper() retourneert de geaggregeerde bovengrenswaarde van  
het vertrouwensinterval voor twee onafhankelijke series waarden. (grp, value  
[, sig [, eq_var]])
```

### Functies voor notatie met twee gewogen kolommen

De volgende functies zijn van toepassing op z-tests waarbij de invoergegevensreeksen worden gegeven in de notatie met twee gewogen kolommen.

ztestw\_conf

**ZTestw\_conf()** retourneert de geaggregeerde z-waarde van het vertrouwensinterval voor een serie waarden.

```
ZTestw_conf() retourneert de geaggregeerde z-waarde van het  
vertrouwensinterval voor een serie waarden. (weight, value [, sigma [, sig]])
```

ztestw\_dif

**ZTestw\_dif()** retourneert het geaggregeerde gemiddelde verschil van de z-toets van een serie waarden.

```
ZTestw_dif() retourneert het geaggregeerde gemiddelde verschil van de z-toets  
van een serie waarden. (weight, value [, sigma])
```

ztestw\_lower

**ZTestw\_lower()** retourneert de geaggregeerde ondergrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

```
ZTestw_lower() retourneert de geaggregeerde ondergrenswaarde van  
het vertrouwensinterval voor twee onafhankelijke series waarden. (weight,  
value [, sigma])
```

ztestw\_sig

**ZTestw\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van z-toets op voor een serie waarden.

```
ZTestw_sig() retourneert het geaggregeerde tweezijdige niveau van  
significantie van z-toets op voor een serie waarden. (weight, value [,  
sigma])
```

ztestw\_sterr

**ZTestw\_sterr()** retourneert de geaggregeerde standaardfout van de z-toets voor een serie waarden.

```
ZTestw_sterr() retourneert de geaggregeerde standaardfout van de z-toets voor  
een serie waarden. (weight, value [, sigma])
```

ztestw\_upper

**ZTestw\_upper()** retourneert de geaggregeerde bovengrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

```
ZTestw_upper() retourneert de geaggregeerde bovengrenswaarde van  
het vertrouwensinterval voor twee onafhankelijke series waarden. (weight,  
value [, sigma])
```

ztestw\_z

**ZTestw\_z()** retourneert de geaggregeerde z-waarde voor een serie waarden.

**ZTestw\_z()** retourneert de geaggregeerde z-waarde voor een serie waarden.  
(weight, value [, sigma])

### ZTest\_z

**ZTest\_z()** retourneert de geaggregeerde z-waarde voor een serie waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

**ZTest\_z**(value[, sigma])

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af.
sigma	De standaarddeviatie kan, indien bekend, worden opgegeven in <b>sigma</b> . Als <b>sigma</b> wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeeld:

```
ZTest_z( value-Testvalue )
```

---

### Zie ook:

p *Voorbeelden van het gebruik van z-test-functies (page 525)*

### ZTest\_sig

**ZTest\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van z-toets op voor een serie waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
ZTest_sig(value[, sigma])
```

Retourgegevenstypen: numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af.
sigma	De standaarddeviatie kan, indien bekend, worden opgegeven in <b>sigma</b> . Als <b>sigma</b> wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeeld:

```
ZTest_sig(Value-TestValue)
```

---

### Zie ook:

*p Voorbeelden van het gebruik van z-test-functies (page 525)*

ZTest\_dif

**ZTest\_dif()** retourneert het geaggregeerde gemiddelde verschil van de z-toets van een serie waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
ZTest_dif(value[, sigma])
```



**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af.
sigma	De standaarddeviatie kan, indien bekend, worden opgegeven in <b>sigma</b> . Als <b>sigma</b> wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTest_dif(Value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 525)*

**ZTest\_sterr**

**ZTest\_sterr()** retourneert de geaggregeerde standaardfout van de z-toets voor een serie waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTest_sterr(value[, sigma])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af.
sigma	De standaarddeviatie kan, indien bekend, worden opgegeven in <b>sigma</b> . Als <b>sigma</b> wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTest_sterr(Value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 525)*

ZTest\_conf

**ZTest\_conf()** retourneert de geaggregeerde z-waarde voor een serie waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTest_conf (value[, sigma[, sig]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af.
sigma	De standaarddeviatie kan, indien bekend, worden opgegeven in <b>sigma</b> . Als <b>sigma</b> wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt.
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTest_conf(Value-TestValue)
```

---

**Zie ook:**

*p Voorbeelden van het gebruik van z-test-functies (page 525)*

ZTest\_lower

**ZTest\_lower()** retourneert de geaggregeerde ondergrenswaarde van het-vertrouwensinterval voor twee onafhankelijke series waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTest_lower (grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
ZTest_lower( Group, value )  
ZTest_lower( Group, value, sig, false )
```

---

**Zie ook:**

*p Voorbeelden van het gebruik van z-test-functies (page 525)*

ZTest\_upper

**ZTest\_upper()** retourneert de geaggregeerde bovengrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
ZTest_upper (grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
ZTest_upper( Group, value )  
ZTest_upper( Group, value, sig, false )
```

### Zie ook:

*p Voorbeelden van het gebruik van z-test-functies (page 525)*

### ZTestw\_z

**ZTestw\_z()** retourneert de geaggregeerde z-waarde voor een serie waarden.

Deze functie is van toepassing op z-testen waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
ZTestw_z (weight, value [, sigma])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De waarden moeten op <b>value</b> worden geretourneerd. Er wordt uitgegaan van een steekproefgemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u die waarde van de steekproefwaarden af.
weight	Elke steekproefwaarde in <b>value</b> kan één of meer keer worden geteld, afhankelijk van de corresponderende waarde van <b>weight</b> .
sigma	De standaarddeviatie kan, indien bekend, worden opgegeven in <b>sigma</b> . Als <b>sigma</b> wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeeld:

```
ZTestw_z( weight, value=TestValue)
```

---

### Zie ook:

*p Voorbeelden van het gebruik van z-test-functies (page 525)*

### ZTestw\_sig

**ZTestw\_sig()** retourneert het geaggregeerde tweezijdige niveau van significantie van z-toets op voor een serie waarden.

Deze functie is van toepassing op z-testen waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
ZTestw_sig (weight, value [, sigma])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De waarden moeten op <b>value</b> worden geretourneerd. Er wordt uitgegaan van een steekproefgemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u die waarde van de steekproefwaarden af.
weight	Elke steekproefwaarde in <b>value</b> kan één of meer keer worden geteld, afhankelijk van de corresponderende waarde van <b>weight</b> .
sigma	De standaarddeviatie kan, indien bekend, worden opgegeven in <b>sigma</b> . Als <b>sigma</b> wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeeld:

```
ZTestw_sig( weight, value-Testvalue)
```

### Zie ook:

p *Voorbeelden van het gebruik van z-test-functies (page 525)*

### ZTestw\_dif

**ZTestw\_dif()** retourneert het geaggregeerde gemiddelde verschil van de z-toets van een serie waarden.

Deze functie is van toepassing op z-testen waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
ZTestw_dif ( weight, value [, sigma])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
value	De waarden moeten op <b>value</b> worden geretourneerd. Er wordt uitgegaan van een steekproefgemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u die waarde van de steekproefwaarden af.
weight	Elke steekproefwaarde in <b>value</b> kan één of meer keer worden geteld, afhankelijk van de corresponderende waarde van <b>weight</b> .
sigma	De standaarddeviatie kan, indien bekend, worden opgegeven in <b>sigma</b> . Als <b>sigma</b> wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTestw_dif( weight, value-TestValue)
```

---

**Zie ook:**

*p Voorbeelden van het gebruik van z-test-functies (page 525)*

ZTestw\_sterr

**ZTestw\_sterr()** retourneert de geaggregeerde standaardfout van de z-toets voor een serie waarden.

Deze functie is van toepassing op z-testen waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTestw_sterr (weight, value [, sigma])
```



**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
value	De waarden moeten op <b>value</b> worden geretourneerd. Er wordt uitgegaan van een steekproefgemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u die waarde van de steekproefwaarden af.
weight	Elke steekproefwaarde in <b>value</b> kan één of meer keer worden geteld, afhankelijk van de corresponderende waarde van <b>weight</b> .
sigma	De standaarddeviatie kan, indien bekend, worden opgegeven in <b>sigma</b> . Als <b>sigma</b> wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTestw_sterr( weight, value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 525)*

ZTestw\_conf

**ZTestw\_conf()** retourneert de geaggregeerde z-waarde van het vertrouwensinterval voor een serie waarden.

Deze functie is van toepassing op z-testen waarbij de invoergegevensreeks wordt opgegeven in twee gewogen kolommen.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTest_conf( weight, value[, sigma[, sig]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. Er wordt uitgegaan van een populatiegemiddelde van 0. Als u met een ander gemiddelde wilt werken, trekt u dat gemiddelde van de steekproefwaarden af.
weight	Elke steekproefwaarde in <b>value</b> kan één of meer keer worden geteld, afhankelijk van de corresponderende waarde van <b>weight</b> .
sigma	De standaarddeviatie kan, indien bekend, worden opgegeven in <b>sigma</b> . Als <b>sigma</b> wordt weggelaten, wordt de huidige standaarddeviatie van de steekproef gebruikt.
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

**Voorbeeld:**

```
ZTestw_conf( weight, value-TestValue)
```

---

**Zie ook:**

p *Voorbeelden van het gebruik van z-test-functies (page 525)*

ZTestw\_lower

**ZTestw\_lower()** retourneert de geaggregeerde ondergrenswaarde van het vertrouwensinterval voor twee onafhankelijke series waarden.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

**Syntaxis:**

```
ZTestw_lower (grp, value [, sig [, eq_var]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingen leiden ertoe dat de functie NULL retourneert.

**Voorbeelden:**

```
ZTestw_lower( Group, Value )  
ZTestw_lower( Group, Value, sig, false )
```

---

**Zie ook:**

*p Voorbeelden van het gebruik van z-test-functies (page 525)*

ZTestw\_upper

**ZTestw\_upper()** retourneert de geaggregeerde bovengrenswaarde van het-vertrouwensinterval voor twee onafhankelijke series waarden.

Deze functie is van toepassing op student t-testen voor onafhankelijke steekproeven.

Als de functie wordt gebruikt in het script voor het laden van gegevens, worden de waarden herhaald over een aantal records, zoals gedefinieerd door een group by-clausule.

Als de functie wordt gebruikt in een diagramuitdrukking, worden de waarden herhaald over de diagramdimensies.

### Syntaxis:

```
ZTestw_upper (grp, value [, sig [, eq_var]])
```

Retourgegevenstypen: numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De steekproefwaarden die moeten worden geëvalueerd. De steekproefwaarden moeten logisch worden gegroepeerd zoals opgegeven door precies twee waarden in <b>group</b> . Als er geen veldnaam voor de steekproefwaarden in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Value</b> .
grp	Het veld met de namen van elk van de twee steekproefgroepen. Als er geen veldnaam voor de groep in het load-script wordt opgegeven, krijgt het veld automatisch de naam <b>Type</b> .
sig	Het tweezijdige niveau van significantie kan worden opgegeven in <b>sig</b> . Als dit wordt weggelaten, wordt <b>sig</b> ingesteld op 0.025, wat resulteert in een betrouwbaarheidsinterval van 95%.
eq_var	Als <b>eq_var</b> wordt opgegeven als False (0), wordt er uitgegaan van ongelijke varianties tussen de twee steekproeven. Als <b>eq_var</b> wordt opgegeven als True (1), wordt er uitgegaan van gelijke varianties tussen de twee steekproeven.

### Beperkingen:

Tekstwaarden, NULL-waarden en ontbrekende waarden in de uitdrukkingswaarde leiden ertoe dat de functie NULL retourneert.

### Voorbeelden:

```
ZTestw_upper( Group, Value )  
ZTestw_upper( Group, Value, sig, false )
```

### Zie ook:

*p Voorbeelden van het gebruik van z-test-functies (page 525)*

### Voorbeelden van statistische testfuncties

Dit gedeelte bevat voorbeelden van statistische testfuncties zoals deze worden toegepast op diagrammen en het script voor het laden van gegevens.

### Voorbeelden van het gebruik van chi2-test-functies in grafieken

De chi2-test-functies worden gebruikt om waarden te zoeken die verband houden met chi2 statistische analyse.

In deze sectie wordt beschreven hoe u visualisaties kunt samenstellen met behulp van steekproefgegevens om de waarden van de chi2-distributietestfuncties te vinden die beschikbaar zijn in Qlik Sense. Raadpleeg de individuele onderwerpen over chi2-test-diagramfuncties voor een beschrijving van syntaxis en argumenten.

### De gegevens voor de steekproeven laden


Er zijn drie sets steekproefgegevens die drie verschillende statistische steekproeven beschrijven die in het script moeten worden geladen.

Doe het volgende:

1. Maak een nieuwe app.
2. Voer het volgende in bij het laden van gegevens:

```
// Sample_1 data is pre-aggregated... Note: make sure you set your DecimalSep='.' at the
top of the script.
sample_1:
LOAD * inline [
Grp,Grade,Count
I,A,15
I,B,7
I,C,9
I,D,20
I,E,26
I,F,19
II,A,10
II,B,11
II,C,7
II,D,15
II,E,21
II,F,16
];
// Sample_2 data is pre-aggregated: If raw data is used, it must be aggregated using
count()...
sample_2:
LOAD * inline [
Sex,Opinion,OpCount
1,2,58
1,1,11
1,0,10
2,2,35
2,1,25
2,0,23 ] (delimiter is ',');
// Sample_3a data is transformed using the crosstable statement...
sample_3a:
crosstable(Gender, Actual) LOAD
Description,
[Men (Actual)] as Men,
[Women (Actual)] as Women;
LOAD * inline [
```



```
Men (Actual),Women (Actual),Description
58,35,Agree
11,25,Neutral
10,23,Disagree ] (delimiter is ',');
// Sample_3b data is transformed using the crosstable statement...
Sample_3b:
crosstable(Gender, Expected) LOAD
Description,
[Men (Expected)] as Men,
[Women (Expected)] as Women;
LOAD * inline [
Men (Expected),Women (Expected),Description
45.35,47.65,Agree
17.56,18.44,Neutral
16.09,16.91,Disagree ] (delimiter is ',');
// Sample_3a and Sample_3b will result in a (fairly harmless) Synthetic Key...
```

3. Klik op  om gegevens te laden.

### De visualisaties van de chi2-test-grafiekmakingsfuncties maken

#### Voorbeeld: Steekproef 1

Doe het volgende:

1. Klik in de editor voor laden van gegevens op  om naar het app-overzicht te gaan en vervolgens op het werkblad te klikken dat u eerder hebt gemaakt.  
De werkbladweergave wordt geopend.
2. Klik op  **Werkblad bewerken** om het werkblad te bewerken.
3. Voeg vanuit **Grafieken** een tabel toe en voeg vanuit **Velden** Grp, Grade en Count toe als dimensies.  
In deze tabel worden de steekproefgegevens weergegeven.
4. Voeg nog een tabel toe met de volgende uitdrukking als dimensie:  
`valueList('p', 'df', 'chi2')`  
Hierbij wordt de functie voor synthetische dimensies gebruikt om labels voor de dimensies te maken met de namen van de drie chi2-test-functies.
5. Voeg de volgende uitdrukking als meting toe aan de tabel:  
`IF(ValueList('p', 'df', 'chi2')='p',Chi2Test_p(Grp,Grade,Count),  
IF(ValueList('p', 'df', 'chi2')='df',Chi2Test_df(Grp,Grade,Count),  
Chi2Test_chi2(Grp,Grade,Count)))`  
Dit heeft als effect dat de resulterende waarde van elke chi2-test-functie in de tabel naast de bijbehorende synthetische dimensie wordt geplaatst.
6. Stel de **getalnotatie** van de meting in op **Getal** en **3 significante cijfers**.



*In de uitdrukking voor de meting zou u ook de volgende uitdrukking kunnen gebruiken: `Pick(Match(ValueList('p', 'df', 'chi2'), 'p', 'df', 'chi2'), Chi2Test_p(Grp, Grade, Count), Chi2Test_df(Grp, Grade, Count), Chi2Test_chi2(Grp, Grade, Count))`*

### Resultaat

De resulterende tabel voor de chi2-test-functies voor de gegevens van Steekproef 1 bevat de volgende waarden:

p	df	Chi2
0.820	5	2.21

### Voorbeeld: Steekproef 2

Doe het volgende:

1. In het werkblad dat u aan het bewerken was in het voorbeeld Steekproef 1, voegt u vanuit **Grafieken** een tabel toe en vanuit **Velden** Sex, Opinion en OpCount als dimensies.
2. Maak een kopie van de resultatentabel uit Steekproef 1 met behulp van de opdrachten **Kopiëren** en **Plakken**. Bewerk de uitdrukking in de meting en vervang de argumenten in alle drie de chi2-test-functies door de namen van de velden die worden gebruikt in de gegevens voor Steekproef 2, bijvoorbeeld: `chi2Test_p(Sex,Opinion,OpCount)`.

### Resultaat

De resulterende tabel voor de chi2-test-functies voor de gegevens van Steekproef 2 bevat de volgende waarden:

p	df	Chi2
0.000309	2	16.2

### Voorbeeld: Steekproef 3

Doe het volgende:

1. Maak nog twee extra tabellen op dezelfde manier als in de voorbeelden voor de gegevens van Steekproef 1 en Steekproef 2. Gebruik in de dimensietabel de volgende velden als dimensies: Gender, Description, Actual en Expected.
2. Gebruik in de resultatentabel de namen van de velden die worden gebruikt in de gegevens voor Steekproef 3, bijvoorbeeld: `chi2Test_p(Gender,Description,Actual,Expected)`.

### Resultaat

De resulterende tabel voor de chi2-test-functies voor de gegevens van Steekproef 3 bevat de volgende waarden:

p	df	Chi2
0.000308	2	16.2

Voorbeelden van het gebruik van chi2-test-functies in het load-script voor gegevens

De chi2-test-functies worden gebruikt om waarden te zoeken die verband houden met chi2 statistische analyse. In deze sectie wordt beschreven hoe u de chi2-distributietestfuncties die beschikbaar zijn in Qlik Sense kunt gebruiken in het script voor het laden van gegevens.

Raadpleeg de individuele onderwerpen over chi2-test-scriptfuncties voor een beschrijving van syntaxis en argumenten.

In dit voorbeeld wordt een tabel gebruikt die het aantal studenten die een graad hebben behaald (A-F) voor twee groepen studenten (I en II) bevat.

Data table

Group	A	B	C	D	E	F
I	15	7	9	20	26	19
II	10	11	7	15	21	16

### De steekproefgegevens laden

Doe het volgende:

1. Maak een nieuwe app.
2. Voer het volgende in de editor voor het laden van gegevens in:  

```
// sample_1 data is pre-aggregated... Note: make sure you set your DecimalSep='.' at the top of the script.
sample_1:
LOAD * inline [
Grp,Grade,Count
I,A,15
I,B,7
I,C,9
I,D,20
I,E,26
I,F,19
II,A,10
II,B,11
II,C,7
II,D,15
II,E,21
II,F,16
];
```
3. Klik op  om gegevens te laden.

U hebt nu de steekproefgegevens geladen.


### De waarden van de chi2-test-functie laden

Nu gaan we de chi2-test-waarden laden op basis van de steekproefgegevens in een nieuwe tabel, gegroepeerd op Grp.

Doe het volgende:



1. Voeg in de editor voor het laden van gegevens het volgende toe aan het uiteinde van het script:  

```
// Sample_1 data is pre-aggregated... Note: make sure you set your DecimalSep='.' at the top of the script.  
Chi2_table:  
LOAD Grp,  
Chi2Test_chi2(Grp, Grade, Count) as chi2,  
Chi2Test_df(Grp, Grade, Count) as df,  
Chi2Test_p(Grp, Grade, Count) as p  
resident Sample_1 group by Grp;
```
2. Klik op  om gegevens te laden.

U hebt nu de chi2-test-waarden geladen in een tabel met de naam Chi2\_table.

### Resultaten

U kunt de resulterende chi2-test-waarden bekijken in de gegevensmodelviewer onder **Voorbeeld**. Deze zouden er als volgt moeten uitzien:

Grp	chi2	df	p
I	16.00	5	0.007
II	9.40	5	0.094

### Een typisch t-test-rapport opstellen

Een typisch t-test-rapport voor cursisten kan tabellen met **Group Statistics** en resultaten van **Independent Samples Test** bevatten.

In de volgende secties gaan we deze tabellen bouwen met Qlik Sense-test-functies die worden toegepast op twee onafhankelijke groepen steekproeven, Observation en Comparison. De bijbehorende tabellen voor deze steekproeven zouden er als volgt uitzien:

Type	N	Mean	Standard Deviation	Standard Error Mean
Comparison	20	11.95	14.61245	3.2674431
Observation	20	27.15	12.507997	2.7968933

### Independent Sample Test

Independent Sample Test

Type	t	df	Sig. (2-tailed)	Mean Difference	Standard Error Difference	95% Confidence Interval of the Difference (Lower)	95% Confidence Interval of the Difference (Upper)
Equal Variance not Assumed	3.534	37.116717335823	0.001	15.2	4.30101	6.48625	23.9137
Equal Variance Assumed	3.534	38	0.001	15.2	4.30101	6.49306	23.9069


### De steekproefgegevens laden

Doe het volgende:

1. Maak een nieuwe app met een nieuw werkblad en open dat werkblad.
2. Voer het volgende in de editor voor laden van gegevens in:



```
Table1:
crosstable LOAD recno() as ID, * inline [
Observation|Comparison
35|2
40|27
12|38
15|31
21|1
14|19
46|1
10|34
28|3
48|1
16|2
30|3
32|2
48|1
31|2
22|1
12|3
39|29
19|37
25|2 ] (delimiter is '|');
```

In dit load-script is **recno()** opgenomen omdat voor **crosstable** drie argumenten vereist zijn. **recno()** levert daarom niet meer dan een extra argument. In dit geval een ID voor elke rij. Zonder dit argument zouden geen steekproefwaarden voor **Comparison** worden geladen.

3. Klik op  om gegevens te laden.

### De tabel Group Statistics maken

Doe het volgende:

1. Klik in de editor voor laden van gegevens op  om naar het app-overzicht te gaan en klik vervolgens op het werkblad dat u eerder hebt gemaakt. Hiermee wordt de werkbladweergave geopend.
2. Klik op  **Werkblad bewerken** om het werkblad te bewerken.
3. Voeg vanuit **Grafieken** een tabel toe en vanuit **Velden** de volgende uitdrukkingen als metingen:

Voorbeelduitdrukkingen

Label	Uitdrukking
N	Count(Value)
Mean	Avg(Value)
Standard Deviation	Stdev(Value)
Standard Error Mean	Sterr(Value)

4. Voeg Type toe als dimensie van de tabel.
5. Klik op **Sorteren** en verplaats Type naar het begin van de sorteerlijst.

### Resultaat


Een tabel Group Statistics voor deze steekproeven zou er als volgt uitzien:

Group statistics

Type	N	Mean	Standard Deviation	Standard Error Mean
Comparison	20	11.95	14.61245	3.2674431
Observation	20	27.15	12.507997	2.7968933

### De tabel Two Independent Sample Student's T-test maken

Doe het volgende:

1. Klik op  **Werkblad bewerken** om het werkblad te bewerken.
2. Voeg de volgende uitdrukking toe als een dimensie van de tabel. `=valueList (Dual('Equal variance not Assumed', 0), Dual('Equal Variance Assumed', 1))`

3. Voeg vanuit **Grafieken** een tabel toe met de volgende uitdrukkingen als metingen:

Voorbeelduitdrukkingen

Label	Uitdrukking
conf	if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_conf(Type, Value),TTest_conf(Type, Value, 0))
t	if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_t(Type, Value),TTest_t(Type, Value, 0))
df	if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_df(Type, Value),TTest_df(Type, Value, 0))
Sig. (2-tailed)	if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_sig(Type, Value),TTest_sig(Type, Value, 0))
Mean Difference	TTest_dif(Type, Value)
Standard Error Difference	if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_sterr(Type, Value),TTest_sterr(Type, Value, 0))
95% Confidence Interval of the Difference (Lower)	if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_lower(Type, Value,(1-(95)/100)/2),TTest_lower (Type, Value,(1-(95)/100)/2, 0))
95% Confidence Interval of the Difference (Upper)	if(ValueList (Dual('Equal Variance not Assumed', 0), Dual('Equal Variance Assumed', 1)),TTest_upper(Type, Value,(1-(95)/100)/2),TTest_upper (Type, Value,(1-(95)/100)/2, 0))

### Resultaat

Independent sample test

Type	t	df	Sig. (2-tailed)	Mean Difference	Standard Error Difference	95% Confidence Interval of the Difference (Lower)	95% Confidence Interval of the Difference (Upper)
Equal Variance not Assumed	3.534	37.116717335823	0.001	15.2	4.30101	6.48625	23.9137
Equal Variance Assumed	3.534	38	0.001	15.2	4.30101	6.49306	23.9069

### Voorbeelden van het gebruik van z-test-functies

De z-test-functies worden gebruikt om waarden te vinden die zijn gekoppeld aan de statistische z-test-analyse voor grote testgegevensvoorbeelden, gewoonlijk groter dan 30, waarbij de variantie bekend is.

In dit gedeelte wordt beschreven hoe visualisaties kunnen worden gebouwd aan de hand van de voorbeeldgegevens, om de waarden te vinden van de z-test-functies die beschikbaar zijn in Qlik Sense. Raadpleeg de individuele onderwerpen over z-test-diagramfuncties voor een beschrijving van syntaxis en argumenten.

### De steekproefgegevens laden

De voorbeeldgegevens die hier worden gebruikt, zijn dezelfde die worden gebruikt in de voorbeelden voor t-test-functies. De grootte van de hoeveelheid voorbeeldgegevens wordt gewoonlijk als te klein beschouwd voor z-testanalyse, maar is voldoende om het gebruik van de verschillende z-test-functies te illustreren in Qlik Sense.

Doe het volgende:

1. Maak een nieuwe app met een nieuw werkblad en open dat werkblad.




*Als u een app gemaakt hebt voor de t-test-functies, kunt u die gebruiken en een nieuw werkblad voor deze functies maken.*

2. Voer het volgende in de editor voor laden van gegevens in:

```
Table1:
crosstable LOAD recno() as ID, * inline [
observation|comparison
35|2
40|27
12|38
15|31
21|1
14|19
46|1
10|34
28|3
48|1
16|2
30|3
32|2
48|1
31|2
22|1
12|3
39|29
19|37
25|2 ] (delimiter is '|');
```



In dit load-script is **recno()** opgenomen omdat voor **crosstable** drie argumenten vereist zijn. **recno()** levert daarom niet meer dan een extra argument. In dit geval een ID voor elke rij. Zonder dit

argument zouden geen steekproefwaarden voor **Comparison** worden geladen.

3. Klik op  om gegevens te laden.

### Visualisaties voor z-test-diagramfuncties maken

Doe het volgende:

1. Klik in de editor voor laden van gegevens op  om naar de app-weergave te gaan en klik vervolgens op het werkblad dat u hebt gemaakt toen u de gegevens laadde. De werkbladweergave wordt geopend.
2. Klik op  **Werkblad bewerken** om het werkblad te bewerken.
3. Voeg vanuit **Diagrammen** een tabel toe en vanuit **Velden** een Type als een dimensie.
4. Voeg de volgende uitdrukkingen aan de tabel toe als metingen.

Voorbeelduitdrukkingen

Label	Uitdrukking
ZTest Conf	ZTest_conf(Value)
ZTest Dif	ZTest_dif(Value)
ZTest Sig	ZTest_sig(Value)
ZTest Sterr	ZTest_sterr(Value)
ZTest Z	ZTest_z(Value)



Mogelijk wilt u de getalnotatie van de metingen aanpassen om betekenisvolle waarden weer te geven. De tabel is gemakkelijker te lezen als u getalnotatie voor de meeste metingen instelt op **Getal>Eenvoudig**, in plaats van op **Auto**. Maar voor ZTest Sig gebruikt u bijvoorbeeld de getalnotatie: **Aangepast** en past u vervolgens het opmaakpatroon aan naar **###**.

### Resultaat

De resulterende tabel voor de z-test-functies voor de voorbeeldgegevens bevat de volgende waarden:

Resultatentabel

Type	ZTest Conf	ZTest Dif	ZTest Sig	ZTest Sterr	ZTest Z
Comparison	6.40	11.95	0.000123	3.27	3.66
Value	5.48	27.15	0.001	2.80	9.71

### Visualisaties voor z-testw-diagramfuncties maken

De z-testw-functies kunnen worden gebruikt als de invoergegevensreeks wordt opgegeven in twee gewogen kolommen. De expressies vereisen een waarde voor het argument weight. In de voorbeelden hier wordt overal de waarde 2 gebruikt, maar u kunt ook een uitdrukking gebruiken, die een waarde definieert voor weight voor elke observatie.

### Voorbeelden en resultaten:

Bij gebruik van dezelfde steekproefgegevens en getalnotatie als voor de z-test-functies, bevat de resulterende tabel voor de z-testw-functies de volgende waarden:

Type	ZTestw Conf	ZTestw Dif	ZTestw Sig	ZTestw Sterr	ZTestw Z
Comparison	3.53	2.95	5.27e-005	1.80	3.88
Value	2.97	34.25	0	4.52	20.49

## Aggregatiefuncties voor tekenreeksen

In deze sectie worden de aggregatiefuncties voor tekenreeksen beschreven.

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### Tekenreeksaggregatiefuncties in het script voor het laden van gegevens

#### Concat

**Concat()** wordt gebruikt om tekenreekswaarden te combineren. Deze scriptfunctie retourneert de geaggregeerde aaneengeschakelde tekenreeks van alle waarden van de uitdrukking, herhaald over een aantal records, zoals vastgelegd door een **group by**-clausule.

```
Concat ([ distinct ] expression [, delimiter [, sort-weight]])
```

#### FirstValue

**FirstValue()** retourneert de waarde die als eerste werd geladen vanuit de records die zijn gedefinieerd door de uitdrukking, gesorteerd door een **group by**-clausule.



*Deze functie is alleen beschikbaar als scriptfunctie.*

```
FirstValue (expression)
```

#### LastValue

**LastValue()** retourneert de waarde die als laatste werd geladen vanuit de records die zijn gedefinieerd door de uitdrukking, gesorteerd door een **group by**-clausule.



*Deze functie is alleen beschikbaar als scriptfunctie.*

```
LastValue (expression)
```

### MaxString

**MaxString()** vindt tekenreekswaarden in de uitdrukking en retourneert de laatste tekstwaarde alfabetisch gesorteerd over een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
MaxString (expression )
```

### MinString

**MinString()** vindt tekenreekswaarden in de uitdrukking en retourneert de eerste tekstwaarde alfabetisch gesorteerd over een aantal records, zoals gedefinieerd door een **group by**-clausule.

```
MinString (expression )
```

## Tekenreeksaggregatiefuncties in diagrammen

De volgende diagramfuncties zijn beschikbaar voor het aggregeren van tekenreeksen in diagrammen.

### Concat

**Concat()** wordt gebruikt om tekenreekswaarden te combineren. De functie retourneert de geaggregeerde aaneenschakeling van tekenreeksen van alle waarden van de uitdrukking in elke dimensie.

```
Concat - diagramfunctie ({ [SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]]  
string[, delimiter[, sort_weight]])
```

### MaxString

**MaxString()** vindt tekenreekswaarden in de uitdrukking of het veld en retourneert de laatste tekstwaarde in alfabetische sorteervolgorde.

```
MaxString - diagramfunctie ({ [SetExpression] [TOTAL [<fld{, fld}>]] } expr)
```

### MinString

**MinString()** vindt tekenreekswaarden in de uitdrukking of het veld en retourneert de eerste tekstwaarde in alfabetische sorteervolgorde.

```
MinString - diagramfunctie ({ [SetExpression] [TOTAL [<fld {, fld}>]] } expr)
```

### Concat

**Concat()** wordt gebruikt om tekenreekswaarden te combineren. Deze scriptfunctie retourneert de geaggregeerde aaneengeschakelde tekenreeks van alle waarden van de uitdrukking, herhaald over een aantal records, zoals vastgelegd door een **group by**-clausule.

### Syntaxis:

```
Concat ([ distinct ] string [, delimiter [, sort-weight]])
```

**Retourgegevenstypen:** tekenreeks

### Argumenten:

De uitdrukking die of het veld dat de tekenreeks bevat die moet worden verwerkt.



### Argumenten

Argument	Beschrijving
string	De uitdrukking die of het veld dat de tekenreeks bevat die moet worden verwerkt.
delimiter	Elke waarde kan worden gescheiden door de tekenreeks die is opgegeven in delimiter.
sort-weight	De volgorde bij aaneenschakelen kan worden bepaald door de waarde van de dimensie <b>sort-weight</b> , indien aanwezig, waarbij de tekenreeks die overeenkomt met de laagste waarde als eerste in de aaneenschakeling verschijnt.
distinct	Als het woord <b>distinct</b> voor de uitdrukking staat, worden alle duplicaten genegeerd.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

### Voorbeelden en resultaten

Voorbeeld	Resultaat	Resultaten nadat ze zijn toegevoegd aan een werkblad
<pre>TeamData: LOAD * inline [ SalesGroup Team Date Amount East Gamma 01/05/2013 20000 East Gamma 02/05/2013 20000 west Zeta 01/06/2013 19000 East Alpha 01/07/2013 25000 East Delta 01/08/2013 14000 west Epsilon 01/09/2013 17000 west Eta 01/10/2013 14000 East Beta 01/11/2013 20000 west Theta 01/12/2013 23000 ] (delimiter is ' ');  Concat1: LOAD SalesGroup,Concat(Team) as TeamConcat1 Resident TeamData Group By SalesGroup;</pre>	SalesGroup  East  West	TeamConcat1  AlphaBetaDeltaGammaGamma  EpsilonEtaThetaZeta
<p>Stel dat de tabel <b>TeamData</b> is geladen zoals in het vorige voorbeeld:</p> <pre>LOAD SalesGroup,Concat(distinct Team,'-') as TeamConcat2 Resident TeamData Group By SalesGroup;</pre>	SalesGroup  East  West	TeamConcat2  Alpha-Beta-Delta-Gamma  Epsilon-Eta-Theta-Zeta

Voorbeeld	Resultaat	Resultaten nadat ze zijn toegevoegd aan een werkblad
<p>Stel dat de tabel <b>TeamData</b> is geladen zoals in het vorige voorbeeld. Doordat het argument voor <b>sort-weight</b> wordt toegevoegd, worden de resultaten gerangschikt op basis van de waarde van de dimensie Amount:</p> <pre>LOAD SalesGroup,Concat(distinct Team,'-',Amount) as TeamConcat2 Resident TeamData Group By SalesGroup;</pre>	<p>SalesGroup</p> <p>East</p> <p>West</p>	<p>TeamConcat2</p> <p>Delta-Beta-Gamma-Alpha</p> <p>Eta-Epsilon-Zeta-Theta</p>

### Concat - diagramfunctie

**Concat()** wordt gebruikt om tekenreekswaarden te combineren. De functie retourneert de geaggregeerde aaneenschakeling van tekenreeksen van alle waarden van de uitdrukking in elke dimensie.

#### Syntaxis:

```
Concat ({ [SetExpression] [DISTINCT] [TOTAL [<fld{, fld}>]] } string[, delimiter [, sort_weight]])
```

**Retourgegevenstypen:** tekenreeks

#### Argumenten:

##### Argumenten

Argument	Beschrijving
string	De uitdrukking die of het veld dat de tekenreeks bevat die moet worden verwerkt.
delimiter	Elke waarde kan worden gescheiden door de tekenreeks die is opgegeven in delimiter.
sort-weight	De volgorde bij aaneenschakelen kan worden bepaald door de waarde van de dimensie <b>sort-weight</b> , indien aanwezig, waarbij de tekenreeks die overeenkomt met de laagste waarde als eerste in de aaneenschakeling verschijnt.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het woord <b>DISTINCT</b> voor de argumenten van de functie wordt geplaatst, worden duplicaten die het resultaat zijn van de evaluatie van de argumenten van de functie genegeerd.

Argument	Beschrijving
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Voorbeelden en resultaten:

Results table

SalesGroup	Amount	Concat(Team)	Concat(TOTAL <SalesGroup> Team)
East	25000	Alpha	AlphaBetaDeltaGammaGamma
East	20000	BetaGammaGamma	AlphaBetaDeltaGammaGamma
East	14000	Delta	AlphaBetaDeltaGammaGamma
West	17000	Epsilon	EpsilonEtaThetaZeta
West	14000	Eta	EpsilonEtaThetaZeta
West	23000	Theta	EpsilonEtaThetaZeta
West	19000	Zeta	EpsilonEtaThetaZeta

Voorbeelden van functies

Voorbeeld	Resultaat
Concat(Team)	De tabel wordt gconstrueerd op basis van de dimensies SalesGroup en Amount, en variaties op de meting Concat(Team). Wanneer het totalenresultaat wordt genegeerd, let dan op dat, hoewel er gegevens zijn voor acht waarden van Team verspreid over twee waarden van SalesGroup, is het enige resultaat van de meting Concat(Team) waarbij meer dan één tekenreekswaarde voor Team wordt aaneengeschakeld in de tabel de rij die de dimensie Amount 20000 bevat, met BetaGammaGamma als resultaat. Dit komt doordat er drie waarden zijn voor het Amount 20000 in de invoergegevens. Alle andere resultaten worden niet aaneengeschakeld als de meting de dimensies omvat omdat er slechts één waarde van Team is voor elke combinatie van SalesGroup en Amount.
Concat (DISTINCT Team, ', ')	Beta, Gamma, omdat de kwalificatie DISTINCT betekent dat het dubbele resultaat van Gamma buiten beschouwing wordt gelaten. Tevens wordt het argument voor het scheidingsteken gedefinieerd als een komma gevolgd door een spatie.

Voorbeeld	Resultaat
Concat (TOTAL <SalesGroup> Team)	Alle tekenreekswaarden voor alle waarden van Team worden aaneengeschakeld als de kwalificatie TOTAL wordt gebruikt. Als de veldselectie <SalesGroup> is opgegeven, worden de resultaten opgesplitst in de twee waarden van de dimensie SalesGroup. Voor de SalesGroupEast zijn de resultaten AlphaBetaDeltaGammaGamma. Voor de SalesGroupWest zijn de resultaten EpsilonEtaThetaZeta.
Concat (TOTAL <SalesGroup> Team, ';', Amount)	Door het argument voor <b>sort-weight</b> toe te voegen: Amount, worden de resultaten gerangschikt op basis van de waarde van de dimensie Amount. De resultaten worden DeltaBetaGammaGammaAlpha en EtaEpsilonZetaTheta.

Gebruikte gegevens in voorbeeld:

```
TeamData:
LOAD * inline [
SalesGroup|Team|Date|Amount
East|Gamma|01/05/2013|20000
East|Gamma|02/05/2013|20000
West|Zeta|01/06/2013|19000
East|Alpha|01/07/2013|25000
East|Delta|01/08/2013|14000
West|Epsilon|01/09/2013|17000
West|Eta|01/10/2013|14000
East|Beta|01/11/2013|20000
West|Theta|01/12/2013|23000
] (delimiter is '|');
```

### FirstValue

**FirstValue()** retourneert de waarde die als eerste werd geladen vanuit de records die zijn gedefinieerd door de uitdrukking, gesorteerd door een **group by**-clausule.



*Deze functie is alleen beschikbaar als scriptfunctie.*

#### Syntaxis:

```
FirstValue ( expr)
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

### Beperkingen:

Als er geen tekstwaarde wordt gevonden, is het resultaat NULL.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

#### Resulterende gegevens

Voorbeeld	Resultaat	Resultaten op een werkblad
<pre>TeamData: LOAD * inline [ SalesGroup Team Date Amount East Gamma 01/05/2013 20000 East Gamma 02/05/2013 20000 West Zeta 01/06/2013 19000 East Alpha 01/07/2013 25000 East Delta 01/08/2013 14000 West Epsilon 01/09/2013 17000 West Eta 01/10/2013 14000 East Beta 01/11/2013 20000 West Theta 01/12/2013 23000 ] (delimiter is ' ');  FirstValue1: LOAD SalesGroup,FirstValue(Team) as FirstTeamLoaded Resident TeamData Group By SalesGroup;</pre>	<p>SalesGroup</p> <p>East</p> <p>West</p>	<p>FirstTeamLoaded</p> <p>Gamma</p> <p>Zeta</p>

### LastValue

**LastValue()** retourneert de waarde die als laatste werd geladen vanuit de records die zijn gedefinieerd door de uitdrukking, gesorteerd door een **group by**-clausule.



*Deze functie is alleen beschikbaar als scriptfunctie.*

### Syntaxis:

```
LastValue ( expr )
```

**Retourgegevenstypen:** dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

### Beperkingen:

Als er geen tekstwaarde wordt gevonden, is het resultaat NULL.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in onze app toe om het resultaat te bekijken.

Als u dezelfde weergave wilt als in de onderstaande resultatenkolom, schakelt u in het eigenschappenvenster onder Sorteren over van Automatisch naar Aangepast en schakelt u numeriek en alfabetisch sorteren uit.

Voorbeeld	Resultaat	Resultaten met aangepast sorteren
<pre>TeamData: LOAD * inline [ SalesGroup Team Date Amount East Gamma 01/05/2013 20000 East Gamma 02/05/2013 20000 West Zeta 01/06/2013 19000 East Alpha 01/07/2013 25000 East Delta 01/08/2013 14000 West Epsilon 01/09/2013 17000 West Eta 01/10/2013 14000 East Beta 01/11/2013 20000 West Theta 01/12/2013 23000 ] (delimiter is ' ');  LastValue1: LOAD SalesGroup,LastValue(Team) as LastTeamLoaded Resident TeamData Group By SalesGroup;</pre>	SalesGroup East West	LastTeamLoaded Beta Theta

### MaxString

**MaxString()** vindt tekenreekswaarden in de uitdrukking en retourneert de laatste tekstwaarde alfabetisch gesorteerd over een aantal records, zoals gedefinieerd door een **group by**-clausule.

### Syntaxis:

```
MaxString ( expr )
```

Retourgegevenstypen: dual

### Argumenten:

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

### Beperkingen:

Als er geen tekstwaarde wordt gevonden, is het resultaat NULL.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Voorbeeld	Resultaat	
<pre>TeamData: LOAD * inline [ SalesGroup Team Date Amount East Gamma 01/05/2013 20000 East Gamma 02/05/2013 20000 West Zeta 01/06/2013 19000 East Alpha 01/07/2013 25000 East Delta 01/08/2013 14000 West Epsilon 01/09/2013 17000 West Eta 01/10/2013 14000 East Beta 01/11/2013 20000 West Theta 01/12/2013 23000 ] (delimiter is ' ');  Concat1: LOAD SalesGroup,MaxString(Team) as MaxString1 Resident TeamData Group By SalesGroup;</pre>	<p>SalesGroup</p> <p>East</p> <p>West</p>	<p>MaxString1</p> <p>Gamma</p> <p>Zeta</p>
<p>Stel dat de tabel <b>TeamData</b> wordt geladen zoals in het voorafgaande voorbeeld, en het load-script voor gegevens de opdracht SET bevat:</p> <pre>SET DateFormat='DD/MM/YYYY';  LOAD SalesGroup,MaxString(Date) as MaxString2 Resident TeamData Group By SalesGroup;</pre>	<p>SalesGroup</p> <p>East</p> <p>West</p>	<p>MaxString2</p> <p>01/11/2013</p> <p>01/12/2013</p>

### MaxString - diagramfunctie

**MaxString()** vindt tekenreekswaarden in de uitdrukking of het veld en retourneert de laatste tekstwaarde in alfabetische sorteervolgorde.

### Syntaxis:

```
MaxString ([SetExpression] [TOTAL [<fld{, fld}>]]) expr)
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {.fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

**Beperkingen:**

Als de uitdrukking geen waarden bevat met een tekenreeksrepresentatie, wordt NULL geretourneerd.

**Voorbeelden en resultaten:**

Resultatentabel

SalesGroup	Amount	MaxString(Team)	MaxString(Date)
East	14000	Delta	2013/08/01
East	20000	Gamma	2013/11/01
East	25000	Alpha	2013/07/01
West	14000	Eta	2013/10/01
West	17000	Epsilon	2013/09/01
West	19000	Zeta	2013/06/01
West	23000	Theta	2013/12/01



### Voorbeelden van functies

Voorbeeld	Resultaat
MaxString (Team)	Er zijn drie waarden van 20000 voor de dimensie Amount: twee voor Gamma (op verschillende datums) en één voor Beta. Het resultaat van de meting MaxString (Team) is daarom Gamma, omdat dit de hoogste waarde in de gesorteerde tekenreeksen is.
MaxString (Date)	2013/11/01 is de grootste waarde voor Date van de drie die aan de dimensie Amount zijn gekoppeld. Hierbij wordt ervan uitgegaan dat uw script de -opdracht bevat <pre>SET SET DateFormat='YYYY-MM-DD';'</pre>

Gebruikte gegevens in voorbeeld:

```
TeamData:
LOAD * inline [
SalesGroup|Team|Date|Amount
East|Gamma|01/05/2013|20000
East|Gamma|02/05/2013|20000
West|Zeta|01/06/2013|19000
East|Alpha|01/07/2013|25000
East|Delta|01/08/2013|14000
West|Epsilon|01/09/2013|17000
West|Eta|01/10/2013|14000
East|Beta|01/11/2013|20000
West|Theta|01/12/2013|23000
] (delimiter is '|');
```

### MinString

**MinString()** vindt tekenreekswaarden in de uitdrukking en retourneert de eerste tekstwaarde alfabetisch gesorteerd over een aantal records, zoals gedefinieerd door een **group by**-clausule.

**Syntaxis:**

```
MinString ( expr )
```

**Retourgegevenstypen:** dual

**Argumenten:**

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

**Beperkingen:**

Als er geen tekstwaarde wordt gevonden, is het resultaat NULL.

### Voorbeelden en resultaten:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

Resulterende gegevens

Voorbeeld	Resultaat	
<pre>TeamData: LOAD * inline [ SalesGroup Team Date Amount East Gamma 01/05/2013 20000 East Gamma 02/05/2013 20000 west Zeta 01/06/2013 19000 East Alpha 01/07/2013 25000 East Delta 01/08/2013 14000 West Epsilon 01/09/2013 17000 west Eta 01/10/2013 14000 East Beta 01/11/2013 20000 west Theta 01/12/2013 23000 ] (delimiter is ' ');  Concat1: LOAD SalesGroup,MinString(Team) as MinString1 Resident TeamData Group By SalesGroup;</pre>	SalesGroup East West	MinString1 Alpha Epsilon
<p>Stel dat de tabel <b>TeamData</b> wordt geladen zoals in het voorafgaande voorbeeld en het load-script voor gegevens de opdracht SET bevat:</p> <pre>SET DateFormat='DD/MM/YYYY';  LOAD SalesGroup,MinString(Date) as MinString2 Resident TeamData Group By SalesGroup;</pre>	SalesGroup East West	MinString2 01/05/2013 01/06/2013

### MinString - diagramfunctie

**MinString()** vindt tekenreekswaarden in de uitdrukking of het veld en retourneert de eerste tekstwaarde in alfabetische sorteervolgorde.

#### Syntaxis:

```
MinString ({ [SetExpression] [TOTAL [<fld {, fld}>]]} expr)
```

Retourgegevenstypen: dual

#### Argumenten:

Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

Argument	Beschrijving
SetExpression	De aggregatiefunctie aggregereert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
TOTAL	<p>Als het woord <b>TOTAL</b> voor de functieargumenten staat, wordt de berekening gemaakt op basis van alle mogelijke waarden bij de huidige selecties, en niet alleen de selecties die betrekking hebben op de huidige dimensiewaarde. Dit betekent dat de dimensies van het diagram worden genegeerd.</p> <p>Wanneer u <b>TOTAL [&lt;fld {fld}&gt;]</b> gebruikt en de <b>TOTAL</b>-kwalificatie wordt gevolgd door een lijst van één of meer veldnamen als subset van de dimensievariabelen van het diagram, maakt u een subset van de totale mogelijke waarden.</p>

### Voorbeelden en resultaten:

#### Voorbeeldgegevens

SalesGroup	Amount	MinString(Team)	MinString(Date)
East	14000	Delta	2013/08/01
East	20000	Beta	2013/05/01
East	25000	Alpha	2013/07/01
West	14000	Eta	2013/10/01
West	17000	Epsilon	2013/09/01
West	19000	Zeta	2013/06/01
West	23000	Theta	2013/12/01

#### Voorbeelden van functies

Voorbeelden	Resultaten
MinString (Team)	Er zijn drie waarden van 20000 voor de dimensie Amount: twee voor Gamma (op verschillende datums) en één voor Beta. Het resultaat van de meting MinString (Team) is daarom Beta, omdat dit de eerste waarde in de gesorteerde tekenreeksen is.
MinString (Date)	2013/11/01 is de eerste waarde voor Date van de drie die aan de dimensie Amount zijn gekoppeld. Hierbij wordt ervan uitgegaan dat uw script de -opdracht bevat SET DateFormat='YYYY-MM-DD';'

### Gebruikte gegevens in voorbeeld:

```
TeamData:
LOAD * inline [
SalesGroup|Team|Date|Amount
East|Gamma|01/05/2013|20000
East|Gamma|02/05/2013|20000
west|Zeta|01/06/2013|19000
```

```
East|Alpha|01/07/2013|25000
East|Delta|01/08/2013|14000
West|Epsilon|01/09/2013|17000
West|Eta|01/10/2013|14000
East|Beta|01/11/2013|20000
West|Theta|01/12/2013|23000
] (delimiter is '|');
```

### Functies voor synthetische dimensies

Een synthetische dimensie wordt in de app gemaakt met waarden die zijn gegenereerd via de functies voor synthetische dimensies en niet rechtstreeks met behulp van velden in het gegevensmodel. Wanneer waarden die zijn gegenereerd door een functie voor synthetische dimensies in een diagram worden gebruikt als een berekende dimensie, levert dit een synthetische dimensie op. Met synthetische dimensies kunt u bijvoorbeeld diagrammen maken met dimensies met waarden die voortkomen uit uw gegevens, oftewel dynamische dimensies.



*Synthetische dimensies worden niet beïnvloed door selecties.*

De volgende functies voor synthetische dimensies kunnen in diagrammen worden gebruikt.

#### ValueList

**ValueList()** retourneert een lijst met waarden die bij gebruik in een berekende dimensie een synthetische dimensie vormen.

```
ValueList - diagramfunctie (v1 {, Expression})
```

#### ValueLoop

**ValueLoop()** retourneert een set herhaalde waarden die bij gebruik in een berekende dimensie een synthetische dimensie vormen.

```
ValueLoop - diagramfunctie (from [, to [, step ]])
```

#### ValueList - diagramfunctie

**ValueList()** retourneert een lijst met waarden die bij gebruik in een berekende dimensie een synthetische dimensie vormen.



*In diagrammen met een synthetische dimensie die is gemaakt met de functie **ValueList** kunt u verwijzen naar de dimensiewaarde die hoort bij een specifieke uitdrukking, door de functie **ValueList** met dezelfde parameters in de diagramuitdrukking nogmaals op te nemen. De functie kan natuurlijk overal in de lay-out worden gebruikt, maar behalve voor synthetische dimensies heeft deze functie alleen nut binnen een aggregatiefunctie.*



*Synthetische dimensies worden niet beïnvloed door selecties.*

#### Syntaxis:

```
ValueList (v1 {, ...})
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

Argument	Beschrijving
v1	Statische waarde (gewoonlijk een tekenreeks, maar kan een getal zijn).
{,...}	Optionele lijst van statische waarden.

**Voorbeelden en resultaten:**

Voorbeelden van functies

Voorbeeld	Resultaat																																				
<pre>ValueList ('Number of Orders', 'Average Order Size', 'Total Amount')</pre>	<p>Als deze bijvoorbeeld worden gebruikt voor het maken van een dimensie in een tabel, leidt dit tot de drie tekensreekswaarden als rijlabels in de tabel. Hier kan vervolgens naar worden verwezen in een uitdrukking.</p>																																				
<pre>=IF( ValueList ('Number of Orders', 'Average Order Size', 'Total Amount') = 'Number of Orders', count (SaleID), IF( ValueList ('Number of Orders', 'Average Order Size', 'Total Amount') = 'Average Order Size', avg (Amount), sum (Amount) ))</pre>	<p>Deze uitdrukking haalt de waarden uit de gemaakte dimensie en verwijst ernaar in een genestelf-opdracht als invoer voor drie aggregatiefuncties:</p> <table border="1"> <thead> <tr> <th colspan="4">ValueList()</th> </tr> <tr> <th>Created dimension</th> <th>Year</th> <th>Added expression</th> <th></th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td>522.00</td> </tr> <tr> <td>Number of Orders</td> <td>2012</td> <td></td> <td>5.00</td> </tr> <tr> <td>Number of Orders</td> <td>2013</td> <td></td> <td>7.00</td> </tr> <tr> <td>Average Order Size</td> <td>2012</td> <td></td> <td>13.20</td> </tr> <tr> <td>Average Order Size</td> <td>2013</td> <td></td> <td>15.43</td> </tr> <tr> <td>Total Amount</td> <td>2012</td> <td></td> <td>66.00</td> </tr> <tr> <td>Total Amount</td> <td>2013</td> <td></td> <td>108.00</td> </tr> </tbody> </table>	ValueList()				Created dimension	Year	Added expression					522.00	Number of Orders	2012		5.00	Number of Orders	2013		7.00	Average Order Size	2012		13.20	Average Order Size	2013		15.43	Total Amount	2012		66.00	Total Amount	2013		108.00
ValueList()																																					
Created dimension	Year	Added expression																																			
			522.00																																		
Number of Orders	2012		5.00																																		
Number of Orders	2013		7.00																																		
Average Order Size	2012		13.20																																		
Average Order Size	2013		15.43																																		
Total Amount	2012		66.00																																		
Total Amount	2013		108.00																																		

**Gegevens die worden gebruikt in voorbeelden:**

```
SalesPeople:
LOAD * INLINE [
SalesID|SalesPerson|Amount|Year
1|1|12|2013
2|1|23|2013
3|1|17|2013
4|2|9|2013
5|2|14|2013
6|2|29|2013
```

```
7|2|4|2013
8|1|15|2012
9|1|16|2012
10|2|11|2012
11|2|17|2012
12|2|7|2012
] (delimiter is '|');
```

### ValueLoop - diagramfunctie

ValueLoop() retourneert een set herhaalde waarden die bij gebruik in een berekende dimensie een synthetische dimensie vormen.

De gegenereerde waarden beginnen met de **from**-waarde en eindigen met de **to**-waarde, met tussenliggende waarden gemaakt in stappen ter grootte van step.



*In diagrammen met een synthetische dimensie die is gemaakt met de functie **ValueLoop** kunt u verwijzen naar de dimensiewaarde die hoort bij een specifieke uitdrukking, door de functie **ValueLoop** met dezelfde parameters in de diagramuitdrukking nogmaals op te nemen. De functie kan natuurlijk overal in de lay-out worden gebruikt, maar behalve voor synthetische dimensies heeft deze functie alleen nut binnen een aggregatiefunctie.*



*Synthetische dimensies worden niet beïnvloed door selecties.*

#### Syntaxis:

```
ValueLoop (from [, to [, step ]])
```

**Retourgegevenstypen:** dual

**Argumenten:**

#### Argumenten

Argumenten	Beschrijving
from	Beginwaarde in de waardeset die wordt gegenereerd.
to	Eindwaarde in de waardeset die wordt gegenereerd.
step	Grootte van de stap tussen waarden.

**Voorbeelden en resultaten:**

#### Voorbeelden van functies

Voorbeeld	Resultaat
ValueLoop (1, 10)	Hiermee wordt een dimensie in een tabel gemaakt, bijvoorbeeld om te gebruiken als genummerde labels. Dit voorbeeld leidt tot waarden genummerd van 1 tot 10. Naar deze waarden kan vervolgens in een uitdrukking worden verwezen.
ValueLoop (2, 10,2)	Dit voorbeeld leidt tot waarden met de nummers 2, 4, 6, 8 en 10 omdat het argument step een waarde van 2 heeft.

### Geneste aggregaties

U kunt situaties tegengekomen waarbij u een aggregatie moet toepassen op het resultaat van een andere aggregatie. Dit wordt het nesten van aggregaties genoemd.

U kunt aggregaties in de meeste diagramuitdrukkingen niet nesten. U kunt aggregaties echter wel nesten als u de kwalificatie **TOTAL** gebruikt in de binnenste aggregatiefunctie.



*Er zijn geneste aggregaties van maximaal 100 niveaus toegestaan.*

### Geneste aggregaties met de kwalificatie TOTAL

#### Voorbeeld:

U wilt de som van het veld **Sales** berekenen, maar alleen transacties opnemen met een **OrderDate** die gelijk zijn aan die van vorig jaar. Het afgelopen jaar kan worden verkregen via de aggregatiefunctie **Max (TOTAL Year (OrderDate))**.

De volgende aggregatie zou het gewenste resultaat retourneren:

```
Sum(If(Year(OrderDate)=Max(TOTAL Year(OrderDate)), Sales))
```

Voor Qlik Sense moet de kwalificatie **TOTAL** worden opgenomen voor dit type van nesten. Dit is vereist voor de gewenste vergelijking. Dit soort geneste aggregaties komt vrij vaak voor en vormt een goede manier van werken.

---

#### Zie ook:

*p Aggr - diagramfunctie (page 543)*

## 5.3 Aggr - diagramfunctie

**Aggr()** retourneert een reeks van waarden voor de uitdrukking die is berekend via de opgegeven dimensie of dimensies. Bijvoorbeeld de maximale waarden van verkopen, per klant, per regio.

De functie **Aggr** wordt gebruikt voor geneste aggregaties, waarbij de eerste parameter (de binnenste aggregatie) één keer per dimensiewaarde wordt berekend. De dimensies worden opgegeven in de tweede parameter (en opeenvolgende parameters).

Bovendien moet de functie **Aggr** worden ingesloten in een buitenste aggregatiefunctie met behulp van de reeks resultaten van de functie **Aggr** als invoer voor de aggregatie waarin de functie is genest.

#### Syntaxis:

```
Aggr ({SetExpression} [DISTINCT] [NODISTINCT] expr, StructuredParameter{, StructuredParameter})
```

Retourgegevenstypen: dual

Argumenten:

### Argumenten

Argument	Beschrijving
expr	Een uitdrukking die bestaat uit een aggregatiefunctie. De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie.
StructuredParameter	<p>StructuredParameter bestaat uit een dimensie en, optioneel, sorteercriteria in de notatie: (Dimension(Sort-type, ordering))</p> <p>De dimensie is een enkel veld en kan geen uitdrukking zijn. Aan de hand van de dimensie wordt de reeks waarden bepaald waarvoor de uitdrukking Aggr wordt berekend.</p> <p>Als sorteercriteria zijn opgenomen, wordt de reeks waarden die door de functie Aggr wordt gemaakt en voor de dimensie wordt berekend, gesorteerd. Dit is belangrijk wanneer de sorteervolgorde van invloed is op het resultaat van de uitdrukking waarin de functie Aggr is opgenomen.</p> <p>Voor details over het gebruik van sorteercriteria raadpleegt u <a href="#">Sorteercriteria toevoegen aan de dimensie in de gestructureerde parameter</a>.</p>
SetExpression	De aggregatiefunctie aggregeert standaard over de set mogelijke records die wordt gedefinieerd door de selectie. Met de uitdrukking Set-analyse kunt u een alternatieve set records definiëren.
DISTINCT	Als het argument van de uitdrukking wordt voorafgegaan door de kwalificatie <b>distinct</b> of als er geen kwalificatie wordt gebruikt, genereert elke afzonderlijke combinatie van dimensiewaarden slechts één retourwaarde. Dit is de normale manier om aggregaties te maken: elke afzonderlijke combinatie van dimensiewaarden levert één lijn in het diagram op.
NODISTINCT	Als het argument van de uitdrukking wordt voorafgegaan door de kwalificatie <b>nodistinct</b> , kan elke combinatie van dimensiewaarden meer dan één retourwaarde genereren, afhankelijk van de onderliggende gegevensstructuur. Als er slechts één dimensie is, retourneert de functie <b>aggr</b> een reeks met hetzelfde aantal elementen als er rijen zijn in de brongegevens.

Elementaire aggregatiefuncties, zoals **Sum**, **Min** en **Avg**, retourneren een enkele numerieke waarde, terwijl de functie Aggr() kan worden vergeleken met het maken van een tijdelijke resultatenset (een virtuele tabel) waarover een andere aggregatie kan worden uitgevoerd. Zo kunt u bijvoorbeeld een gemiddelde verkoopwaarde berekenen door met een **Aggr()**-instructie de verkopen per klant op te tellen en daarna het gemiddelde van de opgetelde resultaten te berekenen: **Avg(TOTAL Aggr(Sum(Sales),Customer))**.





Gebruik de functie `Aggr()` in berekende dimensies als u geneste diagramaggregaties op meerdere niveaus wilt maken.

### Beperkingen:

Elke dimensie in een `Aggr()`-functie moet een enkel veld zijn en kan geen uitdrukking (berekende dimensie) zijn.

### Sorteercriteria toevoegen aan de dimensie in de gestructureerde parameter

In zijn basisvorm is het argument `StructuredParameter` in de syntaxis van de functie `Aggr` een enkele dimensie. De uitdrukking `Aggr(Sum(Sales, Month))` levert de totale verkoopwaarde voor elke maand op. Bij insluiting in een andere aggregatiefunctie kunnen er onverwachte resultaten optreden tenzij sorteercriteria worden gebruikt. Dit komt doordat sommige dimensies numeriek of alfabetisch kunnen worden gesorteerd, enzovoort.

In het argument `StructuredParameter` in de functie `Aggr` kunt u sorteercriteria opgeven voor de dimensie in uw uitdrukking. Op die manier legt u een sorteervolgorde op aan de virtuele tabel die door de functie `Aggr` wordt gemaakt.

Het argument `StructuredParameter` heeft de volgende syntaxis:

```
(FieldName, (Sort-type, Ordering))
```

Gestructureerde parameters kunnen worden genest:

```
(FieldName, (FieldName2, (Sort-type, Ordering)))
```

Het sorteertype kan zijn: `NUMERIC`, `TEXT`, `FREQUENCY` of `LOAD_ORDER`.

De volgende ordeningstypen zijn aan elke sorteertype gekoppeld:

Toegestane ordeningstypen

Sorteertype	Toegestane ordeningstypen
NUMERIC	ASCENDING, DESCENDING of REVERSE
TEXT	ASCENDING, A2Z, DESCENDING, REVERSE of Z2A
FREQUENCY	DESCENDING, REVERSE of ASCENDING
LOAD_ORDER	ASCENDING, ORIGINAL, DESCENDING of REVERSE

De ordeningstypen `REVERSE` en `DESCENDING` zijn equivalent.

Voor sorteertype `TEXT` zijn de ordeningstypen `ASCENDING` en `A2Z` equivalent en zijn `DESCENDING`, `REVERSE` en `Z2A` equivalent.

Voor sorteertype `LOAD_ORDER` zijn de ordeningstypen `ASCENDING` en `ORIGINAL` equivalent.

### Voorbeelden: Diagramuitdrukkingen die gebruikmaken van Aggr

Voorbeelden - Diagramuitdrukkingen

#### Diagramuitdrukking voorbeeld 1

##### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
ProductData: LOAD * inline [ Customer|Product|UnitsSales|UnitPrice Astrida|AA|4|16  
Astrida|AA|10|15 Astrida|BB|9|9 Betacab|BB|5|10 Betacab|CC|2|20 Betacab|DD|25|25  
Canutility|AA|8|15 Canutility|CC|0|19 ] (delimiter is '|');
```

##### Diagramuitdrukking

Maak een KPI-visualisatie in een Qlik Sense-werkblad. Voeg de volgende uitdrukking als meting toe aan de KPI:

```
Avg(Aggr(Sum(UnitsSales*UnitPrice), Customer))
```

##### Resultaat

376.7

##### Uitleg

De uitdrukking `Aggr(Sum(UnitsSales*UnitPrice), Customer)` levert de totale verkoopwaarde per **Customer** op en retourneert een reeks waarden: 295, 715 en 120 voor de drie **Customer**-waarden.

Het komt erop neer dat wij een tijdelijke lijst met waarden hebben samengesteld zonder een expliciete tabel of kolom te hoeven maken die deze waarden bevat.

Deze waarden worden gebruikt als invoer voor de functie **Avg()** om de gemiddelde verkoopwaarde te berekenen: 376.7.

#### Diagramuitdrukking voorbeeld 2

##### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
ProductData: LOAD * inline [ Customer|Product|UnitsSales|UnitPrice Astrida|AA|4|16  
Astrida|AA|10|15 Astrida|BB|10|15 Astrida|BB|9|9 Betacab|BB|5|10 Betacab|BB|7|12  
Betacab|CC|2|22 Betacab|CC|4|20 Betacab|DD|25|25 Canutility|AA|8|15 Canutility|AA|5|11  
Canutility|CC|0|19 ] (delimiter is '|');
```

##### Diagramuitdrukking

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Customer**, **Product**, **UnitPrice** en **UnitSales** als dimensies. Voeg de volgende uitdrukking als meting toe aan de tabel:

```
Aggr(NODISTINCT Max(UnitPrice), Customer, Product)
```

### Resultaat

Customer	Product	UnitPrice	UnitSales	Aggr(NODISTINCT Max(UnitPrice), Customer, Product)
Astrida	AA	15	10	16
Astrida	AA	16	4	16
Astrida	BB	9	9	15
Astrida	BB	15	10	15
Betacab	BB	10	5	12
Betacab	BB	12	7	12
Betacab	CC	20	4	22
Betacab	CC	22	2	22
Betacab	DD	25	25	25
Canutility	AA	11	5	15
Canutility	AA	15	8	15
Canutility	CC	19	0	19

### Uitleg

Een array van waarden: 16, 16, 15, 15, 12, 12, 22, 22, 25, 15, 15 en 19. De kwalificatie **nodistinct** betekent dat de reeks één element bevat voor elke rij in de brongegevens: elk element is de maximale **UnitPrice** per **Customer** en **Product**.

### Diagramuitdrukking voorbeeld 3

#### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
set vNumberOfOrders = 1000; OrderLines: Load RowNo() as OrderLineID, OrderID, OrderDate,
Round((Year(OrderDate)-2005)*1000*Rand()*Rand()*Rand1) as Sales while Rand()<=0.5 or IterNo
()=1; Load * where OrderDate<=Today(); Load Rand() as Rand1, Date(MakeDate(2013)+Floor
((365*4+1)*Rand())) as OrderDate, RecNo() as OrderID Autogenerate vNumberOfOrders;
Calendar: Load distinct Year(OrderDate) as Year, Month(OrderDate) as Month, OrderDate
Resident OrderLines;
```

#### Diagramuitdrukkingen

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Year** en **Month** als dimensies. Voeg de volgende uitdrukkingen aan de tabel toe als metingen:

- `Sum(Sales)`
- `Sum(Aggr( Rangesum(Above(Sum(Sales),0,12)), (Year, (Numeric, Ascending)), (Month, (Numeric, Ascending)) ))` gelabeld als `Structured Aggr()` in de tabel.

### Resultaat

Year	Month	Sum(Sales)	Structured Aggr()
2013	Jan	53495	53495
2013	Feb	48580	102075
2013	Mar	25651	127726
2013	Apr	36585	164311
2013	May	61211	225522
2013	Jun	23689	249211
2013	Jul	42311	291522
2013	Aug	41913	333435
2013	Sep	28886	362361
2013	Oct	25977	388298
2013	Nov	44455	432753
2013	Dec	64144	496897
2014	Jan	67775	67775

### Uitleg

Dit voorbeeld geeft de geaggregeerde waarden weer over een periode van twaalf maanden voor elk jaar in chronologisch oplopende volgorde, en daarmee het deel met de gestructureerde parameters (Numeriek, Oplopend) van de uitdrukking **Aggr()**. Twee specifieke dimensies zijn vereist als gestructureerde parameters: **Year** en **Month**, gesorteerd (1) **Year** (numeriek) en (2) **Month** (numeriek). Deze twee dimensies moeten in de visualisatie van de tabel of het diagram worden gebruikt. Dit is nodig zodat de lijst met dimensies van de functie **Aggr()** overeenkomt met de dimensies van het object dat in de visualisatie wordt gebruikt.

U kunt de resultaten van deze metingen vergelijken in een tabel of in afzonderlijke lijndiagrammen:

- `Sum(Aggr( Rangesum(Above(Sum(Sales),0,12)), (Year), (Month) ))`
- `Sum(Aggr( Rangesum(Above(Sum(Sales),0,12)), (Year, (Numeric, Ascending)), (Month, (Numeric, Ascending)) ))`

Het moet duidelijk zijn dat de gewenste accumulatie van geaggregeerde waarden door de laatstgenoemde uitdrukking wordt uitgevoerd.

Zie ook:

p *Basisaggregatiefuncties (page 325)*

### 5.4 Kleurfuncties

Deze functies kunnen worden gebruikt in uitdrukkingen die verband houden met het instellen en evalueren van de kleureigenschappen van diagramobjecten, alsmede in load-scripts voor gegevens.



*Qlik Sense ondersteunt de kleurfuncties **Color()**, **qliktechblue** en **qliktechgray** om redenen van achterwaartse compatibiliteit, maar het gebruik ervan wordt niet aanbevolen.*

ARGB

**ARGB()** wordt gebruikt in uitdrukkingen om de kleureigenschappen van een diagramobject in te stellen of te evalueren, waarbij de kleur wordt gedefinieerd door een rode component **r**, een groene component **g** een blauwe component **b**, met een alfactor (ondoorzichtigheid) van **alpha**.

```
ARGB (alpha, r, g, b)
```

HSL

**HSL()** wordt gebruikt in uitdrukkingen om de kleureigenschappen van een diagramobject in te stellen of te evalueren, waarbij de kleur wordt gedefinieerd door waarden van **hue**, **saturation** en **luminosity** tussen 0 en 1.

```
HSL (hue, saturation, luminosity)
```

RGB

**RGB()** retourneert een geheel getal dat overeenkomt met de kleurcode van de kleur die is gedefinieerd door de drie parameters: de rode component **r**, de groene component **g** en de blauwe component **b**. Deze componenten moeten beschikken over gehele getalwaarden tussen 0 en 255. De functie kan worden gebruikt in uitdrukkingen voor het instellen of evalueren van de kleureigenschappen van een diagramobject.

```
RGB (r, g, b)
```

Colormix1

**Colormix1()** wordt gebruikt in uitdrukkingen om een ARGB-kleurrepresentatie van een kleurovergang van twee kleuren te retourneren, gebaseerd op een waarde tussen 0 en 1.

```
Colormix1 (Value , ColorZero , ColorOne)
```

Value is een reëel getal tussen 0 en 1.

- Als Value = 0 wordt ColorZero geretourneerd.
- Als Value = 1 wordt ColorOne geretourneerd.
- Als 0 < Value < 1 is, wordt de bijbehorende tussenliggende schaduwkleur geretourneerd.

ColorZero is een geldige RGB-kleurrepresentatie voor de kleur die moet worden toegewezen aan het lage uiteinde van het interval.

ColorOne is een geldige RGB-kleurrepresentatie voor de kleur die moet worden toegewezen aan het hoge uiteinde van het interval.

### Voorbeeld:

```
colormix1(0.5, red(), blue())
```

retourneert:

```
ARGB(255,64,0,64) (purple)
```

### Colormix2

**Colormix2()** wordt gebruikt in om een ARGB-kleurrepresentatie van een kleurovergang van twee kleuren gebaseerd op een waarde tussen -1 en 1, waarbij het mogelijk is een tussenliggende kleur op te geven voor de centrale (0) positie.

```
Colormix2 (Value ,ColorMinusOne , ColorOne[ , ColorZero])
```

Value is een reëel getal tussen -1 en 1.

- Als Value = -1, wordt de eerste kleur geretourneerd.
- Als Value = 1, wordt de tweede kleur geretourneerd.
- Als  $-1 < \text{Value} < 1$  is, wordt de bijbehorende tussenliggende kleurenmix geretourneerd.

ColorMinusOne is een geldige RGB-kleurrepresentatie voor de kleur die moet worden toegewezen aan het lage uiteinde van het interval.

ColorOne is een geldige RGB-kleurrepresentatie voor de kleur die moet worden toegewezen aan het hoge uiteinde van het interval.

ColorZero is een optionele geldige RGB-kleurrepresentatie voor de kleur die moet worden toegewezen aan het midden van de interval.

### SysColor

**SysColor()** retourneert de ARGB-kleurrepresentatie voor de Windows-systeemkleur nr, waarbij nr overeenkomt met de Windows API-functie **GetSysColor(nr)**.

```
SysColor (nr)
```

### ColorMapHue

**ColorMapHue()** retourneert een ARGB-waarde van een kleur uit een kleurenkaart waarbij de tintcomponent van het HSV-kleurmodel wordt gevarieerd. De kleurenkaart begint met rood, doorloopt geel, groen, cyaan, blauw en magenta, en gaat terug naar rood. Voor x moet een waarde tussen 0 en 1 worden opgegeven.

```
ColorMapHue (x)
```

ColorMapJet

**ColorMapJet()** retourneert de ARGB-waarde van een kleur vanuit een kleurenkaart die begint bij blauw en doorloopt van cyaan en geel naar oranje en die daarna terugkeert naar rood. Voor x moet een waarde tussen 0 en 1 worden opgegeven.

**ColorMapJet** (x)

### Vooraf gedefinieerde kleurfuncties

De volgende functies kunnen worden gebruikt in uitdrukkingen voor vooraf gedefinieerde kleuren. Elke functie retourneert een RGB-kleurrepresentatie.

Optioneel kan een parameter voor alfafactor worden opgegeven. In dat geval wordt een ARGB-kleurrepresentatie geretourneerd. Een alfafactor van 0 komt overeen met volledige transparantie en een alfafactor van 255 komt overeen met volledige ondoorzichtigheid. Als geen waarde voor alfa is ingevoerd, wordt uitgegaan van 255.

Vooraf gedefinieerde kleurfuncties

Kleurfunctie	RGB -waarde
black([alpha])	(0,0,0)
blue([alpha])	(0,0,128)
brown([alpha])	(128,128,0)
cyan([alpha])	(0,128,128)
darkgray([alpha])	(128,128,128)
green([alpha])	(0,128,0)
lightblue([alpha])	(0,0,255)
lightcyan([alpha])	(0,255,255)
lightgray([alpha])	(192,192,192)
lightgreen([alpha])	(0,255,0)
lightmagenta([alpha])	(255,0,255)
lightred([alpha])	(255,0,0)
magenta([alpha])	(128,0,128)
red([alpha])	(128,0,0)
white([alpha])	(255,255,255)
yellow([alpha])	(255,255,0)

### Voorbeelden en resultaten:

Voorbeelden en resultaten

Voorbeelden	Resultaten
<code>Blue()</code>	<code>RGB(0,0,128)</code>
<code>Blue(128)</code>	<code>ARGB(128,0,0,128)</code>

## ARGB

**ARGB()** wordt gebruikt in uitdrukkingen om de kleureigenschappen van een diagramobject in te stellen of te evalueren, waarbij de kleur wordt gedefinieerd door een rode component **r**, een groene component **g** een blauwe component **b**, met een alfactor (ondoorzichtigheid) van **alpha**.

### Syntaxis:

```
ARGB(alpha, r, g, b)
```

**Retourgegevenstypen:** dual

### Argumenten:

Argumenten

Argument	Beschrijving
alpha	Transparantiewaarde in het bereik 0 tot 255. 0 komt overeen met volledige transparantie en 255 komt overeen met volledige ondoorzichtigheid.
r, g, b	De componentwaarden voor rood, groen en blauw. Een kleurcomponent van 0 komt overeen met geen bijdrage en een kleurcomponent van 255 met volledige bijdrage.



*Alle argumenten moeten uitdrukkingen zijn die resulteren in een geheel getal in het bereik van 0 tot 255.*

Bij het interpreteren van de numerieke component en het opmaken hiervan in een hexadecimale notatie, zijn de waarden van de kleurcomponenten gemakkelijker te zien. Lichtgroen heeft bijvoorbeeld het getal 4 278 255 360. In hexadecimale notatie is dit FF00FF00. De eerste twee posities 'FF' (255) geven het kanaal **alpha** aan. De volgende twee posities '00' geven de hoeveelheid **red** aan, de volgende twee posities 'FF' geven de hoeveelheid **green** aan en de laatste twee posities 00' geven de hoeveelheid **blue** aan.

## RGB

**RGB()** retourneert een geheel getal dat overeenkomst met de kleurcode van de kleur die is gedefinieerd door de drie parameters: de rode component r, de groene component g en de blauwe component b. Deze componenten moeten beschikken over gehele getalwaarden tussen 0 en 255. De functie kan worden gebruikt in uitdrukkingen voor het instellen of evalueren van de kleureigenschappen van een diagramobject.



### Syntaxis:

**RGB** (r, g, b)

**Retourgegevenstypen:** dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
r, g, b	De componentwaarden voor rood, groen en blauw. Een kleurcomponent van 0 komt overeen met geen bijdrage en een kleurcomponent van 255 met volledige bijdrage.



*Alle argumenten moeten uitdrukkingen zijn die resulteren in een geheel getal in het bereik van 0 tot 255.*

Bij het interpreteren van de numerieke component en het opmaken hiervan in een hexadecimale notatie, zijn de waarden van de kleurcomponenten gemakkelijker te zien. Lichtgroen heeft bijvoorbeeld het getal 4 278 255 360. In hexadecimale notatie is dit FF00FF00. De eerste twee posities 'FF' (255) geven het kanaal **alpha** aan. In de functies **RGB** en **HSL** is dit altijd 'FF' (ondoorzichtig). De volgende twee posities '00' geven de hoeveelheid **red** aan, de volgende twee posities 'FF' geven de hoeveelheid **green** aan en de laatste twee posities '00' geven de hoeveelheid **blue** aan.

Voorbeeld: Diagramuitdrukking

In dit voorbeeld wordt een aangepaste kleur toegepast op een diagram:

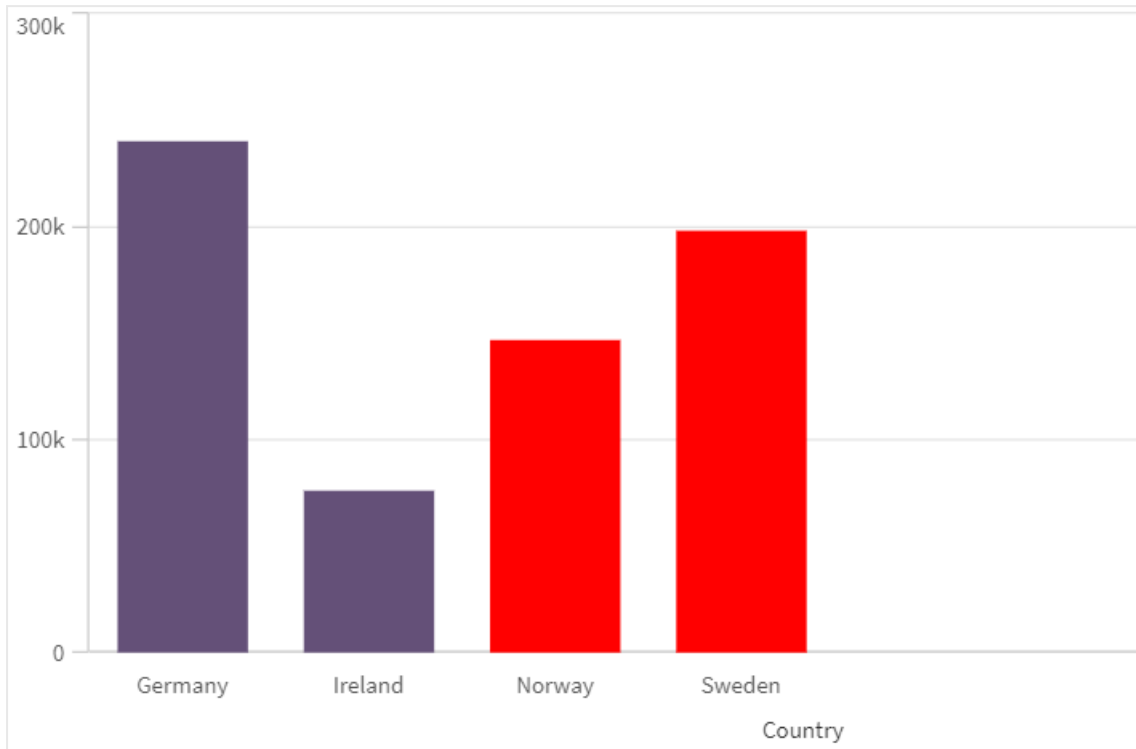
Gebruikte gegevens in dit voorbeeld:

```
ProductSales: Load * Inline [Country,Sales,Budget Sweden,100000,50000 Germany, 125000, 175000  
Norway, 74850, 68500 Ireland, 45000, 48000 Sweden,98000,50000 Germany, 115000, 175000 Norway,  
71850, 68500 Ireland, 31000, 48000 ] (delimiter is ',' );
```

Geef de volgende uitdrukking op in het eigenschappenpaneel **Kleuren en legenda**:

```
If (Sum(Sales)>Sum(Budget),RGB(255,0,0),RGB(100,80,120))
```

Resultaat



Voorbeeld: Load-script

Het volgende voorbeeld toont de overeenkomstige RGB-waarden voor waarden in hex-indeling:

```
Load Text(R & G & B) as Text, RGB(R,G,B) as Color; Load Num#(R,'(HEX)') as R, Num#(G,'(HEX)') as G, Num#(B,'(HEX)') as B Inline [R,G,B 01,02,03 AA,BB,CC];
```

Resultaat

Tekst	Kleur
010203	RGB(1,2,3)
AABBCC	RGB(170,187,204)

## HSL

**HSL()** wordt gebruikt in uitdrukkingen om de kleureigenschappen van een diagramobject in te stellen of te evalueren, waarbij de kleur wordt gedefinieerd door waarden van **hue**, **saturation** en **luminosity** tussen 0 en 1.

**Syntaxis:**

```
HSL (hue, saturation, luminosity)
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

Argument	Beschrijving
hue, saturation, luminosity	De componentwaarden voor hue, saturation en luminosity lopen uiteen van 0 tot 1.



*Alle argumenten moeten uitdrukkingen zijn die resulteren in een geheel getal in het bereik van 0 tot 1.*

Bij het interpreteren van de numerieke component en het opmaken hiervan in een hexadecimale notatie, zijn de RGB -waarden van de kleurcomponenten gemakkelijker te zien. Lichtgroen heeft bijvoorbeeld het getal 4 278 255 360. In hexadecimale notatie is dit FF00FF00 en in RGB (0,255,0). Dit is equivalent aan HSL (80/240, 240/240, 120/240) - een HSL -waarde van (0.33, 1, 0.5).

### 5.5 Voorwaardenfuncties

De voorwaardefuncties evalueren allemaal een voorwaarde en retourneren verschillende antwoorden, afhankelijk van de waarde van de voorwaarde. De functies kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

#### Overzicht voorwaardenfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

##### **alt**

De functie **alt** retourneert de eerste parameter waaraan een geldige numerieke waarde kan worden toegewezen. Als deze niet wordt gevonden, wordt de laatste parameter geretourneerd. U kunt een willekeurig aantal parameters gebruiken.

```
alt (expr1 [ , expr2 , expr3 , ... ] , else)
```

##### **class**

De functie **class** wijst de eerste parameter toe aan een klasse-interval. Het resultaat is een duale waarde met  $a \leq x < b$  als de tekstwaarde, waarbij a en b de boven- en ondergrens van het bereik zijn, en met de ondergrens als numerieke waarde.

```
class (expression, interval [ , label [ , offset ]])
```

### **coalesce**

De functie **coalesce** retourneert de eerste parameter waaraan een geldige non-NULL kan worden toegewezen. U kunt een willekeurig aantal parameters gebruiken.

```
coalesce(expr1 [ , expr2 , expr3 , ...])
```

### **if**

De functie **if** retourneert een waarde afhankelijk van het feit of de opgegeven voorwaarde wordt geëvalueerd als True of False.

```
if (condition , then , else)
```

### **match**

Met de functie **match** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u de numerieke locatie van de overeenkomende uitdrukkingen als resultaat. De vergelijking is hoofdlettergevoelig.

```
match ( str, expr1 [ , expr2,...exprN ])
```

### **mixmatch**

Met de functie **mixmatch** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u de numerieke locatie van de overeenkomende uitdrukkingen als resultaat. De vergelijking is niet hoofdlettergevoelig.

```
mixmatch ( str, expr1 [ , expr2,...exprN ])
```

### **pick**

De functie **pick** retourneert de *n*-de uitdrukking in de lijst.

```
pick (n, expr1 [ , expr2,...exprN])
```

### **wildmatch**

Met de functie **wildmatch** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u het aantal overeenkomende uitdrukkingen als resultaat. Bij deze functie kunt u jokertekens (\* en ?) gebruiken in de tekenreeksen die worden vergeleken. \* komt overeen met een willekeurige reeks tekens. ? komt overeen met een willekeurig enkel teken. De vergelijking is niet hoofdlettergevoelig.

```
wildmatch ( str, expr1 [ , expr2,...exprN ])
```

### **alt**

De functie **alt** retourneert de eerste parameter waaraan een geldige numerieke waarde kan worden toegewezen. Als deze niet wordt gevonden, wordt de laatste parameter geretourneerd. U kunt een willekeurig aantal parameters gebruiken.

#### **Syntaxis:**

```
alt(expr1 [ , expr2 , expr3 , ...] , else)
```

### Argumenten:

Argumenten

Argument	Beschrijving
expr1	De eerste uitdrukking voor controle op een geldige getalsnotatie.
expr2	De tweede uitdrukking voor controle op een geldige getalsnotatie.
expr3	De derde uitdrukking voor controle op een geldige getalsnotatie.
else	Waarde die moet worden geretourneerd als geen van de voorafgaande parameters een geldige getalsnotatie heeft.

De alt -functie wordt vaak gebruikt met interpretatiefuncties voor getallen of datums. Op die manier kan Qlik Sense verschillende datumnotaties in een voorkeursvolgorde testen. De functie kan tevens worden gebruikt voor het afhandelen van NULL-waarden in numerieke uitdrukkingen.

### Voorbeelden:

Voorbeelden

Voorbeeld	Resultaat
<pre>alt( date#( dat , 'YYYY/MM/DD' ),     date#( dat , 'MM/DD/YYYY' ),     date#( dat , 'MM/DD/YY' ),     'No valid date' )</pre>	Deze uitdrukking test of het veld Datum een datum bevat volgens een van de drie opgegeven datumnotaties. Als dit het geval is, wordt een dubbele waarde met de oorspronkelijke tekenreeks en een geldige numerieke representatie van een datum weergegeven. Als geen geldige datum wordt gevonden, wordt de tekst 'No valid date' geretourneerd (zonder geldige numerieke representatie).
<pre>alt(Sales,0) + alt(Margin,0)</pre>	Deze uitdrukking voegt de velden Sales en Margin toe, waarbij eventuele ontbrekende waarden (NULL) worden vervangen door een 0.

## class

De functie **class** wijst de eerste parameter toe aan een klasse-interval. Het resultaat is een duale waarde met  $a \leq x < b$  als de tekstwaarde, waarbij a en b de boven- en ondergrens van het bereik zijn, en met de ondergrens als numerieke waarde.

### Syntaxis:

```
class(expression, interval [ , label [ , offset ]])
```

### Argumenten:

Argumenten

Argument	Beschrijving
interval	Een getal dat de bin-breedte aanduidt.
label	Een willekeurige tekenreeks die de 'x' in de resultaattekst kan vervangen.
offset	Een getal die kan worden gebruikt als verschil met het standaardbeginpunt van de classificatie. Het standaardbeginpunt is meestal 0.

### Voorbeelden:

Voorbeelden

Voorbeeld	Resultaat
<code>class( var,10 ) met var = 23</code>	retourneert '20<=x<30'
<code>class( var,5,'value' ) met var = 23</code>	retourneert '20<= value <25'
<code>class( var,10,'x',5 ) met var = 23</code>	retourneert '15<=x<25'

### Voorbeeld - Load-script dat gebruikmaakt van class

Voorbeeld: load-script

#### Load-script

In dit voorbeeld laden we een tabel die de naam en leeftijd van personen bevat. Wij willen een veld toevoegen dat elke persoon classificeert op basis van een leeftijdsgroep met een interval van tien jaar. De originele brontabel ziet er als volgt uit.

Resultaten

Name	Age
John	25
Karen	42
Yoshi	53

U kunt het classificatieveld voor de leeftijdsgroep toevoegen door een voorafgaande load-opdracht toe te voegen via de functie **class**.

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
LOAD *, class(Age, 10, 'age') As Agegroup; LOAD * INLINE [ Age, Name 25, John 42, Karen 53, Yoshi];
```

### Resultaten

Resultaten

Name	Age	Agegroup
John	25	20 <= age < 30
Karen	42	40 <= age < 50
Yoshi	53	50 <= age < 60

### coalesce

De functie **coalesce** retourneert de eerste parameter waaraan een geldige non-NULL kan worden toegewezen. U kunt een willekeurig aantal parameters gebruiken.

#### Syntaxis:

```
coalesce(expr1[ , expr2 , expr3 , ...])
```

#### Argumenten:

Argumenten

Argument	Beschrijving
expr1	De eerste uitdrukking voor controle op een geldige niet-null getalsnotatie.
expr2	De tweede uitdrukking voor controle op een geldige niet-null getalsnotatie.
expr3	De derde uitdrukking voor controle op een geldige niet-null getalsnotatie.

#### Voorbeelden:

Voorbeelden

Voorbeeld	Resultaat
	Deze uitdrukking wijzigt alle NULL-waarden van een veld in 'N/A'.
<code>coalesce(ProductDescription, ProductName, ProductCode, 'no description available')</code>	Deze uitdrukking selecteert drie verschillende velden met een productomschrijving als sommige velden geen waarden voor het product bevatten. De eerste velden zonder een null-waarde worden in de opgegeven volgorde geretourneerd. Als geen enkel veld een waarde bevat, is het resultaat 'geen omschrijving beschikbaar'.

Voorbeeld	Resultaat
<pre>Coalesce(TextBetween(FileName, '', ''), FileName)</pre>	<p>Deze uitdrukking zorgt ervoor dat de tekst tussen aanhalingstekens van het veld <i>FileName</i> wordt ingekort. Als de opgegeven <i>FileName</i> tussen aanhalingstekens staat, wordt deze verwijderd en wordt de ingesloten <i>FileName</i> zonder aanhalingstekens geretourneerd. Als de functie <i>TextBetween</i> de scheidingstekens niet kan vinden, wordt null geretourneerd. De <b>Coalesce</b> zal deze afwijzen en in plaats daarvan de onbewerkte <i>FileName</i> retourneren.</p>

### if

De functie **if** retourneert een waarde afhankelijk van het feit of de opgegeven voorwaarde wordt geëvalueerd als True of False.

#### Syntaxis:

```
if(condition , then [, else])
```

#### Argumenten

Argument	Beschrijving
condition	Uitdrukking die logisch wordt geïnterpreteerd.
then	Uitdrukking die van elk willekeurig type kan zijn. Als de <i>condition</i> True is, retourneert de functie if de waarde van de uitdrukking <i>then</i> .
else	Uitdrukking die van elk willekeurig type kan zijn. Als de <i>condition</i> False is, retourneert de functie if de waarde van de uitdrukking <i>else</i> .  Deze parameter is optioneel. Als de <i>condition</i> False is, wordt NULL geretourneerd als u else niet hebt opgegeven.

#### Voorbeeld

Voorbeeld	Resultaat
<pre>if( Amount &gt;= 0, 'OK', 'Alarm' )</pre>	<p>Deze uitdrukking test of het aantal een positief getal (0 of groter) is en retourneert 'OK' als dit het geval is. Als het aantal kleiner is dan 0, wordt 'Alarm' geretourneerd.</p>

### Voorbeeld - Load-script dat gebruikmaakt van if

Voorbeeld: Load-script

#### Load-script

If kan worden gebruikt in load-script met andere methoden en objecten, waaronder variabelen. Als u bijvoorbeeld een variabele *threshold* instelt en een veld in het gegevensmodel wilt opnemen op basis van die drempel, kunt u het volgende doen.



Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
Transactions:
Load * Inline [
transaction_id, transaction_date, transaction_amount, transaction_quantity, customer_id, size,
color_code
3750, 20180830, 23.56, 2, 2038593, L, Red
3751, 20180907, 556.31, 6, 203521, m, orange
3752, 20180916, 5.75, 1, 5646471, S, blue
3753, 20180922, 125.00, 7, 3036491, l, Black
3754, 20180922, 484.21, 13, 049681, xs, Red
3756, 20180922, 59.18, 2, 2038593, M, Blue
3757, 20180923, 177.42, 21, 203521, xL, Black
];

set threshold = 100;

/* Create new table called Transaction_Buckets
Compare transaction_amount field from Transaction table to threshold of 100.
Output results into a new field called Compared to Threshold
*/

Transaction_Buckets:
Load
    transaction_id,
    If(transaction_amount > $(threshold),'Greater than $(threshold)','Less than $(threshold)')
as [Compared to Threshold]
Resident Transactions;
```

### Resultaten

Qlik Sense-tabel met de uitvoer van het gebruik van de functie *if* in het load-script.

transaction_id	Vergeleken met drempelwaarde
3750	Kleiner dan 100
3751	Groter dan 100
3752	Kleiner dan 100
3753	Groter dan 100
3754	Groter dan 100
3756	Kleiner dan 100
3757	Groter dan 100

### Voorbeelden - Diagramuitdrukkingen die gebruikmaken van if

Voorbeelden: Diagramuitdrukkingen

#### Diagramuitdrukking 1

##### Load-script

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Nadat u de gegevens hebt geladen, maakt u de diagramuitdrukking in een Qlik Sense-tabel op basis van de onderstaande voorbeelden.

```
MyTable:
LOAD * inline [Date, Location, Incidents
1/3/2016, Beijing, 0
1/3/2016, Boston, 12
1/3/2016, Stockholm, 3
1/3/2016, Toronto, 0
1/4/2016, Beijing, 0
1/4/2016, Boston, 8];
```

Qlik Sense-tabel met voorbeelden van de functie *if* in een diagramuitdrukking.

Datum	Locatie	Incidents	if(Incidents>=10, 'Critical', 'Ok' )	if(Incidents>=10, 'Critical', If(Incidents>=1 and Incidents<10, 'Warning', 'Ok'))
1/3/2016	Beijing	0	Ok	Ok
1/3/2016	Boston	12	Critical	Critical
1/3/2016	Stockholm	3	Ok	Warning
1/3/2016	Toronto	0	Ok	Ok
1/4/2016	Beijing	0	Ok	Ok
1/4/2016	Boston	8	Ok	Waarschuwing

#### Diagramuitdrukking 2

In een nieuwe app voegt u het volgende script toe in een nieuw tabblad van de editor voor laden van gegevens. Vervolgens laadt u de gegevens. U kunt daarna de tabel maken met de onderstaande diagramuitdrukkingen.

```
SET FirstWeekDay=0;
Load
Date(MakeDate(2022)+RecNo()-1) as Date
Autogenerate 14;
```

Qlik Sense-tabel met een voorbeeld van de functie *if* in een diagramuitdrukking.

Date	WeekDay(Date)	If(WeekDay (Date)>=5,'WeekEnd','Normal Day')
1/1/2022	Za	WeekEnd
1/2/2022	Zo	WeekEnd
1/3/2022	Ma	Normal Day
1/4/2022	Di	Normal Day
1/5/2022	Wo	Normal Day
1/6/2022	Do	Normal Day
1/7/2022	Vr	Normal Day
1/8/2022	Za	WeekEnd
1/9/2022	Zo	WeekEnd
1/10/2022	Ma	Normal Day
1/11/2022	Di	Normal Day
1/12/2022	Wo	Normal Day
1/13/2022	Do	Normal Day
1/14/2022	Vr	Normal Day

### match

Met de functie **match** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u de numerieke locatie van de overeenkomende uitdrukkingen als resultaat. De vergelijking is hoofdlettergevoelig.

#### Syntaxis:

```
match( str, expr1 [ , expr2,...exprN ])
```



*Als u een vergelijking wilt gebruiken waarbij geen onderscheid wordt gemaakt tussen hoofdletters en kleine letters, gebruikt u de functie **mixmatch**. Als u een vergelijking wilt gebruiken met jokertekens en waarbij geen onderscheid wordt gemaakt tussen hoofdletters en kleine letters, gebruikt u de functie **wildmatch**.*

### Voorbeeld: Load-script dat gebruikmaakt van match

Voorbeeld: Load-script

#### Load-script

U kunt match gebruiken om een gegevens-subset te laden. U kunt bijvoorbeeld een numerieke waarde als resultaat krijgen voor een uitdrukking in de functie. Vervolgens kunt u de geladen gegevens beperken op basis van de numerieke waarde. Match geeft 0 als resultaat als er geen overeenkomst is. Alle uitdrukkingen zonder overeenkomst hebben in dit voorbeeld daarom 0 als resultaat en worden uitgesloten van de gegevens die zijn geladen door de WHERE-instructie.

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
Transactions: Load * Inline [ transaction_id, transaction_date, transaction_amount,
transaction_quantity, customer_id, size, color_code 3750, 20180830, 23.56, 2, 2038593, L, Red
3751, 20180907, 556.31, 6, 203521, m, orange 3752, 20180916, 5.75, 1, 5646471, s, blue 3753,
20180922, 125.00, 7, 3036491, l, Black 3754, 20180922, 484.21, 13, 049681, xs, Red 3756,
20180922, 59.18, 2, 2038593, M, Blue 3757, 20180923, 177.42, 21, 203521, XL, Black ]; /*
Create new table called Transaction_Buckets Create new fields called customer, and Color code
- Blue and Black Load Transactions table. Match returns 1 for 'Blue', 2 for 'Black'. Does not
return a value for 'blue' because match is case sensitive. Only values that returned numeric
value greater than 0 are loaded by WHERE statement into Transactions_Buckets table. */
Transaction_Buckets: Load customer_id, customer_id as [Customer], color_code as [Color
Code Blue and Black] Resident Transactions where match(color_code, 'Blue', 'Black') > 0;
```

#### Resultaten

Qlik Sense-tabel met de uitvoer van het gebruik van de functie match in het load-script

Color Code Blue and Black	Customer
Black	203521
Black	3036491
Blue	2038593

### Voorbeelden - Diagramuitdrukkingen die gebruikmaken van match

Voorbeelden: Diagramuitdrukkingen

#### Diagramuitdrukking 1

#### Load-script

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Nadat u de gegevens hebt geladen, maakt u de diagramuitdrukking in een Qlik Sense-tabel op basis van de onderstaande voorbeelden.

## 5 Script- en diagramfuncties

```
MyTable: Load * inline [Cities, Count Toronto, 123 Toronto, 234 Toronto, 231 Boston, 32 Boston, 23 Boston, 1341 Beijing, 234 Beijing, 45 Beijing, 235 Stockholm, 938 Stockholm, 39 Stockholm, 189 zurich, 2342 zurich, 9033 zurich, 0039];
```

De eerste uitdrukking in de onderstaande tabel geeft 0 als resultaat voor Stockholm, aangezien 'Stockholm' niet is opgenomen in de lijst met uitdrukkingen in de functie **match**. Het geeft eveneens 0 als resultaat voor 'Zurich' omdat de **match**-vergelijking hoofdlettergevoelig is.

Qlik Sense-tabel met voorbeelden van de functie *match* in een diagramuitdrukking

Cities	match(Cities,'Toronto','Boston','Beijing','Zurich')	match(Cities,'Toronto','Boston','Beijing','Stockholm','Zurich')
Beijing	3	3
Boston	2	2
Stockholm	0	4
Toronto	1	1
zurich	0	5

### Diagramuitdrukking 2

U kunt **match** gebruiken om een uitdrukking op aangepaste wijze te sorteren.

Kolommen worden standaard numeriek of alfabetisch gesorteerd, afhankelijk van de gegevens.

Qlik Sense-tabel met een voorbeeld van de standaard sorteervolgorde

Cities
Beijing
Boston
Stockholm
Toronto
zurich

U kunt de volgorde als volgt veranderen:

1. Open de sectie **Sorteren** in het **eigenschappenvenster**.
2. Schakel automatisch sorteren uit voor de kolom die u aangepast wilt sorteren.
3. Hef de selecties **Numeriek sorteren** en **Alfabetisch sorteren** op.
4. Selecteer **Sorteren op uitdrukking** en voer een uitdrukking in die hierop lijkt:  
`=match( Cities, 'Toronto','Boston','Beijing','Stockholm','zurich')`  
De sorteervolgorde van de kolom Cities wordt gewijzigd.

Klik Sense-tabel met een voorbeeld van het veranderen van de sorteervolgorde met de functie *match*

Cities
Toronto
Boston
Beijing
Stockholm
zurich

U kunt ook de numerieke waarde bekijken die is geretourneerd.

Klik Sense-tabel met een voorbeeld van de numerieke waarden die zijn geretourneerd met de functie *match*

Cities	Cities & ' - ' & match ( Cities, 'Toronto','Boston', 'Beijing','Stockholm','zurich')
Toronto	Toronto - 1
Boston	Boston - 2
Beijing	Beijing - 3
Stockholm	Stockholm - 4
zurich	zurich - 5

### mixmatch

Met de functie **mixmatch** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u de numerieke locatie van de overeenkomende uitdrukkingen als resultaat. De vergelijking is niet hoofdlettergevoelig.

#### Syntaxis:

```
mixmatch( str, expr1 [ , expr2,...exprN ])
```

Als u in plaats daarvan een vergelijking wilt gebruiken waarbij onderscheid wordt gemaakt tussen hoofdletters en kleine letters, gebruikt u de functie **match**. Als u een vergelijking wilt gebruiken met jokertekens en waarbij geen onderscheid wordt gemaakt tussen hoofdletters en kleine letters, gebruikt u de functie **wildmatch**.

### Voorbeeld - Load-script dat gebruikmaakt van mixmatch

Voorbeeld: Load-script

#### Load-script

U kunt **mixmatch** gebruiken om een gegevens-subset te laden. U kunt bijvoorbeeld een numerieke waarde als resultaat krijgen voor een uitdrukking in de functie. Vervolgens kunt u de geladen gegevens beperken

op basis van de numerieke waarde. Mixmatch geeft 0 als resultaat als er geen overeenkomst is. Alle uitdrukkingen zonder overeenkomst hebben in dit voorbeeld daarom 0 als resultaat en worden uitgesloten van de gegevens die zijn geladen door de WHERE-instructie.

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
Load * Inline [ transaction_id, transaction_date, transaction_amount, transaction_quantity,
customer_id, size, color_code 3750, 20180830, 23.56, 2, 2038593, L, Red 3751, 20180907,
556.31, 6, 203521, m, orange 3752, 20180916, 5.75, 1, 5646471, s, blue 3753, 20180922, 125.00,
7, 3036491, l, Black 3754, 20180922, 484.21, 13, 049681, xs, Red 3756, 20180922, 59.18, 2,
2038593, M, Blue 3757, 20180923, 177.42, 21, 203521, XL, Black ]; /* Create new table called
Transaction_Buckets Create new fields called Customer, and Color code - Black, Blue, blue Load
Transactions table. Mixmatch returns 1 for 'Black', 2 for 'Blue'. Also returns 3 for 'blue'
because mixmatch is not case sensitive. Only values that returned numeric value greater than 0
are loaded by WHERE statement into Transactions_Buckets table. */ Transaction_Buckets: Load
customer_id, customer_id as [Customer], color_code as [Color Code - Black, Blue,
blue] Resident Transactions where mixmatch(color_code,'Black','Blue') > 0;
```

### Resultaten

Qlik Sense-tabel met de uitvoer van het gebruik van de functie mixmatch in het load-script.

Color Code Black, Blue, blue	Customer
Black	203521
Black	3036491
Blue	2038593
blue	5646471

### Voorbeelden - Diagramuitdrukkingen die gebruikmaken van mixmatch

Voorbeelden: Diagramuitdrukkingen

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Nadat u de gegevens hebt geladen, maakt u de diagramuitdrukking in een Qlik Sense-tabel op basis van de onderstaande voorbeelden.

#### Diagramuitdrukking 1

```
MyTable: Load * inline [Cities, Count Toronto, 123 Toronto, 234 Toronto, 231 Boston, 32
Boston, 23 Boston, 1341 Beijing, 234 Beijing, 45 Beijing, 235 Stockholm, 938 Stockholm, 39
Stockholm, 189 zurich, 2342 zurich, 9033 zurich, 0039];
```

De eerste uitdrukking in de onderstaande tabel geeft 0 als resultaat voor Stockholm, aangezien 'Stockholm' niet is opgenomen in de lijst met uitdrukkingen in de functie **mixmatch**. Het geeft 4 als resultaat voor 'Zurich' omdat de **mixmatch**-vergelijking niet hoofdlettergevoelig is.

Qlik Sense-tabel met voorbeelden van de functie *mixmatch* in een diagramuitdrukking

Cities	<code>mixmatch(Cities,'Toronto','Boston','Beijing','Zurich')</code>	<code>mixmatch(Cities,'Toronto','Boston','Beijing','Stockholm','Zurich')</code>
Beijing	3	3
Boston	2	2
Stockholm	0	4
Toronto	1	1
zurich	4	5

### Diagramuitdrukking 2

U kunt *mixmatch* gebruiken om een uitdrukking op aangepaste wijze te sorteren.

Kolommen worden standaard alfabetisch of numeriek gesorteerd, afhankelijk van de gegevens.

Qlik Sense-tabel met een voorbeeld van de standaard sorteervolgorde

Cities
Beijing
Boston
Stockholm
Toronto
zurich

U kunt de volgorde als volgt veranderen:

1. Open de sectie **Sorteren** in het **eigenschappenvenster**.
2. Schakel automatisch sorteren uit voor de kolom die u aangepast wilt sorteren.
3. Hef de selecties **Numeriek sorteren** en **Alfabetisch sorteren** op.
4. Selecteer **Sorteren op uitdrukking** en voer de volgende uitdrukking in:  
`=mixmatch(Cities, 'Toronto', 'Boston', 'Beijing', 'Stockholm', 'Zurich')`  
 De sorteervolgorde van de kolom Cities wordt gewijzigd.

Qlik Sense-tabel met een voorbeeld van het veranderen van de sorteervolgorde met de functie *mixmatch*.

Cities
Toronto
Boston
Beijing
Stockholm
zurich



U kunt ook de numerieke waarde bekijken die is geretourneerd.

Qlik Sense-tabel met een voorbeeld van de numerieke waarden die zijn gertourneerd met de functie *mixmatch*.

Cities	Cities & ' - ' & mixmatch ( Cities, 'Toronto','Boston', 'Beijing','Stockholm','Zurich')
Toronto	Toronto - 1
Boston	Boston - 2
Beijing	Beijing - 3
Stockholm	Stockholm - 4
zurich	zurich - 5

### pick

De functie pick retourneert de *n*-de uitdrukking in de lijst.

**Syntaxis:**

```
pick(n, expr1 [ , expr2, ...exprN])
```

**Argumenten:**

#### Argumenten

Argument	Beschrijving
n	n is een geheel getal tussen 1 en N.

**Voorbeeld:**

#### Voorbeeld

Voorbeeld	Resultaat
pick( N, 'A', 'B', 4, 6 )	retourneert 'B' als N = 2 retourneert 4 als N = 3

### wildmatch

Met de functie **wildmatch** vergelijkt u de eerste parameter met alle volgende parameters en krijgt u het aantal overeenkomende uitdrukkingen als resultaat. Bij deze functie kunt u jokertekens ( \* en ? ) gebruiken in de tekenreeksen die worden vergeleken. \* komt overeen met een willekeurige reeks tekens. ? komt overeen met een willekeurig enkel teken. De vergelijking is niet hoofdlettergevoelig.

**Syntaxis:**

```
wildmatch( str, expr1 [ , expr2, ...exprN ] )
```

Als u een vergelijking wilt gebruiken zonder jokertekens, gebruikt u de functie **match** of **mixmatch**.

### Voorbeeld: Load-script dat gebruikmaakt van wildmatch

Voorbeeld: Load-script

#### Load-script

U kunt wildmatch gebruiken om een gegevens-subset te laden. U kunt bijvoorbeeld een numerieke waarde als resultaat krijgen voor een uitdrukking in de functie. Vervolgens kunt u de geladen gegevens beperken op basis van de numerieke waarde. Wildmatch geeft 0 als resultaat als er geen overeenkomst is. Alle uitdrukkingen zonder overeenkomst hebben in dit voorbeeld daarom 0 als resultaat en worden uitgesloten van de gegevens die zijn geladen door de WHERE-instructie.

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Maak de onderstaande tabel in Qlik Sense om de resultaten te bekijken.

```
Transactions: Load * Inline [ transaction_id, transaction_date, transaction_amount,
transaction_quantity, customer_id, size, color_code 3750, 20180830, 23.56, 2, 2038593, L, Red
3751, 20180907, 556.31, 6, 203521, m, orange 3752, 20180916, 5.75, 1, 5646471, s, blue 3753,
20180922, 125.00, 7, 3036491, l, Black 3754, 20180922, 484.21, 13, 049681, xs, Red 3756,
20180922, 59.18, 2, 2038593, M, Blue 3757, 20180923, 177.42, 21, 203521, XL, Black ]; /*
Create new table called Transaction_Buckets Create new fields called customer, and Color code
- Black, Blue, blue, red Load Transactions table. Wildmatch returns 1 for 'Black', 'Blue', and
'blue', and 2 for 'Red'. Only values that returned numeric value greater than 0 are loaded
by WHERE statement into Transactions_Buckets table. */ Transaction_Buckets: Load
customer_id, customer_id as [Customer], color_code as [Color Code Black, Blue, blue,
Red] Resident Transactions where wildmatch(color_code, 'Bl*', 'R??') > 0;
```

#### Resultaten

Qlik Sense-tabel met de uitvoer van het gebruik van de functie *wildmatch* in het load-script

Color Code Black, Blue, blue, Red	Customer
Black	203521
Black	3036491
Blue	2038593
blue	5646471
Red	049681
Red	2038593

### Voorbeelden: Diagramuitdrukkingen die gebruikmaken van wildmatch

Voorbeeld: Diagramuitdrukking

#### Diagramuitdrukking 1

Maak een nieuw tabblad in de editor voor laden van gegevens en laad de volgende gegevens als een inline load. Nadat u de gegevens hebt geladen, maakt u de diagramuitdrukking in een Qlik Sense-tabel op basis van de onderstaande voorbeelden.

```
MyTable: Load * inline [Cities, Count Toronto, 123 Toronto, 234 Toronto, 231 Boston, 32 Boston, 23 Boston, 1341 Beijing, 234 Beijing, 45 Beijing, 235 Stockholm, 938 Stockholm, 39 Stockholm, 189 zurich, 2342 zurich, 9033 zurich, 0039];
```

De eerste uitdrukking in de onderstaande tabel geeft 0 als resultaat voor Stockholm, aangezien 'Stockholm' niet is opgenomen in de lijst met uitdrukkingen in de functie **wildmatch**. Het geeft ook 0 als resultaat voor 'Boston', omdat ? alleen overeenkomt met een enkel teken.

Qlik Sense-tabel met voorbeelden van de functie *wildmatch* in een diagramuitdrukking

Cities	wildmatch(Cities,'Tor*','?ton','Beijing','*urich')	wildmatch(Cities,'Tor*','???ton','Beijing','Stockholm','*urich')
Beijing	3	3
Boston	0	2
Stockholm	0	4
Toronto	1	1
zurich	4	5

#### Diagramuitdrukking 2

U kunt wildmatch gebruiken om een uitdrukking op aangepaste wijze te sorteren.

Kolommen worden standaard numeriek of alfabetisch gesorteerd, afhankelijk van de gegevens.

Qlik Sense-tabel met een voorbeeld van de standaardsorteervolgorde

Cities
Beijing
Boston
Stockholm
Toronto
zurich

U kunt de volgorde als volgt veranderen:

1. Open de sectie **Sorteren** in het **eigenschappenvenster**.
2. Schakel automatisch sorteren uit voor de kolom die u aangepast wilt sorteren.
3. Hef de selecties **Numeriek sorteren** en **Alfabetisch sorteren** op.
4. Selecteer **Sorteren op uitdrukking** en voer een uitdrukking in die hierop lijkt:  
`=wildmatch( Cities, 'Tor*', '???ton', 'Beijing', 'Stockholm', '*urich')`  
De sorteervolgorde van de kolom Cities wordt gewijzigd.

Qlik Sense-tabel met een voorbeeld van het veranderen van de sorteervolgorde met de functie *wildmatch*.

Cities
Toronto
Boston
Beijing
Stockholm
zurich

U kunt ook de numerieke waarde bekijken die is geretourneerd.

Qlik Sense-tabel met een voorbeeld van de numerieke waarden die zijn geretourneerd met de functie *wildmatch*

Cities	Cities & ' - ' & wildmatch ( Cities, 'Tor*', '???ton', 'Beijing', 'Stockholm', '*urich')
Toronto	Toronto - 1
Boston	Boston - 2
Beijing	Beijing - 3
Stockholm	Stockholm - 4
zurich	zurich - 5

## 5.6 Tellerfuncties

In dit hoofdstuk worden functies beschreven waarmee u counters kunt opnemen tijdens de evaluatie van **LOAD**-opdrachten in het script voor het laden van gegevens. De enige functie die kan worden gebruikt in diagramuitdrukkingen is **RowNo()**.

Sommige tellerfuncties hebben geen parameters, maar er moeten altijd haakjes achter worden geplaatst.

### Overzicht tellerfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### autonumber

Deze scriptfunctie retourneert een uniek geheel getal voor elke distinctieve geëvalueerde waarde van *expression* die is gevonden tijdens uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.

```
autonumber (expression[ , AutoID])
```

### autonumberhash128

Deze scriptfunctie berekent een 128-bits hash van de gecombineerde invoerwaarden van de uitdrukking en retourneert een uniek geheel getal voor elke distinctieve hash-waarde die wordt gevonden tijdens de uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.

```
autonumberhash128 (expression {, expression})
```

### autonumberhash256

Deze scriptfunctie berekent een 256-bits hash van de gecombineerde invoerwaarden van de uitdrukking en retourneert een uniek geheel getal voor elke distinctieve hash-waarde die wordt gevonden tijdens de uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.

```
autonumberhash256 (expression {, expression})
```

### IterNo

Deze scriptfunctie retourneert een geheel getal dat aangeeft hoe vaak één record wordt geëvalueerd in een **LOAD**-opdracht met een **while**-clausule. De eerste iteratie heeft nummer 1. De functie **IterNo** is alleen relevant als deze samen met een **while**-clausule wordt gebruikt.

```
IterNo ( )
```

### RecNo

Deze scriptfunctie retourneert een geheel getal voor het nummer van de momenteel gelezen rij van de interne tabel. De eerste record is nummer 1.

```
RecNo ( )
```

### RowNo - script function

Deze functie retourneert een geheel getal voor de positie van de huidige rij in de resulterende interne Qlik Sense-tabel. De eerste rij is nummer 1.

```
RowNo ( )
```

### RowNo - chart function

**RowNo()** retourneert het nummer van de huidige rij in het huidige kolomsegment in een tabel. Voor bitmapdiagrammen wordt met **RowNo()** het nummer van de huidige rij in het equivalent van de standaard tabel voor het diagram geretourneerd.

```
RowNo - diagramfunctie ([TOTAL])
```

## autonumber

Deze scriptfunctie retourneert een uniek geheel getal voor elke distinctieve geëvalueerde waarde van *expression* die is gevonden tijdens uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.



*U kunt alleen verbinding maken met **autonumber**-sleutels die zijn gegenereerd in dezelfde laadbewerking voor gegevens, aangezien het gehele getal wordt gegenereerd op basis van de volgorde waarin de tabel wordt gelezen. Als u sleutels moet gebruiken die ongewijzigd blijven tussen laadbewerkingen voor gegevens, onafhankelijk van de sortering van de brongegevens, moet u de functie **hash128**, **hash160** of **hash256** gebruiken.*

### Syntaxis:

```
autonumber (expression[ , AutoID] )
```

### Argumenten:

Argument	Beschrijving
AutoID	Als de functie <b>autonumber</b> voor verschillende sleutels in het script wordt gebruikt, kunt u meerdere instanties van de teller maken door met de optionele parameter <i>AutoID</i> elke teller een naam te geven.

### Voorbeeld: Een samengestelde sleutel maken

In dit voorbeeld maken we een samengestelde sleutel met de functie **autonumber** om geheugen te sparen. Het voorbeeld is kort vanwege demonstratiedoeleinden, maar zou relevant zijn bij een tabel met een groot aantal rijen.

Voorbeeldgegevens

Region	Year	Month	Sales
North	2014	May	245
North	2014	May	347
North	2014	June	127
South	2014	June	645
South	2013	May	367
South	2013	May	221

De brongegevens worden geladen met behulp van inline-gegevens. Vervolgens voegen we een voorafgaande load-opdracht toe die een samengestelde sleutel maakt op basis van de velden Region, Year en Month.

```
RegionSales:
LOAD *,
AutoNumber(Region&Year&Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Sales
North, 2014, May, 245
North, 2014, May, 347
North, 2014, June, 127
South, 2014, June, 645
South, 2013, May, 367
South, 2013, May, 221
];
```

De resulterende tabel ziet er als volgt uit:

Resultatentabel

Region	Year	Month	Sales	RYMkey
North	2014	May	245	1
North	2014	May	347	1
North	2014	June	127	2
South	2014	June	645	3
South	2013	May	367	4
South	2013	May	221	4

In dit voorbeeld kunt u verwijzen naar de RYMkey, bijvoorbeeld 1, in plaats van de tekenreeks 'North2014May' als u een koppeling met een andere tabel tot stand moet brengen.

Nu laden we op vergelijkbare wijze een brontabel met kosten. De velden Region, Year en Month zijn uitgesloten in de voorafgaande laadbewerking om te voorkomen dat er een synthetische sleutel wordt gemaakt. We maken al een samengestelde sleutel met de functie **autonumber**, waardoor de tabellen worden gekoppeld.

```
RegionCosts:
LOAD Costs,
AutoNumber(Region&Year&Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Costs
South, 2013, May, 167
North, 2014, May, 56
North, 2014, June, 199
South, 2014, June, 64
South, 2013, May, 172
South, 2013, May, 126
];
```

Nu kunnen we een tabelvisualisatie toevoegen aan een werkblad en de velden Region, Year en Month toevoegen, alsmede metingen Som voor de verkopen en kosten. De tabel ziet er dan als volgt uit:

Resultatentabel

Region	Year	Month	Sum([Sales])	Sum([Costs])
Totals	-	-	1952	784
North	2014	June	127	199
North	2014	May	592	56
South	2014	June	645	64
South	2013	May	588	465

### autonumberhash128

Deze scriptfunctie berekent een 128-bits hash van de gecombineerde invoerwaarden van de uitdrukking en retourneert een uniek geheel getal voor elke distinctieve hash-waarde die wordt gevonden tijdens de uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.



*U kunt alleen verbinding maken met **autonumberhash128**-sleutels die zijn gegenereerd in dezelfde laadbewerking voor gegevens, aangezien het gehele getal wordt gegenereerd op basis van de volgorde waarin de tabel wordt gelezen. Als u sleutels moet gebruiken die ongewijzigd blijven tussen laadbewerkingen voor gegevens, onafhankelijk van de sortering van de brongegevens, moet u de functie **hash128**, **hash160** of **hash256** gebruiken.*

#### Syntaxis:

```
autonumberhash128 (expression {, expression})
```

#### Voorbeeld: Een samengestelde sleutel maken

In dit voorbeeld maken we een samengestelde sleutel met de functie **autonumberhash128** om geheugen te sparen. Het voorbeeld is kort vanwege demonstratiedoeleinden, maar zou relevant zijn bij een tabel met een groot aantal rijen.

Voorbeeldgegevens

Region	Year	Month	Sales
North	2014	May	245
North	2014	May	347
North	2014	June	127
South	2014	June	645
South	2013	May	367
South	2013	May	221



De brongegevens worden geladen met behulp van inline-gegevens. Vervolgens voegen we een voorafgaande load-opdracht toe die een samengestelde sleutel maakt op basis van de velden Region, Year en Month.

```
RegionSales:
LOAD *,
AutoNumberHash128(Region, Year, Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Sales
North, 2014, May, 245
North, 2014, May, 347
North, 2014, June, 127
South, 2014, June, 645
South, 2013, May, 367
South, 2013, May, 221
];
```

De resulterende tabel ziet er als volgt uit:

Resultatentabel

Region	Year	Month	Sales	RYMkey
North	2014	May	245	1
North	2014	May	347	1
North	2014	June	127	2
South	2014	June	645	3
South	2013	May	367	4
South	2013	May	221	4

In dit voorbeeld kunt u verwijzen naar de RYMkey, bijvoorbeeld 1, in plaats van de tekenreeks 'North2014May' als u een koppeling met een andere tabel tot stand moet brengen.

Nu laden we op vergelijkbare wijze een brontabel met kosten. De velden Region, Year en Month zijn uitgesloten in de voorafgaande laadbewerking om te voorkomen dat er een synthetische sleutel wordt gemaakt. We maken al een samengestelde sleutel met de functie **autonumberhash128**, waardoor de tabellen worden gekoppeld.

```
RegionCosts:
LOAD Costs,
AutoNumberHash128(Region, Year, Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Costs
South, 2013, May, 167
North, 2014, May, 56
North, 2014, June, 199
South, 2014, June, 64
South, 2013, May, 172
```

```
South, 2013, May, 126
];
```

Nu kunnen we een tabelvisualisatie toevoegen aan een werkblad en de velden Region, Year en Month toevoegen, alsmede metingen Som voor de verkopen en kosten. De tabel ziet er dan als volgt uit:

Resultatentabel

Region	Year	Month	Sum([Sales])	Sum([Costs])
Totals	-	-	1952	784
North	2014	June	127	199
North	2014	May	592	56
South	2014	June	645	64
South	2013	May	588	465

### autonumberhash256

Deze scriptfunctie berekent een 256-bits hash van de gecombineerde invoerwaarden van de uitdrukking en retourneert een uniek geheel getal voor elke distinctieve hash-waarde die wordt gevonden tijdens de uitvoering van het script. Met deze functie kan bijvoorbeeld een compacte geheugenrepresentatie van een complexe sleutel worden gemaakt.



*U kunt alleen verbinding maken met **autonumberhash256**-sleutels die zijn gegenereerd in dezelfde laadbewerking voor gegevens, aangezien het gehele getal wordt gegenereerd op basis van de volgorde waarin de tabel wordt gelezen. Als u sleutels moet gebruiken die ongewijzigd blijven tussen laadbewerkingen voor gegevens, onafhankelijk van de sortering van de brongegevens, moet u de functie **hash128**, **hash160** of **hash256** gebruiken.*

#### Syntaxis:

```
autonumberhash256(expression {, expression})
```

#### Voorbeeld: Een samengestelde sleutel maken

In dit voorbeeld maken we een samengestelde sleutel met de functie **autonumberhash256** om geheugen te sparen. Het voorbeeld is kort vanwege demonstratiedoeleinden, maar zou relevant zijn bij een tabel met een groot aantal rijen.

Voorbeeldtabel

Region	Year	Month	Sales
North	2014	May	245
North	2014	May	347

Region	Year	Month	Sales
North	2014	June	127
South	2014	June	645
South	2013	May	367
South	2013	May	221

De brongegevens worden geladen met behulp van inline-gegevens. Vervolgens voegen we een voorafgaande load-opdracht toe die een samengestelde sleutel maakt op basis van de velden Region, Year en Month.

```
RegionSales:
LOAD *,
AutoNumberHash256(Region, Year, Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Sales
North, 2014, May, 245
North, 2014, May, 347
North, 2014, June, 127
South, 2014, June, 645
South, 2013, May, 367
South, 2013, May, 221
];
```

De resulterende tabel ziet er als volgt uit:

Resultatentabel

Region	Year	Month	Sales	RYMkey
North	2014	May	245	1
North	2014	May	347	1
North	2014	June	127	2
South	2014	June	645	3
South	2013	May	367	4
South	2013	May	221	4

In dit voorbeeld kunt u verwijzen naar de RYMkey, bijvoorbeeld 1, in plaats van de tekenreeks 'North2014May' als u een koppeling met een andere tabel tot stand moet brengen.

Nu laden we op vergelijkbare wijze een brontabel met kosten. De velden Region, Year en Month zijn uitgesloten in de voorafgaande laadbewerking om te voorkomen dat er een synthetische sleutel wordt gemaakt. We maken al een samengestelde sleutel met de functie **autonumberhash256**, waardoor de tabellen worden gekoppeld.

```
RegionCosts:
LOAD Costs,
```

```
AutoNumberHash256(Region, Year, Month) as RYMkey;
```

```
LOAD * INLINE
[ Region, Year, Month, Costs
South, 2013, May, 167
North, 2014, May, 56
North, 2014, June, 199
South, 2014, June, 64
South, 2013, May, 172
South, 2013, May, 126
];
```

Nu kunnen we een tabelvisualisatie toevoegen aan een werkblad en de velden Region, Year en Month toevoegen, alsmede metingen Som voor de verkopen en kosten. De tabel ziet er dan als volgt uit:

Resultatentabel

Region	Year	Month	Sum([Sales])	Sum([Costs])
Totals	-	-	1952	784
North	2014	June	127	199
North	2014	May	592	56
South	2014	June	645	64
South	2013	May	588	465

### IterNo

Deze scriptfunctie retourneert een geheel getal dat aangeeft hoe vaak één record wordt geëvalueerd in een **LOAD**-opdracht met een **while**-clausule. De eerste iteratie heeft nummer 1. De functie **IterNo** is alleen relevant als deze samen met een **while**-clausule wordt gebruikt.

#### Syntaxis:

```
IterNo( )
```

Voorbeelden en resultaten:

#### Voorbeeld:

```
LOAD
  IterNo() as Day,
  Date( StartDate + IterNo() - 1 ) as Date
  while StartDate + IterNo() - 1 <= EndDate;
```

```
LOAD * INLINE
[StartDate, EndDate
2014-01-22, 2014-01-26
];
```

Deze **LOAD**-opdracht genereert één record per datum binnen het bereik dat is gedefinieerd met **StartDate** en **EndDate**.

In dit voorbeeld ziet de resulterende tabel er als volgt uit:

Resultatentabel

Day	Date
1	2014-01-22
2	2014-01-23
3	2014-01-24
4	2014-01-25
5	2014-01-26

### RecNo

Deze scriptfunctie retourneert een geheel getal voor het nummer van de momenteel gelezen rij van de interne tabel. De eerste record is nummer 1.

#### Syntaxis:

```
RecNo ( )
```

In tegenstelling tot bij **RowNo( )**, waarbij rijen in de resulterende Qlik Sense-tabel worden geteld, worden bij **RecNo( )** de records in de tabel met onbewerkte gegevens geteld en gereset als een tabel met onbewerkte gegevens wordt aaneengeschakeld met een andere tabel met onbewerkte gegevens.

#### Voorbeeld: Load-script voor gegevens

Laden van tabellen met onbewerkte gegevens:

```
Tab1:  
LOAD * INLINE  
[A, B  
1, aa  
2, cc  
3, ee];
```

```
Tab2:  
LOAD * INLINE  
[C, D  
5, xx  
4, yy  
6, zz];
```

Record- en rijnummers laden voor geselecteerde rijen:

```
QTab:  
LOAD *,  
RecNo( ),  
RowNo( )
```

```
resident Tab1 where A<>2;
```

```
LOAD
```

```
C as A,
```

```
D as B,
```

```
RecNo( ),
```

```
RowNo( )
```

```
resident Tab2 where A<>5;
```

```
//we don't need the source tables anymore, so we drop them
```

```
Drop tables Tab1, Tab2;
```

De resulterende interne Qlik Sense-tabel:

A	B	RecNo( )	RowNo( )
1	aa	1	1
3	ee	3	2
4	yy	2	3
6	zz	3	4

### RowNo

Deze functie retourneert een geheel getal voor de positie van de huidige rij in de resulterende interne Qlik Sense-tabel. De eerste rij is nummer 1.

#### Syntaxis:

```
RowNo ( [TOTAL] )
```

In tegenstelling tot **RecNo( )**, waarbij records in de tabel met onbewerkte gegevens worden geteld, telt de functie **RowNo( )** geen records mee die zijn uitgesloten met **where**-clausules. Daarnaast wordt deze functie niet opnieuw ingesteld als een tabel met onbewerkte gegevens aan een andere tabel wordt aaneengeschakeld.



*Als u een voorafgaande load gebruikt, dat wil zeggen een aantal gestapelde **LOAD** - opdrachten die uit dezelfde tabel lezen, kunt u **RowNo( )** alleen in de bovenste **LOAD** - opdracht gebruiken. Als u **RowNo( )** in de volgende **LOAD** -opdrachten gebruikt, wordt 0 geretourneerd.*

#### Voorbeeld: Load-script voor gegevens

Laden van tabellen met onbewerkte gegevens:

```
Tab1:
```

```
LOAD * INLINE
```

```
[A, B
```

```
1, aa
```

```
2, cc
```

```
3, ee];
```

```
Tab2:
```

```
LOAD * INLINE
[C, D
5, xx
4, yy
6, zz];
```

Record- en rijnummers laden voor geselecteerde rijen:

```
QTab:
```

```
LOAD *,
RecNo( ),
RowNo( )
resident Tab1 where A<>2;
```

```
LOAD
```

```
C as A,
D as B,
RecNo( ),
RowNo( )
resident Tab2 where A<>5;
```

```
//we don't need the source tables anymore, so we drop them
```

```
Drop tables Tab1, Tab2;
```

De resulterende interne Qlik Sense-tabel:

Resultatentabel

A	B	RecNo( )	RowNo( )
1	aa	1	1
3	ee	3	2
4	yy	2	3
6	zz	3	4

### RowNo - diagramfunctie

**RowNo()** retourneert het nummer van de huidige rij in het huidige kolomsegment in een tabel. Voor bitmapdiagrammen wordt met **RowNo()** het nummer van de huidige rij in het equivalent van de standaard tabel voor het diagram geretourneerd.

Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.

#### Kolomsegmenten

Region	Country	Population	Rank(Population)	
Column segment #1	Americas	Mexico	128,892,753	2
	Americas	Canada	37,742,154	3
	Americas	United States of America	331,002,651	1
Column segment #2	Europe	Sweden	10,099,265	4
	Europe	United Kingdom	67,986,011	2
	Europe	France	65,273,511	3
	Europe	Germany	83,783,942	1



Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.

### Syntaxis:

**RowNo ( [TOTAL] )**

**Retourgegevenstypen:** geheel getal

### Argumenten:

Argument	Beschrijving
TOTAL	Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.

### Voorbeeld: Diagramuitdrukking die gebruikmaakt van RowNo

Voorbeeld - diagramuitdrukking

#### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
Temp:
LOAD * inline [
Customer|Product|OrderNumber|UnitSales|UnitPrice
Astrida|AA|1|4|16
Astrida|AA|7|10|15
Astrida|BB|4|9|9
Betacab|CC|6|5|10
Betacab|AA|5|2|20
Betacab|BB|1|25| 25
Canutility|AA|3|8|15
Canutility|CC|5|4|19
Divadip|CC|2|4|16
Divadip|DD|3|1|25
] (delimiter is '|');
```

#### Diagramuitdrukking

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Customer** en **UnitSales** als dimensies. Voeg **RowNo( )** en **RowNo(TOTAL)** toe als metingen met respectievelijk het label **Rij in segment** en **Row Number**. Voeg de volgende uitdrukking als meting toe aan de tabel:

```
If( RowNo( )=1, 0, UnitSales / Above( UnitSales ) )
```



## Resultaat

Customer	UnitSales	Row in Segment	Row Number	If( RowNo( )=1, 0, UnitSales / Above( UnitSales ))
Astrida	4	1	1	0
Astrida	9	2	2	2.25
Astrida	10	3	3	1.11111111111111
Betacab	2	1	4	0
Betacab	5	2	5	2.5
Betacab	25	3	6	5
Canutility	4	1	7	0
Canutility	8	2	8	2
Divadip	1	1	9	0
Divadip	4	2	10	4

## Uitleg

De kolom **Row in Segment** toont de resultaten 1,2,3 voor het kolomsegment dat de waarden bevat van UnitSales voor klant Astrida. De rijnummering begint vervolgens opnieuw bij 1 voor het volgende kolomsegment, Betacab.

In de kolom **Row Number** worden de dimensies genegeerd vanwege het argument TOTAL voor RowNo() en het aantal rijen in de tabel worden geteld.

Deze uitdrukking retourneert 0 voor de eerste rij in elk kolomsegment, zodat de kolom er als volgt uitziet:

0, 2.25, 1.1111111, 0, 2.5, 5, 0, 2, 0 en 4.

**Zie ook:**

*p Above - diagramfunctie (page 1268)*

## 5.7 Datum- en tijdfuncties

Datum- en tijdfuncties in Qlik Sense worden gebruikt voor het transformeren en converteren van datum- en tijdwaarden. Alle functies kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

Functies zijn gebaseerd op een serienummer voor datum-tijd dat gelijk is aan het aantal dagen sinds 30 december 1899. De waarde van het gehele getal geeft de dag aan en de fractionele waarde geeft de tijd van de dag aan.

In Qlik Sense wordt de numerieke waarde van de parameter gebruikt, dus een getal is geldig als parameter ook als het niet is opgemaakt als een datum of tijd. Als de parameter niet overeenkomt met een numerieke waarde, bijvoorbeeld als het een tekenreeks is, wordt in Qlik Sense geprobeerd om de tekenreeks te interpreteren op basis van de omgevingsvariabelen voor datum en tijd.

Als de tijdnotatie die in de parameter wordt gebruikt niet overeenkomt met de notatie die is ingesteld in de omgevingsvariabelen, is geen correcte interpretatie in Qlik Sense mogelijk. U kunt dit oplossen door de instellingen te wijzigen of gebruik te maken van een interpretatiefunctie.

In de voorbeelden voor elke functie wordt uitgegaan van de standaardnotatie voor tijd en datum, hh:mm:ss en YYYY-MM-DD (ISO 8601).



*Bij Het verwerken van een tijdsstempel met een datum- of tijdfunctie, negeert Qlik Sense zomertijdparameters tenzij de datum- of tijdfunctie een geografische positie bevat.*

*ConvertToLocalTime( filetime('Time.qvd'), 'Paris') gebruikt bijvoorbeeld wel zomertijdparameters terwijl ConvertToLocalTime(filetime('Time.qvd'), 'GMT-01:00') geen zomertijdparameters gebruikt.*

### Overzicht van datum- en tijdfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Uitdrukkingen van tijd in gehele getallen

##### second

Deze functie retourneert een geheel getal dat de seconde representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

```
second (expression)
```

##### minute

Deze functie retourneert een geheel getal dat de minuut representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

```
minute (expression)
```

##### hour

Deze functie retourneert een geheel getal dat het uur representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

```
hour (expression)
```

##### day

Deze functie retourneert een geheel getal dat de dag representeert als de breuk van **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

```
day (expression)
```

### **week**

Deze functie retourneert een geheel getal dat het weeknummer representeert conform ISO 8601. Het weeknummer wordt berekend volgens de datuminterpretatie van de uitdrukking, in overeenstemming met de standaardinterpretatie van getallen.

```
week (expression)
```

### **month**

Deze functie retourneert een duale waarde: een maandnaam zoals vastgelegd in de omgevingsvariabele **MonthNames**, en een geheel getal tussen 1-12. De maand wordt berekend volgens de datuminterpretatie van de uitdrukking, in overeenstemming met de standaardinterpretatie van getallen.

```
month (expression)
```

### **year**

Deze functie retourneert een geheel getal dat het jaar representeert als de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

```
year (expression)
```

### **weekyear**

Deze functie retourneert het jaar waarbij het weeknummer hoort conform ISO 8601. Het weeknummer varieert van 1 tot ongeveer 52.

```
weekyear (expression)
```

### **weekday**

Deze functie retourneert een duale waarde met:

- De naam van een dag zoals gedefinieerd in de omgevingsvariabele **DayNames**.
- Een geheel getal tussen 0 en 6 dat overeenkomt met de nominale dag van de week (0-6).

```
weekday (date)
```

## Tijdstempelfuncties

### **now**

Deze functie retourneert een tijdstempel van de huidige tijd. Deze functie retourneert waarden in de indeling van de systeemvariabele **TimeStamp**. De standaardwaarde voor **timer\_mode** is 1.

```
now ([ timer_mode ])
```

### **today**

Deze functie retourneert de huidige datum. Deze functie retourneert waarden in de indeling van de systeemvariabele `DateFormat`.

```
today ([timer_mode])
```

### **LocalTime**

Deze functie retourneert een tijdstempel met de huidige tijd voor een opgegeven tijdzone.

```
localtime ([timezone [, ignoreDST ]])
```

### Maakfuncties

#### **makedate**

Deze functie retourneert een datum die is berekend voor het jaar **YYYY**, de maand **MM** en de dag **DD**.

```
makedate (YYYY [ , MM [ , DD ] ])
```

#### **makeweekdate**

Deze functie retourneert een datum die is berekend voor het jaar **YYYY**, de week **WW** en de dag van de week **D**.

```
makeweekdate (YYYY [ , WW [ , D ] ])
```

#### **maketime**

Deze functie retourneert een tijd die is berekend voor het uur **hh**, de minuut **mm** en de seconde **ss**.

```
maketime (hh [ , mm [ , ss [ .fff ] ] ])
```

### Andere datumfuncties

#### **AddMonths**

Deze functie retourneert de datum **n** maanden na **startdate** of, als **n** negatief is, de datum **n** maanden vóór **startdate**.

```
addmonths (startdate, n , [ , mode])
```

#### **AddYears**

Deze functie retourneert de datum **n** jaar na **startdate** of, als **n** negatief is, de datum **n** jaar vóór **startdate**.

```
addyears (startdate, n)
```

#### **yeartodate**

Deze functie bepaalt of de invoertijdstempel binnen het jaar valt waarin het script voor het laatst is geladen en retourneert True als dat het geval is en False als dat niet het geval is.

```
yeartodate (date [ , yearoffset [ , firstmonth [ , todaydate] ] ])
```

### TimeZone-functies

#### **timezone**

De functie retourneert de tijdzone zoals gedefinieerd op de computer waarop de Qlik-engine wordt uitgevoerd.

```
timezone ( )
```

#### **GMT**

Deze functie retourneert de huidige Greenwich Mean Time, zoals afgeleid uit de landinstellingen.

```
GMT ( )
```

#### **UTC**

Retourneert de huidige Coordinated Universal Time.

**UTC ( )**

### **daylightsaving**

Retourneert de huidige aanpassing voor zomertijd zoals die in Windows is gedefinieerd.

**daylightsaving ( )**

### **converttolocaltime**

Converteert een UTC- of GMT-tijdstempel naar een lokale tijd met twee representaties. De plaats kan een reeks steden, plaatsen en tijdzones overal ter wereld zijn.

**converttolocaltime** (timestamp [, place [, ignore\_dst=false]])

## Functies voor instellen van tijd

### **setdateyear**

Deze functie neemt een **timestamp** en een **year** als invoer en werkt de **timestamp** bij met het **year** dat is opgegeven in de invoer.

**setdateyear** (timestamp, year)

### **setdateyearmonth**

Deze functie neemt een **timestamp**, een **month** en een **year** als invoer en werkt de **timestamp** bij met het **year** en de **month** die zijn opgegeven in de invoer.

**setdateyearmonth** (timestamp, year, month)

## Functies In...

### **inyear**

Deze functie retourneert True als **timestamp** binnen het jaar valt dat **base\_date** bevat.

**inyear** (date, basedate , shift [, first\_month\_of\_year = 1])

### **inyeartodate**

Deze functie retourneert True als **timestamp** binnen het gedeelte van het jaar ligt dat **base\_date** bevat tot en met de laatste milliseconde van **base\_date**.

**inyeartodate** (date, basedate , shift [, first\_month\_of\_year = 1])

### **inquarter**

Deze functie retourneert True als **timestamp** binnen het kwartaal valt dat **base\_date** bevat.

**inquarter** (date, basedate , shift [, first\_month\_of\_year = 1])

### **inquartertodate**

Deze functie retourneert True als **timestamp** binnen het gedeelte van het kwartaal ligt dat **base\_date** bevat tot en met de laatste milliseconde van **base\_date**.

**inquartertodate** (date, basedate , shift [, first\_month\_of\_year = 1])

### **inmonth**

Deze functie retourneert True als **timestamp** in de maand valt die **base\_date** bevat.

```
inmonth (date, basedate , shift)
```

### **inmonthtodate**

Retourneert True als **date** binnen het gedeelte van de maand ligt dat **basedate** bevat tot en met de laatste milliseconde van **basedate**.

```
inmonthtodate (date, basedate , shift)
```

### **inmonths**

Deze functie bepaalt of een tijdstempel valt binnen dezelfde periode van de maand, twee maanden, kwartaal, tertiaal of half jaar als basisdatum. Het is tevens mogelijk om te bepalen of de tijdstempel binnen een voorafgaande of volgende tijdsperiode valt.

```
inmonths (n, date, basedate , shift [, first_month_of_year = 1])
```

### **inmonthstodate**

Deze functie bepaalt of een tijdstempel binnen het gedeelte van een periode van de maand, twee maanden, kwartaal, tertiaal of half jaar tot en met de laatste milliseconde van **base\_date** valt. Het is tevens mogelijk om te bepalen of de tijdstempel binnen een voorafgaande of volgende tijdsperiode valt.

```
inmonthstodate (n, date, basedate , shift [, first_month_of_year = 1])
```

### **inweek**

Deze functie retourneert True als **timestamp** binnen de week valt die **base\_date** bevat.

```
inweek (date, basedate , shift [, weekstart])
```

### **inweektodate**

Deze functie retourneert True als **timestamp** binnen het gedeelte van de week ligt dat **base\_date** bevat tot en met de laatste milliseconde van **base\_date**.

```
inweektodate (date, basedate , shift [, weekstart])
```

### **inlunarweek**

Deze functie bepaalt of **timestamp** binnen de maanweek valt die **base\_date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week. Met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

```
inlunarweek (date, basedate , shift [, weekstart])
```

### **inlunarweektodate**

Met deze functie wordt bepaald of **timestamp** binnen het gedeelte van de maanweek ligt tot en met de laatste milliseconde van **base\_date**. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

```
inlunarweektodate (date, basedate , shift [, weekstart])
```

### **inday**

Deze functie retourneert True als **timestamp** binnen de dag valt die **base\_timestamp** bevat.

```
inday (timestamp, basetimestamp , shift [, daystart])
```

### **indaytotime**

Deze functie retourneert True als **timestamp** binnen het gedeelte van de dag ligt dat **base\_timestamp** bevat tot en met de exacte milliseconde van **base\_timestamp**.

```
indaytotime (timestamp, basetimestamp , shift [, daystart])
```

## Functies Begin ... Eind

### **yearstart**

Deze functie retourneert een tijdsstempel die overeenkomt met het begin van de eerste dag van het jaar dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
yearstart ( date [, shift = 0 [, first_month_of_year = 1]])
```

### **yearend**

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van het jaar dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
yearend ( date [, shift = 0 [, first_month_of_year = 1]])
```

### **yearname**

Deze functie retourneert een jaar in vier cijfers als weergavewaarde met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van het jaar dat **date** bevat.

```
yearname (date [, shift = 0 [, first_month_of_year = 1]] )
```

### **quarterstart**

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van het kwartaal dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
quarterstart (date [, shift = 0 [, first_month_of_year = 1]])
```

### **quarterend**

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van het kwartaal dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
quarterend (date [, shift = 0 [, first_month_of_year = 1]])
```

### **quartername**

Deze functie retourneert de maanden van het kwartaal (opgemaakt volgens de scriptvariabele **MonthNames**) en het jaar met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van het kwartaal.

```
quartername (date [, shift = 0 [, first_month_of_year = 1]])
```

### monthstart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maand die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
monthstart (date [, shift = 0])
```

### monthend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van de maand die **date** bevat. De standaarduitvoernotatie is de **dateFormat** die is ingesteld in het script.

```
monthend (date [, shift = 0])
```

### monthname

Deze functie retourneert de maand (opgemaakt volgens de scriptvariabele **MonthNames**) en het jaar met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maand.

```
monthname (date [, shift = 0])
```

### monthsstart

Deze functie retourneert een waarde die overeenkomt met de tijdstempel van de eerste milliseconde van de periode van een maand, twee maanden, kwartaal, tertiaal of half jaar die een basisdatum bevat. Het is tevens mogelijk om de tijdstempel voor een voorafgaande of volgende tijdsperiode te bepalen. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
monthsstart (n, date [, shift = 0 [, first_month_of_year = 1]])
```

### monthsend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de periode van een maand, twee maanden, kwartaal, tertiaal of half jaar die een basisdatum bevat. Het is tevens mogelijk om de tijdstempel voor een voorafgaande of volgende tijdsperiode te bepalen.

```
monthsend (n, date [, shift = 0 [, first_month_of_year = 1]])
```

### monthsname

Deze functie retourneert een weergavewaarde die het bereik vertegenwoordigt van de maanden van de periode (geformatteerd volgens de scriptvariabele **MonthNames**) alsmede het jaar. De onderliggende numerieke waarde komt overeen met een tijdstempel van de eerste milliseconde van de periode van een maand, twee maanden, kwartaal, tertiaal of half jaar die een basisdatum bevat.

```
monthsname (n, date [, shift = 0 [, first_month_of_year = 1]])
```

### weekstart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de kalenderweek die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
weekstart (date [, shift = 0 [, weekoffset = 0]])
```



### weekend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag (zondag) van de kalenderweek die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

```
weekend (date [, shift = 0 [,weekoffset = 0]])
```

### weekname

Deze functie retourneert het jaar en weeknummer met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de week die **date** bevat.

```
weekname (date [, shift = 0 [,weekoffset = 0]])
```

### lunarweekstart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maanweek die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

```
lunarweekstart (date [, shift = 0 [,weekoffset = 0]])
```

### lunarweekend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van de maanweek die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

```
lunarweekend (date [, shift = 0 [,weekoffset = 0]])
```

### lunarweekname

Deze functie retourneert een weergavewaarde voor het jaar en maanweeknummer dat overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de week die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

```
lunarweekname (date [, shift = 0 [,weekoffset = 0]])
```

### daystart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel met de eerste milliseconde van de dag in het argument **time**. De standaarduitvoernotatie is de **TimestampFormat** die is ingesteld in het script.

```
daystart (timestamp [, shift = 0 [, dayoffset = 0]])
```

### dayend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de dag in **time**. De standaarduitvoernotatie is de **TimestampFormat** die is ingesteld in het script.

```
dayend (timestamp [, shift = 0 [, dayoffset = 0]])
```

### dayname

Deze functie retourneert een datum met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de dag met **time**.

```
dayname (timestamp [, shift = 0 [, dayoffset = 0]])
```

### Functies voor dagnummering

#### age

De functie **age** retourneert de leeftijd op het tijdstip van de **timestamp** (in voltooide jaren) van iemand die is geboren op **date\_of\_birth**.

```
age (timestamp, date_of_birth)
```

#### networkdays

De functie **networkdays** retourneert het aantal werkdagen (maandag-vrijdag) tussen **start\_date** en **end\_date**, waarbij rekening wordt gehouden met een eventueel opgegeven **holiday**.

```
networkdays (start:date, end_date {, holiday})
```

#### firstworkdate

De functie **firstworkdate** retourneert de laatste begindatum waarbij **no\_of\_workdays** (maandag-vrijdag) kan worden gehaald als niet later wordt gestopt dan **end\_date**, waarbij rekening wordt gehouden met een eventueel opgegeven vakantiedagen. **end\_date** en **holiday** moeten geldige datums of tijdstempels zijn.

```
firstworkdate (end_date, no_of_workdays {, holiday} )
```

#### lastworkdate

De functie **lastworkdate** retourneert de vroegste einddatum waarbij **no\_of\_workdays** (maandag-vrijdag) kan worden gehaald als wordt begonnen op **start\_date**, waarbij rekening wordt gehouden met een eventueel opgegeven **holiday**. **start\_date** en **holiday** moeten geldige datums of tijdstempels zijn.

```
lastworkdate (start_date, no_of_workdays {, holiday})
```

#### daynumberofyear

Deze functie berekent het dagnummer van het jaar waarbinnen een tijdstempel valt. De berekening wordt uitgevoerd vanaf de eerste milliseconde van de eerste dag van het jaar, maar de beginmaand kan worden verschoven.

```
daynumberofyear (date[, firstmonth])
```

#### daynumberofquarter

Deze functie berekent het dagnummer van het kwartaal waarbinnen een tijdstempel valt. Deze functie wordt gebruikt wanneer een masterkalender wordt gemaakt.

```
daynumberofquarter (date[, firstmonth])
```

### addmonths

Deze functie retourneert de datum **n** maanden na **startdate** of, als **n** negatief is, de datum **n** maanden vóór **startdate**.

### Syntaxis:

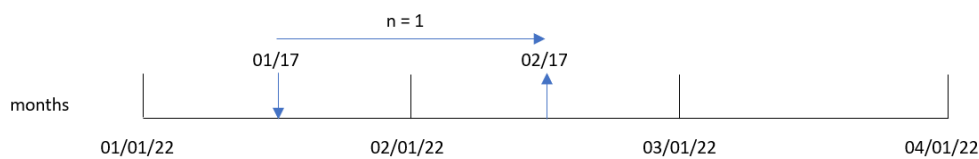
```
AddMonths (startdate, n , [ , mode])
```

### Retourgegevenstypen: dual

De `addmonths()`-functie voegt een gedefinieerd aantal maanden toe aan of trekt dit aantal af, `n`, van een `startdate` en retourneert de resulterende datum.

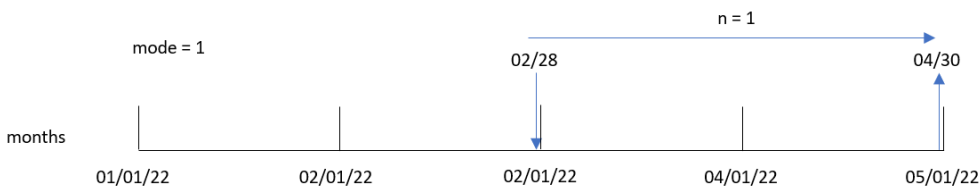
Het argument `mode` heeft invloed op `startdate`-waarden op of na de 28e van de maand. Door het argument `mode` in te stellen op 1, retourneert de `addmonths()`-functie een datum die gelijk is aan de relatieve afstand tot het eind van de maand als de `startdate`.

#### Voorbeelddiagram van de `addmonths()`-functie



Bijvoorbeeld: 28 februari is de laatste dag van de maand. Als de `addmonths()`-functie, met een `mode` van 1, wordt gebruikt om de datum twee maanden later te retourneren, retourneert de functie de laatste datum van april: 30 april.

#### Voorbeelddiagram van de `addmonths()`-functie met `mode=1`



### Argumenten

Argument	Beschrijving
<code>startdate</code>	De startdatum als tijdstempel, bijvoorbeeld '2012-10-12'.
<code>n</code>	Aantal maanden als een positief of negatief geheel getal.
<code>mode</code>	Geeft aan of de maand relatief tot het begin of het eind van de maand is toegevoegd. Standaardmodus is 0 voor toevoegingen relatief tot het begin van de maand. Stel modus in op 1 voor toevoegingen relatief tot het eind van de maand. Als de modus is ingesteld op 1 en de invoerdatum de 28ste of later is, controleert de functie hoeveel dagen er op de startdatum nog resteren voordat het eind van de maand is bereikt. Hetzelfde aantal dagen dat resteert voordat het eind van de maand is bereikt, wordt ingesteld op de geretourneerde datum.

### Wanneer gebruiken

De `addmonths()`-functie wordt doorgaans gebruikt in een uitdrukking om een datum te vinden met een bepaald aantal maanden vóór of na een bepaalde periode.

Bijvoorbeeld: de `addmonths()`-functie kan worden gebruikt om de einddatum van mobiele telefooncontracten te identificeren.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>addmonths ('01/29/2003' ,3)</code>	Retourneert '04/29/2003'.
<code>addmonths ('01/29/2003' ,3,0)</code>	Retourneert '04/29/2003'.
<code>addmonths ('01/29/2003' ,3,1)</code>	Retourneert '04/28/2003'.
<code>addmonths ('01/29/2003' ,1,0)</code>	Retourneert '02/28/2003'.
<code>addmonths ('01/29/2003' ,1,1)</code>	Retourneert '02/26/2003'.
<code>addmonths ('02/28/2003' ,1,0)</code>	Retourneert '03/28/2003'.
<code>addmonths ('02/28/2003' ,1,1)</code>	Retourneert '03/31/2003'.
<code>addmonths ('01/29/2003' ,-3)</code>	Retourneert '10/29/2002'.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (`MM/DD/YYYY`) indeling.
- Het maken van een veld, `two_months_later`, dat de datum retourneert voor twee maanden nadat de transactie plaatsvond.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    addmonths(date,2) as two_months_later
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/10/2020',37.23
```

```
8189,'02/28/2020',17.17
```

```
8190,'04/09/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'12/27/2021',25.12
```

```
8204,'02/02/2022',46.23
```

```
8205,'02/26/2022',84.21
```

```
8206,'03/07/2022',96.24
```

```
8207,'03/11/2022',67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

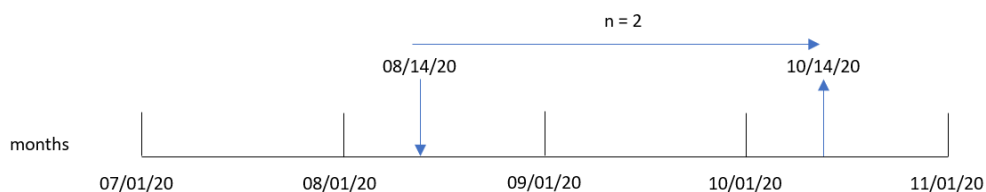
- `date`
- `two_months_later`

Resultatentabel

date	two_months_later
01/10/2020	03/10/2020
02/28/2020	04/28/2020
04/09/2020	06/09/2020
04/16/2020	06/16/2020
05/21/2020	07/21/2020
08/14/2020	10/14/2020
10/07/2020	12/07/2020
12/05/2020	02/05/2021
01/22/2021	03/22/2021
02/03/2021	04/03/2021
03/17/2021	05/17/2021
04/23/2021	06/23/2021
05/04/2021	07/04/2021
06/30/2021	08/30/2021
07/26/2021	09/26/2021
12/27/2021	02/27/2022
02/02/2022	04/02/2022
02/26/2022	04/26/2022
03/07/2022	05/07/2022
03/11/2022	05/11/2022

Het veld `two_months_later` wordt gemaakt in de vorige load-instructie met behulp van de functie `addmonths()`. Het eerste opgegeven argument geeft aan welke datum wordt geëvalueerd. Het tweede argument is het aantal maanden dat moet worden opgeteld bij of afgetrokken van de startdate. In dit geval is de waarde 2 opgegeven.

*Diagram van `addmonths()`-functie, voorbeeld zonder aanvullende argumenten*



Transactie 8193 vond plaats op 14 augustus. Daarom retourneert de `addmonths()`-functie 14 oktober 2020 voor het veld `two_months_later`.

### Voorbeeld 2 - Relatief einde van de maand

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set eindemaandstransacties in 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (`MM/DD/YYYY`) indeling.
- Het maken van een veld, `relative_two_months_prior`, dat de datum retourneert voor de relatieve datum voor het einde van de maand voordat de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    addmonths(date,-2,1) as relative_two_months_prior
  ;

Load
*
Inline
[
id,date,amount
8188,'01/28/2022',37.23
8189,'01/31/2022',57.54
8190,'02/28/2022',17.17
8191,'04/29/2022',88.27
8192,'04/30/2022',57.42
8193,'05/31/2022',53.80
8194,'08/14/2022',82.06
8195,'10/07/2022',40.39
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

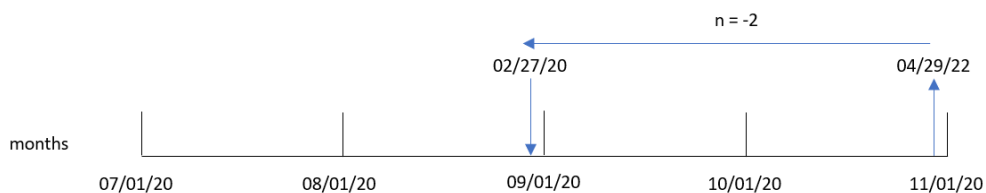
- `date`
- `relative_two_months_prior`

Resultatentabel

date	relative_two_months_prior
01/28/2022	11/27/2021
01/31/2022	11/30/2021
02/28/2022	12/31/2021
04/29/2022	02/27/2022
04/30/2022	02/28/2022
05/31/2022	03/31/2022
08/14/2022	06/14/2022
10/07/2022	08/07/2022

Het veld `relative_two_months_prior` wordt gemaakt in de vorige load-instructie met behulp van de functie `addmonths()`. Het eerste opgegeven argument geeft aan welke datum wordt geëvalueerd. Het tweede argument is het aantal maanden dat moet worden opgeteld bij of afgetrokken van de `startdate`. In dit geval is de waarde `-2` opgegeven. Het laatste argument is de modus met de waarde `1`, die de functie forceert de relatieve datum voor het einde van de maand te berekenen voor alle datums die groter zijn dan of gelijk zijn aan 28.

Diagram van de `addmonths()`-functie, voorbeeld met `n=-2`



Transactie 8191 vindt plaats op 29 april 2022. In eerste instantie zou twee maanden eerder de maand instellen op februari. Vervolgens berekent de functie de relatieve waarde voor het einde van de maand, omdat het derde argument van de functie de modus instelt op `1` en de dagwaarde later is dan de 27e. De functie identificeert dat de 29e de op één na laatste dag van april is en retourneert daarom de op één na laatste dag van februari, de 27e.

### Voorbeeld 3 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.



In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die de datum retourneert voor twee maanden nadat de transactie plaatsvond, wordt gemaakt als een meting in een diagramobject.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/10/2020',37.23
```

```
8189,'02/28/2020',17.17
```

```
8190,'04/09/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'12/27/2021',25.12
```

```
8204,'02/02/2022',46.23
```

```
8205,'02/26/2022',84.21
```

```
8206,'03/07/2022',96.24
```

```
8207,'03/11/2022',67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Maak de volgende meting:

```
=addmonths(date,2)
```

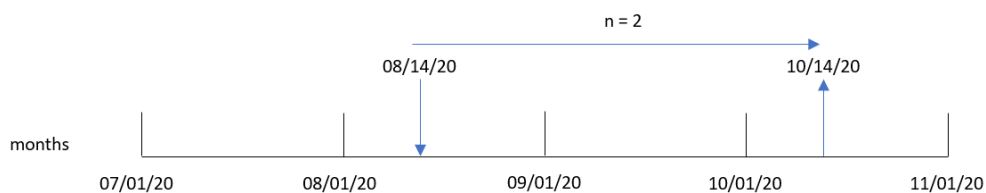
Resultatentabel

date	=addmonths(date,2)
01/10/2020	03/10/2020
02/28/2020	04/28/2020
04/09/2020	06/09/2020

date	=addmonths(date,2)
04/16/2020	06/16/2020
05/21/2020	07/21/2020
08/14/2020	10/14/2020
10/07/2020	12/07/2020
12/05/2020	02/05/2021
01/22/2021	03/22/2021
02/03/2021	04/03/2021
03/17/2021	05/17/2021
04/23/2021	06/23/2021
05/04/2021	07/04/2021
06/30/2021	08/30/2021
07/26/2021	09/26/2021
12/27/2021	02/27/2022
02/02/2022	04/02/2022
02/26/2022	04/26/2022
03/07/2022	05/07/2022
03/11/2022	05/11/2022

De meting `two_months_later` wordt in het diagramobject gemaakt met behulp van de functie `addmonths()`. Het eerste opgegeven argument geeft aan welke datum wordt geëvalueerd. Het tweede argument is het aantal maanden dat moet worden opgeteld bij of afgetrokken van de `startdate`. In dit geval is de waarde 2 opgegeven.

*Diagram van `addmonths()`-functie, diagramobjectvoorbeeld*



Transactie 8193 vond plaats op 14 augustus. Daarom retourneert de `addmonths()`-functie 14 oktober 2020 voor het veld `two_months_later`.

### Voorbeeld 4 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Mobile_Plans`.
- Informatie met de contract-id, begindatum, contractduur en maandelijkse kosten.

De eindgebruiker wil graag een diagramobject dat de einddatum van elk telefooncontract op contract-id weergeeft.

#### Load-script

```
Mobile_Plans:
Load
*
Inline
[
contract_id,start_date,contract_length,monthly_fee
8188,'01/13/2020',18,37.23
8189,'02/26/2020',24,17.17
8190,'03/27/2020',36,88.27
8191,'04/16/2020',24,57.42
8192,'05/21/2020',24,53.80
8193,'08/14/2020',12,82.06
8194,'10/07/2020',18,40.39
8195,'12/05/2020',12,87.21
8196,'01/22/2021',12,95.93
8197,'02/03/2021',18,45.89
8198,'03/17/2021',24,36.23
8199,'04/23/2021',24,25.66
8200,'05/04/2021',12,82.77
8201,'06/30/2021',12,69.98
8202,'07/26/2021',12,76.11
8203,'12/27/2021',36,25.12
8204,'06/06/2022',24,46.23
8205,'07/18/2022',12,84.21
8206,'11/14/2022',12,96.24
8207,'12/12/2022',18,67.67
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- contract\_id
- start\_date
- contract\_length

Maak de volgende meting aan om de einddatum van elk contract te berekenen:

```
=addmonths(start_date,contract_length, 0)
```

Resultatentabel

contract_id	start_date	contract_length	=addmonths(start_date,contract_length,0)
8188	01/13/2020	18	07/13/2021
8189	02/26/2020	24	02/26/2022
8190	03/27/2020	36	03/27/2023
8191	04/16/2020	24	04/16/2022
8192	05/21/2020	24	05/21/2022
8193	08/14/2020	12	08/14/2021
8194	10/07/2020	18	04/07/2022
8195	12/05/2020	12	12/05/2021
8196	01/22/2021	12	01/22/2022
8197	02/03/2021	18	08/03/2022
8198	03/17/2021	24	03/17/2023
8199	04/23/2021	24	04/23/2023
8200	05/04/2021	12	05/04/2022
8201	06/30/2021	12	06/30/2022
8202	07/26/2021	12	07/26/2022
8203	12/27/2021	36	12/27/2024
8204	06/06/2022	24	06/06/2024
8205	07/18/2022	12	07/18/2023
8206	11/14/2022	12	11/14/2023
8207	12/12/2022	18	06/12/2024

### addyears

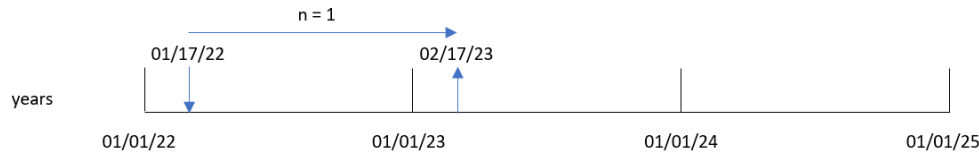
Deze functie retourneert de datum **n** jaar na **startdate** of, als **n** negatief is, de datum **n** jaar vóór **startdate**.

#### Syntaxis:

```
AddYears (startdate, n)
```

### Retourgegevenstypen: dual

Voorbeelddiagram van de `addyears()`-functie



De `addyears()`-functie voegt een gedefinieerd aantal jaar,  $n$ , toe aan of trekt dit aantal af van een `startdate`. Vervolgens wordt de resulterende datum geretourneerd.

#### Argumenten

Argument	Beschrijving
<code>startdate</code>	De startdatum als tijdstempel, bijvoorbeeld '2012-10-12'.
<code>n</code>	Aantal jaren als een positief of negatief geheel getal.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>addyears ('01/29/2010', 3)</code>	Retourneert '01/29/2013'.
<code>addyears ('01/29/2010', -1)</code>	Retourneert '01/29/2009'.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Eenvoudig voorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Er wordt een veld gemaakt, `two_years_later`, dat de datum retourneert voor twee jaar nadat de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    addyears(date,2) as two_years_later
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,'01/10/2020',37.23
8189,'02/28/2020',17.17
8190,'04/09/2020',88.27
8191,'04/16/2020',57.42
8192,'05/21/2020',53.80
8193,'08/14/2020',82.06
8194,'10/07/2020',40.39
8195,'12/05/2020',87.21
8196,'01/22/2021',95.93
8197,'02/03/2021',45.89
8198,'03/17/2021',36.23
8199,'04/23/2021',25.66
8200,'05/04/2021',82.77
8201,'06/30/2021',69.98
8202,'07/26/2021',76.11
8203,'12/27/2021',25.12
8204,'02/02/2022',46.23
8205,'02/26/2022',84.21
8206,'03/07/2022',96.24
8207,'03/11/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

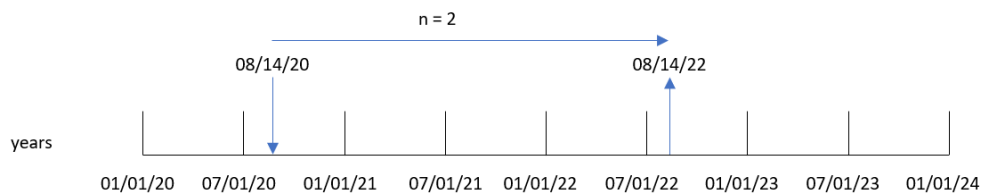
- `date`
- `two_years_later`

Resultatentabel

<code>date</code>	<code>two_years_later</code>
01/10/2020	01/10/2022
02/28/2020	02/28/2022
04/09/2020	04/09/2022
04/16/2020	04/16/2022
05/21/2020	05/21/2022
08/14/2020	08/14/2022
10/07/2020	10/07/2022
12/05/2020	12/05/2022
01/22/2021	01/22/2023
02/03/2021	02/03/2023
03/17/2021	03/17/2023
04/23/2021	04/23/2023
05/04/2021	05/04/2023
06/30/2021	06/30/2023
07/26/2021	07/26/2023
12/27/2021	12/27/2023
02/02/2022	02/02/2024
02/26/2022	02/26/2024
03/07/2022	03/07/2024
03/11/2022	03/11/2024

Het veld `two_years_later` wordt gemaakt in de vorige load-instructie met behulp van de functie `addyears` (). Het eerste opgegeven argument geeft aan welke datum wordt geëvalueerd. Het tweede argument is het aantal jaar dat moet worden opgeteld bij of afgetrokken van de begindatum. In dit geval is de waarde 2 opgegeven.

### Diagram van `addyears()`-functie, basisvoorbeeld



Transactie 8193 vond plaats op 14 augustus 2020. Daarom retourneert de `addyears()`-functie 14 augustus 2022 voor het veld `two_years_later`.

### Voorbeeld 2 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

Maak in een diagramobject een meting aan, `prior_year_date`, die de datum retourneert die één jaar eerder ligt dan de datum waarop de transactie heeft plaatsgevonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/10/2020',37.23
```

```
8189,'02/28/2020',17.17
```

```
8190,'04/09/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```



```
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '12/27/2021', 25.12
8204, '02/02/2022', 46.23
8205, '02/26/2022', 84.21
8206, '03/07/2022', 96.24
8207, '03/11/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Maak de volgende meting aan om de datum te berekenen die één jaar eerder ligt per transactie:

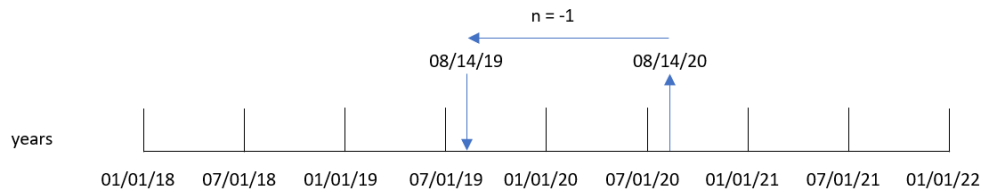
```
=addyears(date, -1)
```

Resultatentabel

date	=addyears(date,-1)
01/10/2020	01/10/2019
02/28/2020	02/28/2019
04/09/2020	04/09/2019
04/16/2020	04/16/2019
05/21/2020	05/21/2019
08/14/2020	08/14/2019
10/07/2020	10/07/2019
12/05/2020	12/05/2019
01/22/2021	01/22/2020
02/03/2021	02/03/2020
03/17/2021	03/17/2020
04/23/2021	04/23/2020
05/04/2021	05/04/2020
06/30/2021	06/30/2020
07/26/2021	07/26/2020
12/27/2021	12/27/2020
02/02/2022	02/02/2021
02/26/2022	02/26/2021
03/07/2022	03/07/2021
03/11/2022	03/11/2021

De meting `one_year_prior` wordt in het diagramobject gemaakt met behulp van de functie `addyears()`. Het eerste opgegeven argument geeft aan welke datum wordt geëvalueerd. Het tweede argument is het aantal jaar dat moet worden opgeteld bij of afgetrokken van de `startdate`. In dit geval is de waarde `-1` opgegeven.

*Diagram van `addyears()`-functie, diagramobjectvoorbeeld*



Transactie 8193 vond plaats op 14 augustus. Daarom retourneert de `addyears()`-functie 14 augustus 2019 voor het veld `one_year_prior`.

### Voorbeeld 3 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `warranties`.
- Informatie met de product-id, aankoopdatum, garantieduur en aankoopprijs.

De eindgebruiker wil graag een diagramobject dat de einddatum van de garantieperiode van elk product op product-id weergeeft.

#### Load-script

warranties:

Load

\*

Inline

[

`product_id,purchase_date,warranty_length,purchase_price`

8188, '01/13/2020', 4, 32000

8189, '02/26/2020', 2, 28000

8190, '03/27/2020', 3, 41000

8191, '04/16/2020', 4, 17000

8192, '05/21/2020', 2, 25000

8193, '08/14/2020', 1, 59000

8194, '10/07/2020', 2, 12000

8195, '12/05/2020', 3, 12000

8196, '01/22/2021', 4, 24000

8197, '02/03/2021', 1, 50000

```
8198, '03/17/2021', 2, 80000
8199, '04/23/2021', 3, 10000
8200, '05/04/2021', 4, 30000
8201, '06/30/2021', 3, 30000
8202, '07/26/2021', 4, 20000
8203, '12/27/2021', 4, 10000
8204, '06/06/2022', 2, 25000
8205, '07/18/2022', 1, 32000
8206, '11/14/2022', 1, 30000
8207, '12/12/2022', 4, 22000
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- product\_id
- purchase\_date
- warranty\_length

Maak de volgende meting aan om de einddatum van de garantie per product te berekenen:

```
=addyears(purchase_date, warranty_length)
```

Resultatentabel

product_id	purchase_date	warranty_length	=addyears(purchase_date,warranty_length)
8188	01/13/2020	4	01/13/2024
8189	02/26/2020	2	02/26/2022
8190	03/27/2020	3	03/27/2023
8191	04/16/2020	4	04/16/2024
8192	05/21/2020	2	05/21/2022
8193	08/14/2020	1	08/14/2021
8194	10/07/2020	2	10/07/2022
8195	12/05/2020	3	12/05/2023
8196	01/22/2021	4	01/22/2025
8197	02/03/2021	1	02/03/2022
8198	03/17/2021	2	03/17/2023
8199	04/23/2021	3	04/23/2024
8200	05/04/2021	4	05/04/2025
8201	06/30/2021	3	06/30/2024
8202	07/26/2021	4	07/26/2025

product_id	purchase_date	warranty_length	=addyears(purchase_date,warranty_length)
8203	12/27/2021	4	12/27/2025
8204	06/06/2022	2	06/06/2024
8205	07/18/2022	1	07/18/2023
8206	11/14/2022	1	11/14/2023
8207	12/12/2022	4	12/12/2026

## age

De functie **age** retourneert de leeftijd op het tijdstip van de **timestamp** (in voltooide jaren) van iemand die is geboren op **date\_of\_birth**.

### Syntaxis:

```
age(timestamp, date_of_birth)
```

Kan een uitdrukking zijn.

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
<b>timestamp</b>	De tijdstempel, of uitdrukking die wordt herleid tot een tijdstempel, tot waar het voltooide aantal jaren moet worden berekend.
<b>date_of_birth</b>	Geboortedatum van de persoon van wie de leeftijd wordt berekend. Kan een uitdrukking zijn.

### Voorbeelden en resultaten:

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

#### Scriptvoorbeelden

Voorbeeld	Resultaat
<code>age('25/01/2014', '29/10/2012')</code>	Retourneert 1.
<code>age('29/10/2014', '29/10/2012')</code>	Retourneert 2.

### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
Employees:
LOAD * INLINE [
Member|DateOfBirth
John|28/03/1989
Linda|10/12/1990
Steve|5/2/1992
Birg|31/3/1993
Raj|19/5/1994
Prita|15/9/1994
Su|11/12/1994
Goran|2/3/1995
Sunny|14/5/1996
Ajoa|13/6/1996
Daphne|7/7/1998
Biffy|4/8/2000
] (delimiter is |);
AgeTable:
Load *,
age('20/08/2015', DateOfBirth) As Age
Resident Employees;
Drop table Employees;
```

De resulterende tabel bevat de geretourneerde waarden van age voor elk van de records in de tabel.

Resultatentabel

Member	DateOfBirth	Age
John	28/03/1989	26
Linda	10/12/1990	24
Steve	5/2/1992	23
Birg	31/3/1993	22
Raj	19/5/1994	21
Prita	15/9/1994	20
Su	11/12/1994	20
Goran	2/3/1995	20
Sunny	14/5/1996	19
Ajoa	13/6/1996	19
Daphne	7/7/1998	17
Biffy	4/8/2000	15

### converttolocaltime

Converteert een UTC- of GMT-tijdstempel naar een lokale tijd met twee representaties. De plaats kan een reeks steden, plaatsen en tijdzones overal ter wereld zijn.

#### Syntaxis:

```
ConvertToLocalTime(timestamp [, place [, ignore_dst=false]])
```

Retourgegevenstypen: dual

Argumenten:

### Argumenten

Argument	Beschrijving
<b>timestamp</b>	De tijdstempel of uitdrukking die wordt herleid tot een tijdstempel waarnaar moet worden geconverteerd.
<b>place</b>	<p>Een plaats of tijdzone uit de tabel met geldige plaatsen en tijdzones hieronder. U kunt ook gebruikmaken van GMT of UTC om de lokale tijd te definiëren. De volgende waarden en tijdverschuivingsbereiken zijn geldig:</p> <ul style="list-style-type: none"> <li>• GMT</li> <li>• GMT-12:00 - GMT-01:00</li> <li>• GMT+01:00 - GMT+14:00</li> <li>• UTC</li> <li>• UTC-12:00 - UTC-01:00</li> <li>• UTC+01:00 - UTC+14:00</li> </ul> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> <i>U kunt alleen standaard tijdverschuivingen gebruiken. Het is niet mogelijk een willekeurige tijdverschuiving te gebruiken, bijvoorbeeld GMT-04:27.</i></p> </div>
<b>ignore_dst</b>	Stel in op True als u DST (zomertijd) wilt negeren.

Bij de resulterende tijd wordt rekening gehouden met zomertijd, tenzij **ignore\_dst** is ingesteld op True.

### Geldige plaatsen en tijdzones

A-C	D-K	L-R	S-Z
Abu Dhabi	Darwin	La Paz	Samoa
Adelaide	Dhaka	Lima	Santiago
Alaska	Eastern Time (US & Canada)	Lisbon	Sapporo
Amsterdam	Edinburgh	Ljubljana	Sarajevo
Arizona	Ekaterinburg	London	Saskatchewan
Astana	Fiji	Madrid	Seoul
Athens	Georgetown	Magadan	Singapore
Atlantic Time (Canada)	Greenland	Mazatlan	Skopje

## 5 Script- en diagramfuncties

A-C	D-K	L-R	S-Z
Auckland	Greenwich Mean Time : Dublin	Melbourne	Sofia
Azores	Guadalajara	Mexico City	Solomon Is.
Baghdad	Guam	Mid-Atlantic	Sri Jayawardenepura
Baku	Hanoi	Minsk	St. Petersburg
Bangkok	Harare	Monrovia	Stockholm
Beijing	Hawaii	Monterrey	Sydney
Belgrade	Helsinki	Moscow	Taipei
Berlin	Hobart	Mountain Time (US & Canada)	Tallinn
Bern	Hong Kong	Mumbai	Tashkent
Bogota	Indiana (East)	Muscat	Tbilisi
Brasilia	International Date Line West	Nairobi	Tehran
Bratislava	Irkutsk	New Caledonia	Tokyo
Brisbane	Islamabad	New Delhi	Urumqi
Brussels	Istanbul	Newfoundland	Warsaw
Bucharest	Jakarta	Novosibirsk	Wellington
Budapest	Jerusalem	Nuku'alofa	West Central Africa
Buenos Aires	Kabul	Osaka	Vienna
Cairo	Kamchatka	Pacific Time (US & Canada)	Vilnius
Canberra	Karachi	Paris	Vladivostok
Cape Verde Is.	Kathmandu	Perth	Volgograd
Caracas	Kolkata	Port Moresby	Yakutsk
Casablanca	Krasnoyarsk	Prague	Yerevan
Central America	Kuala Lumpur	Pretoria	Zagreb
Central Time (US & Canada)	Kuwait	Quito	-
Chennai	Kyiv	Riga	-
Chihuahua	-	Riyadh	-

A-C	D-K	L-R	S-Z
Chongqing	-	Rome	-
Copenhagen	-	-	-

Voorbeelden en resultaten:

### Scriptvoorbeelden

Voorbeeld	Resultaat
<code>ConvertToLocalTime('2007-11-10 23:59:00','Paris')</code>	Retourneert '2007-11-11 00:59:00' en het corresponderende interne tijdstempel.
<code>ConvertToLocalTime(UTC(), 'GMT-05:00')</code>	Retourneert de tijd voor de Noord-Amerikaanse oostkust, bijvoorbeeld New York.
<code>ConvertToLocalTime(UTC(), 'GMT-05:00', True)</code>	Retourneert de tijd voor de Noord-Amerikaanse oostkust, bijvoorbeeld New York, zonder aanpassing voor zomertijd.

## day

Deze functie retourneert een geheel getal dat de dag representeert als de breuk van **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

De functie retourneert de dag van de maand voor een specifieke datum. Dit wordt vaak gebruikt om een dagveld af te leiden als onderdeel van een kalenderdimensie.

### Syntaxis:

**day** (*expression*)

**Retourgegevenstypen:** geheel getal

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>day( 1971-10-12 )</code>	retourneert 12
<code>day( 35648 )</code>	retourneert 6, omdat 35648 = 1997-08-06

## Voorbeeld 1 - gegevensverzameling DateFormat (script)

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.



Het load-script bevat:

- Een gegevensverzameling met de naam `Master_Calendar`. De `DateFormat` systeemvariabele is ingesteld op `DD/MM/JJJJ`.
- Een voorafgaande lading maakt een extra veld met de naam `day_of_month` met behulp van de `day()` functie.
- Een extra veld met de naam `long_date` met behulp van de `date()` functie voor het weergeven van de volledige naam van de maand.

### Load-script

```
SET DateFormat='DD/MM/YYYY';
```

```
Master_Calendar:
```

```
Load
```

```
    date,  
    date(date, 'dd-MMMM-YYYY') as long_date,  
    day(date) as day_of_month
```

```
Inline
```

```
[  
date  
03/11/2022  
03/12/2022  
03/13/2022  
03/14/2022  
03/15/2022  
03/16/2022  
03/17/2022  
03/18/2022  
03/19/2022  
03/20/2022  
03/21/2022  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `long_date`
- `day_of_month`

Resultatentabel

<code>date</code>	<code>long_date</code>	<code>day_of_month</code>
03/11/2022	11-March- 2022	11
03/12/2022	11-March- 2022	12
03/13/2022	13-March- 2022	13

date	long_date	day_of_month
03/14/2022	14-March- 2022	14
03/15/2022	15-March- 2022	15
03/16/2022	16-March- 2022	16
03/17/2022	17-March- 2022	17
03/18/2022	18-March- 2022	18
03/19/2022	19-March- 2022	19
03/20/2022	20-March- 2022	20
03/21/2022	21-March- 2022	21

De dag van maand wordt juist geëvalueerd door de `day()` functie in het script.

### Voorbeeld 2 - ANSI-datums (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met de naam `master_calendar`. De `DateFormat` systeemvariabele `DD/MM/JJJJ` wordt gebruikt. De datums die zijn opgenomen in de gegevensverzameling hebben echter een standaard ANSI-datumnotatie.
- Een voorafgaande lading maakt een extra veld met de naam `day_of_month` met behulp van de `date()` functie.
- Een extra veld met de naam `long_date` met behulp van de `date()` functie voor het weergeven van de datum met de volledige naam van de maand.

#### Load-script

```
SET DateFormat='DD/MM/YYYY';
Master_Calendar:
Load
    date,
    date(date, 'dd-MMMM-YYYY') as long_date,
    day(date) as day_of_month

Inline
[
date
2022-03-11
2022-03-12
2022-03-13
```

```
2022-03-14
2022-03-15
2022-03-16
2022-03-17
2022-03-18
2022-03-19
2022-03-20
2022-03-21
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- long\_date
- day\_of\_month

Resultatentabel

date	long_date	day_of_month
03/11/2022	11-March- 2022	11
03/12/2022	11-March- 2022	12
03/13/2022	13-March- 2022	13
03/14/2022	14-March- 2022	14
03/15/2022	15-March- 2022	15
03/16/2022	16-March- 2022	16
03/17/2022	17-March- 2022	17
03/18/2022	18-March- 2022	18
03/19/2022	19-March- 2022	19
03/20/2022	20-March- 2022	20
03/21/2022	21-March- 2022	21

De dag van maand wordt juist geëvalueerd door de `day()` functie in het script.

### Voorbeeld 3 - Datums zonder notatie (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met de naam `Master_Calendar`. De `dateFormat` systeemvariabele `DD/MM/JJJJ` wordt gebruikt.
- Een voorafgaande lading maakt een extra veld met de naam `day_of_month` met behulp van de `day()` functie.
- De originele datum zonder notatie, met de naam `unformatted_date`.
- Een extra veld met de naam `long_date` met de `date()` wordt gebruikt om de numerieke datum om te zetten in een datumveld met notatie.

### Load-script

```
SET DateFormat='DD/MM/YYYY';
```

```
Master_Calendar:
```

```
Load
```

```
    unformatted_date,  
    date(unformatted_date, 'dd-MMMM-YYYY') as long_date,  
    day(date) as day_of_month
```

```
Inline
```

```
[
```

```
unformatted_date
```

```
44868
```

```
44898
```

```
44928
```

```
44958
```

```
44988
```

```
45018
```

```
45048
```

```
45078
```

```
45008
```

```
45038
```

```
45068
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `unformatted_date`
- `long_date`
- `day_of_month`

Resultatentabel

<code>unformatted_date</code>	<code>long_date</code>	<code>day_of_month</code>
44868	03-November- 2022	3
44898	03-December- 2022	3

unformatted_date	long_date	day_of_month
44928	02-January- 2023	2
44958	01-February- 2023	1
44988	03-March- 2023	3
45008	23-March- 2023	23
45018	02-April- 2023	2
45038	22-April- 2023	22
45048	02-May- 2023	2
45068	22-May- 2023	22
45078	01-June- 2023	1

De dag van maand wordt juist geëvalueerd door de `day()` functie in het script.

### Voorbeeld 4 - Vervalmaand berekenen (diagram)

Load-script en diagramuitdrukking

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met bestellingen die in maart zijn geplaatst, met de naam `orders`. De tabel bevat drie velden.
  - `id`
  - `order_date`
  - `amount`

#### Load-script

Orders:

Load

```
id,  
order_date,  
amount
```

Inline

```
[  
id,order_date,amount  
1,03/01/2022,231.24  
2,03/02/2022,567.28  
3,03/03/2022,364.28  
4,03/04/2022,575.76  
5,03/05/2022,638.68  
6,03/06/2022,785.38
```

```
7,03/07/2022,967.46
8,03/08/2022,287.67
9,03/09/2022,764.45
10,03/10/2022,875.43
11,03/11/2022,957.35
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: order\_date.

Om de leverdatum te berekenen, maakt u deze meting: =day(order\_date+5).

Resultatentabel

order_date	=day(order_date+5)
03/11/2022	16
03/12/2022	17
03/13/2022	18
03/14/2022	19
03/15/2022	20
03/16/2022	21
03/17/2022	22
03/18/2022	23
03/19/2022	24
03/20/2022	25
03/21/2022	26

De day() functie bepaalt op correcte wijze dat een bestelling die op 11 maart is geplaatst op basis van een leveringsperiode van 5 dagen op 16 maart wordt bezorgd.

### dayend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de dag in **time**. De standaarduitvoernotatie is de **TimestampFormat** die is ingesteld in het script.

#### Syntaxis:

```
DayEnd(time[, [period_no[, day_start]])
```

### Wanneer gebruiken

De functie `dayend()` wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening een deel van de dag wordt gebruikt dat nog niet heeft plaatsgevonden. Bijvoorbeeld om de totale kosten te berekenen die gedurende de dag nog gemaakt moeten worden.

**Retourgegevenstypen:** dual

#### Argumenten

Argument	Beschrijving
<b>time</b>	De te evalueren tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de dag aangeeft die <b>time</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande dagen aan en positieve waarden geven volgende dagen aan.
<b>day_start</b>	Als u wilt opgeven dat dagen niet om middernacht beginnen, geeft u een verschuiving aan als een fractie van een dag in <b>day_start</b> . Bijvoorbeeld 0.125 om 03.00 uur aan te geven. Met andere woorden: om de offset te maken, verdeelt u de begintijd met 24 uur. Bijvoorbeeld: om een dag te laten beginnen om 7.00 uur, gebruikt u de fractie 7/24.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/YYYY. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>dayend('01/25/2013 16:45:00')</code>	Returns 01/25/2013 23:59:59. PM
<code>dayend('01/25/2013 16:45:00', -1)</code>	Retourneert 01/24/2013 23:59:59. PM
<code>dayend('01/25/2013 16:45:00', 0, 0.5)</code>	Returns 01/26/2013 11:59:59. PM

### Voorbeeld 1 - Basisscript

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met een lijst met datums wordt geladen in een tabel met de naam "Calendar".
- De standaard `DateFormat`-systeemvariabele (`MM/DD/YYYY`).
- Een voorafgaande lading om een extra veld te maken met de naam `EOD_timestamp` met behulp van de `dayend()`-functie.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Calendar:
```

```
  Load
    date,
    dayend(date) as EOD_timestamp
  ;
```

```
Load
```

```
date
```

```
Inline
```

```
[
```

```
date
```

```
03/11/2022 1:47:15 AM
```

```
03/12/2022 4:34:58 AM
```

```
03/13/2022 5:15:55 AM
```

```
03/14/2022 9:25:14 AM
```

```
03/15/2022 10:06:54 AM
```

```
03/16/2022 10:44:42 AM
```

```
03/17/2022 11:33:30 AM
```

```
03/18/2022 12:58:14 PM
```

```
03/19/2022 4:23:12 PM
```

```
03/20/2022 6:42:15 PM
```

```
03/21/2022 7:41:16 PM
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `EOD_timestamp`

Resultatentabel

<code>date</code>	<code>Tijdstempel einde dag</code>
03/11/2022 1:47:15 AM	3/11/2022 11:59:59 PM



<b>date</b>	<b>Tijdstempel einde dag</b>
03/12/2022 4:34:58 AM	3/12/2022 11:59:59 PM
03/13/2022 5:15:55 AM	3/13/2022 11:59:59 PM
03/14/2022 9:25:14 AM	3/14/2022 11:59:59 PM
03/15/2022 10:06:54 AM	3/15/2022 11:59:59 PM
03/16/2022 10:44:42 AM	3/16/2022 11:59:59 PM
03/17/2022 11:33:30 AM	3/17/2022 11:59:59 PM
03/18/2022 12:58:14 PM	3/18/2022 11:59:59 PM
03/19/2022 4:23:12 PM	3/19/2022 11:59:59 PM
03/20/2022 6:42:15 PM	3/20/2022 11:59:59 PM
03/21/2022 7:41:16 PM	3/21/2022 11:59:59 PM

Zoals u in de bovenstaande tabel kunt zien, wordt het tijdstempel einde dag gegenereerd voor iedere datum in uw gegevensset. Het tijdstempel is de indeling van de systeemvariabele `TimestampFormat` `M/D/YYYY h:mm:ss[.fff] TT`.

### Voorbeeld 2 - `period_no`

#### Load-script en resultaten

##### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

U laadt een gegevensset met serviceboekingen in een tabel genaamd `Services`.

De gegevensset bevat de volgende velden:

- `service_id`
- `service_date`
- `amount`

U maakt twee nieuwe velden in de tabel:

- `deposit_due_date`: De datum waarop de storting moet zijn ontvangen. Dit is het einde van de dag, drie dagen vóór de `service_date`.
- `final_payment_due_date`: De datum waarop de laatste betaling moet zijn ontvangen. Dit is het einde van de dag, zeven dagen na de `service_date`.

De twee velden hierboven zijn gemaakt in een voorafgaande lading met gebruik van de `dayend()`-functie. Ze geven de eerste twee parameters, `time` en `period_no`.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Services:
```

```
  Load
    *,
    dayend(service_date,-3) as deposit_due_date,
    dayend(service_date,7) as final_payment_due_date
  ;
```

```
Load
```

```
service_id,
service_date,
```

```
amount
```

```
Inline
```

```
[
service_id, service_date,amount
1,03/11/2022 9:25:14 AM,231.24
2,03/12/2022 10:06:54 AM,567.28
3,03/13/2022 10:44:42 AM,364.28
4,03/14/2022 11:33:30 AM,575.76
5,03/15/2022 12:58:14 PM,638.68
6,03/16/2022 4:23:12 PM,785.38
7,03/17/2022 6:42:15 PM,967.46
8,03/18/2022 7:41:16 PM,287.67
9,03/19/2022 8:14:15 PM,764.45
10,03/20/2022 9:23:51 PM,875.43
11,03/21/2022 10:04:41 PM,957.35
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- service\_date
- deposit\_due\_date
- final\_payment\_due\_date

Resultatentabel

service_date	deposit_due_date	final_payment_due_date
03/11/2022 9:25:14 AM	3/8/2022 11:59:59 PM	3/18/2022 11:59:59 PM
03/12/2022 10:06:54 AM	3/9/2022 11:59:59 PM	3/19/2022 11:59:59 PM
03/13/2022 10:44:42 AM	3/10/2022 11:59:59 PM	3/20/2022 11:59:59 PM
03/14/2022 11:33:30 AM	3/11/2022 11:59:59 PM	3/21/2022 11:59:59 PM
03/15/2022 12:58:14 PM	3/12/2022 11:59:59 PM	3/22/2022 11:59:59 PM
03/16/2022 4:23:12 PM	3/13/2022 11:59:59 PM	3/23/2022 11:59:59 PM

service_date	deposit_due_date	final_payment_due_date
03/17/2022 6:42:15 PM	3/14/2022 11:59:59 PM	3/24/2022 11:59:59 PM
03/18/2022 7:41:16 PM	3/15/2022 11:59:59 PM	3/25/2022 11:59:59 PM
03/19/2022 8:14:15 PM	3/16/2022 11:59:59 PM	3/26/2022 11:59:59 PM
03/20/2022 9:23:51 PM	3/17/2022 11:59:59 PM	3/27/2022 11:59:59 PM
03/21/2022 10:04:41 PM	3/18/2022 11:59:59 PM	3/28/2022 11:59:59 PM

De waarden van de nieuwe velden hebben de `TimestampFormat M/D/YYYY h:mm:ss[.fff] TT`. Omdat de functie `dayend()` is gebruikt, zijn de tijdstempelwaarden allemaal de laatste milliseconde van de dag.

De vervaldatumwaarden voor de storting zijn drie dagen vóór de servicedatum, omdat het tweede argument dat in de `dayend()`-functie wordt doorgegeven negatief is.

De vervaldatumwaarden voor de laatste betaling zijn zeven dagen na de servicevervaldatum, omdat het tweede argument dat in de `dayend()`-functie wordt doorgegeven negatief is.

### Voorbeeld 3 - day\_start-script

#### Load-script en resultaten

##### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

De gegevensset en het scenario die in dit voorbeeld worden gebruikt, zijn dezelfde als in het vorige voorbeeld.

Net als in het vorige voorbeeld, maakt u twee nieuwe velden:

- `deposit_due_date`: De datum waarop de storting moet zijn ontvangen. Dit is het einde van de dag, drie dagen vóór de `service_date`.
- `final_payment_due_date`: De datum waarop de laatste betaling moet zijn ontvangen. Dit is het einde van de dag, zeven dagen na de `service_date`.

Uw bedrijf wil echter volgens een beleid werken waarbij de werkdag om 5:00 PM begint en de volgende dag om 5:00 PM eindigt. Uw bedrijf kan dan de transacties bewaken die binnen die werkuren plaatsvinden.

Om deze vereisten te realiseren, worden de twee bovenstaande velden gemaakt in een voorafgaande load met de `dayend()`-functie en de drie parameters, `time`, `period_no` en `day_start`.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Services:
```

```
  Load  
  *,  
  dayend(service_date,-3,17/24) as deposit_due_date,
```

```
    dayend(service_date,7,17/24) as final_payment_due_date
;
Load
service_id,
service_date,
amount
Inline
[
service_id, service_date,amount
1,03/11/2022 9:25:14 AM,231.24
2,03/12/2022 10:06:54 AM,567.28
3,03/13/2022 10:44:42 AM,364.28
4,03/14/2022 11:33:30 AM,575.76
5,03/15/2022 12:58:14 PM,638.68
6,03/16/2022 4:23:12 PM,785.38
7,03/17/2022 6:42:15 PM,967.46
8,03/18/2022 7:41:16 PM,287.67
9,03/19/2022 8:14:15 PM,764.45
10,03/20/2022 9:23:51 PM,875.43
11,03/21/2022 10:04:41 PM,957.35
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- service\_date
- deposit\_due\_date
- final\_payment\_due\_date

Resultatentabel

service_date	deposit_due_date	final_payment_due_date
03/11/2022 9:25:14 AM	3/8/2022 4:59:59 PM	3/18/2022 4:59:59 PM
03/12/2022 10:06:54 AM	3/9/2022 4:59:59 PM	3/19/2022 4:59:59 PM
03/13/2022 10:44:42 AM	3/10/2022 4:59:59 PM	3/20/2022 4:59:59 PM
03/14/2022 11:33:30 AM	3/11/2022 4:59:59 PM	3/21/2022 4:59:59 PM
03/15/2022 12:58:14 PM	3/12/2022 4:59:59 PM	3/22/2022 4:59:59 PM
03/16/2022 4:23:12 PM	3/13/2022 4:59:59 PM	3/23/2022 4:59:59 PM
03/17/2022 6:42:15 PM	3/14/2022 4:59:59 PM	3/24/2022 4:59:59 PM
03/18/2022 7:41:16 PM	3/15/2022 4:59:59 PM	3/25/2022 4:59:59 PM
03/19/2022 8:14:15 PM	3/16/2022 4:59:59 PM	3/26/2022 4:59:59 PM
03/20/2022 9:23:51 PM	3/17/2022 4:59:59 PM	3/27/2022 4:59:59 PM
03/21/2022 10:04:41 PM	3/18/2022 4:59:59 PM	3/28/2022 4:59:59 PM

De datums blijven zoals in voorbeeld 2, maar de datums hebben nu een tijdstempel van de laatste milliseconde vóór 5:00 PM, omdat de waarde van het derde argument, `day_start`, dat in de `dayend()`-functie is doorgegeven, 17/24 was.

### Voorbeeld 4 - diagramvoorbeeld

#### Load-script en diagramuitdrukking

##### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

De gegevensset en het scenario die in dit voorbeeld worden gebruikt, zijn dezelfde als in de vorige twee voorbeelden. Uw bedrijf wil graag volgens een beleid werken waarbij de werkdag om 5:00 PM begint en de volgende dag om 5:00 PM eindigt.

Net als in het vorige voorbeeld, maakt u twee nieuwe velden:

- `deposit_due_date`: De datum waarop de storting moet zijn ontvangen. Dit is het einde van de dag, drie dagen vóór de `service_date`.
- `final_payment_due_date`: De datum waarop de laatste betaling moet zijn ontvangen. Dit is het einde van de dag, zeven dagen na de `service_date`.

##### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Services:
```

```
Load
```

```
service_id,
```

```
service_date,
```

```
amount
```

```
Inline
```

```
[
```

```
service_id, service_date, amount
```

```
1,03/11/2022 9:25:14 AM,231.24
```

```
2,03/12/2022 10:06:54 AM,567.28
```

```
3,03/13/2022 10:44:42 AM,364.28
```

```
4,03/14/2022 11:33:30 AM,575.76
```

```
5,03/15/2022 12:58:14 PM,638.68
```

```
6,03/16/2022 4:23:12 PM,785.38
```

```
7,03/17/2022 6:42:15 PM,967.46
```

```
8,03/18/2022 7:41:16 PM,287.67
```

```
9,03/19/2022 8:14:15 PM,764.45
```

```
10,03/20/2022 9:23:51 PM,875.43
```

```
11,03/21/2022 10:04:41 PM,957.35
```

```
];
```

##### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

```
service_date.
```

Om het `deposit_due_date`-veld te maken, maakt u deze meting:

```
=dayend(service_date,-3,17/24).
```

Vervolgens maakt u om het `final_payment_due_date`-veld te maken, deze meting:

```
=dayend(service_date,7,17/24).
```

Resultatentabel

<code>service_date</code>	<code>=dayend(service_date,-3,17/24)</code>	<code>=dayend(service_date,7,17/24)</code>
03/11/2022	3/8/2022 16:59:59 PM	3/18/2022 16:59:59 PM
03/12/2022	3/9/2022 16:59:59 PM	3/19/2022 16:59:59 PM
03/13/2022	3/10/2022 16:59:59 PM	3/20/2022 16:59:59 PM
03/14/2022	3/11/2022 16:59:59 PM	3/21/2022 16:59:59 PM
03/15/2022	3/12/2022 16:59:59 PM	3/22/2022 16:59:59 PM
03/16/2022	3/13/2022 16:59:59 PM	3/23/2022 16:59:59 PM
03/17/2022	3/14/2022 16:59:59 PM	3/24/2022 16:59:59 PM
03/18/2022	3/15/2022 16:59:59 PM	3/25/2022 16:59:59 PM
03/19/2022	3/16/2022 16:59:59 PM	3/26/2022 16:59:59 PM
03/20/2022	3/17/2022 16:59:59 PM	3/27/2022 16:59:59 PM
03/21/2022	3/18/2022 16:59:59 PM	3/28/2022 16:59:59 PM

De waarden van de nieuwe velden hebben de `TimestampFormat` `M/D/YYYY h:mm:ss[.fff] TT`. Omdat de functie `dayend()` is gebruikt, zijn de tijdstempelwaarden allemaal de laatste milliseconde van de dag.

De vervaldatumwaarden voor de belating zijn drie dagen vóór de servicedatum, omdat het tweede argument dat in de `dayend()`-functie wordt doorgegeven negatief is.

De vervaldatumwaarden voor de laatste betaling zijn zeven dagen na de servicevervaldatum, omdat het tweede argument dat in de `dayend()`-functie wordt doorgegeven negatief is.

De datums hebben nu een tijdstempel van de laatste milliseconde vóór 5:00 PM omdat de waarde van het derde argument, `day_start`, dat in de `dayend()`-functie is doorgegeven, `17/24` was.

### daylightsaving

Retourneert de huidige aanpassing voor zomertijd zoals die in Windows is gedefinieerd.

#### Syntaxis:

```
DaylightSaving( )
```

**Retourgegevenstypen:** dual

#### Voorbeeld:

```
daylightsaving( )
```

## dayname

Deze functie retourneert een datum met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de dag met **time**.

### Syntaxis:

```
DayName (time[, period_no [, day_start]])
```

Retourgegevenstypen: dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
<b>time</b>	De te evalueren tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de dag aangeeft die <b>time</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande dagen aan en positieve waarden geven volgende dagen aan.
<b>day_start</b>	Als u wilt opgeven dat dagen niet om middernacht beginnen, geeft u een verschuiving aan als een fractie van een dag in <b>day_start</b> . Bijvoorbeeld 0.125 om 03.00 uur aan te geven.

### Voorbeelden en resultaten:

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

#### Scriptvoorbeelden

Voorbeeld	Resultaat
<code>dayname('25/01/2013 16:45:00')</code>	Retourneert 25/01/2013.
<code>dayname('25/01/2013 16:45:00', -1)</code>	Retourneert 24/01/2013.
<code>dayname('25/01/2013 16:45:00', 0, 0.5 )</code>	Retourneert 25/01/2013. Als de volledige tijdstempel wordt weergegeven, verschijnt de onderliggende numerieke waarde die overeenkomt met '25/01/2013 12:00:00.000'.

### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

## 5 Script- en diagramfuncties

In dit voorbeeld wordt de naam van de dag gemaakt aan de hand van de tijdstempel die het einde van de dag na elke factuurdatum in de tabel aangeeft.

```
TempTable:
LOAD RecNo() as InvID, * Inline [
InvDate
28/03/2012
10/12/2012
5/2/2013
31/3/2013
19/5/2013
15/9/2013
11/12/2013
2/3/2014
14/5/2014
13/6/2014
7/7/2014
4/8/2014
];
```

```
InvoiceData:
LOAD *,
DayName(InvDate, 1) AS DName
Resident TempTable;
Drop table TempTable;
```

De resulterende tabel bevat de oorspronkelijke datums en een kolom met de retourwaarde van de functie `dayname()`. U kunt de volledige tijdstempel weergeven door de opmaak op te geven in het eigenschappenvenster.

Resultatentabel

InvDate	DName
28/03/2012	29/03/2012 00:00:00
10/12/2012	11/12/2012 00:00:00
5/2/2013	07/02/2013 00:00:00
31/3/2013	01/04/2013 00:00:00
19/5/2013	20/05/2013 00:00:00
15/9/2013	16/09/2013 00:00:00
11/12/2013	12/12/2013 00:00:00
2/3/2014	03/03/2014 00:00:00
14/5/2014	15/05/2014 00:00:00
13/6/2014	14/06/2014 00:00:00
7/7/2014	08/07/2014 00:00:00
4/8/2014	05/08/2014 00:00:00



## daynumberofquarter

Deze functie berekent het dagnummer van het kwartaal waarbinnen een tijdstempel valt. Deze functie wordt gebruikt wanneer een masterkalender wordt gemaakt.

### Syntaxis:

```
DayNumberOfQuarter(timestamp[, start_month])
```

Retourgegevenstypen: geheel getal

### Argumenten

Argument	Beschrijving
<b>timestamp</b>	De te evalueren datum of tijdstempel.
<b>start_month</b>	Door een <b>start_month</b> tussen 2 en 12 op te geven (1 indien weggelaten), kunt u het begin van het jaar naar voren verplaatsen naar de eerste dag van een andere maand. Als u bijvoorbeeld wilt werken met een boekjaar dat begint op 1 maart, geeft u <b>start_month</b> = 3 op.

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

### Voorbeelden van functies

Voorbeeld	Resultaat
DayNumberOfQuarter('12/09/2014')	Retourneert 74, het dagnummer van het huidige kwartaal.
DayNumberOfQuarter('12/09/2014', 3)	Retourneert 12, het dagnummer van het huidige kwartaal. In dit geval begint het eerste kwartaal met maart (omdat bij start_month 3 is opgegeven). Dit betekent dat het huidige kwartaal het derde kwartaal is, dat begint op 1 september.

## Voorbeeld 1 - Januari begin van het jaar (script)

Load-script en resultaten

### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een eenvoudige gegevensset met een lijst met datums, die wordt geladen in een tabel met de naam calendar. De standaard DateFormat-systeemvariabele mm/dd/yyyy wordt gebruikt.

- Een voorafgaande lading maakt een extra veld met de naam `DayNrQtr` met behulp van de `DayNumberOfQuarter()` functie.

Naast de datum worden er geen extra parameters aan de functie toegevoegd.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Calendar:
```

```
Load
```

```
    date,  
    DayNumberOfQuarter(date) as DayNrQtr  
    ;
```

```
Load
```

```
date
```

```
Inline
```

```
[  
date  
01/01/2022  
01/10/2022  
01/31/2022  
02/01/2022  
02/10/2022  
02/28/2022  
03/01/2022  
03/31/2022  
04/01/2022  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `daynrqtr`

Resultatentabel

<code>date</code>	<code>daynrqtr</code>
01/01/2022	1
01/10/2022	10
01/31/2022	31
02/01/2022	32
02/10/2022	41
02/28/2022	59
03/01/2022	61

date	daynrqtr
03/31/2022	91
04/01/2022	1

De eerste dag van het jaar is 1 januari omdat er geen tweede argument is doorgegeven in de `DayNumberOfQuarter()`-functie.

1 Januari is de eerste dag van het kwartaal en 1 februari is de 32e dag van het kwartaal. 31 maart is de 91e en laatste dag van het kwartaal. 1 April is de eerste dag van het tweede kwartaal.

### Voorbeeld 2 - Februari begin van het jaar (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele `mm/dd/yyyy` wordt gebruikt.
- Een `start_month`-argument dat begint op 1 februari. Dit stelt het financiële jaar in op 1 februari.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Calendar:
```

```
Load
    date,
    DayNumberOfQuarter(date,2) as DayNrQtr
;
```

```
Load
```

```
date
```

```
Inline
```

```
[
```

```
date
```

```
01/01/2022
```

```
01/10/2022
```

```
01/31/2022
```

```
02/01/2022
```

```
02/10/2022
```

```
02/28/2022
```

```
03/01/2022
```

```
03/31/2022
```

```
04/01/2022
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- daynrqtr

Resultatentabel

date	daynrqtr
01/01/2022	62
01/10/2022	71
01/31/2022	92
02/01/2022	1
02/10/2022	10
02/28/2022	28
03/01/2022	30
03/31/2022	60
04/01/2022	61

De eerste dag van het jaar is 1 februari omdat het tweede argument dat is doorgegeven in de `DayNumberOfQuarter()`-functie 2 is.

Het eerste kwartaal van het jaar loopt van februari tot en met april en het vierde kwartaal van november tot en met januari. Dit wordt weergegeven in de tabel met resultaten. Hierbij is 1 februari de eerste dag van het kwartaal en 31 januari is de 92e en laatste dag van het kwartaal.

### Voorbeeld 3 - Januari begin van het jaar (diagram)

Load-script en diagramuitdrukking

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele `mm/dd/yyyy` wordt gebruikt.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De waarde van de dag van het kwartaal wordt berekend via een meting in een diagramobject.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Calendar:  
Load  
date  
Inline  
[  
date  
01/01/2022  
01/10/2022  
01/31/2022  
02/01/2022  
02/10/2022  
02/28/2022  
03/01/2022  
03/31/2022  
04/01/2022  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Maak de volgende meting:

```
=daynumberofquarter(date)
```

Resultatentabel

date	=daynumberofquarter(date)
01/01/2022	1
01/10/2022	10
01/31/2022	31
02/01/2022	32
02/10/2022	41
02/28/2022	59
03/01/2022	61
03/31/2022	91
04/01/2022	1

De eerste dag van het jaar is 1 januari omdat er geen tweede argument is doorgegeven in de `DayNumberOfQuarter()`-functie.

1 Januari is de eerste dag van het kwartaal en 1 februari is de 32e dag van het kwartaal. 31 maart is de 91e en laatste dag van het kwartaal. 1 April is de eerste dag van het tweede kwartaal.

### Voorbeeld 4 - Februari begin van het jaar (diagram)

Load-script en diagramuitdrukking

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele `mm/dd/yyyy` wordt gebruikt.
- Het financiële jaar loopt van 1 februari tot en met 31 januari.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De waarde van de dag van het kwartaal wordt berekend via een meting in een diagramobject.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Calendar:
```

```
Load
```

```
date
```

```
InLine
```

```
[
```

```
date
```

```
01/01/2022
```

```
01/10/2022
```

```
01/31/2022
```

```
02/01/2022
```

```
02/10/2022
```

```
02/28/2022
```

```
03/01/2022
```

```
03/31/2022
```

```
04/01/2022
```

```
];
```

#### Diagramobject

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting:

```
=daynumberofquarter(date,2)
```

### Resultaten

Resultatentabel

date	=daynumberofquarter(date,2)
01/01/2022	62
01/10/2022	71
01/31/2022	92
02/01/2022	1
02/10/2022	10
02/28/2022	28
03/01/2022	30
03/31/2022	60
04/01/2022	61

De eerste dag van het jaar is 1 februari omdat het tweede argument dat is doorgegeven in de `DayNumberOfQuarter()`-functie 2 is.

Het eerste kwartaal van het jaar loopt van februari tot en met april en het vierde kwartaal van november tot en met januari. Dit wordt aangeduid in de tabel met resultaten. Hierbij is 1 februari de eerste dag van het kwartaal en 31 januari is de 92e en laatste dag van het kwartaal.

### daynumberofyear

Deze functie berekent het dagnummer van het jaar waarbinnen een tijdstempel valt. De berekening wordt uitgevoerd vanaf de eerste milliseconde van de eerste dag van het jaar, maar de beginmaand kan worden verschoven.

#### Syntaxis:

```
DayNumberOfYear(timestamp[,start_month])
```

Retourgegevenstypen: geheel getal

Argumenten

Argument	Beschrijving
timestamp	De te evalueren datum of tijdstempel.
start_month	Door een <b>start_month</b> tussen 2 en 12 op te geven (1 indien weggelaten), kunt u het begin van het jaar naar voren verplaatsen naar de eerste dag van een andere maand. Als u bijvoorbeeld wilt werken met een boekjaar dat begint op 1 maart, geeft u <b>start_month</b> = 3 op.

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>DayNumberOfYear( '12/09/2014' )</code>	Retourneert 256, het dagnummer geteld vanaf de eerste dag van het jaar.
<code>DayNumberOfYear( '12/09/2014', 3 )</code>	Retourneert 196, het nummer van de dag geteld vanaf 1 maart.

### Voorbeeld 1 - Januari begin van het jaar (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een eenvoudige gegevensset met een lijst met datums, die wordt geladen in een tabel met de naam `calendar`. De standaard `DateFormat`-systeemvariabele `mm/dd/yyyy` wordt gebruikt.
- Een voorafgaande lading maakt een extra veld met de naam `daynryear` met behulp van de `DayNumberOfYear()` functie.

Naast de datum worden er geen extra parameters aan de functie toegevoegd.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
calendar:
```

```
Load
```

```
    date,  
    DayNumberOfYear(date) as daynryear  
;
```

```
Load
```

```
date
```

```
Inline
```

```
[
```

```
date
```

```
01/01/2022
```

```
01/10/2022
```

```
01/31/2022
```

```
02/01/2022
```

```
02/10/2022
```

```
06/30/2022
```

```
07/26/2022
```

```
10/31/2022
```



```
11/01/2022  
12/31/2022  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- daynryear

Resultatentabel

date	daynryear
01/01/2022	1
01/10/2022	10
01/31/2022	31
02/01/2022	32
02/10/2022	41
06/30/2022	182
07/26/2022	208
10/31/2022	305
11/01/2022	306
12/31/2022	366

De eerste dag van het jaar is 1 januari omdat er geen tweede argument is doorgegeven in de `DayNumberOfYear()`-functie.

1 Januari is de eerste dag van het kwartaal en 1 februari is de 32e dag van het jaar. 30 juni is de 182e dag en 31 december is de 366e en laatste dag van het jaar.

### Voorbeeld 2 - November begin van het jaar (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele `mm/dd/yyyy` wordt gebruikt
- Een `start_month`-argument dat begint op 1 november. Dit stelt het financiële jaar in op 1 november.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Calendar:
Load
    date,
    DayNumberOfYear(date,11) as daynryear
;

Load
date
Inline
[
date
01/01/2022
01/10/2022
01/31/2022
02/01/2022
02/10/2022
06/30/2022
07/26/2022
10/31/2022
11/01/2022
12/31/2022
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- daynryear

Resultatentabel

date	daynryear
01/01/2022	62
01/10/2022	71
01/31/2022	92
02/01/2022	93
02/10/2022	102
06/30/2022	243
07/26/2022	269
10/31/2022	366
11/01/2022	1
12/31/2022	61

De eerste dag van het jaar is 1 november omdat het tweede argument dat is doorgegeven in de `DayNumberOfYear()`-functie 11 is.

1 Januari is de eerste dag van het kwartaal en 1 februari is de 32e dag van het jaar. 30 juni is de 182e dag en 31 december is de 366e en laatste dag van het jaar.

### Voorbeeld 3 - Januari begin van het jaar (diagram)

Load-script en diagramuitdrukking

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele `mm/dd/yyyy` wordt gebruikt.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De waarde van de dag van het kwartaal wordt berekend via een meting in een diagramobject.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Calendar:  
Load  
date  
Inline  
[  
date  
01/01/2022  
01/10/2022  
01/31/2022  
02/01/2022  
02/10/2022  
06/30/2022  
07/26/2022  
10/31/2022  
11/01/2022  
12/31/2022  
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting:

```
=daynumberofyear(date)
```

Resultatentabel

date	=daynumberofyear(date)
01/01/2022	1
01/10/2022	10
01/31/2022	31
02/01/2022	32
02/10/2022	41
06/30/2022	182
07/26/2022	208
10/31/2022	305
11/01/2022	306
12/31/2022	366

De eerste dag van het jaar is 1 januari omdat er geen tweede argument is doorgegeven in de `DayNumberOfYear()`-functie.

1 Januari is de eerste dag van het jaar en 1 februari is de 32e dag van het jaar. 30 juni is de 182e dag en 31 december is de 366e en laatste dag van het jaar.

### Voorbeeld 4 - November begin van het jaar (diagram)

Load-script en diagramuitdrukking

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele `mm/dd/yyyy` wordt gebruikt.
- Het financiële jaar loopt van 1 november tot en met 31 oktober.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De waarde van de dag van het jaar wordt berekend via een meting in een diagramobject.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
Calendar:
Load
date
Inline
[
```

```
date
01/01/2022
01/10/2022
01/31/2022
02/01/2022
02/10/2022
06/30/2022
07/26/2022
10/31/2022
11/01/2022
12/31/2022
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Maak de volgende meting:

```
=daynumberofyear(date)
```

Resultatentabel

date	=daynumberofyear(date,11)
01/01/2022	62
01/10/2022	71
01/31/2022	92
02/01/2022	93
02/10/2022	102
06/30/2022	243
07/26/2022	269
10/31/2022	366
11/01/2022	1
12/31/2022	61

De eerste dag van het jaar is 1 november omdat het tweede argument dat is doorgegeven in de `DayNumberOfYear()`-functie 11 is.

Het financiële jaar loopt tussen november en oktober. Dit wordt weergegeven in de tabel met resultaten. Hierbij is 1 november de eerste dag van het jaar en 31 oktober is de 366e en laatste dag van het jaar.

### daystart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel met de eerste milliseconde van de dag in het argument **time**. De standaarduitvoernotatie is de **TimestampFormat** die is ingesteld in het script.

### Syntaxis:

```
DayStart(time[, [period_no[, day_start]])
```

Retourgegevenstypen: dual

#### Argumenten

Argument	Beschrijving
<b>time</b>	De te evalueren tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de dag aangeeft die <b>time</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande dagen aan en positieve waarden geven volgende dagen aan.
<b>day_start</b>	Als u wilt opgeven dat dagen niet om middernacht beginnen, geeft u een verschuiving aan als een fractie van een dag in <b>day_start</b> . Bijvoorbeeld 0.125 om 03.00 uur aan te geven. Met andere woorden: om de offset te maken, verdeelt u de begintijd met 24 uur. Bijvoorbeeld: om een dag te laten beginnen om 7.00 uur, gebruikt u de fractie 7/24.

### Wanneer gebruiken

De functie `daystart()` wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van de dag wordt gebruikt dat al is geweest. Het zou bijvoorbeeld kunnen worden gebruikt om het totale salaris te berekenen dat werknemers op een dag tot dusver hebben verdiend.

Deze voorbeelden gebruiken de tijdstempelopmaak 'M/D/YYYY h:mm:ss[.fff] TT'. De tijdstempelopmaak wordt opgegeven in de opdracht `SET Timestamp` boven aan uw load-script voor gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>daystart('01/25/2013 4:45:00 PM')</code>	Retourneert 1/25/2013 12:00:00 AM.
<code>daystart('1/25/2013 4:45:00 PM', -1)</code>	Retourneert 1/24/2013 12:00:00 AM.
<code>daystart('1/25/2013 16:45:00', 0, 0.5 )</code>	Retourneert 1/25/2013 12:00:00 PM.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Eenvoudig voorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een eenvoudige gegevensset met een lijst met datums, die wordt geladen in een tabel met de naam `Calendar`.
- De standaard `TimestampFormat`-systeemvariabele (`M/D/YYYY h:mm:ss[.fff] TT`) wordt gebruikt.
- Een voorafgaande lading die een extra veld maakt met de naam `SOD_timestamp` met behulp van de `daystart()`functie.

Naast de datum worden er geen extra parameters aan de functie toegevoegd.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Calendar:
```

```
  Load
    date,
    daystart(date) as SOD_timestamp
  ;
```

```
Load
```

```
date
```

```
Inline
```

```
[
```

```
date
```

```
03/11/2022 1:47:15 AM
```

```
03/12/2022 4:34:58 AM
```

```
03/13/2022 5:15:55 AM
```

```
03/14/2022 9:25:14 AM
```

```
03/15/2022 10:06:54 AM
```

```
03/16/2022 10:44:42 AM
```

```
03/17/2022 11:33:30 AM
```

```
03/18/2022 12:58:14 PM
```

```
03/19/2022 4:23:12 PM
```

```
03/20/2022 6:42:15 PM
```

```
03/21/2022 7:41:16 PM
```

];

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- SOD\_timestamp

Resultatentabel

date	SOD_timestamp
03/11/2022 1:47:15 AM	3/11/2022 12:00:00 AM
03/12/2022 4:34:58 AM	3/12/2022 12:00:00 AM
03/13/2022 5:15:55 AM	3/13/2022 12:00:00 AM
03/14/2022 9:25:14 AM	3/14/2022 12:00:00 AM
03/15/2022 10:06:54 AM	3/15/2022 12:00:00 AM
03/16/2022 10:44:42 AM	3/16/2022 12:00:00 AM
03/17/2022 11:33:30 AM	3/17/2022 12:00:00 AM
03/18/2022 12:58:14 PM	3/18/2022 12:00:00 AM
03/19/2022 4:23:12 PM	3/19/2022 12:00:00 AM
03/20/2022 6:42:15 PM	3/20/2022 12:00:00 AM
03/21/2022 7:41:16 PM	3/21/2022 12:00:00 AM

Zoals in de bovenstaande tabel te zien is, wordt de eind van de dag-tijdstempel gegenereerd voor iedere datum in uw gegevensset. Het tijdstempel is de indeling van de systeemvariabele `TimestampFormat` `M/D/YYYY h:mm:ss[.fff] TT`.

### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met parkeerboetes, die wordt geladen in de tabel `Fines`. De gegevensset bevat de volgende velden:
  - id
  - due\_date



- number\_plate
- amount
- Een voorafgaande lading met de `daystart()`-functie en met de drie parameters: `time`, `period_no` en `day_start`. Deze voorafgaande lading maakt de volgende twee nieuwe datumvelden aan:
  - Een `early_repayment_period`-datumveld dat zeven dagen vóór de vervaldatum van de betaling begint.
  - Een `late_penalty_period`-datumveld dat veertien dagen na de vervaldatum van de betaling begint.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Fines:
```

```
    Load
        *,
        daystart(due_date,-7) as early_repayment_period,
        daystart(due_date,14) as late_penalty_period
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id, due_date, number_plate, amount
1,02/11/2022, 573RJG,50.00
2,03/25/2022, SC41854,50.00
3,04/14/2022, 8EHZ378,50.00
4,06/28/2022, 8HSS198,50.00
5,08/15/2022, 1221665,50.00
6,11/16/2022, EAK473,50.00
7,01/17/2023, KD6822,50.00
8,03/22/2023, 1GGLB,50.00
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- due\_date
- early\_repayment\_period
- late\_penalty\_period

Resultatentabel

due_date	early_repayment_period	late_penalty_period
02/11/2022 9:25:14 AM	2/4/2022 12:00:00 AM	2/25/2022 12:00:00 AM
03/15/2022 10:06:54 AM	3/18/2022 12:00:00 AM	4/8/2022 12:00:00 AM
04/14/2022 10:44:42 AM	4/7/2022 12:00:00 AM	4/28/2022 12:00:00 AM

due_date	early_repayment_period	late_penalty_period
06/28/2022 11:33:30 AM	6/21/2022 12:00:00 AM	7/12/2022 12:00:00 AM
08/15/2022 12:58:14 PM	8/8/2022 12:00:00 AM	8/29/2022 12:00:00 AM
11/16/2022 4:23:12 PM	11/9/2022 12:00:00 AM	11/30/2022 12:00:00 AM
01/17/2023 6:42:15 PM	1/10/2023 12:00:00 AM	1/31/2023 12:00:00 AM
03/22/2023 7:41:16 PM	3/15/2023 12:00:00 AM	4/5/2023 12:00:00 AM

De waarden van de nieuwe velden hebben de `TimestampFormat` `M/DD/YYYY tt`. Omdat de `daystart()`-functie is gebruikt, zijn de tijdstempelwaarden allemaal de eerste milliseconde van de dag.

De vroege betaalperiode-waarden zijn zeven dagen vóór de vervaldatum, omdat het tweede argument dat in de `daystart()`-functie wordt doorgegeven negatief is.

De late betaalperiode-waarden zijn veertien dagen na de vervaldatum, omdat het tweede argument dat in de `daystart()`-functie wordt doorgegeven positief is.

### Voorbeeld 3 - day\_start

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het vorige voorbeeld.
- Dezelfde voorafgaande lading als het vorige voorbeeld.

In dit voorbeeld laten we de werkdag elke dag beginnen en eindigen om 7:00 AM.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Fines:
```

```
  Load
```

```
    *
```

```
    daystart(due_date,-7,7/24) as early_repayment_period,
```

```
    daystart(due_date,14, 7/24) as late_penalty_period
```

```
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id, due_date, number_plate, amount
```

```
1,02/11/2022, 573RJG,50.00
```

```
2,03/25/2022, SC41854,50.00
```

```
3,04/14/2022, 8EHZ378,50.00
4,06/28/2022, 8HSS198,50.00
5,08/15/2022, 1221665,50.00
6,11/16/2022, EAK473,50.00
7,01/17/2023, KD6822,50.00
8,03/22/2023, 1GGLB,50.00
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- due\_date
- early\_repayment\_period
- late\_penalty\_period

Resultatentabel

due_date	early_repayment_period	late_penalty_period
02/11/2022	2/3/2022 7:00:00 AM	2/24/2022 7:00:00 AM
03/25/2022	3/17/2022 7:00:00 AM	4/7/2022 7:00:00 AM
04/14/2022	4/6/2022 7:00:00 AM	4/27/2022 7:00:00 AM
06/28/2022	6/20/2022 7:00:00 AM	7/11/2022 7:00:00 AM
08/15/2022	8/7/2022 7:00:00 AM	8/28/2022 7:00:00 AM
11/16/2022	11/8/2022 7:00:00 AM	11/29/2022 7:00:00 AM
01/17/2023	1/9/2023 7:00:00 AM	1/30/2023 7:00:00 AM
03/22/2023	3/14/2023 7:00:00 AM	4/4/2023 7:00:00 AM

De datums hebben nu een tijdstempel van 7:00 AM, omdat de waarde van het day\_start-argument dat in de daystart()-functie is doorgegeven, 7/24 was. Dit stelt het begin van de dag in op 7:00 AM.

Omdat het due\_date-veld geen tijdstempel heeft, wordt dit behandeld als 12:00 AM, waardoor de dag nog deel uitmaakt van de vorige dag omdat de dagen beginnen en eindigen om 7:00 AM. Daarom begint de vroege-terugbetalingsperiode voor een boete die op 11 februari betaald moet zijn, op 3 februari om 7:00 AM.

### Voorbeeld 4 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Dit voorbeeld gebruikt dezelfde gegevensset en hetzelfde scenario als het vorige voorbeeld.

Maar alleen de originele Fines-tabel wordt in de applicatie geladen waarbij de twee extra vervaldatumwaarden worden berekend in een diagramobject.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Fines:
```

```
    Load
```

```
*
```

```
Inline
```

```
[
```

```
id, due_date, numer_plate, amount
```

```
1,02/11/2022 9:25:14 AM, 573RJG,50.00
```

```
2,03/25/2022 10:06:54 AM, SC41854,50.00
```

```
3,04/14/2022 10:44:42 AM, 8EHZ378,50.00
```

```
4,06/28/2022 11:33:30 AM, 8HSS198,50.00
```

```
5,08/15/2022 12:58:14 PM, 1221665,50.00
```

```
6,11/16/2022 4:23:12 PM, EAK473,50.00
```

```
7,01/17/2023 6:42:15 PM, KD6822,50.00
```

```
8,03/22/2023 7:41:16 PM, 1GGLB,50.00
```

```
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: due\_date.
2. Om het early\_repayment\_period-veld te maken, maakt u de volgende meting: =daystart(due\_date,-7,7/24)
3. Om het late\_penalty\_period-veld te maken, maakt u de volgende meting: =daystart(due\_date,14,7/24)

Resultatentabel

due_date	=daystart(due_date,-7,7/24)	=daystart(due_date,14,7/24)
02/11/2022 9:25:14 AM	2/4/2022 7:00:00 AM	2/25/2022 7:00:00 AM
03/15/2022 10:06:54 AM	3/18/2022 7:00:00 AM	4/8/2022 7:00:00 AM
04/14/2022 10:44:42 AM	4/7/2022 7:00:00 AM	4/28/2022 7:00:00 AM
06/28/2022 11:33:30 AM	6/21/2022 7:00:00 AM	7/12/2022 7:00:00 AM
08/15/2022 12:58:14 PM	8/8/2022 7:00:00 AM	8/29/2022 7:00:00 AM
11/16/2022 4:23:12 PM	11/9/2022 7:00:00 AM	11/30/2022 7:00:00 AM
01/17/2023 6:42:15 PM	1/10/2023 7:00:00 AM	1/31/2023 7:00:00 AM
03/22/2023 7:41:16 PM	3/15/2023 7:00:00 AM	4/5/2023 7:00:00 AM

De waarden van de nieuwe velden hebben de `TimestampFormat M/D/YYYY h:mm:ss[.fff] TT`. Omdat de `daystart()`-functie is gebruikt, komen de tijdstempelwaarden allemaal overeen met de eerste milliseconde van de dag.

De vroege betaalperiode-waarden zijn zeven dagen vóór de vervaldatum, omdat het tweede argument dat in de `daystart()`-functie is doorgegeven, negatief was.

De late betaalperiode-waarden zijn veertien dagen na de vervaldatum, omdat het tweede argument dat in de `daystart()`-functie is doorgegeven, positief was.

De datums hebben een tijdstempel van 7:00 AM, omdat de waarde van het derde argument dat in de `daystart()`-functie, `day_start`, is doorgegeven, `7/24` was.

### firstworkdate

De functie **firstworkdate** retourneert de laatste begindatum waarbij **no\_of\_workdays** (maandag-vrijdag) kan worden gehaald als niet later wordt gestopt dan **end\_date**, waarbij rekening wordt gehouden met een eventueel opgegeven vakantiedagen. **end\_date** en **holiday** moeten geldige datums of tijdstempels zijn.

#### Syntaxis:

```
firstworkdate(end_date, no_of_workdays {, holiday} )
```

**Retourgegevenstypen:** geheel getal

#### Argumenten:

##### Argumenten

Argument	Beschrijving
<b>end_date</b>	De tijdstempel van de te evalueren einddatum.
<b>no_of_workdays</b>	Het aantal te bereiken werkdagen.
<b>holiday</b>	Vakantieperioden die moeten worden uitgesloten van werkdagen. Een vakantie wordt aangegeven als een tekenreeks met constante datum. U kunt meerdere vakantiedatums opgeven, gescheiden door komma's.  <b>Voorbeeld:</b> '12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014'

Voorbeelden en resultaten:

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

### Scriptvoorbeelden

Voorbeeld	Resultaat
firstworkdate ('29/12/2014', 9)	Retourneert '17/12/2014.
firstworkdate ('29/12/2014', 9, '25/12/2014', '26/12/2014')	Retourneert 15/12/2014 omdat rekening is gehouden met een vakantieperiode van twee dagen.

#### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
ProjectTable:
LOAD *, recno() as InvID, INLINE [
EndDate
28/03/2015
10/12/2015
5/2/2016
31/3/2016
19/5/2016
15/9/2016
] ;
NrDays:
Load *,
FirstWorkDate(EndDate,120) As StartDate
Resident ProjectTable;
Drop table ProjectTable;
```

De resulterende tabel bevat de geretourneerde waarden van FirstWorkDate voor elk van de records in de tabel.

### Resultatentabel

InvID	EndDate	StartDate
1	28/03/2015	13/10/2014
2	10/12/2015	26/06/2015
3	5/2/2016	24/08/2015
4	31/3/2016	16/10/2015
5	19/5/2016	04/12/2015
6	15/9/2016	01/04/2016

## GMT

Deze functie retourneert de huidige Greenwich Mean Time, zoals afgeleid uit de landinstellingen. Deze functie retourneert waarden in de notatie van de `TimestampFormat`-systeemvariabele.

Wanneer de app opnieuw wordt geladen, wordt iedere load-scripttabel, variabele of diagramobject die/dat gebruikmaakt van de GMT-functie aangepast aan de actuele Greenwich Mean Time zoals afgeleid uit de systeemklok.

### Syntaxis:

```
GMT ( )
```

### Retourgegevenstypen: dual

Deze voorbeelden gebruiken de tijdstempelopmaak M/D/YYYY h:mm:ss[.fff] TT. De datumopmaak wordt opgegeven in de opdracht SET TimestampFormat boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Voorbeelden van functies

Voorbeeld	Resultaat
GMT()	3/28/2022 2:47:36 PM

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/YYYY. De datumopmaak wordt opgegeven in de opdracht SET DateFormat in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Variabele (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad. In dit voorbeeld voor de actuele Greenwich Mean Time met de GMT-functie ingesteld als een variabele in het load-script.

#### Load-script

```
LET vGMT = GMT();
```

### Resultaten

Laad de gegevens en maak een werkblad. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.

Voeg deze meting toe aan het tekstvak:

```
=VGMT
```

Het tekstvak moet een regel tekst met een datum en een tijd bevatten, zoals hieronder:

```
3/28/2022 2:47:36 PM
```

### Voorbeeld 2 - November begin van het jaar (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met bibliotheekboeken die al hadden moeten zijn ingeleverd, die wordt geladen in de tabel overdue. De standaard dateFormat-systeemvariabele mm/dd/yyyy wordt gebruikt.
- Er wordt een nieuw veld gemaakt, days\_overdue, waarmee wordt berekend hoeveel dagen elk boek te laat is.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Overdue:
```

```
Load
    *,
    Floor(GMT()-due_date) as days_overdue
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
cust_id,book_id,due_date
1,4,01/01/2021,
2,24,01/10/2021,
6,173,01/31/2021,
31,281,02/01/2021,
86,265,02/10/2021,
52,465,06/30/2021,
26,537,07/26/2021,
92,275,10/31/2021,
27,455,11/01/2021,
```



```
27,46,12/31/2021  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- due\_date
- book\_id
- days\_overdue

Resultatentabel

due_date	book_id	days_overdue
01/01/2021	4	455
01/10/2021	24	446
01/31/2021	173	425
02/01/2021	281	424
02/10/2021	265	415
06/30/2021	465	275
07/26/2021	537	249
10/31/2021	275	152
11/01/2021	455	151
12/31/2021	46	91

De waarden in het veld `days_overdue` worden berekend door het verschil te vinden tussen de actuele Greenwich Mean Time, met behulp van de `GMT()`-functie, en de oorspronkelijke vervaldatum. Om alleen het aantal dagen te berekenen, worden de resultaten afgerond op het dichtstbijzijnde hele getal met behulp van de `Floor()`-functie.

### Voorbeeld 3 - Diagramobject (diagram)

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad. Het load-script bevat dezelfde gegevensset als het vorige voorbeeld. De standaard `dateFormat`-systeemvariabele `mm/dd/yyyy` wordt gebruikt.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De waarde van het aantal te laat-dagen wordt berekend via een meting in een diagramobject.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

Overdue:

Load

\*

Inline

[

cust\_id,book\_id,due\_date

1,4,01/01/2021,

2,24,01/10/2021,

6,173,01/31/2021,

31,281,02/01/2021,

86,265,02/10/2021,

52,465,06/30/2021,

26,537,07/26/2021,

92,275,10/31/2021,

27,455,11/01/2021,

27,46,12/31/2021

];

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- due\_date
- book\_id

Maak de volgende meting:

=Floor(GMT() - due\_date)

Resultatentabel

due_date	book_id	=Floor(GMT()-due_date)
01/01/2021	4	455
01/10/2021	24	446
01/31/2021	173	425
02/01/2021	281	424
02/10/2021	265	415
06/30/2021	465	275
07/26/2021	537	249
10/31/2021	275	152
11/01/2021	455	151
12/31/2021	46	91

De waarden in het veld `days_overdue` worden berekend door het verschil te vinden tussen de actuele Greenwich Mean Time, met behulp van de `GMT()`-functie, en de oorspronkelijke vervaldatum. Om alleen het aantal dagen te berekenen, worden de resultaten afgerond op het dichtstbijzijnde hele getal met behulp van de `Floor()`-functie.

### hour

Deze functie retourneert een geheel getal dat het uur representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

#### Syntaxis:

```
hour (expression)
```

**Retourgegevenstypen:** geheel getal

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/YYYY. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>hour('09:14:36')</code>	De opgegeven tekenreeks wordt impliciet geconverteerd in een tijdstempel omdat het overeenkomt met de indeling van het tijdstempel dat is gedefinieerd in de variabele <code>TimestampFormat</code> . De expressie retourneert 9.
<code>hour('0.5555')</code>	De expressie retourneert 13 (omdat $0,5555 = 13:19:55$ is)

### Voorbeeld 1 - Variabele (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met transactie per tijdstempel
- De standaard `Timestamp`-systeemvariabele (`M/D/YYYY h:mm:ss[.fff] TT`).

Maak een veld, `hour` dat berekent wanneer de aankopen hebben plaatsgevonden.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
  Load
    *,
    hour(date) as hour
  ;
Load
*
Inline
[
id,date,amount
9497,'2022-01-05 19:04:57',47.25,
9498,'2022-01-03 14:21:53',51.75,
9499,'2022-01-03 05:40:49',73.53,
9500,'2022-01-04 18:49:38',15.35,
9501,'2022-01-01 22:10:22',31.43,
9502,'2022-01-05 19:34:46',13.24,
9503,'2022-01-04 22:58:34',74.34,
9504,'2022-01-06 11:29:38',50.00,
9505,'2022-01-02 08:35:54',36.34,
9506,'2022-01-06 08:49:09',74.23
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `hour`

Resultatentabel

<code>date</code>	<code>hour</code>
2022-01-01 22:10:22	22
2022-01-02 08:35:54	8
2022-01-03 05:40:49	5
2022-01-03 14:21:53	14
2022-01-04 18:49:38	18

date	hour
2022-01-04 22:58:34	22
2022-01-05 19:04:57	19
2022-01-05 19:34:46	19
2022-01-06 08:49:09	8
2022-01-06 11:29:38	11

De waarden in het uurveld zijn gemaakt met de `hour()`-functie en geven de datum door als de uitdrukking in het voorafgaande load-instructie.

### Voorbeeld 2 - Diagramobject (diagram)

Load-script en diagramuitdrukking

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De standaard `Timestamp`-systeemvariabele (`M/D/YYYY h:mm:ss[.fff] TT`).

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De `hour`-waarden worden berekend via een meting in een diagramobject.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
9497,'2022-01-05 19:04:57',47.25,
```

```
9498,'2022-01-03 14:21:53',51.75,
```

```
9499,'2022-01-03 05:40:49',73.53,
```

```
9500,'2022-01-04 18:49:38',15.35,
```

```
9501,'2022-01-01 22:10:22',31.43,
```

```
9502,'2022-01-05 19:34:46',13.24,
```

```
9503,'2022-01-04 22:58:34',74.34,
```

```
9504,'2022-01-06 11:29:38',50.00,
```

```
9505,'2022-01-02 08:35:54',36.34,
```

```
9506,'2022-01-06 08:49:09',74.23
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Maak de volgende meting om het uur te berekenen:

=hour(date)

Resultatentabel

due_date	=uur(datum)
2022-01-01 22:10:22	22
2022-01-02 08:35:54	8
2022-01-03 05:40:49	5
2022-01-03 14:21:53	14
2022-01-04 18:49:38	18
2022-01-04 22:58:34	22
2022-01-05 19:04:57	19
2022-01-05 19:34:46	19
2022-01-06 08:49:09	8
2022-01-06 11:29:38	11

De waarden voor hour worden gemaakt met behulp van de hour()-functie en geven de datum door als de uitdrukking in een meting voor het diagramobject.

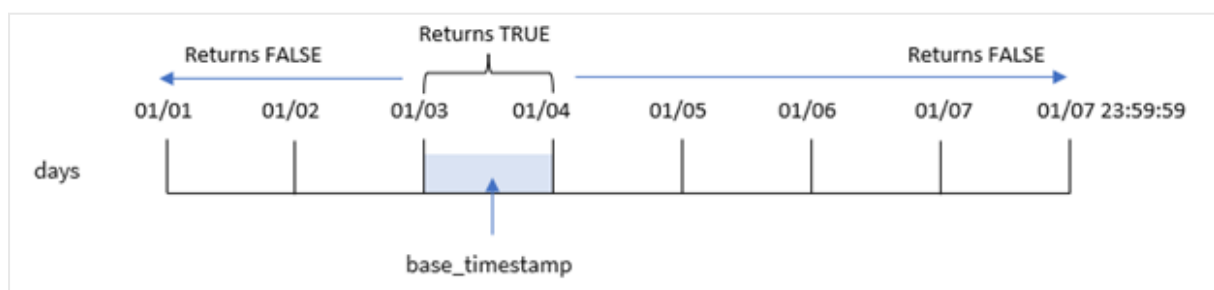
### inday

Deze functie retourneert True als **timestamp** binnen de dag valt die **base\_timestamp** bevat.

#### Syntaxis:

```
InDay (timestamp, base_timestamp, period_no[, day_start])
```

Diagram van de inday-functie



De functie `inday()` gebruikt het argument `base_timestamp` om te identificeren onder welke dag het tijdstempel valt. De begintijd van de dag is standaard middernacht, maar u kunt de begintijd van de dag wijzigen door het argument `day_start` van de functie `inday()` te gebruiken. Zodra deze dag is gedefinieerd, retourneert de functie de booleaanse waarden bij het vergelijken van de opgegeven tijdstempelwaarden van die dag.

### Wanneer gebruiken

De functie `inday()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if expression`. Hierdoor wordt een aggregatie of berekening geretourneerd, afhankelijk van het feit of een geëvalueerde datum plaatsvond op de dag van de desbetreffende tijdstempel.

De functie `inday()` bijvoorbeeld kan worden gebruikt om alle materieel te identificeren die op een bepaalde dag is geproduceerd.

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

#### Argumenten

Argument	Beschrijving
<code>timestamp</code>	De datum en tijd die u wilt vergelijken met <code>base_timestamp</code> .
<code>base_timestamp</code>	Datum en tijd op basis waarvan de tijdstempel wordt geëvalueerd.
<code>period_no</code>	De dag kan worden verschoven met <code>period_no</code> . <code>period_no</code> is een geheel getal, waarbij de waarde 0 de dag aangeeft die <code>base_timestamp</code> bevat. Negatieve waarden in <code>period_no</code> geven voorafgaande dagen aan en positieve waarden geven volgende dagen aan.
<code>day_start</code>	Als u wilt werken met dagen die niet om middernacht beginnen, kunt u een verschuiving als een fractie van een dag opgeven in <code>day_start</code> , bijvoorbeeld 0,125 voor 3 uur in de ochtend.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/YYYY. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze

landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>inday ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', 0)</code>	Retourneert True
<code>inday ('01/12/2006 12:23:00 PM', '01/13/2006 12:00:00 AM', 0)</code>	Retourneert False
<code>inday ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', -1)</code>	Retourneert False
<code>inday ('01/11/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', -1)</code>	Retourneert True
<code>inday ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', 0, 0.5)</code>	Retourneert False
<code>inday ('01/12/2006 11:23:00 AM', '01/12/2006 12:00:00 AM', 0, 0.5)</code>	Retourneert True

### Voorbeeld 1 - Load-instructie (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met transacties per tijdstempel die wordt geladen in een tabel met de naam `Transactions`.
- Een datumveld dat wordt weergegeven in de `Timestamp`-systeemvariabele (`M/D/YYYY h:mm:ss [.fff] TT`)-opmaak.
- Een voorafgaande lading met de functie `inday()` die is ingesteld als het veld `in_day`.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
  Load
    *,
    inday(date,'01/05/2022 12:00:00 AM', 0) as in_day
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
9497, '01/01/2022 7:34:46 PM', 13.24
```

```
9498, '01/01/2022 10:10:22 PM', 31.43
```

```
9499, '01/02/2022 8:35:54 AM', 36.34
```

```
9500, '01/03/2022 2:21:53 PM', 51.75
```



```
9501, '01/04/2022 6:49:38 PM', 15.35
9502, '01/04/2022 10:58:34 PM', 74.34
9503, '01/05/2022 5:40:49 AM', 73.53
9504, '01/05/2022 11:29:38 AM', 50.00
9505, '01/05/2022 7:04:57 PM', 47.25
9506, '01/06/2022 8:49:09 AM', 74.23
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_day

Resultatentabel

date	in_day
01/01/2022 7:34:46 PM	0
01/01/2022 10:10:22 PM	0
01/02/2022 8:35:54 AM	0
01/03/2022 2:21:53 PM	0
01/04/2022 6:49:38 PM	0
01/04/2022 10:58:34 PM	0
01/05/2022 5:40:49 AM	-1
01/05/2022 11:29:38 AM	-1
01/05/2022 7:04:57 PM	-1
01/06/2022 8:49:09 AM	0

Het veld `in_day` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `in_day()` en geeft het datumveld door, een vastgelegde tijdstempel voor 5 januari en een `period_no` of 0 als de argumenten van de functie.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

De load-instructie gebruikt dezelfde gegevensset en hetzelfde scenario als die in het eerste voorbeeld.

Maar in dit voorbeeld is de taak om te berekenen of de transactiedatum twee dagen vóór 5 januari plaatsvond.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
    Load
        *,
        inday(date, '01/05/2022 12:00:00 AM', -2) as in_day
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
9497, '01/01/2022 7:34:46 PM', 13.24
9498, '01/01/2022 10:10:22 PM', 31.43
9499, '01/02/2022 8:35:54 AM', 36.34
9500, '01/03/2022 2:21:53 PM', 51.75
9501, '01/04/2022 6:49:38 PM', 15.35
9502, '01/04/2022 10:58:34 PM', 74.34
9503, '01/05/2022 5:40:49 AM', 73.53
9504, '01/05/2022 11:29:38 AM', 50.00
9505, '01/05/2022 7:04:57 PM', 47.25
9506, '01/06/2022 8:49:09 AM', 74.23
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_day

Resultatentabel

date	in_day
01/01/2022 7:34:46 PM	0
01/01/2022 10:10:22 PM	0
01/02/2022 8:35:54 AM	0
01/03/2022 2:21:53 PM	-1
01/04/2022 6:49:38 PM	0
01/04/2022 10:58:34 PM	0
01/05/2022 5:40:49 AM	0
01/05/2022 11:29:38 AM	0
01/05/2022 7:04:57 PM	0
01/06/2022 8:49:09 AM	0

In dit voorbeeld bepaalt de functie of elk van de transactiedatums op 3 januari plaatsvonden omdat een `period_no` of `-2` is gebruikt als het `offset`-argument in de functie `in_day()`. Dit kan worden geverifieerd in de uitvoertabel waar één transactie een booleaans resultaat van `WAAR` retourneert.

### Voorbeeld 3 - day\_start-script

Load-script en resultaten

#### Overzicht

De load-instructie gebruikt dezelfde gegevensset en hetzelfde scenario als die in de voorgaande voorbeelden.

Maar in dit voorbeeld is het bedrijfsbeleid dat de werkdag begint en eindigt om 07.00 uur.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';

Transactions:
  Load
    *,
    in_day(date,'01/05/2022 12:00:00 AM', 0, 7/24) as in_day
  ;

Load
*
Inline
[
id,date,amount
9497,'01/01/2022 7:34:46 PM',13.24
9498,'01/01/2022 10:10:22 PM',31.43
9499,'01/02/2022 8:35:54 AM',36.34
9500,'01/03/2022 2:21:53 PM',51.75
9501,'01/04/2022 6:49:38 PM',15.35
9502,'01/04/2022 10:58:34 PM',74.34
9503,'01/05/2022 5:40:49 AM',73.53
9504,'01/05/2022 11:29:38 AM',50.00
9505,'01/05/2022 7:04:57 PM',47.25
9506,'01/06/2022 8:49:09 AM',74.23
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_day

Resultatentabel

date	in_day
01/01/2022 7:34:46 PM	0
01/01/2022 10:10:22 PM	0
01/02/2022 8:35:54 AM	0
01/03/2022 2:21:53 PM	0
01/04/2022 6:49:38 PM	-1
01/04/2022 10:58:34 PM	-1
01/05/2022 5:40:49 AM	-1
01/05/2022 11:29:38 AM	0
01/05/2022 7:04:57 PM	0
01/06/2022 8:49:09 AM	0

De functie bepaalt of elk van de transactiedatums plaatsvonden op 4 januari vanaf 07.00 uur en 5 januari vóór 07.00 uur, omdat het argument `start_day` van `7/24`, dat 07.00 uur is, is gebruikt in de functie `inday()`.

Dit kan worden geverifieerd in de uitvoertabel waarbij transacties die na 07.00 uur op 4 januari plaatsvinden een booleaans resultaat van `WAAR` retourneren en transacties die plaatsvinden na 07.00 uur op 5 januari een booleaans resultaat van `ONWAAR`.

### Voorbeeld 4 - diagramobject

Load-script en diagramuitdrukking

#### Overzicht

De load-instructie gebruikt dezelfde gegevensset en hetzelfde scenario als die in het eerste voorbeeld.de vorige voorbeelden.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. U maakt een berekening om te bepalen of een transactie plaatsvindt op 5 januari door een meting te maken in een diagramobject.

#### Load-script

Transactions:

Load

\*

Inline

[

id,date,amount

9497,'01/01/2022 7:34:46 PM',13.24

9498,'01/01/2022 10:10:22 PM',31.43

9499,'01/02/2022 8:35:54 AM',36.34

9500,'01/03/2022 2:21:53 PM',51.75

```
9501, '01/04/2022 6:49:38 PM', 15.35
9502, '01/04/2022 10:58:34 PM', 74.34
9503, '01/05/2022 5:40:49 AM', 73.53
9504, '01/05/2022 11:29:38 AM', 50.00
9505, '01/05/2022 7:04:57 PM', 47.25
9506, '01/06/2022 8:49:09 AM', 74.23
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- date

Maak de volgende meting om te berekenen of een transactie plaatsvindt op 5 januari:

```
=inday(date, '01/05/2022 12:00:00 AM', 0)
```

Resultatentabel

date	inday(date, '01/05/2022 12:00:00 AM', 0)
01/01/2022 7:34:46 PM	0
01/01/2022 10:10:22 PM	0
01/02/2022 8:35:54 AM	0
01/03/2022 2:21:53 PM	0
01/04/2022 6:49:38 PM	0
01/04/2022 10:58:34 PM	0
01/05/2022 5:40:49 AM	-1
01/05/2022 11:29:38 AM	-1
01/05/2022 7:04:57 PM	-1
01/06/2022 8:49:09 AM	0

### Voorbeeld 5 - Scenario

Load-script en resultaten

#### Overzicht

In dit voorbeeld is vastgesteld dat producten die op 5 januari zijn geproduceerd defect zijn vanwege een materieelfout. De eindgebruiker wil graag een diagramobject dat op datum de status toont van geproduceerde producten die 'defect' of 'foutloos' zijn en de kosten van de producten die op 5 januari zijn geproduceerd.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling die wordt geladen in een tabel met de naam 'Producten'.
- De tabel bevat de volgende velden:
  - product ID
  - manufacture time
  - cost price

### Load-script

```
Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
9497,'01/01/2022 7:34:46 PM',13.24
9498,'01/01/2022 10:10:22 PM',31.43
9499,'01/02/2022 8:35:54 AM',36.34
9500,'01/03/2022 2:21:53 PM',51.75
9501,'01/04/2022 6:49:38 PM',15.35
9502,'01/04/2022 10:58:34 PM',74.34
9503,'01/05/2022 5:40:49 AM',73.53
9504,'01/05/2022 11:29:38 AM',50.00
9505,'01/05/2022 7:04:57 PM',47.25
9506,'01/06/2022 8:49:09 AM',74.23
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

```
=dayname(manufacture_date)
```

Maak de volgende metingen:

- =if(only(InDay(manufacture\_date,makedate(2022,01,05),0)),'Defective','Faultless')
- =sum(cost\_price)

Stel de **Getalnotatie** van de meting in op **Geld**.

Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

dayname (manufacture_date)	=if(only(InDay(manufacture_date,makedate (2022,01,05),0)),'Defective','Faultless')	=sum(cost_ price)
01/01/2022	Foutloos	44.67
01/02/2022	Foutloos	36.34
01/03/2022	Foutloos	51.75

dayname (manufacture_date)	=if(only(InDay(manufacture_date,makedate (2022,01,05),0)), 'Defective', 'Faultless')	=sum(cost_ price)
01/04/2022	Foutloos	89.69
01/05/2022	Defect	170.78
01/06/2022	Foutloos	74.23

De functie `inday()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat op 5 januari is geproduceerd, retourneert de functie `inday()` een booleaanse waarde voor WAAR en markeert de producten als 'Defect'. Voor elk product dat de waarde voor ONWAAR retourneert, en dus niet op die dag is geproduceerd, wordt het product als 'Foutloos' gemarkeerd.

### indaytotime

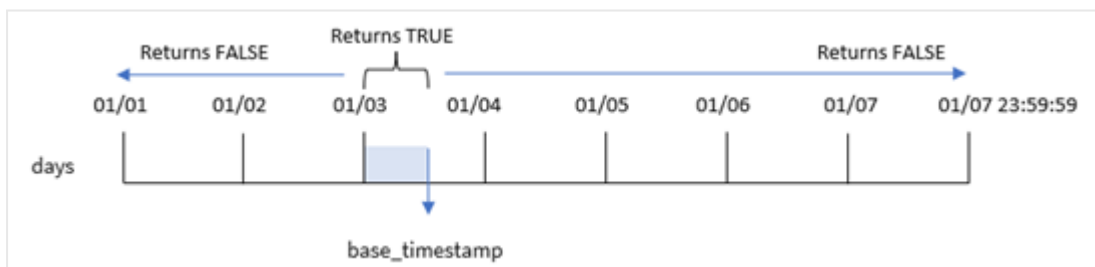
Deze functie retourneert True als **timestamp** binnen het gedeelte van de dag ligt dat **base\_timestamp** bevat tot en met de exacte milliseconde van **base\_timestamp**.

#### Syntaxis:

**InDayToTime** (timestamp, base\_timestamp, period\_no[, day\_start])

De `indaytotime()`-functie retourneert een booleaans resultaat afhankelijk van het feit wanneer een tijdstempelwaarde plaatsvindt tijdens het segment van de dag. De begingrens van dit segment is het begin van de dag, die standaard is ingesteld op middernacht. Het begin van de dag dit kan worden aangepast via het argument `day_start` van de functie `indaytotime()`. De eindgrens van het dagsegment wordt bepaald door een `base_timestamp`-argument van de functie.

Diagram van `indaytotime`-functie.



#### Wanneer gebruiken

De functie `indaytotime()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if` expression. De functie `indaytotime()` retourneert een aggregatie of berekening afhankelijk van het feit of een tijdstempel aanwezig is in het segment van de dag tot en met de tijd van de basistijdstempel.

De functie `indaytotime()` kan bijvoorbeeld worden gebruikt om de som van ticketverkoppen te tonen voor shows die tot nu toe vandaag hebben plaatsgevonden.

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

### Argumenten

Argument	Beschrijving
timestamp	De datum en tijd die u wilt vergelijken met base_timestamp.
base_timestamp	Datum en tijd op basis waarvan de tijdstempel wordt geëvalueerd.
period_no	De dag kan worden verschoven met period_no. period_no is een geheel getal, waarbij de waarde 0 de dag aangeeft die base_timestamp bevat. Negatieve waarden in period_no geven voorafgaande dagen aan en positieve waarden geven volgende dagen aan.
day_start	(Optioneel) Als u wilt werken met dagen die niet om middernacht beginnen, kunt u een verschuiving als een fractie van een dag opgeven in day_start. Bijvoorbeeld gebruik 0.125 voor 3 AM

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/YYYY. De datumopmaak wordt opgegeven in de opdracht SET DateFormat in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeelden van functies

Voorbeeld	Resultaat
indaytotime ('01/12/2006 12:23:00 PM', '01/12/2006 11:59:00 PM', 0)	Retourneert True
indaytotime ('01/12/2006 12:23:00 PM', '01/12/2006 12:00:00 AM', 0)	Retourneert False
indaytotime '01/11/2006 12:23:00 PM', '01/12/2006 11:59:00 PM', -1)	Retourneert True

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:



- Een gegevensset met transacties voor de periode tussen 4 en 5 januari.
- Een datumveld dat wordt weergegeven in de `Timestamp`-systeemvariabele (`M/D/YYYY h:mm:ss [.fff] TT`)-opmaak.
- Een voorafgaande lading met de functie `indaytotime()` dat is ingesteld als `'in_day_to_time'`, het veld dat bepaalt of de transacties vóór 09.00 AM plaatsvinden.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';

Transactions:
  Load
    *,
    indaytotime(date,'01/05/2022 9:00:00 AM',0) as in_day_to_time
  ;

Load
*
Inline
[
id,date,amount
8188,'01/04/2022 3:41:54 AM',25.66
8189,'01/04/2022 4:19:43 AM',87.21
8190,'01/04/2022 4:53:47 AM',53.80
8191,'01/04/2022 8:38:53 AM',69.98
8192,'01/04/2022 10:37:52 AM',57.42
8193,'01/04/2022 1:54:10 PM',45.89
8194,'01/04/2022 5:53:23 PM',82.77
8195,'01/04/2022 8:13:26 PM',36.23
8196,'01/04/2022 10:00:49 PM',76.11
8197,'01/05/2022 7:45:37 AM',82.06
8198,'01/05/2022 8:44:36 AM',17.17
8199,'01/05/2022 11:26:08 AM',40.39
8200,'01/05/2022 6:43:08 PM',37.23
8201,'01/05/2022 10:54:10 PM',88.27
8202,'01/05/2022 11:09:09 PM',95.93
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

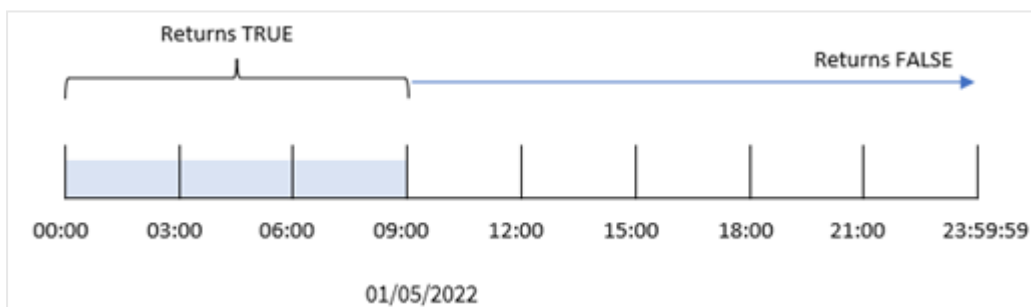
- `date`
- `in_day_to_time`

Resultatentabel

<code>date</code>	<code>in_day_to_time</code>
01/04/2022 3:41:54 AM	0
01/04/2022 4:19:43 AM	0

date	in_day_to_time
01/04/2022 04:53:47 AM	0
01/04/2022 8:38:53 AM	0
01/04/2022 10:37:52 AM	0
01/04/2022 1:54:10 PM	0
01/04/2022 5:53:23 PM	0
01/04/2022 8:13:26 PM	0
01/04/2022 10:00:49 PM	0
01/05/2022 7:45:37 AM	-1
01/05/2022 8:44:36 AM	-1
01/05/2022 11:26:08 AM	0
01/05/2022 6:43:08 PM	0
01/05/2022 10:54:10 PM	0
01/05/2022 11:09:09 PM	0

Voorbeeld 1: diagram van `indaytotime`-functie met limiet van 9:00 AM.



Het `in_day_to_time` field wordt gemaakt in de voorafgaande load-instructie met behulp van de `indaytotime()`-functie en geeft het datumveld door, een vastgelegde tijdstempel voor 09.00 uur op 5 januari en een offset van 0 als de argumenten van de functie. Alle transacties die plaatsvinden tussen middernacht en 09.00 uur op 5 januari retourneren TRUE.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

De load-instructie gebruikt dezelfde gegevensset en hetzelfde scenario als die in het eerste voorbeeld.

Maar in dit voorbeeld berekent u of de transactiedatum één dag vóór 09.00 uur op 5 januari plaatsvond.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';

Transactions:
  Load
    *,
    indaytotime(date,'01/05/2022 9:00:00 AM', -1) as in_day_to_time
  ;

Load
*
Inline
[
id,date,amount
8188,'01/04/2022 3:41:54 AM',25.66
8189,'01/04/2022 4:19:43 AM',87.21
8190,'01/04/2022 4:53:47 AM',53.80
8191,'01/04/2022 8:38:53 AM',69.98
8192,'01/04/2022 10:37:52 AM',57.42
8193,'01/04/2022 1:54:10 PM',45.89
8194,'01/04/2022 5:53:23 PM',82.77
8195,'01/04/2022 8:13:26 PM',36.23
8196,'01/04/2022 10:00:49 PM',76.11
8197,'01/05/2022 7:45:37 AM',82.06
8198,'01/05/2022 8:44:36 AM',17.17
8199,'01/05/2022 11:26:08 AM',40.39
8200,'01/05/2022 6:43:08 PM',37.23
8201,'01/05/2022 10:54:10 PM',88.27
8202,'01/05/2022 11:09:09 PM',95.93
];
```

### Resultaten

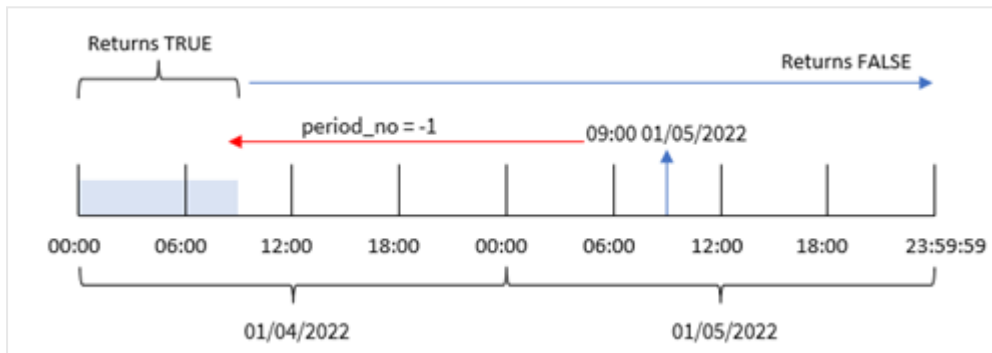
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_day\_to\_time

Resultatentabel	
date	in_day_to_time
01/04/2022 3:41:54 AM	-1
01/04/2022 4:19:43 AM	-1
01/04/2022 04:53:47 AM	-1
01/04/2022 8:38:53 AM	-1
01/04/2022 10:37:52 AM	0
01/04/2022 1:54:10 PM	0
01/04/2022 5:53:23 PM	0

date	in_day_to_time
01/04/2022 8:13:26 PM	0
01/04/2022 10:00:49 PM	0
01/05/2022 7:45:37 AM	0
01/05/2022 8:44:36 AM	0
01/05/2022 11:26:08 AM	0
01/05/2022 6:43:08 PM	0
01/05/2022 10:54:10 PM	0
01/05/2022 11:09:09 PM	0

Voorbeeld 2: diagram van *indaytotime*-functie met transacties vanaf 4 januari.



In dit voorbeeld bepaalt de functie of elk van de transactiedatums vóór 09.00 uur op 4 januari plaatsvonden omdat een offset-argument van -1 is gebruikt in de *indaytotime()*-functie. Dit kan worden geverifieerd in de uitvoertabel waar een transactie een booleaans resultaat van TRUE retourneert.

### Voorbeeld 3 - day\_start-script

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

Maar in dit voorbeeld is het bedrijfsbeleid dat de werkdag begint en eindigt om 8 AM.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

Transactions:

```
Load
```

```
*,
```

```
indaytotime(date,'01/05/2022 9:00:00 AM', 0,8/24) as in_day_to_time
```

```
;
```

```
Load
```

```
*
Inline
[
id,date,amount
8188,'01/04/2022 3:41:54 AM',25.66
8189,'01/04/2022 4:19:43 AM',87.21
8190,'01/04/2022 4:53:47 AM',53.80
8191,'01/04/2022 8:38:53 AM',69.98
8192,'01/04/2022 10:37:52 AM',57.42
8193,'01/04/2022 1:54:10 PM',45.89
8194,'01/04/2022 5:53:23 PM',82.77
8195,'01/04/2022 8:13:26 PM',36.23
8196,'01/04/2022 10:00:49 PM',76.11
8197,'01/05/2022 7:45:37 AM',82.06
8198,'01/05/2022 8:44:36 AM',17.17
8199,'01/05/2022 11:26:08 AM',40.39
8200,'01/05/2022 6:43:08 PM',37.23
8201,'01/05/2022 10:54:10 PM',88.27
8202,'01/05/2022 11:09:09 PM',95.93
];
```

### Resultaten

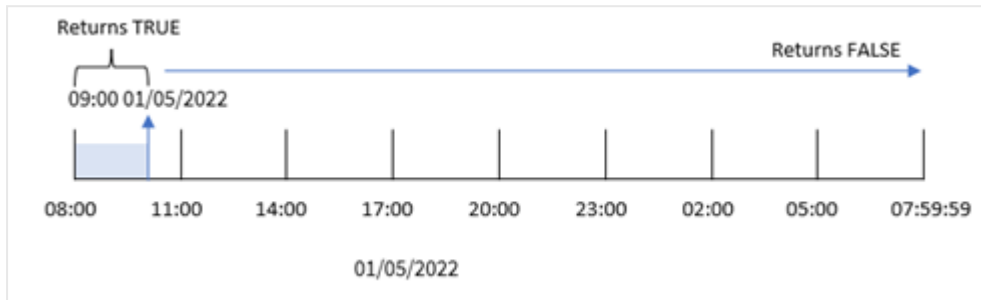
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_day\_to\_time

Resultatentabel	
date	in_day_to_time
01/04/2022 3:41:54 AM	0
01/04/2022 4:19:43 AM	0
01/04/2022 04:53:47 AM	0
01/04/2022 8:38:53 AM	0
01/04/2022 10:37:52 AM	0
01/04/2022 1:54:10 PM	0
01/04/2022 5:53:23 PM	0
01/04/2022 8:13:26 PM	0
01/04/2022 10:00:49 PM	0
01/05/2022 7:45:37 AM	0
01/05/2022 8:44:36 AM	-1
01/05/2022 11:26:08 AM	0
01/05/2022 6:43:08 PM	0

date	in_day_to_time
01/05/2022 10:54:10 PM	0
01/05/2022 11:09:09 PM	0

Voorbeeld 3: diagram van *indaytotime*-functie met transacties vanaf 08.00 uur tot 09.00 uur.



Omdat het argument *start\_day* van 8/24, dat overeenkomt met 8:00 AM, wordt gebruikt in de functie *indaytotime()*, begint en eindigt elke dag om 8:00 AM. Daarom zal de functie *indaytotime()* een booleaanse waarde voor WAAR retourneren voor elke transactie die plaatsvond tussen 8:00 AM en 9:00 AM op 5 januari.

### Voorbeeld 4 - diagramobject

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. U maakt een berekening om te bepalen of een transactie plaatsvindt op 5 januari vóór 09.00 uur door een meting te maken in een diagramobject.

#### Load-script

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,'01/04/2022 3:41:54 AM',25.66
8189,'01/04/2022 4:19:43 AM',87.21
8190,'01/04/2022 4:53:47 AM',53.80
8191,'01/04/2022 8:38:53 AM',69.98
8192,'01/04/2022 10:37:52 AM',57.42
8193,'01/04/2022 1:54:10 PM',45.89
8194,'01/04/2022 5:53:23 PM',82.77
8195,'01/04/2022 8:13:26 PM',36.23
8196,'01/04/2022 10:00:49 PM',76.11
8197,'01/05/2022 7:45:37 AM',82.06
```

```
8198, '01/05/2022 8:44:36 AM', 17.17
8199, '01/05/2022 11:26:08 AM', 40.39
8200, '01/05/2022 6:43:08 PM', 37.23
8201, '01/05/2022 10:54:10 PM', 88.27
8202, '01/05/2022 11:09:09 PM', 95.93
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

date.

Maak de volgende meting om te berekenen of een transactie plaatsvindt op 5 januari vóór 09.00 uur:

```
=indaytotime(date, '01/05/2022 9:00:00 AM', 0)
```

date	Resultatentabel
	inday(date, '01/05/2022 12:00:00 AM', 0)
01/04/2022 3:41:54 AM	0
01/04/2022 4:19:43 AM	0
01/04/2022 04:53:47 AM	0
01/04/2022 8:38:53 AM	0
01/04/2022 10:37:52 AM	0
01/04/2022 1:54:10 PM	0
01/04/2022 5:53:23 PM	0
01/04/2022 8:13:26 PM	0
01/04/2022 10:00:49 PM	0
01/05/2022 7:45:37 AM	-1
01/05/2022 8:44:36 AM	-1
01/05/2022 11:26:08 AM	0
01/05/2022 6:43:08 PM	0
01/05/2022 10:54:10 PM	0
01/05/2022 11:09:09 PM	0

De meting `in_day_to_time` wordt gemaakt in het diagramobject met behulp van de functie `indaytotime()` en geeft het datumveld door, een vastgelegde tijdstempel voor 09.00 uur op 5 januari en een offset van 0 als de argumenten van de functie. Alle transacties die plaatsvinden tussen middernacht en 09.00 uur op 5 januari retourneren TRUE. Dit wordt gevalideerd in de resultatentabel.

### Voorbeeld 5 - Scenario

Load-script en resultaten

#### Overzicht

In dit voorbeeld wordt een gegevensverzameling met ticketverkopten voor een lokale bioscoop geladen in een tabel met de naam Ticket\_Sales. Vandaag is het 3 mei 2022 en het is 11:00 AM.

De gebruiker wil graag een KPI-diagramobject gebruiken om de omzet te tonen van alle shows die tot dusver vandaag hebben plaatsgevonden.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Ticket_Sales:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
sale ID, show time, ticket price
```

```
1,05/01/2022 09:30:00 AM,10.50
```

```
2,05/03/2022 05:30:00 PM,21.00
```

```
3,05/03/2022 09:30:00 AM,10.50
```

```
4,05/03/2022 09:30:00 AM,31.50
```

```
5,05/03/2022 09:30:00 AM,10.50
```

```
6,05/03/2022 12:00:00 PM,42.00
```

```
7,05/03/2022 12:00:00 PM,10.50
```

```
8,05/03/2022 05:30:00 PM,42.00
```

```
9,05/03/2022 08:00:00 PM,31.50
```

```
10,05/04/2022 10:30:00 AM,31.50
```

```
11,05/04/2022 12:00:00 PM,10.50
```

```
12,05/04/2022 05:30:00 PM,10.50
```

```
13,05/05/2022 05:30:00 PM,21.00
```

```
14,05/06/2022 12:00:00 PM,21.00
```

```
15,05/07/2022 09:30:00 AM,42.00
```

```
16,05/07/2022 10:30:00 AM,42.00
```

```
17,05/07/2022 10:30:00 AM,10.50
```

```
18,05/07/2022 05:30:00 PM,10.50
```

```
19,05/08/2022 05:30:00 PM,21.00
```

```
20,05/11/2022 09:30:00 AM,10.50
```

```
];
```

#### Resultaten

Doe het volgende:

1. Maak een KPI-object.
2. Maak met behulp van de functie `indaytotime()` een meting die de som toont van alle ticketverkopten voor shows die tot dusver vandaag hebben plaatsgevonden:



```
=sum(if(indaytotime([show time],'05/03/2022 11:00:00 AM'),0),[ticket price],0))
```

3. Maak een label voor het KPI-object, 'Current Revenue'.
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Het totaal van ticketverkopten tot 11:00 AM op 3 mei 2022 is \$52,50.

De functie `indaytotime()` retourneert een booleaanse waarde bij het vergelijken van de showtijden van elk van de ticketverkopten met de huidige tijd ('05/03/2022 11:00:00 AM'). Voor elke show op 3 mei vóór 11:00 AM retourneert de functie `indaytotime()` een booleaanse waarde WAAR en de ticketprijs wordt meegenomen in het somtotaal.

### inlunarweek

Deze functie bepaalt of **timestamp** binnen de maanweek valt die **base\_date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week. Met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

#### Syntaxis:

```
InLunarWeek (timestamp, base_date, period_no[, first_week_day])
```

**Retourgegevenstypen:** Booleaanse waarde



*In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.*

De `inlunarweek()`-functie bepaalt welke maanweek de `base_date` in valt. Er wordt vervolgens een booleaanse waarde geretourneerd zodra de functie heeft bepaald of iedere tijdstempelwaarde binnen dezelfde maanweek valt als de `base_date`.

*Diagram van de `inlunarweek()`-functie*



### Wanneer gebruiken

De functie `inlunarweek()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een IF-uitdrukking. Dit retourneert een aggregatie of berekening afhankelijk van of de geëvalueerde datum plaatsvond in de betreffende maanweek.

## 5 Script- en diagramfuncties

De functie `inlunarweek()` kan bijvoorbeeld worden gebruikt om alle materieel te identificeren die in een bepaalde maanweek is geproduceerd.

### Argumenten

Argument	Beschrijving
<code>timestamp</code>	De datum die u wilt vergelijken met <code>base_date</code> .
<code>base_date</code>	Datum op basis waarvan de maanweek wordt geëvalueerd.
<code>period_no</code>	De maanweek kan worden verschoven met <code>period_no</code> . <code>period_no</code> is een geheel getal, waarbij de waarde 0 de maanweek aangeeft die <code>base_date</code> bevat. Negatieve waarden in <code>period_no</code> geven voorafgaande maanweken aan en positieve waarden geven volgende maanweken aan.
<code>first_week_day</code>	Een verschuiving kan groter of kleiner zijn dan nul. Hiermee wordt het begin van het jaar gewijzigd met het opgegeven aantal dagen en/of fracties van een dag.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>inlunarweek('01/12/2013', '01/14/2013', 0)</code>	Retourneert <code>TRUE</code> , omdat de waarde van <code>timestamp</code> , 01/12/2013, in de week van 01/08/2013 tot 01/14/2013 valt.
<code>inlunarweek('01/12/2013', '01/07/2013', 0)</code>	Retourneert <code>FALSE</code> , omdat de <code>base_date</code> 01/07/2013 in de maanweek valt die is ingesteld op 01/01/2013 tot 01/07/2013.
<code>inlunarweek('01/12/2013', '01/14/2013', -1)</code>	Retourneert <code>FALSE</code> . Door het opgeven van een waarde van <code>period_no</code> als -1 wordt de week naar de vorige week verschoven, van 01/01/2013 t/m 01/07/2013.
<code>inlunarweek('01/07/2013', '01/14/2013', -1)</code>	Retourneert <code>TRUE</code> . In vergelijking met het voorafgaande voorbeeld ligt de <code>timestamp</code> in de volgende week daarna als wordt rekening gehouden met de terugwaartse verschuiving.
<code>inlunarweek('01/11/2006', '01/08/2006', 0, 3)</code>	Retourneert <code>FALSE</code> . Door een waarde van 3 op te geven voor <code>first_week_day</code> , wordt het begin van het jaar berekend vanaf 01/04/2013. Daarom valt de waarde van <code>base_date</code> in de eerste week en de waarde van <code>timestamp</code> in de week van 01/11/2013 t/m 01/17/2013.

De functie `inlunarweek()` wordt vaak gebruikt in combinatie met de volgende functies:

### Gerelateerde functies

Functie	Interactie
<code>lunarweekname</code> (page 855)	Deze functie wordt gebruikt om het maanweeknummer te bepalen van het jaar waarin een invoerdatum valt.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met transacties voor de maand januari die wordt geladen in de tabel `Transactions`.
- Het datumveld is opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

Maak een veld, `in_lunar_week`, dat bepaalt of de transacties in dezelfde maanweek plaatsvonden als 10 januari.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
    *,
    inlunarweek(date,'01/10/2022', 0) as in_lunar_week
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8183,'1/5/2022',42.32
```

```
8184,'1/6/2022',68.22
```

```
8185,'1/7/2022',15.25
```

```
8186, '1/8/2022', 25.26
8187, '1/9/2022', 37.23
8188, '1/10/2022', 37.23
8189, '1/11/2022', 17.17
8190, '1/12/2022', 88.27
8191, '1/13/2022', 57.42
8192, '1/14/2022', 53.80
8193, '1/15/2022', 82.06
8194, '1/16/2022', 87.21
8195, '1/17/2022', 95.93
8196, '1/18/2022', 45.89
8197, '1/19/2022', 36.23
8198, '1/20/2022', 25.66
8199, '1/21/2022', 82.77
8200, '1/22/2022', 69.98
8201, '1/23/2022', 76.11
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

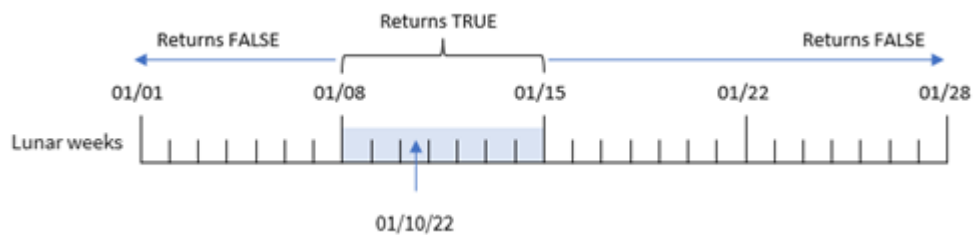
- date
- in\_lunar\_week

Resultatentabel

date	in_lunar_week
1/5/2022	0
1/6/2022	0
1/7/2022	0
1/8/2022	-1
1/9/2022	-1
1/10/2022	-1
1/11/2022	-1
1/12/2022	-1
1/13/2022	-1
1/14/2022	-1
1/15/2022	0
1/16/2022	0
1/17/2022	0
1/18/2022	0

date	in_lunar_week
1/19/2022	0
1/20/2022	0
1/21/2022	0
1/22/2022	0
1/23/2022	0

*in1lunarweek()-functie, basisvoorbeeld*



Het veld `in_lunar_week` wordt gemaakt in de voorafgaande load-instructie met behulp van de `in1lunarweek()`-functie en geeft vervolgens het volgende door als de argumenten van de functie:

- Het `date`-veld
- Een vastgelegde datum van 10 januari als de `base_date`
- Een `period_no` van 0

Omdat maanweken op 1 januari beginnen, zou 10 januari in de maanweek vallen die begint op 8 januari en eindigt op 14 januari. Daarom zouden alle transacties die tussen deze datums in januari plaatsvinden, de booleaanse waarde `TRUE` retourneren. Dit wordt gevalideerd in de resultatentabel.

### Voorbeeld 2 - `period_no`

Voorbeelden en resultaten:

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het datumveld is opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

Maak een veld, `2_lunar_weeks_later`, dat bepaalt of de transacties in dezelfde maanweek plaatsvonden als 10 januari.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
    Load
        *,
        inlunarweek(date,'01/10/2022', 2) as [2_lunar_weeks_later]
    ;

Load
*
Inline
[
id,date,amount
8183,'1/5/2022',42.32
8184,'1/6/2022',68.22
8185,'1/7/2022',15.25
8186,'1/8/2022',25.26
8187,'1/9/2022',37.23
8188,'1/10/2022',37.23
8189,'1/11/2022',17.17
8190,'1/12/2022',88.27
8191,'1/13/2022',57.42
8192,'1/14/2022',53.80
8193,'1/15/2022',82.06
8194,'1/16/2022',87.21
8195,'1/17/2022',95.93
8196,'1/18/2022',45.89
8197,'1/19/2022',36.23
8198,'1/20/2022',25.66
8199,'1/21/2022',82.77
8200,'1/22/2022',69.98
8201,'1/23/2022',76.11
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

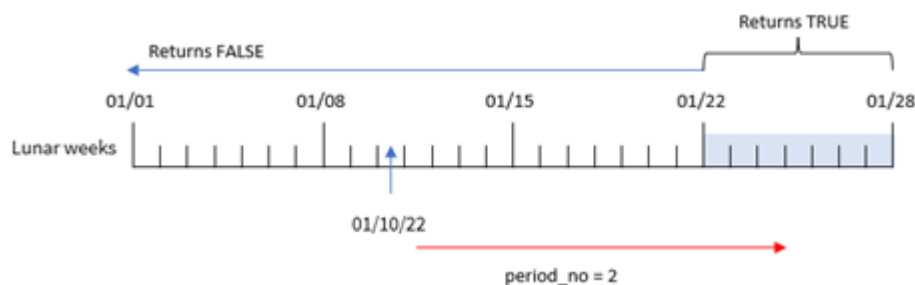
- date
- 2\_lunar\_weeks\_later

Resultatentabel

date	2_lunar_weeks_later
1/5/2022	0
1/6/2022	0
1/7/2022	0
1/8/2022	0

date	2_lunar_weeks_later
1/9/2022	0
1/10/2022	0
1/11/2022	0
1/12/2022	0
1/13/2022	0
1/14/2022	0
1/15/2022	0
1/16/2022	0
1/17/2022	0
1/18/2022	0
1/19/2022	0
1/20/2022	0
1/21/2022	0
1/22/2022	-1
1/23/2022	-1

*in1lunarweek()*-functie, voorbeeld van *period\_no*



In dit voorbeeld definieert de functie de week die begint op 22 januari als de maanweek op basis waarvan transacties moeten worden gevalideerd omdat een *period\_no* van 2 is gebruikt als het offset-argument in de *in1lunarweek()*-functie. Daarom zal elke transactie die plaatsvindt tussen 22 januari en 28 januari een booleaans resultaat van *TRUE* retourneren.

### Voorbeeld 3 - first\_week\_day

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script gebruikt dezelfde gegevensset en scenario als het eerste voorbeeld. Maar in dit voorbeeld stellen we het begin van de maanweek in op 6 januari.

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele `mm/dd/yyyy` wordt gebruikt.
- Een `first_week_day`-argument van 5. Dit stelt het begin van de maanweken in op 5 januari.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inlunarweek(date,'01/10/2022', 0,5) as in_lunar_week
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8183,'1/5/2022',42.32
```

```
8184,'1/6/2022',68.22
```

```
8185,'1/7/2022',15.25
```

```
8186,'1/8/2022',25.26
```

```
8187,'1/9/2022',37.23
```

```
8188,'1/10/2022',37.23
```

```
8189,'1/11/2022',17.17
```

```
8190,'1/12/2022',88.27
```

```
8191,'1/13/2022',57.42
```

```
8192,'1/14/2022',53.80
```

```
8193,'1/15/2022',82.06
```

```
8194,'1/16/2022',87.21
```

```
8195,'1/17/2022',95.93
```

```
8196,'1/18/2022',45.89
```

```
8197,'1/19/2022',36.23
```

```
8198,'1/20/2022',25.66
```

```
8199,'1/21/2022',82.77
```

```
8200,'1/22/2022',69.98
```

```
8201,'1/23/2022',76.11
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

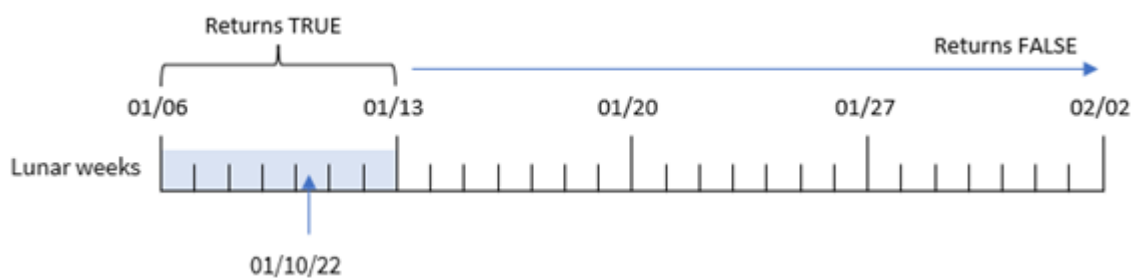
- `date`
- `in_lunar_week`



Resultatentabel

date	in_lunar_week
1/5/2022	0
1/6/2022	-1
1/7/2022	-1
1/8/2022	-1
1/9/2022	-1
1/10/2022	-1
1/11/2022	-1
1/12/2022	-1
1/13/2022	0
1/14/2022	0
1/15/2022	0
1/16/2022	0
1/17/2022	0
1/18/2022	0
1/19/2022	0
1/20/2022	0
1/21/2022	0
1/22/2022	0
1/23/2022	0

*in1lunarweek()-functie, first\_week\_day example*



In dit geval wordt het begin van de maanweekkalender verschoven naar 6 januari omdat in het `first_week_date`-argument van 5 wordt gebruikt in de functie `in1lunarweek()`. Daarom valt 10 januari in de maanweek die begint op 6 januari en eindigt op 12 januari. Iedere transactie die tussen deze twee datums valt, retourneert een booleaanse waarde voor `TRUE`.

### Voorbeeld 4 - Diagramobject

Load-script en diagramuitdrukking:

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het datumveld is opgegeven in de `dateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt of transacties plaatsvonden in dezelfde maand als 10 januari, is gemaakt als meting in een diagramobject van de toepassing.

#### Load-script

```
SET dateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8183,'1/5/2022',42.32
```

```
8184,'1/6/2022',68.22
```

```
8185,'1/7/2022',15.25
```

```
8186,'1/8/2022',25.26
```

```
8187,'1/9/2022',37.23
```

```
8188,'1/10/2022',37.23
```

```
8189,'1/11/2022',17.17
```

```
8190,'1/12/2022',88.27
```

```
8191,'1/13/2022',57.42
```

```
8192,'1/14/2022',53.80
```

```
8193,'1/15/2022',82.06
```

```
8194,'1/16/2022',87.21
```

```
8195,'1/17/2022',95.93
```

```
8196,'1/18/2022',45.89
```

```
8197,'1/19/2022',36.23
```

```
8198,'1/20/2022',25.66
```

```
8199,'1/21/2022',82.77
```

```
8200,'1/22/2022',69.98
```

```
8201,'1/23/2022',76.11
```

```
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `date`.

Maak de volgende meting om te berekenen of een transactie plaatsvindt in de maand waarin 10 januari valt:

```
= inlunarweek(date, '01/10/2022', 0)
```

Resultatentabel

date	=inlunarweek(date,'01/10/2022', 0)
1/5/2022	0
1/6/2022	0
1/7/2022	0
1/8/2022	-1
1/9/2022	-1
1/10/2022	-1
1/11/2022	-1
1/12/2022	-1
1/13/2022	-1
1/14/2022	-1
1/15/2022	0
1/16/2022	0
1/17/2022	0
1/18/2022	0
1/19/2022	0
1/20/2022	0
1/21/2022	0
1/22/2022	0
1/23/2022	0

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking:

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Products`.
- Informatie met daarin `product-id`, `productiedatum` en `kostprijs`.

Er is vastgesteld dat producten die in de maand van 12 januari zijn geproduceerd defect zijn vanwege een materieel fout. De eindgebruiker wil graag een diagramobject dat op maandnaam de status toont of de geproduceerde producten 'defect' of 'foutloos' zijn en de kosten van de producten die in die maand zijn geproduceerd.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
product_id,manufacture_date,cost_price
```

```
8183,'1/5/2022',42.32
```

```
8184,'1/6/2022',68.22
```

```
8185,'1/7/2022',15.25
```

```
8186,'1/8/2022',25.26
```

```
8187,'1/9/2022',37.23
```

```
8188,'1/10/2022',37.23
```

```
8189,'1/11/2022',17.17
```

```
8190,'1/12/2022',88.27
```

```
8191,'1/13/2022',57.42
```

```
8192,'1/14/2022',53.80
```

```
8193,'1/15/2022',82.06
```

```
8194,'1/16/2022',87.21
```

```
8195,'1/17/2022',95.93
```

```
8196,'1/18/2022',45.89
```

```
8197,'1/19/2022',36.23
```

```
8198,'1/20/2022',25.66
```

```
8199,'1/21/2022',82.77
```

```
8200,'1/22/2022',69.98
```

```
8201,'1/23/2022',76.11
```

```
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een dimensie om de maandnamen weer te geven:  
`=lunarweekname(manufacture_date)`
3. Maak een dimensie om te identificeren welke van de producten defect zijn en welke foutloos zijn met behulp van de `inlunarweek()`-functie:  
`=if(only(inlunarweek(manufacture_date,makedate(2022,01,12),0)), 'Defective','Faultless')`
4. Maak een meting om de `cost_price` van de producten op te tellen:

=sum(cost\_price)

5. Stel de **Getalnotatie** van de meting in op **Geld**.
6. Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

lunarweekname (manufacture_date)	=if(only(inlunarweek(manufacture_date,makedate(2022,01,12),0)), 'Defect','Foutloos')	sum(cost_price)
2022/01	Foutloos	\$125.79
2022/02	Defect	\$316.38
2022/03	Foutloos	\$455.75
2022/04	Foutloos	\$146.09

De functie `inlunarweek()` retourneert een booleaanse waarde bij het evalueren van de productiedatum van elk product. Voor elk product dat wordt geproduceerd in de maand waarin 10 januari valt, retourneert de `inlunarweek()`-functie een booleaanse waarde voor `TRUE` en markeert de producten als 'Defect'. Voor elk product dat de waarde voor `FALSE` retourneert, en dus niet in die week is geproduceerd, wordt het product als 'Foutloos' gemarkeerd.

### inlunarweektodate

Met deze functie wordt bepaald of **timestamp** binnen het gedeelte van de maand ligt tot en met de laatste milliseconde van **base\_date**. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

#### Syntaxis:

**InLunarWeekToDate** (timestamp, base\_date, period\_no [, first\_week\_day])

**Retourgegevenstypen:** Booleaanse waarde



*In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.*

Voorbeelddiagram van de `inlunarweektodate()`-functie



## 5 Script- en diagramfuncties

De `inlunarweektodate()`-functie functioneert als het eindpunt van de maanweek. Daarentegen bepaalt de `inlunarweek()`-functie in welke maanweek de `base_date` valt. Bijvoorbeeld: als de `base_date` 5 januari was, dan zou een tijdstempel tussen 1 januari en 5 januari een booleaans resultaat voor `TRUE` retourneren. Datums zoals 6 en 7 januari en zouden een booleaans resultaat voor `FALSE` retourneren.

### Argumenten

Argument	Beschrijving
<b>timestamp</b>	De datum die u wilt vergelijken met <b>base_date</b> .
<b>base_date</b>	Datum op basis waarvan de maanweek wordt geëvalueerd.
<b>period_no</b>	De maanweek kan worden verschoven met <b>period_no</b> . <b>period_no</b> is een geheel getal, waarbij de waarde 0 de maanweek aangeeft die <b>base_date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande maanweken aan en positieve waarden geven volgende maanweken aan.
<b>first_week_day</b>	Een verschuiving kan groter of kleiner zijn dan nul. Hiermee wordt het begin van het jaar gewijzigd met het opgegeven aantal dagen en/of fracties van een dag.

### Wanneer gebruiken

De functie `inlunarweektodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een IF-uitdrukking. De `inlunarweektodate()`-functie zou worden gebruikt wanneer de gebruiker wil dat de berekening een aggregatie of berekening retourneert, afhankelijk van of de geëvalueerde datum in een bepaald deel van de week in kwestie heeft plaatsgevonden.

De functie `inlunarweektodate()` kan bijvoorbeeld worden gebruikt om alle materieel te identificeren die in een bepaalde week t/m een bepaalde datum is geproduceerd.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>inlunarweektodate('01/12/2013', '01/13/2013', 0)</code>	Retourneert <code>TRUE</code> , omdat de waarde van <code>timestamp</code> , 01/12/2013, in het deel van de week van 01/08/2013 tot 01/13/2013 valt.
<code>inlunarweektodate('01/12/2013', '01/11/2013', 0)</code>	Retourneert <code>FALSE</code> omdat de waarde <code>timestamp</code> later is dan de waarde <code>base_date</code> hoewel de twee datums in dezelfde maanweek vallen vóór 01/12/2012.
<code>inlunarweektodate('01/12/2006', '01/05/2006', 1)</code>	Retourneert <code>TRUE</code> . Als een waarde van 1 wordt opgegeven voor <code>period_no</code> , wordt de <code>base_date</code> één week vooruit geschoven, zodat de waarde van <code>timestamp</code> in het deel van de maanweek valt.

De functie `inlunarweektodate()` wordt vaak gebruikt in combinatie met de volgende functies:

### Gerelateerde functies

Functie	Interactie
<i>lunarweekname</i> (page 855)	Deze functie wordt gebruikt om het maanweeknummer te bepalen van het jaar waarin een invoerdatum valt.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor de maand januari die wordt geladen in de tabel `Transactions`. De standaard `DateFormat`-systeemvariabele `mm/dd/yyyy` wordt gebruikt.
- Maak een veld, `in_lunar_week_to_date`, dat bepaalt welke transacties in de maanweek tot de datum 10 januari plaatsvonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
    *,
    inlunarweektoday(date,'01/10/2022', 0) as in_lunar_week_to_date
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/10/2022',37.23
```

```
8189,'1/17/2022',17.17
```

```
8190,'1/26/2022',88.27
```

```
8191,'1/12/2022',57.42
```

```
8192,'1/19/2022',53.80
```

```
8193, '1/21/2022', 82.06
8194, '1/1/2022', 40.39
8195, '1/27/2022', 87.21
8196, '1/11/2022', 95.93
8197, '1/29/2022', 45.89
8198, '1/31/2022', 36.23
8199, '1/18/2022', 25.66
8200, '1/23/2022', 82.77
8201, '1/15/2022', 69.98
8202, '1/4/2022', 76.11
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

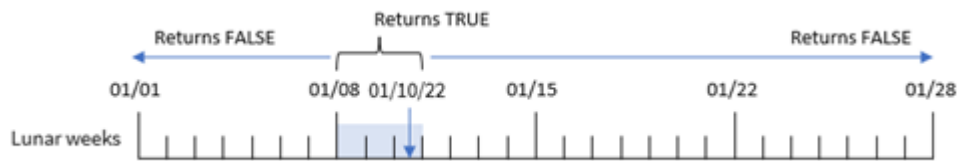
- date
- in\_lunar\_week\_to\_date

Resultatentabel

date	in_lunar_week_to_date
1/1/2022	0
1/4/2022	0
1/10/2022	-1
1/11/2022	0
1/12/2022	0
1/15/2022	0
1/17/2022	0
1/18/2022	0
1/19/2022	0
1/21/2022	0
1/23/2022	0
1/26/2022	0
1/27/2022	0
1/29/2022	0
1/31/2022	0



*inlunarweektodate()-functie, geen extra argumenten*



Het `in_lunar_week_to_date`-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de `inlunarweektodate()`-functie en geeft het `date`-veld door, een vastgelegde tijdstempel voor 10 januari als onze `base_date` en een offset van 0 als de argumenten van de functie.

Omdat maanweken beginnen op 1 januari, zou 10 januari in de maakweek vallen die op 8 januari begint, en omdat we de `inlunarweektodate()`-functie gebruiken zou die maanweek eindigen op de 10e. Daarom zouden alle transacties die tussen deze datums in januari plaatsvinden, de booleaanse waarde `TRUE` retourneren. Dit wordt gevalideerd in de resultatentabel.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. Maar in dit voorbeeld, is de taak om een veld te maken, `2_lunar_weeks_later`, dat bepaalt of de transacties al dan niet twee weken na de maanweek van 1 januari plaatsvonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
Transactions:
  Load
    *,
    inlunarweektodate(date,'01/10/2022', 2) as [2_lunar_weeks_later]
  ;
Load
*
Inline
[
id,date,amount
8188,'1/10/2022',37.23
8189,'1/17/2022',17.17
8190,'1/26/2022',88.27
8191,'1/12/2022',57.42
8192,'1/19/2022',53.80
8193,'1/21/2022',82.06
8194,'1/1/2022',40.39
8195,'1/27/2022',87.21
```

```
8196, '1/11/2022', 95.93
8197, '1/29/2022', 45.89
8198, '1/31/2022', 36.23
8199, '1/18/2022', 25.66
8200, '1/23/2022', 82.77
8201, '1/15/2022', 69.98
8202, '1/4/2022', 76.11
];
```

### Resultaten

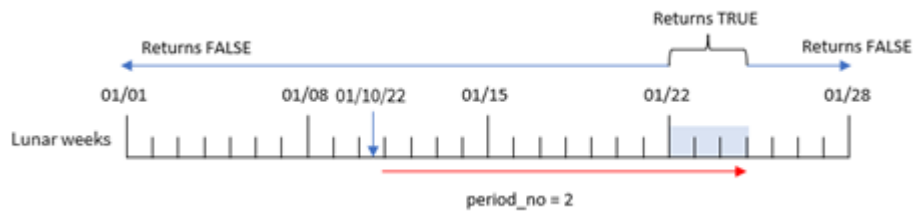
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- 2\_lunar\_weeks\_later

Resultatentabel

date	2_lunar_weeks_later
1/1/2022	0
1/4/2022	0
1/10/2022	0
1/11/2022	0
1/12/2022	0
1/15/2022	0
1/17/2022	0
1/18/2022	0
1/19/2022	0
1/21/2022	0
1/23/2022	-1
1/26/2022	0
1/27/2022	0
1/29/2022	0
1/31/2022	0

*in\_lunarweektodate()-functie, voorbeeld van period\_no*



In deze instantie bepaalt de `in_lunarweektodate()`-functie dat de maanweek tot 10 januari gelijk is aan drie dagen (8, 9 en 10 januari). Omdat er een `period_no` van 2 is gebruikt als het offset-argument, wordt deze maanweek met 14 dagen verschoven. Dit zorgt er daarom voor dat de definitie van deze driedaagse maanweek bestaat uit 22, 23 en 24 januari. Iedere transactie die plaatsvindt tussen 22 januari en 24 januari zal een booleaans resultaat van `TRUE` retourneren.

### Voorbeeld 3 - first\_week\_day

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- De standaard `DateFormat`-systeemvariabele `mm/dd/yyyy` wordt gebruikt.
- Een `first_week_date`-argument van 3. Dit stelt het begin van de maanweken in op 3 januari

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,  
in_lunarweek(date,'01/10/2022', 0,3) as in_lunar_week_to_date  
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/10/2022',37.23
```

```
8189,'1/17/2022',17.17
```

```
8190,'1/26/2022',88.27
```

```
8191,'1/12/2022',57.42
```

```
8192,'1/19/2022',53.80
```

```
8193,'1/21/2022',82.06
```

```
8194,'1/1/2022',40.39
```

```
8195, '1/27/2022', 87.21
8196, '1/11/2022', 95.93
8197, '1/29/2022', 45.89
8198, '1/31/2022', 36.23
8199, '1/18/2022', 25.66
8200, '1/23/2022', 82.77
8201, '1/15/2022', 69.98
8202, '1/4/2022', 76.11
];
```

### Resultaten

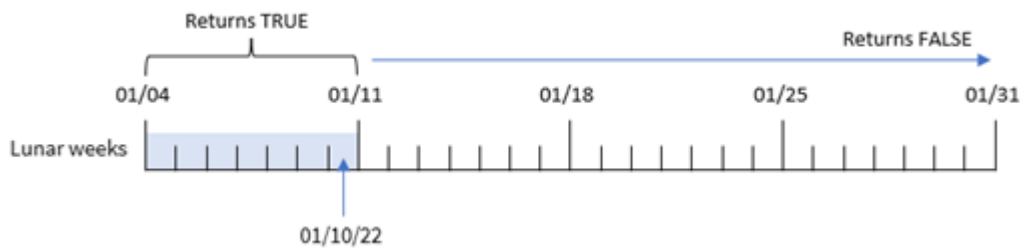
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_lunar\_week\_to\_date

Resultatentabel

date	in_lunar_week_to_date
1/1/2022	0
1/4/2022	-1
1/10/2022	-1
1/11/2022	0
1/12/2022	0
1/15/2022	0
1/17/2022	0
1/18/2022	0
1/19/2022	0
1/21/2022	0
1/23/2022	0
1/26/2022	0
1/27/2022	0
1/29/2022	0
1/31/2022	0

*in1lunarweektodate()-functie, first\_week\_day example*



In deze instantie loopt de eerste maanweek van 3 januari t/m 10 januari omdat het the `first_week_date`-argument 3 is gebruikt in de `in1lunarweek()`-functie. Omdat 10 januari tevens de `base_date` is, retourneert iedere transactie die tussen deze twee datums valt, een booleaanse waarde van `TRUE`.

### Voorbeeld 4 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt of transacties plaatsvonden in de maanweek tot 10 januari, is gemaakt als meting in een diagramobject van de toepassing.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/10/2022',37.23
```

```
8189,'1/17/2022',17.17
```

```
8190,'1/26/2022',88.27
```

```
8191,'1/12/2022',57.42
```

```
8192,'1/19/2022',53.80
```

```
8193,'1/21/2022',82.06
```

```
8194,'1/1/2022',40.39
```

```
8195,'1/27/2022',87.21
```

```
8196,'1/11/2022',95.93
```

```
8197,'1/29/2022',45.89
```

```
8198,'1/31/2022',36.23
```

```
8199,'1/18/2022',25.66
```

## 5 Script- en diagramfuncties

```
8200, '1/23/2022', 82.77  
8201, '1/15/2022', 69.98  
8202, '1/4/2022', 76.11  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Maak de volgende meting:

```
=inlunarweektodate(date, '01/10/2022', 0)
```

Resultatentabel

date	=inlunarweektodate(date,'01/10/2022', 0)
1/1/2022	0
1/4/2022	0
1/10/2022	-1
1/11/2022	0
1/12/2022	0
1/15/2022	0
1/17/2022	0
1/18/2022	0
1/19/2022	0
1/21/2022	0
1/23/2022	0
1/26/2022	0
1/27/2022	0
1/29/2022	0
1/31/2022	0

*inlunarweektodate()-functie, diagramobjectvoorbeeld*



De meting `in_lunar_week_to_date` wordt gemaakt in het diagramobject met behulp van de `inlunarweektodate()`-functie en geeft het datumveld door, een vastgelegde tijdstempel voor 10 januari als onze `base_date` en een offset van 0 als de argumenten van de functie.

Omdat maanweken beginnen op 1 januari, zou 10 januari in de maakweek vallen die op 8 januari begint. Omdat we de `inlunarweektodate()`-functie gebruiken zou die maanweek eindigen op de 10e. Daarom zouden alle transacties die tussen deze datums in januari plaatsvinden, de booleaanse waarde `TRUE` retourneren. Dit wordt gevalideerd in de resultatentabel.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukkingen

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Products`.
- Informatie met daarin `product-id`, `productiedatum` en `kostprijs`.

Er is vastgesteld dat producten die in de maanweek van 12 januari zijn geproduceerd defect zijn vanwege een materiële fout. Op 13 januari is het probleem opgelost. De eindgebruiker wil graag een diagramobject dat op week de status toont of de geproduceerde producten 'defect' of 'foutloos' zijn en de kosten van de producten die in die week zijn geproduceerd.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff]';
```

```
Products:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
product_id,manufacture_date,cost_price
```

```
8188,'01/02/2022 12:22:06',37.23
```

```
8189,'01/05/2022 01:02:30',17.17
```

```
8190,'01/06/2022 15:36:20',88.27
```

```
8191,'01/08/2022 10:58:35',57.42
```

```
8192,'01/09/2022 08:53:32',53.80
```

```
8193,'01/10/2022 21:13:01',82.06
```

```
8194,'01/11/2022 00:57:13',40.39
```

```
8195,'01/12/2022 09:26:02',87.21
```

```
8196,'01/13/2022 15:05:09',95.93
```

```
8197,'01/14/2022 18:44:57',45.89
```

```
8198,'01/15/2022 06:10:46',36.23
```

```
8199,'01/16/2022 06:39:27',25.66
```

```
8200,'01/17/2022 10:44:16',82.77
```

```
8201,'01/18/2022 18:48:17',69.98
```

```
8202,'01/26/2022 04:36:03',76.11
```

```
8203, '01/27/2022 08:07:49', 25.12
8204, '01/28/2022 12:24:29', 46.23
8205, '01/30/2022 11:56:56', 84.21
8206, '01/30/2022 14:40:19', 96.24
8207, '01/31/2022 05:28:21', 67.67
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een dimensie om de weeknamen weer te geven:  
`=weekname(manufacture_date)`
3. Maak vervolgens een dimensie om te maken die de `inlunarweektodate()`-functie gebruikt om te identificeren welke van de producten defect zijn en welke foutloos zijn:  
`=if(inlunarweektodate(manufacture_date,makedate(2022,01,12),0),'Defective','Faultless')`
4. Maak een meting om de `cost_price` van de producten op te tellen:  
`=sum(cost_price)`
5. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

<code>=lunarweekname (manufacture_date)</code>	<code>=if(InLunarWeekToDate(manufacture_date,makedate (2022,01,12),0),'Defective','Faultless')</code>	<code>=Sum(cost_ price)</code>
2022/01	Foutloos	\$142.67
2022/02	Defect	\$320.88
2022/02	Foutloos	\$141.82
2022/03	Foutloos	\$214.64
2022/04	Foutloos	\$147.46
2022/05	Foutloos	\$248.12

De functie `inlunarweektodate()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor degenen die een booleaanse waarde van `TRUE` retourneren, markeert het de producten als 'defective'. Voor elk product dat een waarde retourneert van `FALSE`, en dus niet gemaakt in de maand tot en met 12 januari, markeert het de producten als 'Faultless'.

### inmonth

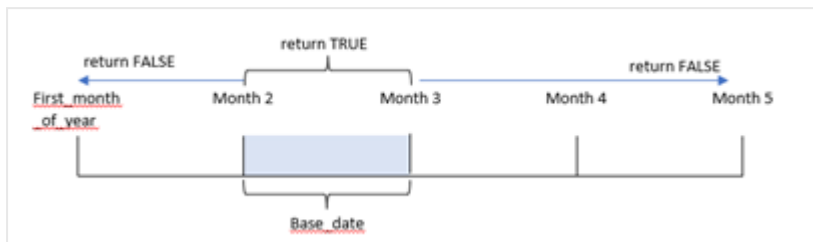
Deze functie retourneert `True` als **timestamp** in de maand valt die **base\_date** bevat.

#### Syntaxis:

```
InMonth (timestamp, base_date, period_no)
```



Diagram van *inmonth*-functie.



Met andere woorden, de functie `inmonth()` bepaalt of een reeks datums in deze maand valt en retourneert een booleaanse waarde op basis van een `base_date` die de maand identificeert.

### Wanneer gebruiken

De functie `inmonth()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if` expression. Hierdoor wordt een aggregatie of berekening geretourneerd, afhankelijk van of een datum plaatsvond in de maand, inclusief de desbetreffende datum.

De functie `inmonth()` bijvoorbeeld kan worden gebruikt om alle materieel te identificeren die in een specifieke maand is geproduceerd.

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

### Argumenten

Argument	Beschrijving
tijdstempel	De datum die u wilt vergelijken met <code>base_date</code> .
<code>base_date</code>	Datum op basis waarvan de maand wordt geëvalueerd. Het is belangrijk op te merken dat de <code>base_date</code> elke dag binnen een maand kan zijn.
<code>period_no</code>	De maand kan worden verschoven met <code>period_no</code> . <code>period_no</code> is een geheel getal, waarbij de waarde 0 de maand aangeeft die <code>base_date</code> bevat. Negatieve waarden in <code>period_no</code> geven voorafgaande maanden aan en positieve waarden geven volgende maanden aan.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze

landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>inmonth ('25/01/2013', '01/01/2013', 0)</code>	Retourneert True
<code>inmonth('25/01/2013', '23/04/2013', 0)</code>	Retourneert False
<code>inmonth ('25/01/2013', '01/01/2013', -1)</code>	Retourneert False
<code>inmonth ('25/12/2012', '17/01/2013', -1)</code>	Retourneert True

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor de eerste helft van 2022.
- Een voorafgaande lading met een extra variabele, 'in\_month', die bepaalt of transacties in april hebben plaatsgevonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    inmonth(date,'04/01/2022', 0) as in_month
  ;
Load
*
Inline
[
id,date,amount
8188,'1/10/2022',37.23
8189,'1/14/2022',17.17
8190,'1/20/2022',88.27
8191,'1/22/2022',57.42
8192,'2/1/2022',53.80
8193,'2/2/2022',82.06
8194,'2/20/2022',40.39
8195,'4/11/2022',87.21
8196,'4/13/2022',95.93
```

```
8197, '4/15/2022', 45.89
8198, '4/25/2022', 36.23
8199, '5/20/2022', 25.66
8200, '5/22/2022', 82.77
8201, '6/19/2022', 69.98
8202, '6/22/2022', 76.11
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_month

Voorbeelden van functies

date	in_month
1/10/2022	0
1/14/2022	0
1/20/2022	0
1/22/2022	0
2/1/2022	0
2/2/2022	0
2/20/2022	0
4/11/2022	-1
4/13/2022	-1
4/15/2022	-1
4/25/2022	-1
5/20/2022	0
5/22/2022	0
6/19/2022	0
6/22/2022	0

Het veld 'in\_month' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `inmonth()` en geeft het datumveld door, een vastgelegde datum van 1 april, als onze `base_date` en een `period_no` van 0 als de argumenten van de functie.

De `base_date` identificeert de maand die het booleaanse resultaat TRUE zal opleveren. Daarom retourneren alle transacties die in april hebben plaatsgevonden TRUE, wat wordt gevalideerd in de resultatentabel.

### Voorbeeld 2 - period\_no

Load-script en resultaten

### Overzicht

Dezelfde gegevensset en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld maakt u echter een veld '2\_months\_prior' aan dat bepaalt of de transacties twee maanden voor april hebben plaatsgevonden.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
Load
    *,
    inmonth(date,'04/01/2022', -2) as [2_months_prior]
Inline
[
id,date,amount
8188,'1/10/2022',37.23
8189,'1/14/2022',17.17
8190,'1/20/2022',88.27
8191,'1/22/2022',57.42
8192,'2/1/2022',53.80
8193,'2/2/2022',82.06
8194,'2/20/2022',40.39
8195,'4/11/2022',87.21
8196,'4/13/2022',95.93
8197,'4/15/2022',45.89
8198,'4/25/2022',36.23
8199,'5/20/2022',25.66
8200,'5/22/2022',82.77
8201,'6/19/2022',69.98
8202,'6/22/2022',76.11
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- 2\_months\_prior

Voorbeelden van functies

date	2_months_prior
1/10/2022	0
1/14/2022	0
1/20/2022	0
1/22/2022	0

date	2_months_prior
2/1/2022	-1
2/2/2022	-1
2/20/2022	-1
4/11/2022	0
4/13/2022	0
4/15/2022	0
4/25/2022	0
5/20/2022	0
5/22/2022	0
6/19/2022	0
6/22/2022	0

Als u -2 gebruikt als het `period_no`-argument in de `inmonth()`-functie, verschuift de maand gedefinieerd door het `base_date`-argument naar twee maanden eerder. In dit voorbeeld verandert het de gedefinieerde maand van april in februari.

Daarom zal elke transactie die in februari plaatsvindt een booleaans resultaat van TRUE retourneren.

### Voorbeeld 3 - Diagramobject

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensset en het scenario uit de vorige voorbeelden worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die bepaalt of transacties plaatsvonden in april is gemaakt als meting in een diagramobject van de toepassing.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/10/2022',37.23
```

```
8189,'1/14/2022',17.17
```

```
8190,'1/20/2022',88.27
```

```
8191,'1/22/2022',57.42
```

```
8192,'2/1/2022',53.80
```

```
8193, '2/2/2022', 82.06
8194, '2/20/2022', 40.39
8195, '4/11/2022', 87.21
8196, '4/13/2022', 95.93
8197, '4/15/2022', 45.89
8198, '4/25/2022', 36.23
8199, '5/20/2022', 25.66
8200, '5/22/2022', 82.77
8201, '6/19/2022', 69.98
8202, '6/22/2022', 76.11
];
```

### Diagramobject

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

date

Maak de volgende meting om te berekenen of een transactie plaatsvindt in april:

```
=inmonth(date, '04/01/2022', 0)
```

### Resultaten

Voorbeelden van functies

date	=inmonth(date, '04/01/2022', 0)
1/10/2022	0
1/14/2022	0
1/20/2022	0
1/22/2022	0
2/1/2022	0
2/2/2022	0
2/20/2022	0
4/11/2022	-1
4/13/2022	-1
4/15/2022	-1
4/25/2022	-1
5/20/2022	0
5/22/2022	0
6/19/2022	0
6/22/2022	0

### Voorbeeld 4 - Scenario

Load-script en resultaten

#### Overzicht

In dit voorbeeld wordt een gegevensset geladen in de tabel 'Products'. De tabel bevat de volgende velden:

- Product-id
- Productiedatum
- Kostprijs

Door een fout in de apparatuur waren producten die in de maand juli 2022 zijn geproduceerd defect. Het probleem is op 27 juli 2022 opgelost.

De eindgebruiker wil graag een diagram dat op maand de status toont van producten die zijn geproduceerd als 'defect' (booleaans WAAR) of 'foutloos' (booleaans NIET WAAR) en de kosten van de producten die in die maand zijn geproduceerd.

#### Load-script

Products:

Load

\*

Inline

[

product\_id,manufacture\_date,cost\_price

8188,'1/19/2022',37.23

8189,'1/7/2022',17.17

8190,'2/28/2022',88.27

8191,'2/5/2022',57.42

8192,'3/16/2022',53.80

8193,'4/1/2022',82.06

8194,'5/7/2022',40.39

8195,'5/16/2022',87.21

8196,'6/15/2022',95.93

8197,'6/26/2022',45.89

8198,'7/9/2022',36.23

8199,'7/22/2022',25.66

8200,'7/23/2022',82.77

8201,'7/27/2022',69.98

8202,'8/2/2022',76.11

8203,'8/8/2022',25.12

8204,'8/19/2022',46.23

8205,'9/26/2022',84.21

8206,'10/14/2022',96.24

8207,'10/29/2022',67.67

];

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

=monthname(manufacture\_date)

Maak de volgende metingen:

- =sum(cost\_price)
- =if(only(inmonth(manufacture\_date,makedate(2022,07,01),0)), 'Defective', 'Faultless')

1. Stel de **Getalnotatie** van de meting in op **Geld**.
2. Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

monthname (manufacture_date)	=if(only(inmonth(manufacture_date,makedate (2022,07,01),0)), 'Defective', 'Faultless')	sum(cost_ price)
januari 2022	Foutloos	\$54.40
februari 2022	Foutloos	\$145.69
maart 2022	Foutloos	\$53.80
april 2022	Foutloos	\$82.06
mei 2022	Foutloos	\$127.60
juni 2022	Foutloos	\$141.82
juli 2022	Defect	\$214.64
augustus 2022	Foutloos	\$147.46
september 2022	Foutloos	\$84.21
oktober 2022	Foutloos	\$163.91

De functie `inmonth()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat in juli 2022 is geproduceerd, retourneert de functie `inmonth()` een booleaanse waarde voor Waar en markeert de producten als 'Defect'. Voor elk product dat de waarde voor Niet waar retourneert, en dus niet op in juli is geproduceerd, wordt het product als 'Foutloos' gemarkeerd.

### inmonths

Deze functie bepaalt of een tijdstempel valt binnen dezelfde periode van de maand, twee maanden, kwartaal, tertiaal of half jaar als basisdatum. Het is tevens mogelijk om te bepalen of de tijdstempel binnen een voorafgaande of volgende tijdsperiode valt.

#### Syntaxis:

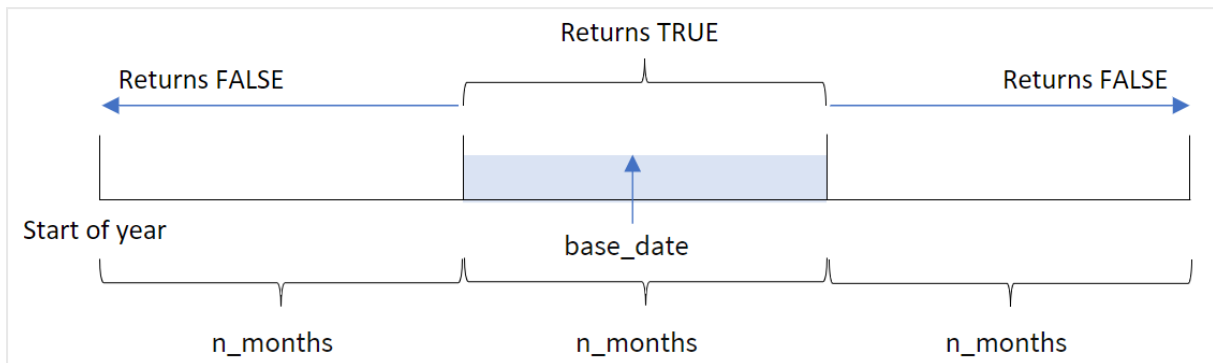
```
InMonths (n_months, timestamp, base_date, period_no [, first_month_of_year])
```

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.



Diagram van de `inmonths()`-functie



De functie `inmonths()` verdeelt het jaar in segmenten op basis van het opgegeven `n_months`-argument. Het bepaalt vervolgens of een geëvalueerde tijdstempel in hetzelfde segment valt als het `base_date`-argument. Als er echter een `period_no`-argument is opgegeven, bepaalt de functie of de tijdstempels in een vorige of volgende periode vallen ten opzichte van de `base_date`.

De volgende segmenten van het jaar zijn beschikbaar in de functie als `n_month`-argumenten:

n_month-argumenten	
Periode	Aantal maanden
maandelijks	1
tweemaandelijks	2
kwartaal	3
tertiaal	4
halfjaarlijks	6

### Wanneer gebruiken

De functie `inmonths()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if` expression. Door de `inmonths()`-functie te gebruiken, kunt u de periode selecteren die u wilt laten evalueren. Bijvoorbeeld: door de gebruiker producten te laten identificeren die in de maand, het kwartaal of de jaarhelft van een bepaalde periode zijn geproduceerd.

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

### Argumenten

Argument	Beschrijving
<code>n_months</code>	Het aantal maanden dat de periode definieert. Een geheel getal of uitdrukking die wordt herleid tot een geheel getal dat een van de volgende waarden moet hebben: 1 (equivalent aan de functie <code>inmonth()</code> ), 2 (2 maanden), 3 (equivalent aan de functie <code>inquarter()</code> ), 4 (tertiaal) of 6 (half jaar).

Argument	Beschrijving
<b>timestamp</b>	De datum die u wilt vergelijken met <b>base_date</b> .
<b>base_date</b>	Datum op basis waarvan de periode wordt geëvalueerd.
<b>period_no</b>	De periode kan worden verschoven met <b>period_no</b> , een geheel getal of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de periode aangeeft die <b>base_date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande perioden aan en positieve waarden geven volgende perioden aan.
<b>first_month_of_year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> .

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden `first_month_of_year`

Month	Waarde
Februari	2
Maart	3
April	4
May	5
Juni	6
Juli	7
Augustus	8
September	9
Oktober	10
November	11
December	12

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze

landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>inmonths(4, '01/25/2013', '04/25/2013', 0)</code>	Retourneert TRUE. Omdat de waarde van timestamp, 25/01/2013, binnen de periode van vier maanden van 01/01/2013 t/m 30/04/2013 ligt, waarin de waarde van base_date, 25/04/2013 ligt.
<code>inmonths(4, '05/25/2013', '04/25/2013', 0)</code>	Retourneert FALSE. Omdat 25/05/2013 buiten dezelfde periode als het voorafgaande voorbeeld ligt.
<code>inmonths(4, '11/25/2012', '02/01/2013', -1)</code>	Retourneert TRUE. Omdat de waarde van period_no, -1, de zoekperiode één periode van vier maanden (de waarde van n-months) terugschuift, waardoor de zoekperiode valt van 01/09/2012 t/m 31/12/2012.
<code>inmonths(4, '05/25/2006', '03/01/2006', 0, 3)</code>	Retourneert TRUE. Aangezien de waarde van first_month_of_year is ingesteld op 3, waardoor de zoekperiode 03/01/2006 tot 07/30/2006 wordt in plaats van 01/01/2006 tot 04/30/2006.

### Voorbeeld 1 - Geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 wordt in de tabel Transactions geladen.
- Een voorafgaande lading met een extra variabele, in\_months, die bepaalt of transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei 2022 valt.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    inmonths(3,date,'05/15/2022', 0) as in_months
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
8188,'2/19/2022',37.23
8189,'3/7/2022',17.17
8190,'3/30/2022',88.27
8191,'4/5/2022',57.42
8192,'4/16/2022',53.80
8193,'5/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/22/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_months

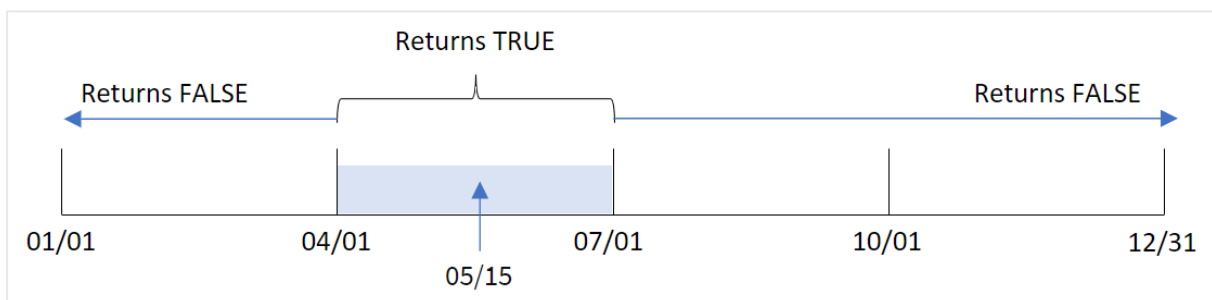
Resultatentabel

date	in_months
2/19/2022	0
3/7/2022	0
3/30/2022	0
4/5/2022	-1
4/16/2022	-1
5/1/2022	-1
5/7/2022	-1
5/22/2022	-1
6/15/2022	-1
6/26/2022	-1
7/9/2022	0

date	in_months
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

Het veld `in_months` wordt gemaakt in de vorige load-instructie met behulp van de functie `inmonths()`. Het eerste opgegeven argument is 3, waarbij het jaar wordt verdeeld in kwartaalsegmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd; het datumveld in dit voorbeeld. Het derde argument is een vastgestelde datum voor 15 mei. Dit is de `base_date` en een `period_no` van 0 is het laatste argument.

*Diagram van de `inmonths()`-functie met kwartaalsegmenten*



Mei valt in het tweede kwartaal van het jaar. Daarom zal elke transactie die plaatsvindt tussen 1 april en 30 juni een booleaans resultaat voor TRUE retourneren. Dit wordt gevalideerd in de resultatentabel.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 wordt in de tabel `transactions` geladen.
- Een voorafgaande lading met een extra variabele, `previous_quarter`, die bepaalt of transacties hebben plaatsgevonden in het kwartaal voorafgaand aan het kwartaal waarin 15 mei 2022 valt.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    inmonths(3,date,'05/15/2022', -1) as previous_quarter
  ;

Load
*
Inline
[
id,date,amount
8188,'2/19/2022',37.23
8189,'3/7/2022',17.17
8190,'3/30/2022',88.27
8191,'4/5/2022',57.42
8192,'4/16/2022',53.80
8193,'5/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/22/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_quarter

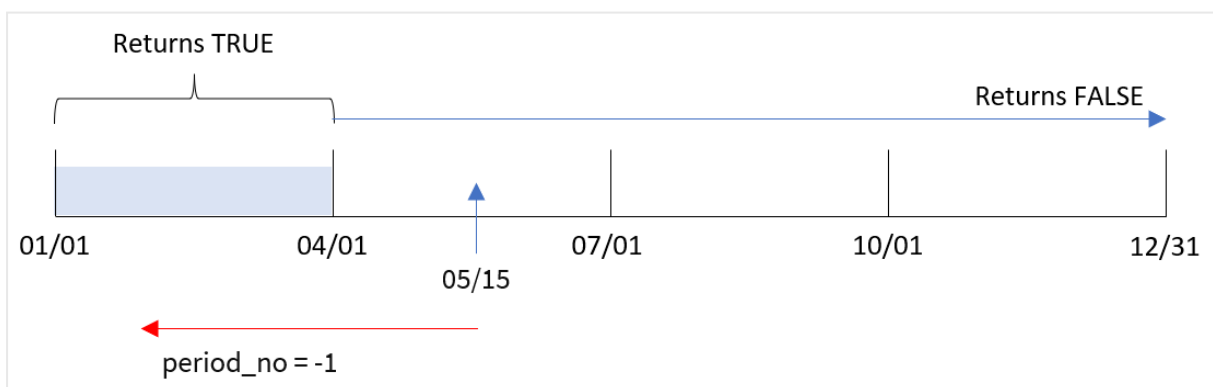
Resultatentabel

date	vorig kwartaal
2/19/2022	-1
3/7/2022	-1
3/30/2022	-1
4/5/2022	0

date	vorig kwartaal
4/16/2022	0
5/1/2022	0
5/7/2022	0
5/22/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

De functie stelt vast of transacties in het eerste kwartaal van het jaar hebben plaatsgevonden met -1 als het `period_no`-argument in de `inmonths()`-functie. 15 Mei is de `base_date` en valt in het tweede kwartaal van het jaar (april-juni).

Diagram van de `inmonths()`-functie met kwartaalsegmenten en de `period_no` ingesteld op -1



Daarom zal elke transactie die plaatsvindt tussen januari en maart een booleaans resultaat voor TRUE retourneren.

### Voorbeeld 3 - first\_month\_of\_year

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 wordt in de tabel Transactions geladen.
- Een voorafgaande lading met een extra variabele, `in_months`, die bepaalt of transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei 2022 valt.

In dit voorbeeld is maart volgens het organisatiebeleid de eerste maand van het boekjaar.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inmonths(3,date,'05/15/2022', 0, 3) as in_months
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,'2/19/2022',37.23
8189,'3/7/2022',17.17
8190,'3/30/2022',88.27
8191,'4/5/2022',57.42
8192,'4/16/2022',53.80
8193,'5/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/22/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```



### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

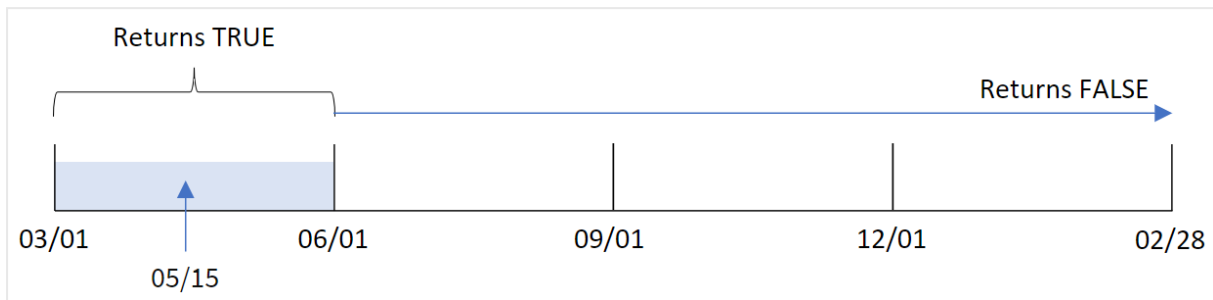
- date
- in\_months

Resultatentabel

date	in_months
2/19/2022	0
3/7/2022	-1
3/30/2022	-1
4/5/2022	-1
4/16/2022	-1
5/1/2022	-1
5/7/2022	-1
5/22/2022	-1
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

Door 3 te gebruiken als het `first_month_of_year`-argument in de `inmonths()`-functie, begint het jaar op 1 maart. De `inmonths()`-functie verdeelt het jaar vervolgens in kwartalen. Maa-mei, jun-aug, sep-nov, dec-feb. Daarom valt 15 mei in het eerste kwartaal van het jaar (maart-mei).

Diagram van functie `inmonths()` met maart ingesteld als de eerste maand van het jaar



Een transactie die in deze maanden plaatsvindt, retourneert een booleaans resultaat voor TRUE.

### Voorbeeld 4 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensset en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die bepaalt welke transacties plaatsvonden in hetzelfde kwartaal als 15 mei, wordt gemaakt als een meting in een diagram in de app.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'2/19/2022',37.23
```

```
8189,'3/7/2022',17.17
```

```
8190,'3/30/2022',88.27
```

```
8191,'4/5/2022',57.42
```

```
8192,'4/16/2022',53.80
```

```
8193,'5/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/22/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- date

Om te berekenen of transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei valt, maakt u de volgende meting:

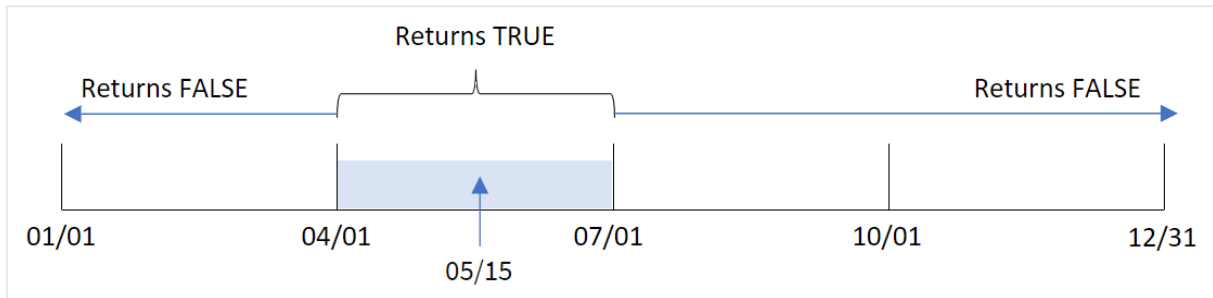
```
=inmonths(3,date,'05/15/2022', 0)
```

Resultatentabel

date	=inmonths(3,date,'05/15/2022', 0)
2/19/2022	0
3/7/2022	0
3/30/2022	0
4/5/2022	-1
4/16/2022	-1
5/1/2022	-1
5/7/2022	-1
5/22/2022	-1
6/15/2022	-1
6/26/2022	-1
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

Het veld `in_months` wordt in een diagram gemaakt met behulp van de functie `inmonths()`. Het eerste opgegeven argument is 3, waarbij het jaar wordt verdeeld in kwartaalsegmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd; het datumveld in dit voorbeeld. Het derde argument is een vastgestelde datum voor 15 mei. Dit is de `base_date` en een `period_no` van 0 is het laatste argument.

*Diagram van de `inmonths()`-functie met kwartaalsegmenten*



Mei valt in het tweede kwartaal van het jaar. Daarom zal elke transactie die plaatsvindt tussen 1 april en 30 juni een booleaans resultaat voor TRUE retourneren. Dit wordt gevalideerd in de resultatentabel.

### Voorbeeld 5 - Scenario

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die in de tabel `Products` wordt geladen.
- De tabel bevat de volgende velden:
  - product ID
  - producttype
  - productiedatum
  - cost price

De eindgebruiker wil een diagram dat de kosten per producttype weergeeft van de producten die in het eerste segment van 2021 zijn geproduceerd. De gebruiker wil de duur van dit segment kunnen definiëren.

#### Load-script

```
SET vPeriod = 1;
```

```
Products:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
product_id,product_type,manufacture_date,cost_price
```

```
8188,product A,'2/19/2022',37.23
```

```
8189,product D,'3/7/2022',17.17
8190,product C,'3/30/2022',88.27
8191,product B,'4/5/2022',57.42
8192,product D,'4/16/2022',53.80
8193,product D,'5/1/2022',82.06
8194,product A,'5/7/2022',40.39
8195,product B,'5/22/2022',87.21
8196,product C,'6/15/2022',95.93
8197,product B,'6/26/2022',45.89
8198,product C,'7/9/2022',36.23
8199,product D,'7/22/2022',25.66
8200,product D,'7/23/2022',82.77
8201,product A,'7/27/2022',69.98
8202,product A,'8/2/2022',76.11
8203,product B,'8/8/2022',25.12
8204,product B,'8/19/2022',46.23
8205,product B,'9/26/2022',84.21
8206,product C,'10/14/2022',96.24
8207,product D,'10/29/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad.

Aan het begin van het load-script is een variabele, `vPeriod`, gemaakt die wordt gekoppeld aan het besturingselement voor variabeleninput.

Doe het volgende:

1. Klik in het bedrijfsmiddelenvenster op **Aangepaste objecten**.
2. Selecteer **Qlik Dashboard-bundel** en maak een **Variabeleninput**-object.
3. Voer een titel in voor het diagramobject.
4. Selecteer onder **Variabele** de optie **vPeriod** als de naam en stel het object in om als een **Vervolgkeuzelijst** te worden weergegeven.
5. Klik onder **Waarden** op **Dynamische waarden**. Voer het volgende in:  
`= '1~month|2~bi-month|3~quarter|4~tertiaal|6~half-year'`.
6. Voeg een nieuwe tabel toe aan het werkblad.
7. Voeg in het eigenschappenvenster onder **Gegevens** `product_type` toe als een dimensie.
8. Voeg de volgende uitdrukking als een meting toe:  
`=sum(if(inmonths($(vPeriod),manufacture_date,makedate(2022,01,01)),0),cost_price,0))`
9. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

product_type	=sum(if(inmonths(\$(vPeriod),manufacture_date,makedate(2022,01,01)),0),cost_price,0))
product A	\$88.27

product_type	=sum(if(inmonths(\$(vPeriod),manufacture_date,makedate(2022,01,01),0),cost_price,0))
product B	\$37.23
product C	\$17.17
product D	\$0.00

De `inmonths()`-functie gebruikt de invoer van de gebruiker als argument om de omvang van het beginsegment van het jaar te definiëren. De functie passeert in de productiedatum van elk van de producten als het tweede argument van de `inmonths()`-functie. Door 1 januari als het derde argument in de `inmonths()`-functie te gebruiken, zullen producten met productiedatums die in het beginsegment van het jaar vallen, een booleaanse waarde voor TRUE retourneren. Daarom zal de somfunctie de kosten van die producten optellen.

### inmonthstodate

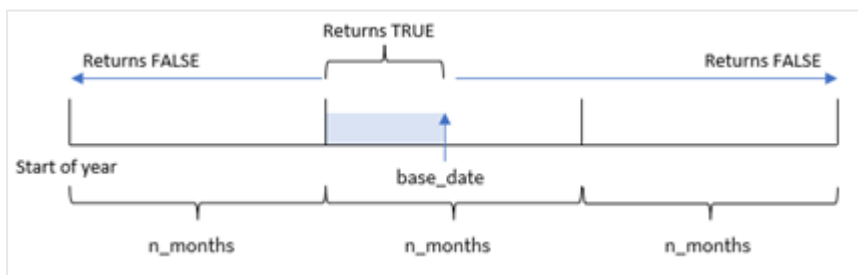
Deze functie bepaalt of een tijdstempel binnen het gedeelte van een periode van de maand, twee maanden, kwartaal, tertiaal of half jaar tot en met de laatste milliseconde van `base_date` valt. Het is tevens mogelijk om te bepalen of de tijdstempel binnen een voorafgaande of volgende tijdsperiode valt.

#### Syntaxis:

```
InMonths (n_months, timestamp, base_date, period_no[, first_month_of_year ])
```

**Retourgegevenstypen:** Booleaanse waarde

*Diagram van inmonthstodate-functie.*



#### Argumenten

Argument	Beschrijving
<b>n_months</b>	Het aantal maanden dat de periode definieert. Een geheel getal of uitdrukking die wordt herleid tot een geheel getal dat een van de volgende waarden moet hebben: 1 (equivalent aan de functie <code>inmonth()</code> ), 2 (2 maanden), 3 (equivalent aan de functie <code>inquarter()</code> ), 4 (tertiaal) of 6 (half jaar).
<b>timestamp</b>	De datum die u wilt vergelijken met <b>base_date</b> .

Argument	Beschrijving
<b>base_date</b>	Datum op basis waarvan de periode wordt geëvalueerd.
<b>period_no</b>	De periode kan worden verschoven met <b>period_no</b> , een geheel getal of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de periode aangeeft die <b>base_date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande perioden aan en positieve waarden geven volgende perioden aan.
<b>first_month_of_year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> .

In de `inmonthstodate()`-functie functioneert de `base_date` als het eindpunt van een specifiek jaarsegment waar het onderdeel van is.

Als het jaar bijvoorbeeld in drie segmenten is verdeeld en de `base_date` 15 mei was, zou iedere tijdstempel tussen het begin van januari en het eind van april een booleaans resultaat `FALSE` retourneren. Datums tussen 1 mei en 15 mei zouden `TRUE` retourneren. De rest van het jaar zou `FALSE` retourneren.

*Diagram van booleaans resultatenbereik van de `inmonthstodate`-functie.*



De volgende segmenten van het jaar zijn beschikbaar in de functie als `n_month`-argumenten:

n_month-argumenten	
Periode	Aantal maanden
maandelijks	1
tweemaandelijks	2
kwartaal	3
viermaandelijks	4
halfjaarlijks	6

### Wanneer gebruiken

De functie `inmonthstodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if` expression. Door de `inmonthstodate()`-functie te gebruiken, kunt u de periode selecteren die u wilt laten evalueren. Bijvoorbeeld: door een invoervariabele te geven waarmee de gebruiker producten kan identificeren die in de maand, het kwartaal of de jaarhelft van een periode tot een bepaalde datum zijn geproduceerd.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>inmonthstodate(4, '01/25/2013', '04/25/2013', 0)</code>	Retourneert True omdat de waarde van de timestamp, 01/25/2013, binnen de periode van vier maanden van 01/01/2013 tot het einde van 04/25/2013 ligt, waarin de waarde van <code>base_date</code> , 04/25/2013 ligt.
<code>inmonthstodate(4, '04/26/2013', '04/25/2006', 0)</code>	Retourneert False, omdat 04/26/2013 buiten dezelfde periode als het voorafgaande voorbeeld ligt.
<code>inmonthstodate(4, '09/25/2005', '02/01/2006', -1)</code>	Retourneert True, omdat de waarde van <code>period_no</code> , -1, de zoekperiode één periode van vier maanden (de waarde van <code>n-months</code> ) terugschuift, waardoor de zoekperiode van 01/09/2005 t/m 02/01/2006 valt.
<code>inmonthstodate(4, '04/25/2006', '06/01/2006', 0, 3)</code>	Retourneert True, omdat de waarde van <code>first_month_of_year</code> is ingesteld op 3, waardoor de zoekperiode van 03/01/2006 t/m 06/01/2006 valt in plaats van van 05/01/2006 t/m 06/01/2006.

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- Een datumveld dat wordt weergegeven in de `DateFormat`-systeemvariabele met de notatie (MM/DD/YYYY).
- Een voorgaande load-instructie met:



- De `inmonthstodate()`-functie die is ingesteld als het veld `in_months_to_date`. Dit bepaalt welke transacties hebben plaatsgevonden in het kwartaal tot 15 mei 2022.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
    *,
    inmonthstodate(3,date,'05/15/2022', 0) as in_months_to_date
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_months\_to\_date

Resultatentabel

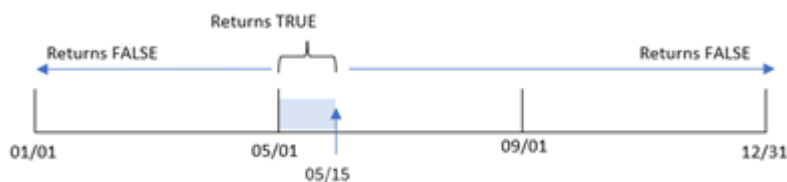
date	in_months_to_date
1/7/2022	0
1/19/2022	0

date	in_months_to_date
2/5/2022	0
2/28/2022	0
3/16/2022	0
4/1/2022	-1
5/7/2022	-1
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

Het veld `in_months_to_date` wordt gemaakt in de vorige load-instructie met behulp van de functie `inmonthstodate()`.

Het eerste opgegeven argument is 3, waarbij het jaar wordt verdeeld in kwartaalsegmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd. Het derde argument is een vastgestelde datum voor 15 mei. Dit is de `base_date` die de eindgrens van het segment definieert. Een `period_no` van 0 is het laatste argument.

*Diagram van `inmonthstodate`-functie zonder aanvullende argumenten.*



Elke transactie die tussen 1 april en 15 mei plaatsvindt, retourneert een booleaans resultaat voor TRUE. Transactiedatums buiten die periode retourneren FALSE.

### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld wordt echter een veld gemaakt, `previous_qtr_to_date`, dat bepaalt of de transacties plaatsvonden in een kwartaal vóór 15 mei.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
inmonthstodate(3,date,'05/15/2022', -1) as previous_qtr_to_date
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_qtr\_to\_date

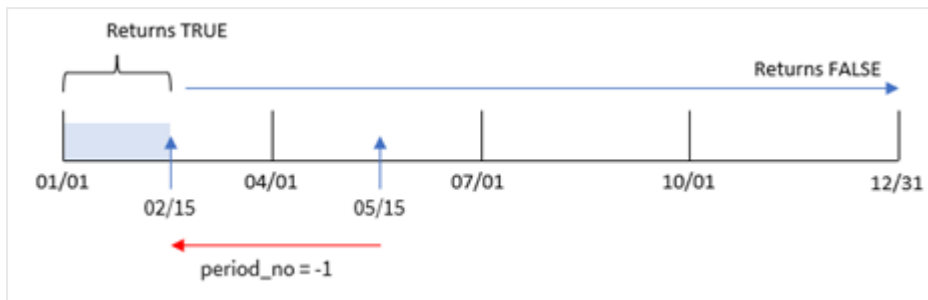
Resultatentabel

date	previous_qtr_to_date
1/7/2022	-1
1/19/2022	-1
2/5/2022	-1
2/28/2022	0
3/16/2022	0
4/1/2022	0
5/7/2022	0
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

Door -1 als het `period_no`-argument te gebruiken in de `inmonthstodate()`-functie, verschuift de functie de grenzen van het jaarsegment in de vergelijking met een kwartaal.

15 mei valt in het tweede kwartaal van het jaar en dus komt het segment aanvankelijk overeen met de ingestelde periode tussen 1 april en 15 mei. Het `period_no`-argument verschuift dit segment met een negatieve periode van drie maanden. De datumgrenzen worden 1 januari t/m 15 februari.

Diagram van functie `inmonthstodate` waarbij `period_no` is ingesteld op -1.



Daarom zal elke transactie die plaatsvindt tussen 1 januari en 15 februari een booleaans resultaat voor TRUE retourneren.

### Voorbeeld 3 - first\_month\_of\_year

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is maart volgens het organisatiebeleid de eerste maand van het boekjaar.

Maak een veld, `in_months_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde kwartaal tot en met 15 mei 2022.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
    *,
    inmonthstodate(3,date,'05/15/2022', 0,3) as in_months_to_date
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_months\_to\_date

Resultatentabel

date	previous_qtr_to_date
1/7/2022	0
1/19/2022	0
2/5/2022	0
2/28/2022	0
3/16/2022	-1
4/1/2022	-1
5/7/2022	-1
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

## 5 Script- en diagramfuncties

Door 3 te gebruiken als het `first_month_of_year`-argument in de `inmonthstodate()`-functie, begint het jaar op 1 maart en wordt het jaar vervolgens in kwartalen verdeeld op basis van het eerste argument dat is opgegeven. Daarom zijn de kwartaalsegmenten:

- Maa-mei
- Jun-aug
- Sep-nov
- Dec-feb

De `base_date` van 15 mei segmenteert vervolgens het kwartaal van maart tot mei door de eindgrens in te stellen op 15 mei.

*Diagram van de `inmonthstodate`-functie waarbij maart is ingesteld als de eerste maand van het jaar.*



Daarom zal elke transactie die plaatsvindt tussen 1 maart en 15 mei een booleaans resultaat voor TRUE retourneren, terwijl transacties met datums buiten deze grenzen een waarde voor FALSE zullen retourneren.

### Voorbeeld 4 - diagramvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld wordt de gegevensverzameling ongewijzigd in de app geladen. De taak bestaat erin een berekening te maken die bepaalt of transacties plaatsvonden in hetzelfde kwartaal als het kwartaal van 15 mei als meting in een diagram in de app.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191, '2/5/2022', 57.42
8192, '3/16/2022', 53.80
8193, '4/1/2022', 82.06
8194, '5/7/2022', 40.39
8195, '5/16/2022', 87.21
8196, '6/15/2022', 95.93
8197, '6/26/2022', 45.89
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

date

Om te berekenen of transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei valt, maakt u de volgende meting:

```
=inmonthstodate(3,date,'05/15/2022', 0)
```

Resultatentabel

date	=inmonthstodate(3,date,'05/15/2022', 0)
1/7/2022	0
1/19/2022	0
2/5/2022	0
2/28/2022	0
3/16/2022	0
4/1/2022	-1
5/7/2022	-1
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0

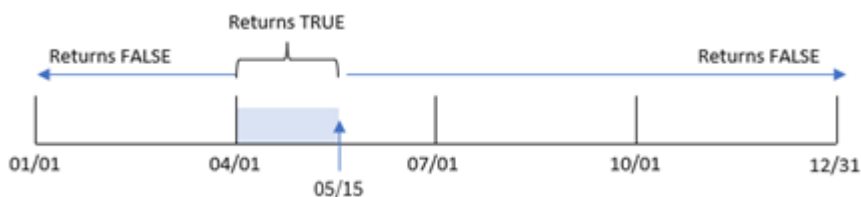


date	=inmonthstodate(3,date,'05/15/2022', 0)
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

De meting 'in\_months\_to\_date' wordt in het diagram gemaakt met behulp van de `inmonthstodate()`-functie.

Het eerste opgegeven argument is 3, waarbij het jaar wordt verdeeld in kwartaalsegmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd. Het derde argument is een vastgestelde datum voor 15 mei. Dit is de `base_date` die de eindgrens van het segment definieert. Een `period_no` van 0 is het laatste argument.

Diagram van de `inmonthstodate`-functie met kwartaalsegmenten.



Elke transactie die tussen 1 april en 15 mei plaatsvindt, retourneert een booleaans resultaat voor TRUE. Transactiedatums buiten dat segment retourneren FALSE.

### Voorbeeld 5 - Scenario

Load-script en resultaten

#### Overzicht

In dit voorbeeld wordt een gegevensset geladen in de tabel 'sales'. De tabel bevat de volgende velden:

- Product-id
- Producttype
- Verkoopdatum
- Verkoopprijs

De eindgebruiker wil een diagram dat de verkoop van producten op producttype weergeeft van producten die in de periode voorafgaand aan 24 december 2022 zijn verkocht. De gebruiker wil de duur van deze periode kunnen definiëren.

### Load-script

```
SET vPeriod = 1;

Products:
Load
*
Inline
[
product_id,product_type,sales_date,sales_price
8188,product A,'9/19/2022',37.23
8189,product D,'10/27/2022',17.17
8190,product C,'10/30/2022',88.27
8191,product B,'10/31/2022',57.42
8192,product D,'11/16/2022',53.80
8193,product D,'11/28/2022',82.06
8194,product A,'12/2/2022',40.39
8195,product B,'12/5/2022',87.21
8196,product C,'12/15/2022',95.93
8197,product B,'12/16/2022',45.89
8198,product C,'12/19/2022',36.23
8199,product D,'12/22/2022',25.66
8200,product D,'12/23/2022',82.77
8201,product A,'12/24/2022',69.98
8202,product A,'12/24/2022',76.11
8203,product B,'12/26/2022',25.12
8204,product B,'12/27/2022',46.23
8205,product B,'12/27/2022',84.21
8206,product C,'12/28/2022',96.24
8207,product D,'12/29/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad.

Aan het begin van het load-script is een variabele, `vPeriod`, gemaakt die wordt gekoppeld aan het besturingselement voor variabelen invoer.

Doe het volgende:

1. Klik in het bedrijfsmiddelenvenster op **Aangepaste objecten**.
2. Selecteer **Qlik Dashboard-bundel** en voeg een **variabele-invoer** toe aan uw werkblad.
3. Voer een titel in voor het diagram.
4. Selecteer onder **Variabele** de optie **vPeriod** als de naam en stel het object in om als een **Vervolgkeuzelijst** te worden weergegeven.
5. Klik onder **Waarden** op **Dynamische waarden**. Voer het volgende in:

= '1~month|2~bi-month|3~quarter|4~tertia1|6~half-year'.

6. Voeg een nieuwe tabel toe aan het werkblad.
7. Voeg in het eigenschappenvenster onder **Gegevens** product\_type toe als een dimensie.
8. Voeg de volgende uitdrukking als een meting toe:  
`=sum(if(inmonthstodate$(vPeriod),sales_date,makedate(2022,12,24),0),sales_price,0))`
9. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

product_type	=sum(if(inmonthstodate\$(vPeriod),sales_date,makedate(2022,12,24),0),sales_price,0))
product A	\$186.48
product B	\$190.52
product C	\$220.43
product D	\$261.46

De `inmonthstodate()`-functie gebruikt de invoer van de gebruiker als argument om de omvang van het beginsegment van het jaar te definiëren.

De functie passeert in de verkoopdatum van elk van de producten als het tweede argument van de `inmonthstodate()`-functie. Door 24 december te gebruiken als het derde argument in de `inmonthstodate()`-functie, retourneren producten met een verkoopdatum die binnen de gedefinieerde periode tot en met 24 december valt, een booleaanse waarde voor TRUE. De somfunctie telt de verkoop van deze producten bij elkaar op.

### inmonthtodate

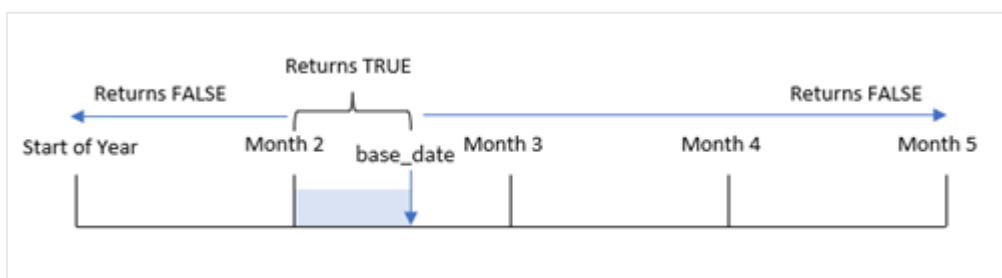
Retourneert True als **date** binnen het gedeelte van de maand ligt dat **basedate** bevat tot en met de laatste milliseconde van **basedate**.

#### Syntaxis:

**InMonthToDate** (timestamp, base\_date, period\_no)

**Retourgegevenstypen:** Booleaanse waarde

*Diagram van inmonthtodate-functie.*



De `inmonthtoday()`-functie identificeert een geselecteerde maand als een segment. De begingrenswaarde is het begin van de maand. De eindgrens kan worden ingesteld als een latere datum in de maand. Dit bepaalt vervolgens of een reeks datums al dan niet in dit segment vallen en retourneert een booleaanse waarde voor TRUE of FALSE.

### Argumenten

Argument	Beschrijving
<code>timestamp</code>	De datum die u wilt vergelijken met <code>base_date</code> .
<code>base_date</code>	Datum op basis waarvan de maand wordt geëvalueerd.
<code>period_no</code>	De maand kan worden verschoven met <code>period_no</code> . <code>period_no</code> is een geheel getal, waarbij de waarde 0 de maand aangeeft die <code>base_date</code> bevat. Negatieve waarden in <code>period_no</code> geven voorafgaande maanden aan en positieve waarden geven volgende maanden aan.

### Wanneer gebruiken

De functie `inmonthtoday()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if` expression. De `inmonthtoday()`-functie retourneert een aggregatie of berekening die afhankelijk is van of een datum plaatsvond in de maand tot en met de desbetreffende datum.

De functie `inmonthtoday()` kan bijvoorbeeld worden gebruikt om alle materieel te identificeren die in een maand tot en met een specifieke datum is geproduceerd.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>inmonthtoday ('01/25/2013', '25/01/2013', 0)</code>	Retourneert True
<code>inmonthtoday ('01/25/2013', '24/01/2013', 0)</code>	Retourneert False
<code>inmonthtoday ('01/25/2013', '28/02/2013', -1)</code>	Retourneert True

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 wordt in de tabel Transactions geladen.
- Een datumveld dat wordt weergegeven in de DateFormat-systeemvariabele (met de notatie MM/DD/YYYY).
- Een voorgaande load-instructie met:
  - De inmonthtodate()-functie die is ingesteld als het veld in\_month\_to\_date. Dit bepaalt welke transacties plaatsvonden tussen 1 juli en 26 juli 2022.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
    *,
    inmonthtodate(date,'07/26/2022', 0) as in_month_to_date
;
Load
*
Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_month\_to\_date

Resultatentabel

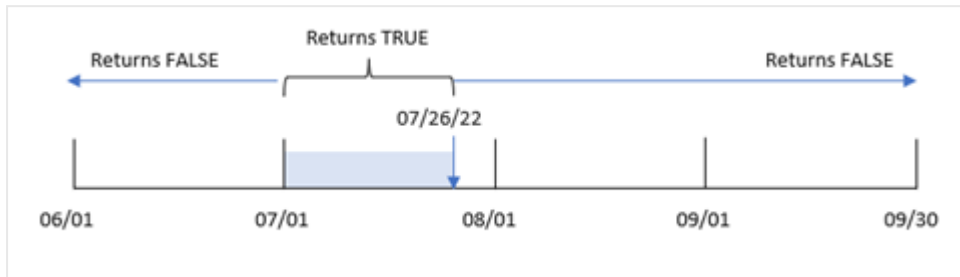
date	in_month_to_date
1/7/2022	0
1/19/2022	0
2/5/2022	0
2/28/2022	0
3/16/2022	0
4/1/2022	0
5/7/2022	0
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	-1
7/22/2022	-1
7/23/2022	-1
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

Het veld `in_month_to_date` wordt gemaakt in de vorige load-instructie met behulp van de functie `inmonthtoday()`.

Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum, 26 juli, die de `base_date` is. Dit `base_date`-argument identificeert welke maand is gesegmenteerd en wat de eindgrens van dat segment is.

Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen maanden voor of na de gesegmenteerde maand vergelijkt.

Diagram van `inmonthtodate`-functie zonder aanvullende argumenten.



Elke transactie die tussen 1 juli en 26 juli plaatsvindt, retourneert een booleaans resultaat TRUE. Alle transacties die in juli na 26 juli plaatsvinden, retourneren een booleaans resultaat FALSE. Hetzelfde geldt voor transacties in een andere maand van het jaar.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld wordt een veld gemaakt, `six_months_prior`, dat bepaalt of de transacties zes hele maanden vóór 1 juli en 26 juli plaatsvonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
*,
inmonthtodate(date,'07/26/2022', -6) as six_months_prior
;
Load
*
Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
```

```
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- six\_months\_prior

Resultatentabel

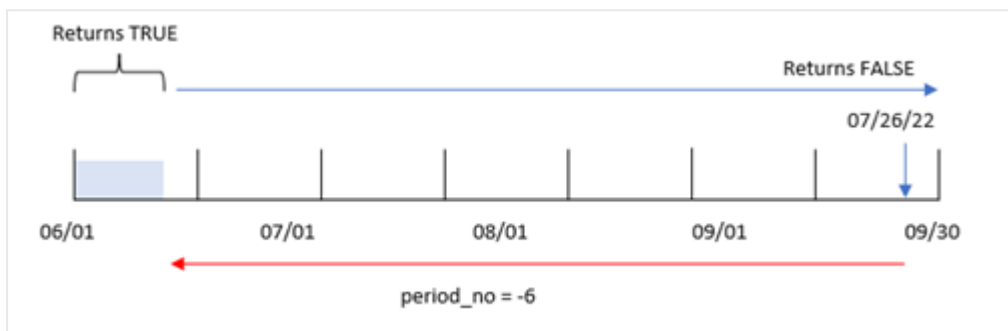
date	six_months_prior
1/7/2022	-1
1/19/2022	-1
2/5/2022	0
2/28/2022	0
3/16/2022	0
4/1/2022	0
5/7/2022	0
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0



date	six_months_prior
10/14/2022	0
10/29/2022	0

Door -6 als het `period_no`-argument te gebruiken in de `inmonthtodate()`-functie, verschuiven de grenzen van het maandsegment in de vergelijking met zes maanden. In eerste instantie is het maandsegment gelijk aan de periode tussen 1 juli en 26 juli. De `period_no` verschuift dit segment vervolgens naar zes maanden eerder en de datumgrenzen worden verschoven en vallen tussen 1 januari en 26 januari.

Diagram van functie `inmonthtodate` waarbij `period_no` is ingesteld op -6.



Elke transactie die tussen 1 januari en 26 januari plaatsvindt, retourneert een booleaans resultaat TRUE.

### Voorbeeld 3 - Diagramvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld wordt de ongewijzigde gegevensverzameling in de app geladen. De taak is om een berekening te maken die bepaalt of transacties plaatsvonden tussen 1 juli en 26 juli als meting in een diagram van de app.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193, '4/1/2022', 82.06
8194, '5/7/2022', 40.39
8195, '5/16/2022', 87.21
8196, '6/15/2022', 95.93
8197, '6/26/2022', 45.89
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

date

Om te berekenen of transacties plaatsgevonden tussen 1 juli en 26 juli, maakt u de volgende meting:

```
=inmonthtodate(date, '07/26/2022', 0)
```

Resultatentabel

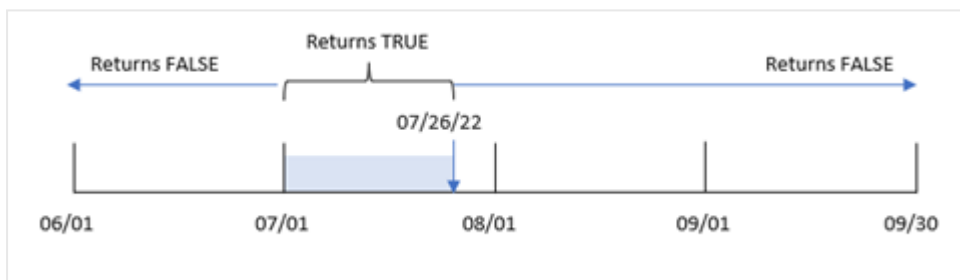
date	=inmonthtodate(date, '07/26/2022', 0)
1/7/2022	0
1/19/2022	0
2/5/2022	0
2/28/2022	0
3/16/2022	0
4/1/2022	0
5/7/2022	0
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	-1
7/22/2022	-1
7/23/2022	-1
7/27/2022	0

date	=inmonthtoday(date,'07/26/2022', 0)
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

De veldmeting `in_month_to_date` wordt in een diagram gemaakt met behulp van de functie `inmonthtoday()`.

Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum, 26 juli, die de `base_date` is. Dit `base_date`-argument identificeert welke maand is gesegmenteerd en wat de eindgrens van dat segment is. Een `period_no` van 0 is het laatste argument. Dit betekent dat de functie geen maanden voor of na de gesegmenteerde maand vergelijkt.

Diagram van `inmonthtoday`-functie zonder aanvullende argumenten.



Elke transactie die tussen 1 juli en 26 juli plaatsvindt, retourneert een booleaans resultaat TRUE. Alle transacties die in juli na 26 juli plaatsvinden, retourneren een booleaans resultaat FALSE. Hetzelfde geldt voor transacties in een andere maand van het jaar.

### Voorbeeld 4 - Scenario

Load-script en resultaten

#### Overzicht

In dit voorbeeld wordt een gegevensset geladen in de tabel 'Products'. De tabel bevat de volgende velden:

- Product-id
- Productiedatum
- Kostprijs

Door een fout in de apparatuur waren producten die in de maand juli 2022 zijn geproduceerd defect. Het probleem is op 27 juli 2022 opgelost.

De eindgebruiker wil graag een diagram dat op maand de status toont van producten die zijn geproduceerd als 'defect' (booleaans WAAR) of 'foutloos' (booleaans NIET WAAR) en de kosten van de producten die in die maand zijn geproduceerd.

### Load-script

```
Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- =monthname(manufacture\_date)
- =if(Inmonthtoday(manufacture\_date,makedate(2022,07,26),0),'Defective','Faultless')

Maak deze meting om de som van de kosten van de producten te berekenen:

```
=sum(cost_price)
```

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

monthname (manufacture_date)	if(Inmonthtodate(manufacture_date,makedate (2022,07,26),0),'Defect','Foutloos')	Sum(cost_ price)
januari 2022	Foutloos	\$54.40
februari 2022	Foutloos	\$145.69
maart 2022	Foutloos	\$53.80
april 2022	Foutloos	\$82.06
mei 2022	Foutloos	\$127.60
juni 2022	Foutloos	\$141.82
juli 2022	Defect	\$144.66
juli 2022	Foutloos	\$69.98
augustus 2022	Foutloos	\$147.46
september 2022	Foutloos	\$84.21
oktober 2022	Foutloos	\$163.91

De functie `inmonthtodate()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product.

Voor de datums die een booleaanse waarde voor TRUE retourneren, wordt het product gemarkeerd als Defect. Voor elk product dat de waarde voor FALSE retourneert, en dus niet op in de maand tot en met 26 juli is geproduceerd, wordt het product als Foutloos gemarkeerd.

### inquarter

Deze functie retourneert True als **timestamp** binnen het kwartaal valt dat **base\_date** bevat.

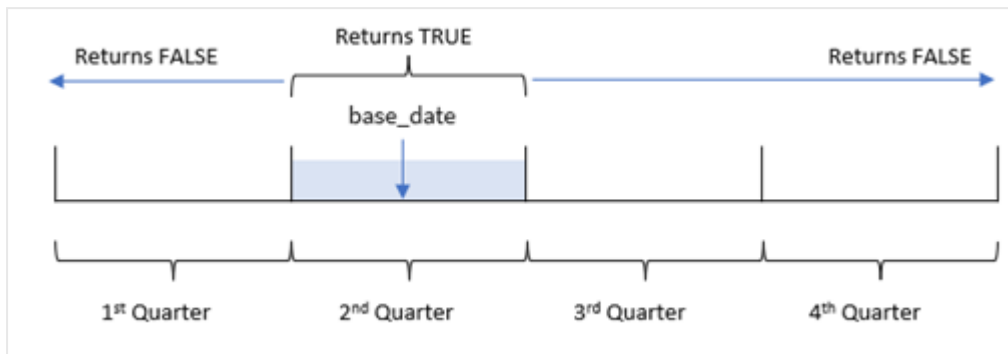
#### Syntaxis:

```
InQuarter (timestamp, base_date, period_no[, first_month_of_year])
```

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

Diagram van het bereik van de `inquarter()`-functie



Met andere woorden: de `inquarter()`-functie verdeelt het jaar in vier gelijke kwartalen tussen 1 januari en 31 december. U kunt het `first_month_of_year`-argument gebruiken om te veranderen welke maand als de eerste maand wordt gezien in uw app. Op basis van dat argument veranderen ook de kwartalen. De `base_date`-functie identificeert welk kwartaal moet worden gebruikt als de vergelijkingsdatum voor de functie. Ten slotte retourneert de functie een booleaans resultaat wanneer de datumwaarden met dat segment worden vergeleken.

### Wanneer gebruiken

De functie `inquarter()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if` expression. Hierdoor wordt een aggregatie of berekening geretourneerd, afhankelijk van of een datum al dan niet in het geselecteerde kwartaal valt.

De `inquarter()`-functie kan bijvoorbeeld worden gebruikt om alle apparatuur te identificeren die in een kwartaalsegment is geproduceerd op basis van de datums waarop de apparatuur is geproduceerd.

### Argumenten

Argument	Beschrijving
<code>timestamp</code>	De datum die u wilt vergelijken met <code>base_date</code> .
<code>base_date</code>	Datum op basis waarvan het kwartaal wordt geëvalueerd.
<code>period_no</code>	Het kwartaal kan worden verschoven met <code>period_no</code> . <code>period_no</code> is een geheel getal, waarbij de waarde 0 het kwartaal aangeeft dat <code>base_date</code> bevat. Negatieve waarden in <code>period_no</code> geven voorafgaande kwartalen aan en positieve waarden geven volgende kwartalen aan.
<code>first_month_of_year</code>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <code>first_month_of_year</code> .

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden first\_month\_of\_year

Month	Waarde
Februari	2
Maart	3
April	4
May	5
Juni	6
Juli	7
Augustus	8
September	9
Oktober	10
November	11
December	12

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>inquarter ('01/25/2013', '01/01/2013', 0)</code>	Retourneert TRUE
<code>inquarter ('01/25/2013', '04/01/2013', 0)</code>	Retourneert FALSE
<code>inquarter ('01/25/2013', '01/01/2013', -1)</code>	Retourneert FALSE
<code>inquarter ('12/25/2012', '01/01/2013', -1)</code>	Retourneert TRUE
<code>inquarter ('01/25/2013', '03/01/2013', 0, 3)</code>	Retourneert FALSE
<code>inquarter ('03/25/2013', '03/01/2013', 0, 3)</code>	Retourneert TRUE

### Voorbeeld 1 - Geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `transactions`.
- Een voorafgaande lading met de `inquarter()`-functie die is ingesteld als het `in_quarter`-veld en die bepaalt welke transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei 2022 valt.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        inquarter (date,'05/15/2022', 0) as in_quarter
    ;

Load
*
Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```



### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

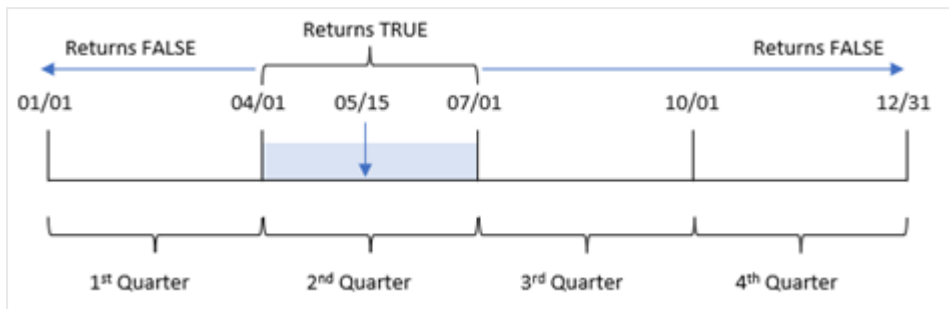
- date
- in\_quarter

Resultatentabel

date	in_quarter
1/7/2022	0
1/19/2022	0
2/5/2022	0
2/28/2022	0
3/16/2022	0
4/1/2022	-1
5/7/2022	-1
5/16/2022	-1
6/15/2022	-1
6/26/2022	-1
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

Het veld `in_quarter` wordt gemaakt in de vorige load-instructie met behulp van de functie `inquarter()`. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 15 mei die identificeert welk kwartaal als de vergelijker moet worden gedefinieerd. Een `period_no` van 0 is het laatste argument en zorgt ervoor dat de `inquarter()`-functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

Diagram van de `inquarter()`-functie met 15 mei als de basisdatum



Elke transactie die tussen 1 april en het eind van 30 juni plaatsvindt, retourneert een booleaans resultaat TRUE.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `transactions`.
- Een voorafgaande lading met de `inquarter()`-functie die is ingesteld als het `previous_quarter`-veld en die bepaalt welke transacties in het kwartaal hebben plaatsgevonden voorafgaand aan het kwartaal waarin 15 mei 2022 valt.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*,
inquarter (date,'05/15/2022', -1) as previous_qtr
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197, '6/26/2022', 45.89
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_qtr

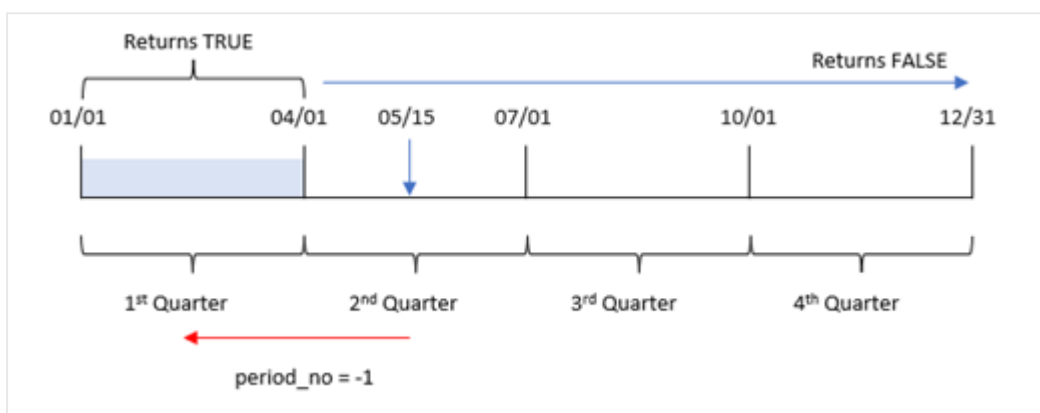
Resultatentabel

date	previous_qtr
1/7/2022	-1
1/19/2022	-1
2/5/2022	-1
2/28/2022	-1
3/16/2022	-1
4/1/2022	0
5/7/2022	0
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0

date	previous_qtr
9/26/2022	0
10/14/2022	0
10/29/2022	0

Door -1 als het `period_no`-argument in de `inquarter()`-functie te gebruiken, verschuift de functie de grenzen van het vergelijkingskwartaal terug met een heel een kwartaal. 15 Mei valt in het tweede kwartaal van het jaar en dus komt het segment aanvankelijk overeen met de ingestelde periode tussen 1 april en 30 juni. De `period_no` compenseert dit segment vervolgens drie maanden eerder, waardoor de datumgrenzen van 1 januari tot 30 maart worden.

Diagram van de `inquarter()`-functie met 15 mei als de basisdatum



Daarom zal elke transactie die plaatsvindt tussen 1 januari en 30 maart een booleaans resultaat voor TRUE retourneren.

### Voorbeeld 3 - first\_month\_of\_year

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `transactions`.
- Een voorafgaande lading met de `inquarter()`-functie die is ingesteld als het `in_quarter`-veld en die bepaalt welke transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei 2022 valt.

In dit voorbeeld is maart volgens het organisatiebeleid echter de eerste maand van het boekjaar.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    inquarter (date,'05/15/2022', 0, 3) as in_quarter
  ;
Load
*
Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_qtr

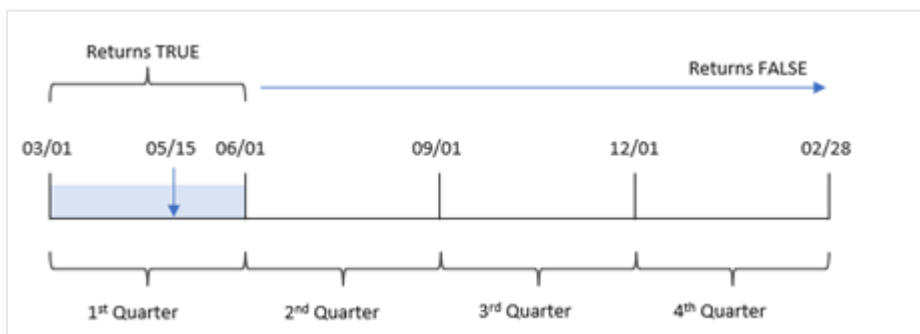
Resultatentabel

date	previous_qtr
1/7/2022	0
1/19/2022	0
2/5/2022	0
2/28/2022	0

date	previous_qtr
3/16/2022	-1
4/1/2022	-1
5/7/2022	-1
5/16/2022	-1
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

Door 3 te gebruiken als het `first_month_of_year`-argument in de `inquarter()`-functie, wordt 1 maart ingesteld als het begin van het jaar en wordt het jaar vervolgens in kwartalen verdeeld. Daardoor zijn de kwartaalsegmenten maa-mei, jun-aug, sep-nov en dec-feb. De `base_date` van 15 mei stelt het maa-mei-kwartaal in als het vergelijkingskwartaal voor de functie.

*Diagram van functie `inquarter()` met maart ingesteld als de eerste maand van het jaar*



Daarom zal elke transactie die plaatsvindt tussen 1 maart en 31 mei een booleaans resultaat voor TRUE retourneren.

### Voorbeeld 4 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `transactions`.
- Een voorafgaande lading met de `inquarter()`-functie die is ingesteld als het `in_quarter`-veld en die bepaalt welke transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei 2022 valt.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- date

Om te berekenen of transacties in hetzelfde kwartaal hebben plaatsgevonden als het kwartaal waarin 15 mei valt, maakt u de volgende meting:

```
=inquarter(date, '05/15/2022', 0)
```

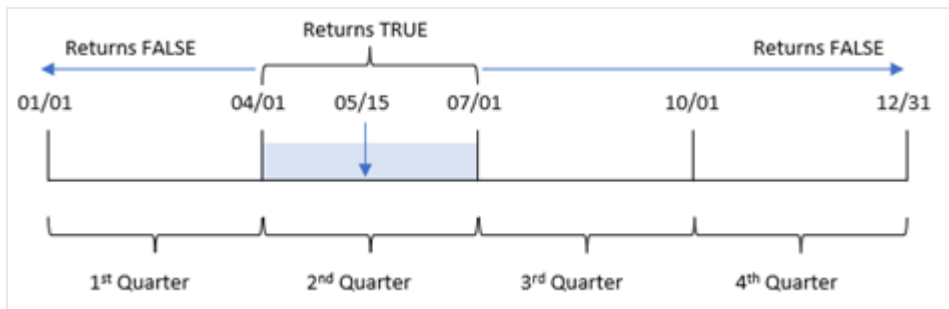
Resultatentabel

date	in_quarter
1/7/2022	0
1/19/2022	0
2/5/2022	0
2/28/2022	0
3/16/2022	0
4/1/2022	-1
5/7/2022	-1
5/16/2022	-1
6/15/2022	-1
6/26/2022	-1
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

De meting `in_quarter` wordt in het diagram gemaakt met behulp van de functie `inquarter()`. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 15 mei die identificeert welk kwartaal als de vergelijker moet worden gedefinieerd. Een `period_no` van 0 is het laatste argument en zorgt ervoor dat de `inquarter()`-functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.



Diagram van de `inquarter()`-functie met 15 mei als de basisdatum



Elke transactie die tussen 1 april en het eind van 30 juni plaatsvindt, retourneert een booleaans resultaat TRUE.

### Voorbeeld 5 - Scenario

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die in de tabel `Products` wordt geladen.
- De tabel bevat de volgende velden:
  - product ID
  - producttype
  - productiedatum
  - cost price

Er is vastgesteld dat producten die in het kwartaal van 15 mei 2022 zijn geproduceerd, defect zijn vanwege een materieelfout. De eindgebruiker wil graag een diagram dat op kwartaalnaam de status toont van welke geproduceerde producten 'defect' of 'foutloos' zijn en de kosten van de producten die in die maand zijn geproduceerd.

#### Load-script

```
Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
```

```
8194, '5/7/2022', 40.39
8195, '5/16/2022', 87.21
8196, '6/15/2022', 95.93
8197, '6/26/2022', 45.89
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

```
=quartername(manufacture_date)
```

Maak de volgende metingen:

- `=if(only(InQuarter(manufacture_date,makedate(2022,05,15),0)), 'Defective', 'Faultless')`, om te identificeren welke van de producten defect zijn en welke foutloos zijn met behulp van de `inquarter()`-functie.
- `=sum(cost_price)`, om de som van de kosten van elk product te tonen.

### Doe het volgende:

1. Stel de **Getalnotatie** van de meting in op **Geld**.
2. Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

quartername (manufacture_date)	=if(only(InQuarter(manufacture_date,makedate(2022,05,15),0)), 'Defective', 'Faultless')	Sum(cost_price)
jan-mrt 2022	Foutloos	253.89
april-jun 2022	Defect	351.48
juli-sept 2022	Foutloos	446.31
okt-dec 2022	Foutloos	163.91

De functie `inquarter()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat in het kwartaal waarin 15 mei valt, wordt geproduceerd, retourneert de `inquarter()`-functie een booleaanse waarde TRUE en markeert het de producten als Defect. Voor elk product dat de waarde FALSE retourneert, en dus niet in dat kwartaal is geproduceerd, wordt het product als Foutloos gemarkeerd.

## inquartertodate

Deze functie retourneert True als **timestamp** binnen het gedeelte van het kwartaal ligt dat **base\_date** bevat tot en met de laatste milliseconde van **base\_date**.

### Syntaxis:

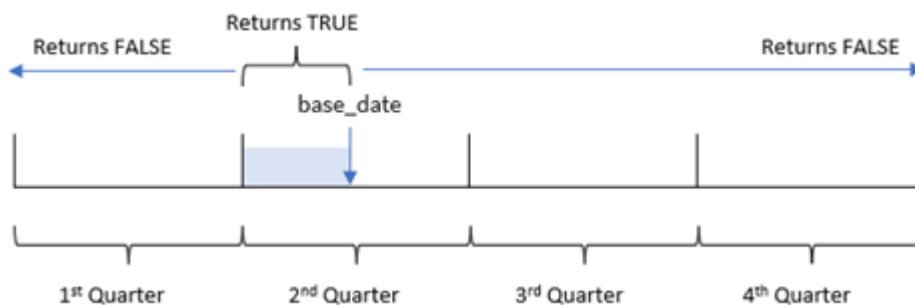
```
InQuarterToDate (timestamp, base_date, period_no [, first_month_of_year])
```

**Retourgegevenstypen:** Booleaanse waarde



In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

Diagram van functie inquartertodate



De functie `inquartertodate()` verdeelt het jaar in vier gelijke kwartalen tussen 1 januari en 31 december (of het door de gebruiker gedefinieerde begin van het jaar en de bijbehorende einddatum). Met de `base_date` zal de functie dan een bepaald kwartaal segmenteren, waarbij de `base_date` zowel welk kwartaal als de maximaal toegestane datum voor dat kwartaalsegment aangeeft. Ten slotte retourneert de functie een booleaans resultaat bij het vergelijken van de voorgeschreven datumwaarden met dat segment.

### Argumenten

Argument	Beschrijving
<b>timestamp</b>	De datum die u wilt vergelijken met <b>base_date</b> .
<b>base_date</b>	Datum op basis waarvan het kwartaal wordt geëvalueerd.
<b>period_no</b>	Het kwartaal kan worden verschoven met <b>period_no</b> . <b>period_no</b> is een geheel getal, waarbij de waarde 0 het kwartaal aangeeft dat <b>base_date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande kwartalen aan en positieve waarden geven volgende kwartalen aan.
<b>first_month_of_year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> .

### Wanneer gebruiken

De functie `inquartertodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if`-uitdrukking. De `inquartertodate()`-functie wordt gebruikt om een aggregatie of berekening te retourneren afhankelijk van of een geëvalueerde datum plaatsvond in het kwartaal tot en met de betreffende datum.

De functie `inquartertodate()` bijvoorbeeld kan worden gebruikt om alle materieel te identificeren die in een kwartaal tot en met een specifieke datum is geproduceerd.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>inquartertodate('01/25/2013', '03/25/2013', 0)</code>	Retourneert <code>TRUE</code> , aangezien de waarde van <code>timestamp</code> , 01/25/2013, binnen de periode ligt van drie maanden vanaf 01/01/2013 tot 03/25/2013, waarin de waarde van <code>base_date</code> , 03/25/2013, ligt.
<code>inquartertodate('04/26/2013', '03/25/2013', 0)</code>	Retourneert <code>FALSE</code> , aangezien 04/26/2013 buiten dezelfde periode als het vorige voorbeeld ligt.
<code>inquartertodate('02/25/2013', '06/09/2013', -1)</code>	Retourneert <code>TRUE</code> , aangezien de waarde van <code>period_no</code> , -1, de zoekperiode één periode van drie maanden terugschuift (een kwartaal van het jaar). Dit maakt de zoekperiode 01/01/2013 tot 03/09/2013.
<code>inquartertodate('03/25/2006', '04/15/2006', 0, 2)</code>	Retourneert <code>TRUE</code> , aangezien de waarde van <code>first_month_of_year</code> is ingesteld op 2, waardoor de zoekperiode 02/01/2006 tot 04/15/2006 wordt in plaats van 04/01/2006 tot 04/15/2006.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.
- Het aanmaken van een veld, in\_quarter\_to\_date, dat bepaalt welke transacties hebben plaatsgevonden in het kwartaal tot 15 mei 2022.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
```

```
    *,
```

```
    inquartertoday(date,'05/15/2022', 0) as in_quarter_to_date
```

```
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

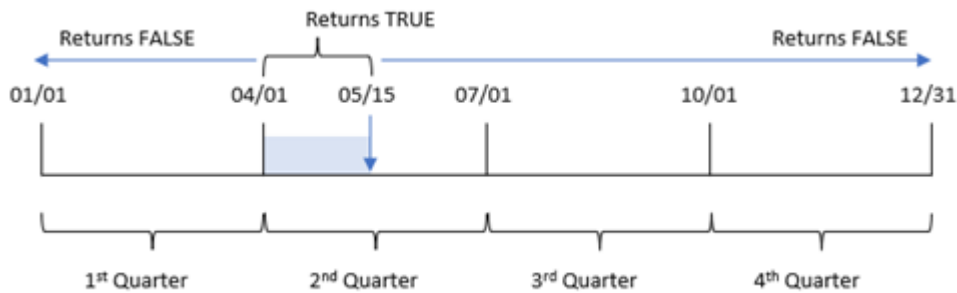
- date
- in\_quarter\_to\_date

Resultatentabel

date	in_quarter_to_date
1/7/2022	0
1/19/2022	0
2/5/2022	0
2/28/2022	0
3/16/2022	0
4/1/2022	-1
5/7/2022	-1
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

Het veld `in_quarter_to_date` wordt gemaakt in de vorige load-instructie met behulp van de functie `inquartertoday()`. Het eerste opgegeven argument geeft aan welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 15 mei, de `base_date` die aangeeft welk kwartaal moet worden gesegmenteerd en de eindgrens van dat segment definieert. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

Diagram van functie `inquartertodate`, geen aanvullende argumenten



Elke transactie die tussen 1 april en 15 mei plaatsvindt, retourneert een booleaans resultaat van `TRUE`. Transactiedatums van 16 mei en later zullen `FALSE` retourneren, net als alle transacties vóór 1 april.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `previous_qtr_to_date`, dat bepaalt welke transacties een volledig kwartaal voor het einde van het kwartaalsegment op 15 mei 2022 hebben plaatsgevonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,  
inquartertodate(date,'05/15/2022', -1) as previous_qtr_to_date  
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197, '6/26/2022', 45.89
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_qtr\_to\_date

Resultatentabel

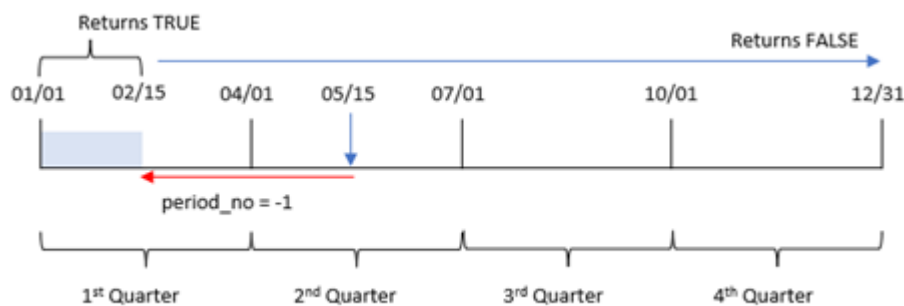
date	previous_qtr_to_date
1/7/2022	-1
1/19/2022	-1
2/5/2022	-1
2/28/2022	0
3/16/2022	0
4/1/2022	0
5/7/2022	0
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0



date	previous_qtr_to_date
9/26/2022	0
10/14/2022	0
10/29/2022	0

Een `period_no`-waarde van -1 geeft aan dat de functie `inquartertoday` () het invoerkwartaalsegment vergelijkt met het voorgaande kwartaal. 15 mei valt in het tweede kwartaal van het jaar, dus het segment komt aanvankelijk overeen met tussen 1 april en 15 mei. De `period_no` compenseert dit segment vervolgens drie maanden eerder, waardoor de datumgrenzen van 1 januari tot 15 februari worden.

*Diagram van functie `inquartertoday`, `period_no` example*



Daarom zal elke transactie die plaatsvindt tussen 1 januari en 15 februari een booleaans resultaat van TRUE retourneren.

### Voorbeeld 3 - `first_month_of_year`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `in_quarter_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde kwartaal tot en met 15 mei 2022.

In dit voorbeeld stellen we maart in als de eerste maand van het fiscale jaar.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,
inquartertodate(date,'05/15/2022', 0,3) as in_quarter_to_date
;
Load
*
Inline
[
id,date,amount
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
8198,'7/9/2022',36.23
8199,'7/22/2022',25.66
8200,'7/23/2022',82.77
8201,'7/27/2022',69.98
8202,'8/2/2022',76.11
8203,'8/8/2022',25.12
8204,'8/19/2022',46.23
8205,'9/26/2022',84.21
8206,'10/14/2022',96.24
8207,'10/29/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_quarter\_to\_date

Resultatentabel

date	in_quarter_to_date
1/7/2022	0
1/19/2022	0
2/5/2022	0
2/28/2022	0
3/16/2022	-1
4/1/2022	-1
5/7/2022	-1

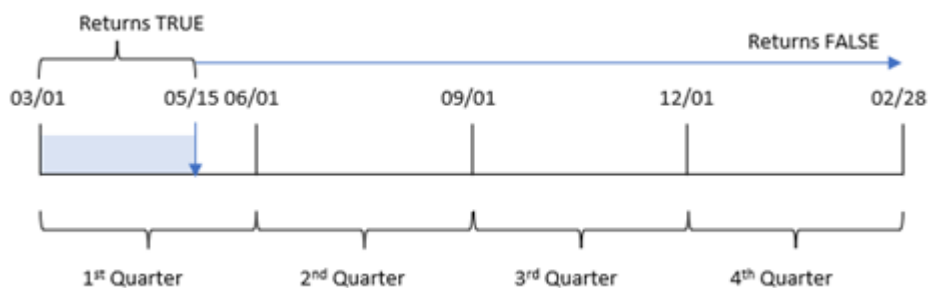
date	in_quarter_to_date
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

Door 3 te gebruiken als het `first_month_of_year`-argument in de `inquartertoday()`-functie, begint de functie het jaar op 1 maart en verdeelt het jaar vervolgens in kwartalen. Daarom zijn de kwartaalsegmenten:

- maart tot mei
- juni tot augustus
- september tot november
- December tot februari

De `base_date` van 15 mei segmenteert vervolgens het kwartaal van maart tot mei door de eindgrens in te stellen op 15 mei.

*Diagram van functie `inquartertoday`, `first_month_of_year` example*



Daarom zal elke transactie die plaatsvindt tussen 1 maart en 15 mei een booleaans resultaat van `TRUE` retourneren, terwijl transacties met datums buiten deze grenzen een waarde van `FALSE` zullen retourneren.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt welke transacties plaatsvonden in hetzelfde kwartaal als 15 mei is gemaakt als een meting in het diagramobject.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/19/2022',37.23
```

```
8189,'1/7/2022',17.17
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:date.

Maak de volgende meting:

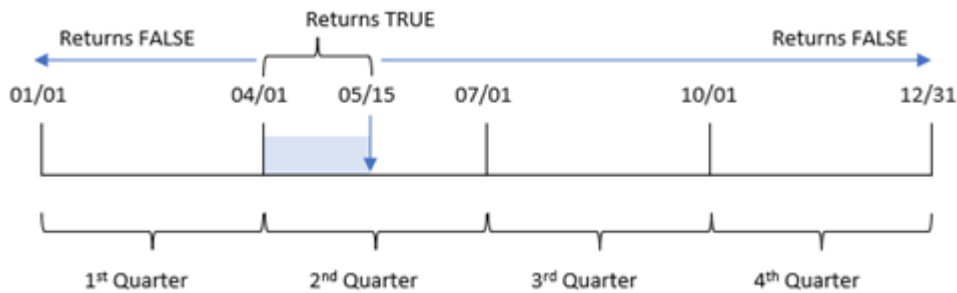
```
=inquartertoday(date,'05/15/2022',0)
```

Resultatentabel

date	=inquartertoday(date,'05/15/2022', 0)
1/7/2022	0
1/19/2022	0
2/5/2022	0
2/28/2022	0
3/16/2022	0
4/1/2022	-1
5/7/2022	-1
5/16/2022	0
6/15/2022	0
6/26/2022	0
7/9/2022	0
7/22/2022	0
7/23/2022	0
7/27/2022	0
8/2/2022	0
8/8/2022	0
8/19/2022	0
9/26/2022	0
10/14/2022	0
10/29/2022	0

De meting `in_quarter_to_date` wordt in een diagramobject gemaakt met behulp van de functie `inquartertoday()`. Het eerste argument is het datumveld dat wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 15 mei, de `base_date` die aangeeft welk kwartaal moet worden gesegmenteerd en de eindgrens van dat segment definieert. Een `van 0` is het laatste argument, wat betekent dat de functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt. `period_no`

Diagram van functie `inquartertoday`, diagramobjectvoorbeeld



Elke transactie die tussen 1 april en 15 mei plaatsvindt, retourneert een booleaans resultaat van `TRUE`. Transacties op 16 mei en later zullen `FALSE` retourneren, net als alle transacties vóór 1 april.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Products`.
- Informatie over product-id, productiedatum en kostprijs.

Op 15 mei 2022 werd tijdens het productieproces een fout in de apparatuur vastgesteld en opgelost. Producten die tot die datum in dat kwartaal zijn geproduceerd, zullen defect zijn. De eindgebruiker wil graag een diagramobject dat op kwartaalnaam de status toont of het product 'defect' of 'foutloos' is en de kosten van producten die zijn geproduceerd in dat kwartaal.

#### Load-script

```
Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
8188,'1/19/2022',37.23
8189,'1/7/2022',17.17
8190,'2/28/2022',88.27
8191,'2/5/2022',57.42
8192,'3/16/2022',53.80
8193,'4/1/2022',82.06
8194,'5/7/2022',40.39
8195,'5/16/2022',87.21
8196,'6/15/2022',95.93
8197,'6/26/2022',45.89
```

```
8198, '7/9/2022', 36.23
8199, '7/22/2022', 25.66
8200, '7/23/2022', 82.77
8201, '7/27/2022', 69.98
8202, '8/2/2022', 76.11
8203, '8/8/2022', 25.12
8204, '8/19/2022', 46.23
8205, '9/26/2022', 84.21
8206, '10/14/2022', 96.24
8207, '10/29/2022', 67.67
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel. Maak een dimensie om de kwartaalnamen weer te geven:  
`=quartername(manufacture_date)`
2. Maak vervolgens een dimensie om te identificeren welke van de producten defect zijn en welke foutloos zijn:  
`=if(inquartertodate(manufacture_date,makedate(2022,05,15),0),'Defective','Faultless')`
3. Maak een meting om de `cost_price` van de producten op te tellen:  
`=sum(cost_price)`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

quartername (manufacture_date)	if(inquartertodate(manufacture_date,makedate (2022,05,15),0),'Defective','Faultless')	Sum(cost_ price)
jan-mrt 2022	Foutloos	\$253.89
april-jun 2022	Foutloos	\$229.03
april-jun 2022	Defect	\$122.45
juli-sept 2022	Foutloos	\$446.31
okt-dec 2022	Foutloos	\$163.91

De functie `inquartertodate()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor degenen die een booleaanse waarde van `TRUE` retourneren, markeert het de producten als 'defective'. Voor elk product dat een waarde retourneert van `FALSE`, en dus niet gemaakt in het kwartaal tot en met 15 mei, markeert het de producten als 'Faultless'.

### inweek

Deze functie retourneert `True` als **timestamp** binnen de week valt die **base\_date** bevat.

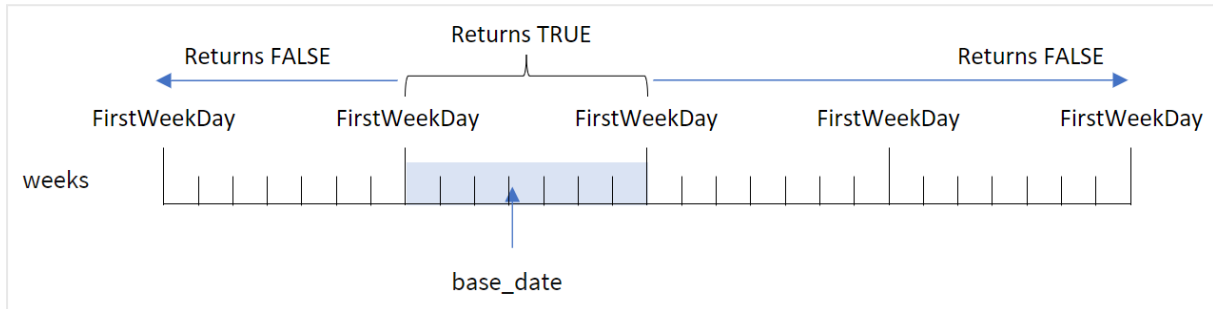
#### Syntaxis:

```
InWeek (timestamp, base_date, period_no[, first_week_day])
```

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

Diagram van het bereik van de `inweek()`-functie



De `inweek()`-functie gebruikt het argument `base_date` om te identificeren in welke zevendaagse periode dag valt. De begindag van de week wordt gebaseerd op de systeemvariabele `FirstWeekDay`. U kunt echter ook de eerste dag van de week wijzigen door het `first_week_day`-argument in de `inweek()`-functie te gebruiken.

Nadat de geselecteerde week is gedefinieerd, retourneert de functie booleaanse resultaten bij het vergelijken van de voorgeschreven datumwaarden met dat weeksegment.

### Wanneer gebruiken

De functie `inweek()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if` expression. De `inweek()`-functie retourneert een aggregatie of berekening die afhankelijk is van of een geëvalueerde datum in de week valt met de geselecteerde datum van het `base_date`-argument.

De `inweek()`-functie kan bijvoorbeeld worden gebruikt om alle materieel te identificeren dat in een specifieke week is geproduceerd.

### Argumenten

Argument	Beschrijving
<b>timestamp</b>	De datum die u wilt vergelijken met <code>base_date</code> .
<b>base_date</b>	Datum op basis waarvan de week wordt geëvalueerd.
<b>period_no</b>	De week kan worden verschoven met <b>period_no</b> . <b>period_no</b> is een geheel getal, waarbij de waarde 0 de week aangeeft die <b>base_date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorgaande weken aan en positieve waarden geven volgende weken aan.
<b>first_week_day</b>	Standaard is de eerste dag van de week zondag (zoals bepaald door de systeemvariabele <code>FirstWeekDay</code> ), beginnend om middernacht tussen zaterdag en zondag. De parameter <b>first_week_day</b> vervangt de variabele <b>FirstWeekDay</b> . Geef aan dat de week op een andere dag begint door een vlag tussen 0 en 6 op te geven.



waarden first\_week\_day

Dag	Waarde
Maandag	0
Dinsdag	1
Woensdag	2
Donderdag	3
Vrijdag	4
Zaterdag	5
Zondag	6

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>inweek ('01/12/2006', '01/14/2006', 0)</code>	Retourneert TRUE
<code>inweek ('01/12/2006', '01/20/2006', 0 )</code>	Retourneert FALSE
<code>inweek ('01/12/2006', '01/14/2006', -1 )</code>	Retourneert FALSE
<code>inweek ('01/07/2006', '01/14/2006', -1)</code>	Retourneert TRUE

Voorbeeld	Resultaat
<code>inweek ( '01/12/2006', '01/09/2006', 0, 3)</code>	Retourneert FALSE omdat <code>first_week_day</code> is gespecificeerd als 3 (donderdag), waardoor 01/12/2006 de eerste dag van de week is na de week waarin 01/09/2006 valt.

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

### Gerelateerde onderwerpen

Onderwerp	Standaardvlag/-waarde	Beschrijving
<i>FirstWeekDay</i> (page 219)	6 / zondag	Definieert de begindag van elke week.

## Voorbeeld 1 - Geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor de maand januari 2022 die wordt geladen in de tabel `Transactions`.
- De systeemvariabele `FirstWeekDay` die is ingesteld op 6 (zondag).
- Een voorgaande lading die het volgende bevat:
  - De `inweek()`-functie die is ingesteld als het `in_week`-veld dat bepaalt welke transacties hebben plaatsgevonden in de week tot en met 14 januari 2022.
  - De `weekday()`-functie, ingesteld als het veld `week_day`, toont welke dag van de week overeenkomt met iedere datum.

### Load-script

```
SET FirstWeekDay=6;  
SET DateFormat='MM/DD/YYYY';
```

`Transactions:`

```
Load  
*,  
weekday(date) as week_day,  
inweek(date,'01/14/2022', 0) as in_week  
;  
  
Load  
*  
Inline  
[  
id,date,amount  
8188,'01/02/2022',37.23
```

```
8189, '01/05/2022', 17.17
8190, '01/06/2022', 88.27
8191, '01/08/2022', 57.42
8192, '01/09/2022', 53.80
8193, '01/10/2022', 82.06
8194, '01/11/2022', 40.39
8195, '01/12/2022', 87.21
8196, '01/13/2022', 95.93
8197, '01/14/2022', 45.89
8198, '01/15/2022', 36.23
8199, '01/16/2022', 25.66
8200, '01/17/2022', 82.77
8201, '01/18/2022', 69.98
8202, '01/26/2022', 76.11
8203, '01/27/2022', 25.12
8204, '01/28/2022', 46.23
8205, '01/29/2022', 84.21
8206, '01/30/2022', 96.24
8207, '01/31/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- week\_day
- in\_week

Resultatentabel

date	week_day	in_week
01/02/2022	Zo	0
01/05/2022	Wo	0
01/06/2022	Do	0
01/08/2022	Za	0
01/09/2022	Zo	-1
01/10/2022	Ma	-1
01/11/2022	Di	-1
01/12/2022	Wo	-1
01/13/2022	Do	-1
01/14/2022	Vr	-1
01/15/2022	Za	-1
01/16/2022	Zo	0

date	week_day	in_week
01/17/2022	Ma	0
01/18/2022	Di	0
01/26/2022	Wo	0
01/27/2022	Do	0
01/28/2022	Vr	0
01/29/2022	Za	0
01/30/2022	Zo	0
01/31/2022	Ma	0

Het veld `in_week` wordt gemaakt in de vorige load-instructie met behulp van de functie `inweek()`. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum, 14 januari, wat de `base_date` is. Het `base_date`-argument werkt met de `FirstweekDay`-systeemvariabele om de week te bepalen waarmee moet worden vergeleken. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

De systeemvariabele `FirstweekDay` bepaalt dat weken beginnen op een zondag en eindigen op een zaterdag. Daarom zou januari worden opgedeeld in weken volgens het onderstaande diagram, waarbij de datums tussen 9 en 15 januari de geldige periode vormen voor de `inweek()` berekening:

Diagram van kalender met het bereik van de `inweek()`-functie gemarkeerd

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Elke transactie die tussen 9 januari en 15 januari plaatsvindt, retourneert een booleaans resultaat voor TRUE.

### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- De systeemvariabele `Firstweekday` die is ingesteld op 6 (zondag).
- Een voorgaande lading die het volgende bevat:
  - De `inweek()`-functie die is ingesteld als het `prev_week`-veld dat bepaalt welke transacties hebben plaatsgevonden in de week voorafgaand aan de week waarin 14 januari 2022 valt.

- De `weekday()`-functie, ingesteld als het veld `week_day`, toont welke dag van de week overeenkomt met iedere datum.

### Load-script

```
SET FirstWeekDay=6;
SET DateFormat='MM/DD/YYYY';

Transactions:
    Load
        *,
        weekday(date) as week_day,
        inweek(date, '01/14/2022', -1) as prev_week
    ;
Load
*
Inline
[
id,date,amount
8188,'01/02/2022',37.23
8189,'01/05/2022',17.17
8190,'01/06/2022',88.27
8191,'01/08/2022',57.42
8192,'01/09/2022',53.80
8193,'01/10/2022',82.06
8194,'01/11/2022',40.39
8195,'01/12/2022',87.21
8196,'01/13/2022',95.93
8197,'01/14/2022',45.89
8198,'01/15/2022',36.23
8199,'01/16/2022',25.66
8200,'01/17/2022',82.77
8201,'01/18/2022',69.98
8202,'01/26/2022',76.11
8203,'01/27/2022',25.12
8204,'01/28/2022',46.23
8205,'01/29/2022',84.21
8206,'01/30/2022',96.24
8207,'01/31/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week_day`
- `prev_week`

Resultatentabel

<b>date</b>	<b>week_day</b>	<b>prev_week</b>
01/02/2022	Zo	-1
01/05/2022	Wo	-1
01/06/2022	Do	-1
01/08/2022	Za	-1
01/09/2022	Zo	0
01/10/2022	Ma	0
01/11/2022	Di	0
01/12/2022	Wo	0
01/13/2022	Do	0
01/14/2022	Vr	0
01/15/2022	Za	0
01/16/2022	Zo	0
01/17/2022	Ma	0
01/18/2022	Di	0
01/26/2022	Wo	0
01/27/2022	Do	0
01/28/2022	Vr	0
01/29/2022	Za	0
01/30/2022	Zo	0
01/31/2022	Ma	0

Door -1 als het `period_no`-argument in de `inweek()`-functie te gebruiken, verschuift de functie de grenzen van de vergelijkingsweek terug met zeven dagen. Met een `period_no` van 0, zou de week tussen 9 en 15 januari vallen. Maar in dit voorbeeld verschuift de `period_no` van -1 de begin- en eindgrens van dit segment terug met één week. De datumgrenzen worden 2 t/m 8 januari.

Diagram van kalender met het bereik van de `inweek()`-functie gemarkeerd

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Daarom zal elke transactie die plaatsvindt tussen 2 januari en 8 januari een booleaans resultaat voor TRUE retourneren.

### Voorbeeld 3 - first\_week\_day

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- De systeemvariabele `FirstWeekDay` die is ingesteld op 6 (zondag).
- Een voorgaande lading die het volgende bevat:
  - De `inweek()`-functie die is ingesteld als het `in_week`-veld dat bepaalt welke transacties hebben plaatsgevonden in de week tot en met 14 januari 2022.



- De `weekday()`-functie, ingesteld als het veld `week_day`, toont welke dag van de week overeenkomt met iedere datum.

### Load-script

```
SET FirstWeekDay=6;
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    weekday(date) as week_day,
    inweek(date,'01/14/2022', 0, 0) as in_week
  ;

Load
*
Inline
[
id,date,amount
8188,'01/02/2022',37.23
8189,'01/05/2022',17.17
8190,'01/06/2022',88.27
8191,'01/08/2022',57.42
8192,'01/09/2022',53.80
8193,'01/10/2022',82.06
8194,'01/11/2022',40.39
8195,'01/12/2022',87.21
8196,'01/13/2022',95.93
8197,'01/14/2022',45.89
8198,'01/15/2022',36.23
8199,'01/16/2022',25.66
8200,'01/17/2022',82.77
8201,'01/18/2022',69.98
8202,'01/26/2022',76.11
8203,'01/27/2022',25.12
8204,'01/28/2022',46.23
8205,'01/29/2022',84.21
8206,'01/30/2022',96.24
8207,'01/31/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week_day`
- `in_week`

Resultatentabel

<b>date</b>	<b>week_day</b>	<b>in_week</b>
01/02/2022	Zo	0
01/05/2022	Wo	0
01/06/2022	Do	0
01/08/2022	Za	0
01/09/2022	Zo	0
01/10/2022	Ma	-1
01/11/2022	Di	-1
01/12/2022	Wo	-1
01/13/2022	Do	-1
01/14/2022	Vr	-1
01/15/2022	Za	-1
01/16/2022	Zo	-1
01/17/2022	Ma	0
01/18/2022	Di	0
01/26/2022	Wo	0
01/27/2022	Do	0
01/28/2022	Vr	0
01/29/2022	Za	0
01/30/2022	Zo	0
01/31/2022	Ma	0

Door 0 te gebruiken als het `first_week_day`-argument in de `inweek()`-functie, wordt de `FirstweekDay`-systeemvariabele vervangen en maandag ingesteld als de eerste dag van de week.

Diagram van kalender met het bereik van de `inweek()`-functie gemarkeerd

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Daarom zal elke transactie die plaatsvindt tussen 10 en 16 januari een booleaans resultaat voor TRUE retourneren.

### Voorbeeld 4 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. Maak een meting in de resultatentabel om te bepalen welke transacties plaatsvonden in de week van 14 januari 2022.

#### Load-script

```
SET FirstweekDay=6;  
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
*
Inline
[
id,date,amount
8188,'01/02/2022',37.23
8189,'01/05/2022',17.17
8190,'01/06/2022',88.27
8191,'01/08/2022',57.42
8192,'01/09/2022',53.80
8193,'01/10/2022',82.06
8194,'01/11/2022',40.39
8195,'01/12/2022',87.21
8196,'01/13/2022',95.93
8197,'01/14/2022',45.89
8198,'01/15/2022',36.23
8199,'01/16/2022',25.66
8200,'01/17/2022',82.77
8201,'01/18/2022',69.98
8202,'01/26/2022',76.11
8203,'01/27/2022',25.12
8204,'01/28/2022',46.23
8205,'01/29/2022',84.21
8206,'01/30/2022',96.24
8207,'01/31/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- date

Maak de volgende metingen:

- =inweek (date, '01/14/2022', 0) om te bepalen of transacties in de week van 14 januari plaatsvonden.
- =weekday(date) om aan te geven welke dag van de week overeenkomt met elke datum.

Resultatentabel

date	week_day	=inweek (date,'01/14/2022',0)
01/02/2022	Zo	0
01/05/2022	Wo	0
01/06/2022	Do	0
01/08/2022	Za	0
01/09/2022	Zo	-1
01/10/2022	Ma	-1

date	week_day	=inweek (date,'01/14/2022',0)
01/11/2022	Di	-1
01/12/2022	Wo	-1
01/13/2022	Do	-1
01/14/2022	Vr	-1
01/15/2022	Za	-1
01/16/2022	Zo	0
01/17/2022	Ma	0
01/18/2022	Di	0
01/26/2022	Wo	0
01/27/2022	Do	0
01/28/2022	Vr	0
01/29/2022	Za	0
01/30/2022	Zo	0
01/31/2022	Ma	0

De meting `in_week` wordt in het diagram gemaakt met behulp van de `inweek()`-functie. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum, 14 januari, wat de `base_date` is. Het `base_date`-argument werkt met de `FirstweekDay`-systeemvariabele om de week te bepalen waarmee moet worden vergeleken. Een `period_no` van 0 is het laatste argument.

De systeemvariabele `FirstweekDay` bepaalt dat weken beginnen op een zondag en eindigen op een zaterdag. Daarom zou januari worden opgedeeld in weken volgens het onderstaande diagram, waarbij de datums tussen 9 en 15 januari de geldige periode vormen voor de `inweek()` berekening:

Diagram van kalender met het bereik van de `inweek()`-functie gemarkeerd

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Elke transactie die tussen 9 januari en 15 januari plaatsvindt, retourneert een booleaans resultaat voor TRUE.

### Voorbeeld 5 - Scenario

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die in de tabel `Products` wordt geladen.
- De tabel bevat de volgende velden:
  - product ID
  - producttype
  - productiedatum
  - cost price

Er is vastgesteld dat producten die in de week van 12 januari zijn geproduceerd defect zijn vanwege een materiële fout. De eindgebruiker wil graag een diagram dat op week de status toont welke geproduceerde producten de status 'defect' of 'foutloos' hadden en de kosten van de producten die in die week zijn geproduceerd.

### Load-script

```
Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
8188,'01/02/2022',37.23
8189,'01/05/2022',17.17
8190,'01/06/2022',88.27
8191,'01/08/2022',57.42
8192,'01/09/2022',53.80
8193,'01/10/2022',82.06
8194,'01/11/2022',40.39
8195,'01/12/2022',87.21
8196,'01/13/2022',95.93
8197,'01/14/2022',45.89
8198,'01/15/2022',36.23
8199,'01/16/2022',25.66
8200,'01/17/2022',82.77
8201,'01/18/2022',69.98
8202,'01/26/2022',76.11
8203,'01/27/2022',25.12
8204,'01/28/2022',46.23
8205,'01/29/2022',84.21
8206,'01/30/2022',96.24
8207,'01/31/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- =weekname(manufacture\_date)

Maak de volgende metingen:

- =if(only(inweek(manufacture\_date,makedate(2022,01,12),0)), 'defective', 'Faultless'), om te identificeren welke van de producten defect zijn en welke foutloos zijn met behulp van de inweek()-functie.
- =sum(cost\_price), om de som van de kosten van elk product te tonen.

### Doe het volgende:

1. Stel de **Getalnotatie** van de meting in op **Geld**.
2. Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

weekname (manufacture_date)	=if(only(inweek(manufacture_date,makedate(2022,01,12),0)), 'Defect','Foutloos')	Sum(cost_price)
2022/02	Foutloos	200.09
2022/03	Defect	441.51
2022/04	Foutloos	178.41
2022/05	Foutloos	231.67
2022/06	Foutloos	163.91

De functie `inweek()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat op 12 januari is geproduceerd, retourneert de `inweek()`-functie een booleaanse waarde voor TRUE en markeert de producten als 'Defect'. Voor elk product dat de waarde voor FALSE retourneert, en dus niet in die week is geproduceerd, wordt het product als 'Foutloos' gemarkeerd.

### inweektodate

Deze functie retourneert True als **timestamp** binnen het gedeelte van de week ligt dat **base\_date** bevat tot en met de laatste milliseconde van **base\_date**.

#### Syntaxis:

```
InWeekToDate (timestamp, base_date, period_no [, first_week_day])
```

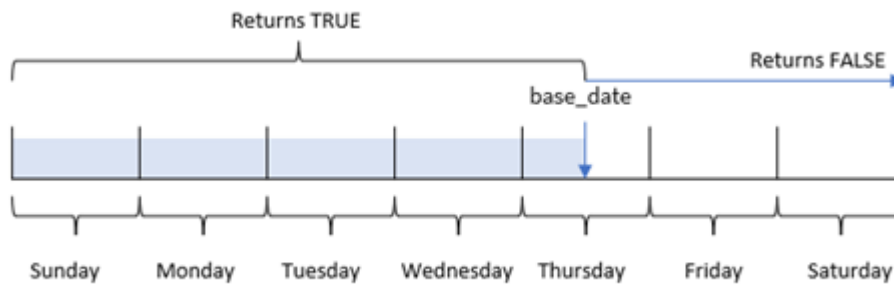
**Retourgegevenstypen:** Booleaanse waarde



*In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.*



Diagram van functie `inweektodate`



De functie `inweektodate()` gebruikt de parameter `base_date` om een maximale grensdatum van een weeksegment te identificeren, evenals de bijbehorende datum voor het begin van de week, die is gebaseerd op de systeemvariabele `FirstWeekDay` (of door de gebruiker gedefinieerde parameter `first_week_day`). Zodra dit weeksegment is gedefinieerd, retourneert de functie booleaanse resultaten bij het vergelijken van de voorgeschreven datumwaarden met dat segment.

### Wanneer gebruiken

De functie `inweektodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if`-uitdrukking. Dit retourneert een aggregatie of berekening afhankelijk van of een geëvalueerde datum plaatsvond in de week tot en met een bepaalde datum.

De functie `inweektodate()` kan bijvoorbeeld worden gebruikt om alle verkopen te berekenen die in een bepaalde week tot een bepaalde datum zijn gedaan.

#### Argumenten

Argument	Beschrijving
<code>timestamp</code>	De datum die u wilt vergelijken met <code>base_date</code> .
<code>base_date</code>	Datum op basis waarvan de week wordt geëvalueerd.
<code>period_no</code>	De week kan worden verschoven met <code>period_no</code> . <code>period_no</code> is een geheel getal, waarbij de waarde 0 de week aangeeft die <code>base_date</code> bevat. Negatieve waarden in <code>period_no</code> geven voorgaande weken aan en positieve waarden geven volgende weken aan.
<code>first_week_day</code>	Standaard is de eerste dag van de week zondag (zoals bepaald door de systeemvariabele <code>FirstWeekDay</code> ), beginnend om middernacht tussen zaterdag en zondag. De parameter <code>first_week_day</code> vervangt de variabele <code>FirstWeekDay</code> . Geef aan dat de week op een andere dag begint door een vlag tussen 0 en 6 op te geven.  Gebruik voor een week die begint op maandag en eindigt op zondag een vlag van 0 voor maandag, 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag.

### Voorbeelden van functies

Voorbeeld	Interactie
inweektodate ( '01/12/2006' , '01/12/2006' , 0 )	Retourneert TRUE.
inweektodate ( '01/12/2006' , '01/11/2006' , 0 )	Retourneert FALSE.
inweektodate ( '01/12/2006' , '01/18/2006' , -1 )	Retourneert FALSE. omdat period_no is opgegeven als -1, vormt 01/11/2006 de daadwerkelijke datum waartegen timestamp wordt afgezet.
inweektodate ( '01/11/2006' , '01/12/2006' , 0, 3 )	Retourneert FALSE, omdat first_week_day is opgegeven als 3 (donderdag), waardoor 01/12/2006 de eerste dag van de week wordt na de week die 01/12/2006 bevat.

Deze onderwerpen kunnen u helpen bij het werken met deze functie:

### Gerelateerde onderwerpen

Onderwerp	Standaardvlag/-waarde	Beschrijving
<i>FirstWeekDay</i> (page 219)	6 / zondag	Definieert de begindag van elke week.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht SET DateFormat in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor de maand januari 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het gegevensveld in de `TimestampFormat='M/D/YYYY h:mm:ss[.fff]'`-indeling.
- Het aanmaken van een veld, `in_week_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in de week tot en met 14 januari 2022.
- Het maken van een extra veld met de naam `weekday` met behulp van de functie `weekday()`. Dit nieuwe veld is gemaakt om aan te geven welke dag van de week overeenkomt met elke datum.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff]';
SET FirstWeekDay=6;
Transactions:
    Load
        *,
        weekday(date) as week_day,
        inweektodate(date,'01/14/2022', 0) as in_week_to_date
    ;
Load
*
Inline
[
id,date,amount
8188,'2022-01-02 12:22:06',37.23
8189,'2022-01-05 01:02:30',17.17
8190,'2022-01-06 15:36:20',88.27
8191,'2022-01-08 10:58:35',57.42
8192,'2022-01-09 08:53:32',53.80
8193,'2022-01-10 21:13:01',82.06
8194,'2022-01-11 00:57:13',40.39
8195,'2022-01-12 09:26:02',87.21
8196,'2022-01-13 15:05:09',95.93
8197,'2022-01-14 18:44:57',45.89
8198,'2022-01-15 06:10:46',36.23
8199,'2022-01-16 06:39:27',25.66
8200,'2022-01-17 10:44:16',82.77
8201,'2022-01-18 18:48:17',69.98
8202,'2022-01-26 04:36:03',76.11
8203,'2022-01-27 08:07:49',25.12
8204,'2022-01-28 12:24:29',46.23
8205,'2022-01-30 11:56:56',84.21
8206,'2022-01-30 14:40:19',96.24
8207,'2022-01-31 05:28:21',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week_day`

- `in_week_to_date`

Resultatentabel

<code>date</code>	<code>week_day</code>	<code>in_week_to_date</code>
2022-01-02 12:22:06	Zo	0
2022-01-05 01:02:30	Wo	0
2022-01-06 15:36:20	Do	0
2022-01-08 10:58:35	Za	0
2022-01-09 08:53:32	Zo	-1
2022-01-10 21:13:01	Ma	-1
2022-01-11 00:57:13	Di	-1
2022-01-12 09:26:02	Wo	-1
2022-01-13 15:05:09	Do	-1
2022-01-14 18:44:57	Vr	-1
2022-01-15 06:10:46	Za	0
2022-01-16 06:39:27	Zo	0
2022-01-17 10:44:16	Ma	0
2022-01-18 18:48:17	Di	0
2022-01-26 04:36:03	Wo	0
2022-01-27 08:07:49	Do	0
2022-01-28 12:24:29	Vr	0
2022-01-30 11:56:56	Zo	0
2022-01-30 14:40:19	Zo	0
2022-01-31 05:28:21	Ma	0

Het veld `in_week_to_date` wordt gemaakt in de vorige load-instructie met behulp van de functie `in_week_to_date` `dateinweektodate()`. Het eerste verstrekte argument geeft aan welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 14 januari, de `base_date` die aangeeft welke week moet worden gesegmenteerd en de eindgrens van dat segment definieert. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

De systeemvariabele `Firstweekday` bepaalt dat weken beginnen op een zondag en eindigen op een zaterdag. Daarom zou januari worden opgedeeld in weken volgens het onderstaande diagram, waarbij de datums tussen 9 en 14 januari de geldige periode vormen voor de berekening `inweektodate()`:

*Kalenderdiagram met transactiedatums die een booleaans resultaat van WAAR zouden opleveren*

Sun	Mon	Tue	Wed	Thur	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Elke transactie die tussen 9 januari en 14 januari plaatsvindt, retourneert een booleaans resultaat van TRUE. Transacties voor en na de datums retourneren een booleaans resultaat van FALSE.

### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `prev_week_to_date`, dat bepaalt welke transacties een volledige week voor het einde van het weeksegment op 14 januari 2022 hebben plaatsgevonden.
- Het maken van een extra veld met de naam `weekday` met behulp van de functie `weekday()`. Dit is om aan te geven welke dag van de week overeenkomt met elke datum.

#### Load-script

```
SET FirstWeekDay=6;
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff]';
Transactions:
  Load
    *,
    weekday(date) as week_day,
    inweektodate(date, '01/14/2022', -1) as prev_week_to_date
  ;
Load
*
Inline
[
id,date,amount
```

```
8188, '2022-01-02 12:22:06', 37.23
8189, '2022-01-05 01:02:30', 17.17
8190, '2022-01-06 15:36:20', 88.27
8191, '2022-01-08 10:58:35', 57.42
8192, '2022-01-09 08:53:32', 53.80
8193, '2022-01-10 21:13:01', 82.06
8194, '2022-01-11 00:57:13', 40.39
8195, '2022-01-12 09:26:02', 87.21
8196, '2022-01-13 15:05:09', 95.93
8197, '2022-01-14 18:44:57', 45.89
8198, '2022-01-15 06:10:46', 36.23
8199, '2022-01-16 06:39:27', 25.66
8200, '2022-01-17 10:44:16', 82.77
8201, '2022-01-18 18:48:17', 69.98
8202, '2022-01-26 04:36:03', 76.11
8203, '2022-01-27 08:07:49', 25.12
8204, '2022-01-28 12:24:29', 46.23
8205, '2022-01-30 11:56:56', 84.21
8206, '2022-01-30 14:40:19', 96.24
8207, '2022-01-31 05:28:21', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- week\_day
- prev\_week\_to\_date

Resultatentabel

date	week_day	prev_week_to_date
2022-01-02 12:22:06	Zo	-1
2022-01-05 01:02:30	Wo	-1
2022-01-06 15:36:20	Do	-1
2022-01-08 10:58:35	Za	0
2022-01-09 08:53:32	Zo	0
2022-01-10 21:13:01	Ma	0
2022-01-11 00:57:13	Di	0
2022-01-12 09:26:02	Wo	0
2022-01-13 15:05:09	Do	0
2022-01-14 18:44:57	Vr	0
2022-01-15 06:10:46	Za	0

date	week_day	prev_week_to_date
2022-01-16 06:39:27	Zo	0
2022-01-17 10:44:16	Ma	0
2022-01-18 18:48:17	Di	0
2022-01-26 04:36:03	Wo	0
2022-01-27 08:07:49	Do	0
2022-01-28 12:24:29	Vr	0
2022-01-30 11:56:56	Zo	0
2022-01-30 14:40:19	Zo	0
2022-01-31 05:28:21	Ma	0

Een `period_no`-waarde van -1 geeft aan dat de functie `inweektodate` () het invoerkwartaalsegment vergelijkt met de voorgaande week. Het weeksegment komt aanvankelijk overeen met tussen 9 januari en 14 januari. De `period_no` verschuift vervolgens zowel de begin- als de eindgrens van dit segment naar een week eerder, waardoor de datumgrenzen van 2 januari tot 7 januari worden.

*Kalenderdiagram met transactiedatums die een Booleaans resultaat van TRUE zouden opleveren*

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Daarom zal elke transactie die plaatsvindt tussen 2 januari en 8 januari (niet inclusief 8 januari zelf) een booleaans resultaat van `TRUE` retourneren.

### Voorbeeld 3 - `first_week_day`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `in_week_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in de week tot en met 14 januari 2022.
- Het maken van een extra veld met de naam `weekday` met behulp van de functie `weekday()`. Dit is om aan te geven welke dag van de week overeenkomt met elke datum.

In dit voorbeeld beschouwen we maandag als de eerste dag van de week.

### Load-script

```
SET FirstWeekDay=6;
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff]';

Transactions:
    Load
        *,
        weekday(date) as week_day,
        inweektodate(date,'01/14/2022', 0, 0) as in_week_to_date
    ;
Load
*
Inline
[
id,date,amount
8188,'2022-01-02 12:22:06',37.23
8189,'2022-01-05 01:02:30',17.17
8190,'2022-01-06 15:36:20',88.27
8191,'2022-01-08 10:58:35',57.42
8192,'2022-01-09 08:53:32',53.80
8193,'2022-01-10 21:13:01',82.06
8194,'2022-01-11 00:57:13',40.39
8195,'2022-01-12 09:26:02',87.21
8196,'2022-01-13 15:05:09',95.93
8197,'2022-01-14 18:44:57',45.89
8198,'2022-01-15 06:10:46',36.23
8199,'2022-01-16 06:39:27',25.66
8200,'2022-01-17 10:44:16',82.77
8201,'2022-01-18 18:48:17',69.98
8202,'2022-01-26 04:36:03',76.11
8203,'2022-01-27 08:07:49',25.12
8204,'2022-01-28 12:24:29',46.23
8205,'2022-01-30 11:56:56',84.21
8206,'2022-01-30 14:40:19',96.24
8207,'2022-01-31 05:28:21',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `week_day`



- `in_week_to_date`

Resultatentabel

<b>date</b>	<b>week_day</b>	<b>in_week_to_date</b>
2022-01-02 12:22:06	Zo	0
2022-01-05 01:02:30	Wo	0
2022-01-06 15:36:20	Do	0
2022-01-08 10:58:35	Za	0
2022-01-09 08:53:32	Zo	0
2022-01-10 21:13:01	Ma	-1
2022-01-11 00:57:13	Di	-1
2022-01-12 09:26:02	Wo	-1
2022-01-13 15:05:09	Do	-1
2022-01-14 18:44:57	Vr	-1
2022-01-15 06:10:46	Za	0
2022-01-16 06:39:27	Zo	0
2022-01-17 10:44:16	Ma	0
2022-01-18 18:48:17	Di	0
2022-01-26 04:36:03	Wo	0
2022-01-27 08:07:49	Do	0
2022-01-28 12:24:29	Vr	0
2022-01-30 11:56:56	Zo	0
2022-01-30 14:40:19	Zo	0
2022-01-31 05:28:21	Ma	0

Door 0 te gebruiken als het `first_week_day`-argument in de `inweektoday()`-functie, vervangt het functieargument de `FirstweekDay`-systeemvariabele en stelt het maandag in als de eerste dag van de week.

*Kalenderdiagram met transactiedatums die een booleaans resultaat van WAAR zouden opleveren*

Mon	Tue	Wed	Thu	Fri	Sat	Sun
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	17
24	25	26	27	28	29	30
31						

Daarom zal elke transactie die plaatsvindt tussen 10 en 14 januari een booleaans resultaat van TRUE retourneren, terwijl transacties met datums buiten deze grenzen een waarde van FALSE zullen retourneren.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt welke transacties plaatsvonden in de week tot en met 14 januari 2022 is gemaakt als een meting in het diagramobject.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'2022-01-02 12:22:06',37.23
```

```
8189,'2022-01-05 01:02:30',17.17
```

```
8190,'2022-01-06 15:36:20',88.27
```

```
8191,'2022-01-08 10:58:35',57.42
```

```
8192,'2022-01-09 08:53:32',53.80
```

```
8193,'2022-01-10 21:13:01',82.06
```

```
8194,'2022-01-11 00:57:13',40.39
```

```
8195,'2022-01-12 09:26:02',87.21
```

```
8196, '2022-01-13 15:05:09', 95.93
8197, '2022-01-14 18:44:57', 45.89
8198, '2022-01-15 06:10:46', 36.23
8199, '2022-01-16 06:39:27', 25.66
8200, '2022-01-17 10:44:16', 82.77
8201, '2022-01-18 18:48:17', 69.98
8202, '2022-01-26 04:36:03', 76.11
8203, '2022-01-27 08:07:49', 25.12
8204, '2022-01-28 12:24:29', 46.23
8205, '2022-01-30 11:56:56', 84.21
8206, '2022-01-30 14:40:19', 96.24
8207, '2022-01-31 05:28:21', 67.67
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:  
date.
2. Om te berekenen of transacties in dezelfde week tot 14 januari hebben plaatsgevonden, maakt u de volgende meting aan:  
=inweektodate(date, '01/14/2022', 0)
3. Om aan te geven welke dag van de week overeenkomt met elke datum, maakt u een aanvullende meting:  
=weekday(date)

Resultatentabel

date	week_day	in_week_to_date
2022-01-02 12:22:06	Zo	0
2022-01-05 01:02:30	Wo	0
2022-01-06 15:36:20	Do	0
2022-01-08 10:58:35	Za	0
2022-01-09 08:53:32	Zo	-1
2022-01-10 21:13:01	Ma	-1
2022-01-11 00:57:13	Di	-1
2022-01-12 09:26:02	Wo	-1
2022-01-13 15:05:09	Do	-1
2022-01-14 18:44:57	Vr	-1
2022-01-15 06:10:46	Za	0
2022-01-16 06:39:27	Zo	0

date	week_day	in_week_to_date
2022-01-17 10:44:16	Ma	0
2022-01-18 18:48:17	Di	0
2022-01-26 04:36:03	Wo	0
2022-01-27 08:07:49	Do	0
2022-01-28 12:24:29	Vr	0
2022-01-30 11:56:56	Zo	0
2022-01-30 14:40:19	Zo	0
2022-01-31 05:28:21	Ma	0

Het veld `in_week_to_date` wordt in een diagramobject gemaakt als een meting met behulp van de functie `inweektodate()`. Het eerste opgegeven argument geeft aan welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 14 januari, de `base_date` die aangeeft welke week moet worden gesegmenteerd en de eindgrens van dat segment definieert. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen kwartalen voor of na het gesegmenteerde kwartaal vergelijkt.

De systeemvariabele `Firstweekday` bepaalt dat weken beginnen op een zondag en eindigen op een zaterdag. Daarom zou januari worden opgedeeld in weken volgens het onderstaande diagram, waarbij de datums tussen 9 en 14 januari de geldige periode vormen voor de `inweektodate()` berekening:

*Kalenderdiagram met transactiedatums die een Booleaans resultaat van WAAR zouden opleveren*

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Elke transactie die tussen 9 januari en 14 januari plaatsvindt, retourneert een booleaans resultaat van `TRUE`. Transacties voor en na de datums retourneren een booleaans resultaat van `FALSE`.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Products`.
- Informatie over product-id, productiedatum en kostprijs.

Er is vastgesteld dat producten die in de week van 12 januari zijn geproduceerd defect zijn vanwege een materieelfout. Op 13 januari is het probleem opgelost. De eindgebruiker wil graag een diagramobject dat op week de status toont of de geproduceerde producten 'defect' of 'foutloos' zijn en de kosten van de producten die in die week zijn geproduceerd.

#### Load-script

```
Products:
Load
*
Inline
[
product_id,manufacture_date,cost_price
8188,'2022-01-02 12:22:06',37.23
8189,'2022-01-05 01:02:30',17.17
8190,'2022-01-06 15:36:20',88.27
8191,'2022-01-08 10:58:35',57.42
8192,'2022-01-09 08:53:32',53.80
8193,'2022-01-10 21:13:01',82.06
8194,'2022-01-11 00:57:13',40.39
8195,'2022-01-12 09:26:02',87.21
8196,'2022-01-13 15:05:09',95.93
8197,'2022-01-14 18:44:57',45.89
8198,'2022-01-15 06:10:46',36.23
8199,'2022-01-16 06:39:27',25.66
8200,'2022-01-17 10:44:16',82.77
8201,'2022-01-18 18:48:17',69.98
8202,'2022-01-26 04:36:03',76.11
8203,'2022-01-27 08:07:49',25.12
8204,'2022-01-28 12:24:29',46.23
8205,'2022-01-30 11:56:56',84.21
8206,'2022-01-30 14:40:19',96.24
8207,'2022-01-31 05:28:21',67.67
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel. Maak een dimensie om de weeknamen weer te geven:  
`=weekname(manufacture_date)`
2. Maak vervolgens een dimensie om te identificeren welke van de producten defect zijn en welke foutloos zijn:  
`=if(inweektodate(manufacture_date,makedate(2022,01,12),0),'Defective','Faultless')`
3. Maak een meting om de `cost_price` van de producten op te tellen:  
`=sum(cost_price)`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

<code>weekname(manufacture_date)</code>	<code>if(inweektodate(manufacture_date,makedate(2022,01,12),0),'Defective','Faultless')</code>	<code>Sum(cost_price)</code>
2022/02	Foutloos	\$200.09
2022/03	Defect	\$263.46
2022/03	Foutloos	\$178.05
2022/04	Foutloos	\$178.41
2022/05	Foutloos	\$147.46
2022/06	Foutloos	\$248.12

De functie `inweektodate()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor degenen die een booleaanse waarde van `TRUE` retourneren, markeert het de producten als 'defective'. Voor elk product dat een waarde retourneert van `FALSE`, en dus niet gemaakt in de week tot en met 12 januari, markeert het de producten als 'Faultless'.

### inyear

Deze functie retourneert `True` als **timestamp** binnen het jaar valt dat **base\_date** bevat.

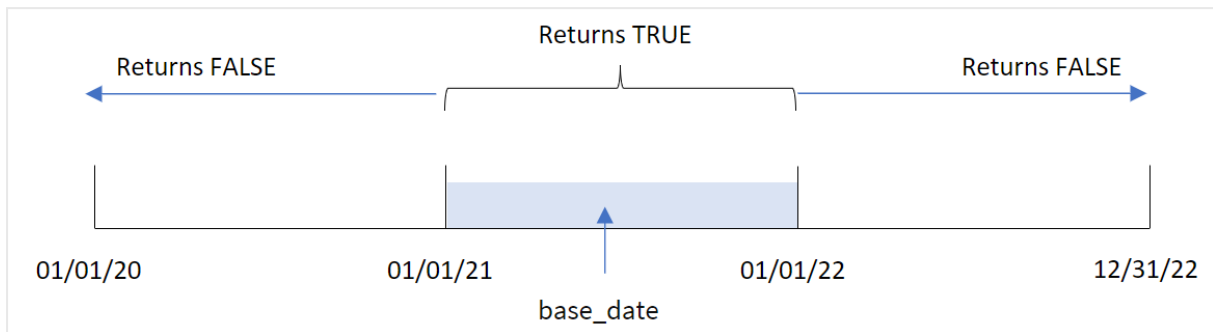
#### Syntaxis:

```
InYear (timestamp, base_date, period_no [, first_month_of_year])
```

**Retourgegevenstypen:** Booleaanse waarde

In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.

Diagram van het bereik van de `inyear()`-functie



De `inyear()`-functie retourneert een booleaans resultaat wanneer de geselecteerde datumwaarde worden vergeleken met een jaar dat is gedefinieerd door de `base_date`.

### Wanneer gebruiken

De functie `inyear()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if expression`. Hierdoor wordt een aggregatie of berekening geretourneerd, afhankelijk van het feit of een geëvalueerde datum in het desbetreffende jaar plaatsvond. De `inyear()`-functie kan bijvoorbeeld worden gebruikt om alle verkoop te identificeren die in een bepaald jaar heeft plaatsgevonden.

### Argumenten

Argument	Beschrijving
<b>timestamp</b>	De datum die u wilt vergelijken met <code>base_date</code> .
<b>base_date</b>	Datum op basis waarvan het jaar wordt geëvalueerd.
<b>period_no</b>	Het jaar kan worden verschoven met <b>period_no</b> . <b>period_no</b> is een geheel getal, waarbij de waarde 0 het jaar aangeeft dat <code>base_date</code> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande jaren aan en positieve waarden geven volgende jaren aan.
<b>first_month_of_year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> .

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden `first_month_of_year`

Month	Waarde
Februari	2
Maart	3
April	4
May	5

Month	Waarde
Juni	6
Juli	7
Augustus	8
September	9
Oktober	10
November	11
December	12

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>inyear ('01/25/2013', '01/01/2013', 0 )</code>	Retourneert TRUE
<code>inyear ('01/25/2012', '01/01/2013', 0)</code>	Retourneert FALSE
<code>inyear ('01/25/2013', '01/01/2013', -1)</code>	Retourneert FALSE
<code>inyear ('01/25/2012', '01/01/2013', -1 )</code>	Retourneert TRUE
<code>inyear ('01/25/2013', '01/01/2013', 0, 3)</code>	Retourneert TRUE De waarde van de <code>base_date</code> en de <code>first_month_of_year</code> specificeren dat een tijdstempel tussen 01/03/2012 en 02/28/2013 moet vallen
<code>inyear ('03/25/2013', '07/01/2013', 0, 3 )</code>	Retourneert TRUE



### Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in de tabel Transactions.
- Een voorafgaande lading met de `inyear()`-functie die is ingesteld als het `in_year`-veld en die bepaalt welke transacties in hetzelfde jaar hebben plaatsgevonden als het jaar waarin 26 juli 2021 valt.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
Transactions:
  Load
    *,
    inyear(date,'07/26/2021', 0) as in_year
  ;
Load
*
Inline
[
id,date,amount
8188,'01/13/2020',37.23
8189,'02/26/2020',17.17
8190,'03/27/2020',88.27
8191,'04/16/2020',57.42
8192,'05/21/2020',53.80
8193,'08/14/2020',82.06
8194,'10/07/2020',40.39
8195,'12/05/2020',87.21
8196,'01/22/2021',95.93
8197,'02/03/2021',45.89
8198,'03/17/2021',36.23
8199,'04/23/2021',25.66
8200,'05/04/2021',82.77
8201,'06/30/2021',69.98
8202,'07/26/2021',76.11
8203,'12/27/2021',25.12
8204,'06/06/2022',46.23
8205,'07/18/2022',84.21
8206,'11/14/2022',96.24
8207,'12/12/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

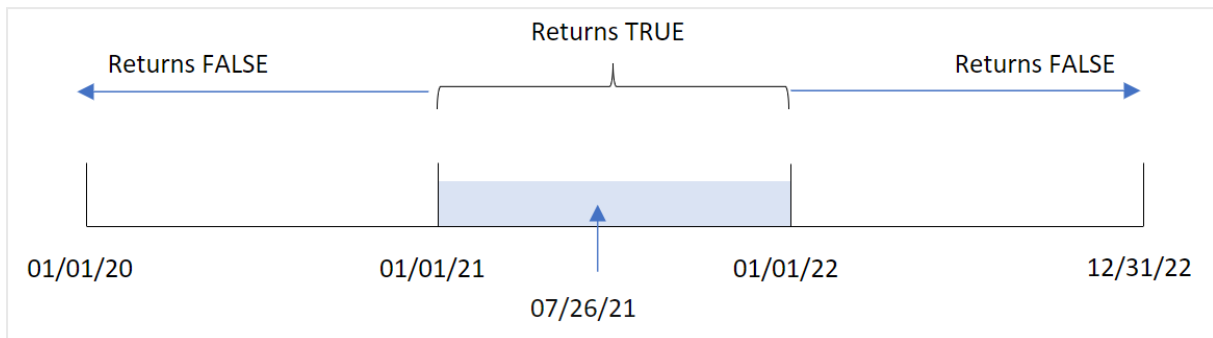
- date
- in\_year

Resultatentabel

date	in_year
01/13/2020	0
02/26/2020	0
03/27/2020	0
04/16/2020	0
05/21/2020	0
08/14/2020	0
10/07/2020	0
12/05/2020	0
01/22/2021	-1
02/03/2021	-1
03/17/2021	-1
04/23/2021	-1
05/04/2021	-1
06/30/2021	-1
07/26/2021	-1
12/27/2021	-1
06/06/2022	0
07/18/2022	0
11/14/2022	0
12/12/2022	0

Het veld `in_year` wordt gemaakt in de vorige load-instructie met behulp van de functie `inyear()`. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 26 juli 2021. Dit is de `base_date` die het jaar bepaalt waarmee de vergelijking moet worden gedaan. Een `period_no` van 0 is het laatste argument, wat betekent dat de `inyear()`-functie geen jaren voor of na het jaar vergelijkt.

Diagram van het bereik van de `inyear()`-functie met 26 juli als de basisdatum



Een transactie die in 2021 plaatsvindt, retourneert een booleaans resultaat voor TRUE.

### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in de tabel Transactions.
- Een voorafgaande lading met de `inyear()`-functie die is ingesteld als het `previous_year`-veld en die bepaalt welke transacties in hetzelfde jaar hebben plaatsgevonden als het jaar waarin 26 juli 2021 valt.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
Transactions:
  Load
    *,
    inyear(date,'07/26/2021', -1) as previous_year
  ;
Load
*
Inline
[
id,date,amount
8188,'01/13/2020',37.23
8189,'02/26/2020',17.17
8190,'03/27/2020',88.27
8191,'04/16/2020',57.42
8192,'05/21/2020',53.80
8193,'08/14/2020',82.06
8194,'10/07/2020',40.39
8195,'12/05/2020',87.21
8196,'01/22/2021',95.93
```

```
8197, '02/03/2021', 45.89
8198, '03/17/2021', 36.23
8199, '04/23/2021', 25.66
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '12/27/2021', 25.12
8204, '06/06/2022', 46.23
8205, '07/18/2022', 84.21
8206, '11/14/2022', 96.24
8207, '12/12/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_year

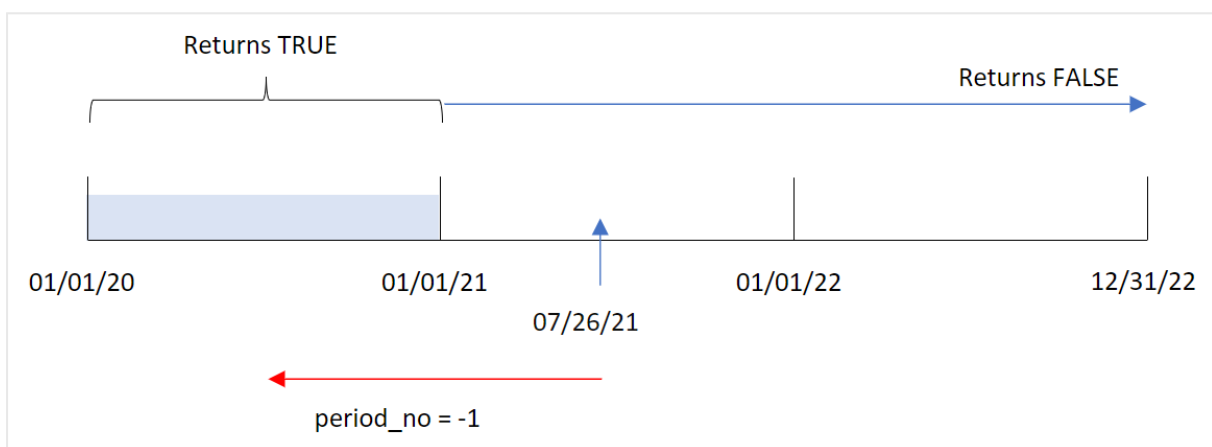
Resultatentabel

date	previous_year
01/13/2020	-1
02/26/2020	-1
03/27/2020	-1
04/16/2020	-1
05/21/2020	-1
08/14/2020	-1
10/07/2020	-1
12/05/2020	-1
01/22/2021	0
02/03/2021	0
03/17/2021	0
04/23/2021	0
05/04/2021	0
06/30/2021	0
07/26/2021	0
12/27/2021	0
06/06/2022	0

date	previous_year
07/18/2022	0
11/14/2022	0
12/12/2022	0

Met -1 als het `period_no`-argument in de `inyear()`-functie verschuiven de grenzen van het jaar waarmee de vergelijking moet worden gedaan met een heel jaar. 2021 wordt aanvankelijk geïdentificeerd als het jaar voor de vergelijking. De `period_no` verschuift het jaar waarmee moet worden vergeleken met één jaar waardoor 2020 het jaar is voor de vergelijking.

Diagram van het bereik van de `inyear()`-functie met het argument voor de `period_no` ingesteld op -1



Daarom zal een transactie die in 2020 plaatsvindt, een booleaans resultaat voor TRUE retourneren.

### Voorbeeld 3 - first\_month\_of\_year

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in de tabel Transactions.
- Een voorafgaande lading met de `inyear()`-functie die is ingesteld als het `in_year`-veld en die bepaalt welke transacties in hetzelfde jaar hebben plaatsgevonden als het jaar waarin 26 juli 2021 valt.

In dit voorbeeld is maart volgens het organisatiebeleid echter de eerste maand van het financiële boekjaar.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
Transactions:
    Load
        *,
        inyear(date,'07/26/2021', 0, 3) as in_year
    ;
Load
*
Inline
[
id,date,amount
8188,'01/13/2020',37.23
8189,'02/26/2020',17.17
8190,'03/27/2020',88.27
8191,'04/16/2020',57.42
8192,'05/21/2020',53.80
8193,'08/14/2020',82.06
8194,'10/07/2020',40.39
8195,'12/05/2020',87.21
8196,'01/22/2021',95.93
8197,'02/03/2021',45.89
8198,'03/17/2021',36.23
8199,'04/23/2021',25.66
8200,'05/04/2021',82.77
8201,'06/30/2021',69.98
8202,'07/26/2021',76.11
8203,'12/27/2021',25.12
8204,'06/06/2022',46.23
8205,'07/18/2022',84.21
8206,'11/14/2022',96.24
8207,'12/12/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_year

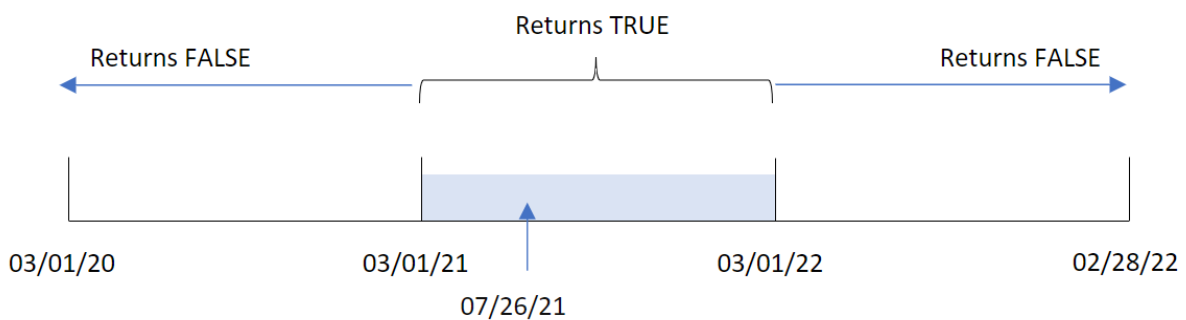
Resultatentabel

date	in_year
01/13/2020	0
02/26/2020	0
03/27/2020	0
04/16/2020	0

date	in_year
05/21/2020	0
08/14/2020	0
10/07/2020	0
12/05/2020	0
01/22/2021	0
02/03/2021	0
03/17/2021	-1
04/23/2021	-1
05/04/2021	-1
06/30/2021	-1
07/26/2021	-1
12/27/2021	-1
06/06/2022	0
07/18/2022	0
11/14/2022	0
12/12/2022	0

Met 3 als het `first_month_of_year`-argument in de `inyear()`-functie begint het jaar op 1 maart en eindigt het jaar aan het eind van februari.

*Diagram van het tijdsbereik van de `inyear()`-functie met maart ingesteld als de eerste maand van het jaar*



Daarom zal elke transactie die plaatsvindt tussen 1 maart 2021 en 1 maart 2022 een booleaans resultaat voor TRUE retourneren.

### Voorbeeld 4 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die bepaalt of transacties al dan niet hebben plaatsgevonden in hetzelfde jaar als 26 juli 2021, wordt als meting in een diagramobject van de applicatie gemaakt.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
Transactions:
Load
*
Inline
[
id,date,amount
8188,'01/13/2020',37.23
8189,'02/26/2020',17.17
8190,'03/27/2020',88.27
8191,'04/16/2020',57.42
8192,'05/21/2020',53.80
8193,'08/14/2020',82.06
8194,'10/07/2020',40.39
8195,'12/05/2020',87.21
8196,'01/22/2021',95.93
8197,'02/03/2021',45.89
8198,'03/17/2021',36.23
8199,'04/23/2021',25.66
8200,'05/04/2021',82.77
8201,'06/30/2021',69.98
8202,'07/26/2021',76.11
8203,'12/27/2021',25.12
8204,'06/06/2022',46.23
8205,'07/18/2022',84.21
8206,'11/14/2022',96.24
8207,'12/12/2022',67.67
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- date

Om te berekenen of transacties in hetzelfde jaar hebben plaatsgevonden als het jaar waarin 26 juli 2021 valt, maakt u de volgende meting:



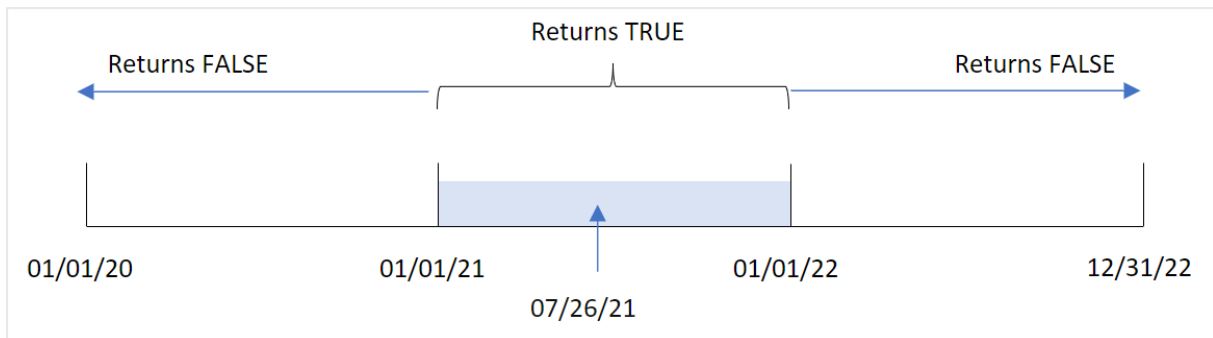
- `=inyear(date, '07/26/2021', 0)`

Resultatentabel

date	=inyear(date,'07/26/2021',0)
01/13/2020	0
02/26/2020	0
03/27/2020	0
04/16/2020	0
05/21/2020	0
08/14/2020	0
10/07/2020	0
12/05/2020	0
01/22/2021	-1
02/03/2021	-1
03/17/2021	-1
04/23/2021	-1
05/04/2021	-1
06/30/2021	-1
07/26/2021	-1
12/27/2021	-1
06/06/2022	0
07/18/2022	0
11/14/2022	0
12/12/2022	0

Het veld `in_year` wordt in een diagram gemaakt met behulp van de functie `inyear()`. Het eerste argument identificeert welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 26 juli 2021. Dit is de `base_date` die het jaar bepaalt waarmee de vergelijking moet worden gedaan. Een `period_no` van 0 is het laatste argument, wat betekent dat de `inyear()`-functie geen jaren voor of na het jaar vergelijkt.

Diagram van het bereik van de `inyear()`-functie met 27 juli als de basisdatum



Een transactie die in 2021 plaatsvindt, retourneert een booleaans resultaat voor TRUE.

### Voorbeeld 5 - Scenario

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die in de tabel `Products` wordt geladen.
- De tabel bevat de volgende velden:
  - product ID
  - producttype
  - productiedatum
  - cost price

De eindgebruiker wil een diagramobject dat, per producttype, de kosten weergeeft van de producten die in 2021 zijn geproduceerd.

#### Load-script

```
Products:
Load
*
Inline
[
product_id,product_type,manufacture_date,cost_price
8188,product A,'01/13/2020',37.23
8189,product B,'02/26/2020',17.17
8190,product B,'03/27/2020',88.27
8191,product C,'04/16/2020',57.42
8192,product D,'05/21/2020',53.80
8193,product D,'08/14/2020',82.06
8194,product C,'10/07/2020',40.39
8195,product B,'12/05/2020',87.21
```

```
8196,product A,'01/22/2021',95.93
8197,product B,'02/03/2021',45.89
8198,product C,'03/17/2021',36.23
8199,product C,'04/23/2021',25.66
8200,product B,'05/04/2021',82.77
8201,product D,'06/30/2021',69.98
8202,product D,'07/26/2021',76.11
8203,product D,'12/27/2021',25.12
8204,product C,'06/06/2022',46.23
8205,product C,'07/18/2022',84.21
8206,product A,'11/14/2022',96.24
8207,product B,'12/12/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

- product\_type

Om de som van elk product dat in 2021 is geproduceerd te berekenen, maakt u de volgende meting:

- =sum(if(InYear(manufacture\_date,makedate(2021,01,01),0),cost\_price,0))

### Doe het volgende:

1. Stel de **Getalnotatie** van de meting in op **Geld**.
2. Onder **Uiterlijk** schakelt u **Totaal** uit.

Resultatentabel

product_type	=sum(if(InYear(manufacture_date,makedate(2021,01,01),0),cost_price,0))
product A	\$95.93
product B	\$128.66
product C	\$61.89
product D	\$171.21

De functie `inyear()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat in 2021 is geproduceerd, retourneert de `inyear()`-functie een booleaanse waarde voor TRUE en wordt de som van de `cost_price` weergegeven.

### inyeartodate

Deze functie retourneert True als **timestamp** binnen het gedeelte van het jaar ligt dat **base\_date** bevat tot en met de laatste milliseconde van **base\_date**.

### Syntaxis:

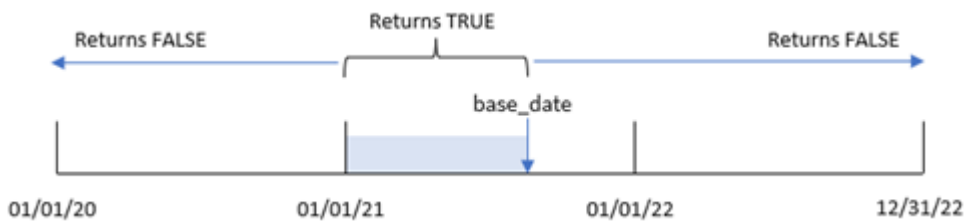
```
InYearToDate (timestamp, base_date, period_no[, first_month_of_year])
```

**Retourgegevenstypen:** Booleaanse waarde



In Qlik Sense wordt de booleaanse waarde *Waar* vertegenwoordigd door *-1* en de waarde *Onwaar* door *0*.

Diagram van functie `inyeartodate`



De functie `inyeartodate()` segmenteert een bepaald gedeelte van het jaar met de `base_date`, waarbij de maximaal toegestane datum voor dat jaarsegment wordt aangegeven. De functie evalueert vervolgens of een datumveld of waarde in dit segment valt en retourneert een booleaans resultaat.

Argumenten

Argument	Beschrijving
<b>timestamp</b>	De datum die u wilt vergelijken met <b>base_date</b> .
<b>base_date</b>	Datum op basis waarvan het jaar wordt geëvalueerd.
<b>period_no</b>	Het jaar kan worden verschoven met <b>period_no</b> . <b>period_no</b> is een geheel getal, waarbij de waarde 0 het jaar aangeeft dat <b>base_date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande jaren aan en positieve waarden geven volgende jaren aan.
<b>first_month_of_year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> .

### Wanneer gebruiken

De functie `inyeartodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een `if`-uitdrukking. Dit retourneert een aggregatie of berekening afhankelijk van of een geëvalueerde datum plaatsvond in het jaar tot en met de betreffende datum.

De functie `inyeartodate()` bijvoorbeeld kan worden gebruikt om alle materieel te identificeren die in een jaar tot en met een specifieke datum is geproduceerd.

Deze voorbeelden gebruiken de datumopmaak `MM/DD/YYYY`. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>inyeartodate ('01/25/2013', '02/01/2013', 0)</code>	Retourneert <code>TRUE</code> .
<code>inyeartodate ('01/25/2012', '01/01/2013', 0)</code>	Retourneert <code>FALSE</code> .
<code>inyeartodate ('01/25/2012', '02/01/2013', -1)</code>	Retourneert <code>TRUE</code> .
<code>inyeartodate ('11/25/2012', '01/31/2013', 0, 4)</code>	Retourneert <code>TRUE</code> . De waarde van <code>timestamp</code> valt binnen het fiscale jaar dat begint in de vierde maand en vóór de waarde van <code>base_date</code> .
<code>inyeartodate ('3/31/2013', '01/31/2013', 0, 4)</code>	Retourneert <code>FALSE</code> . Retourneert <code>Vergeleken</code> met het voorafgaande voorbeeld ligt de waarde van nog steeds binnen het fiscale jaar, maar valt deze na de waarde van , dus valt deze buiten het gedeelte van het jaar <code>timestampbase_date</code>

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (`MM/DD/YYYY`) indeling.
- Het aanmaken van een veld, `in_year_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in het jaar tot en met 26 juli 2021.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    inyeartodate(date,'07/26/2021', 0) as in_year_to_date
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190,'03/27/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'06/14/2020',82.06
```

```
8194,'08/07/2020',40.39
```

```
8195,'09/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'07/27/2021',25.12
```

```
8204,'06/06/2022',46.23
```

```
8205,'07/18/2022',84.21
```

```
8206,'11/14/2022',96.24
```

```
8207,'12/12/2022',67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `in_year_to_date`

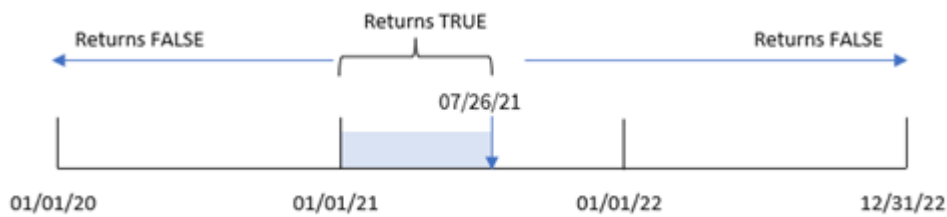
Resultatentabel

<b>date</b>	<b>in_year_to_date</b>
01/13/2020	0
02/26/2020	0
03/27/2020	0
04/16/2020	0
05/21/2020	0
06/14/2020	0
08/07/2020	0
09/05/2020	0
01/22/2021	-1
02/03/2021	-1
03/17/2021	-1
04/23/2021	-1
05/04/2021	-1
06/30/2021	-1
07/26/2021	-1
07/27/2021	0
06/06/2022	0
07/18/2022	0
11/14/2022	0
12/12/2022	0

Het veld `in_year_to_date` wordt gemaakt in de vorige load-instructie met behulp van de functie `inyeartodate()`. Het eerste opgegeven argument geeft aan welk veld wordt geëvalueerd.

Het tweede argument is een vastgestelde datum voor 26 juli 2021, de `base_date` die de eindgrens van het jaarsegment aangeeft. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen jaren voor of na het gesegmenteerde jaar vergelijkt.

Diagram van functie `inyeartodate`, geen aanvullende argumenten



Elke transactie die tussen 1 januari en 26 juli plaatsvindt, retourneert een booleaans resultaat van `TRUE`. Transactiedatums vóór 2021 en na 26 juli 2021 retourneren `FALSE`.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `previous_year_to_date`, dat bepaalt welke transacties een volledig jaar voor het einde van het jaarsegment dat eindigt op 26 juli 2021 hebben plaatsgevonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*
inyeartodate(date,'07/26/2021', -1) as previous_year_to_date
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190,'03/27/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'06/14/2020',82.06
```

```
8194,'08/07/2020',40.39
```

```
8195,'09/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```



```
8199, '04/23/2021', 25.66
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '07/27/2021', 25.12
8204, '06/06/2022', 46.23
8205, '07/18/2022', 84.21
8206, '11/14/2022', 96.24
8207, '12/12/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

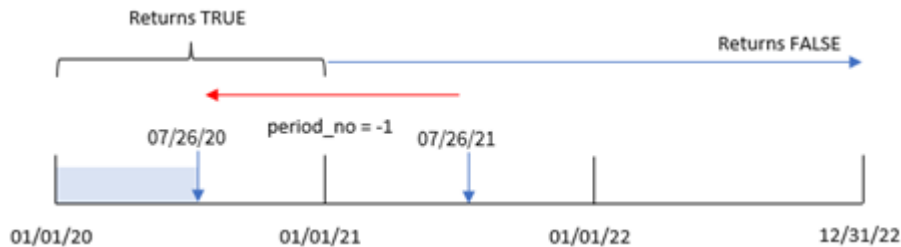
- date
- previous\_year\_to\_date

Resultatentabel

date	previous_year_to_date
01/13/2020	-1
02/26/2020	-1
03/27/2020	-1
04/16/2020	-1
05/21/2020	-1
06/14/2020	-1
08/07/2020	0
09/05/2020	0
01/22/2021	0
02/03/2021	0
03/17/2021	0
04/23/2021	0
05/04/2021	0
06/30/2021	0
07/26/2021	0
07/27/2021	0
06/06/2022	0
07/18/2022	0
11/14/2022	0
12/12/2022	0

Een `period_no`-waarde van -1 geeft aan dat de functie `inyeartodate ()` het invoerkwartaalsegment vergelijkt met het voorgaande jaar. Met een invoerdatum van 26 juli 2021, werd het segment van 1 januari 2021 tot 26 juli 2021 aanvankelijk geïdentificeerd als het jaar tot heden. De `period_no` compenseert dit segment vervolgens een volledig jaar eerder, waardoor de datumgrenzen van 1 januari tot 26 juli 2020 worden.

Diagram van functie `inyeartodate`, `period_no` example



Daarom zal elke transactie die plaatsvindt tussen 1 januari en 26 juli 2020 een booleaans resultaat van TRUE retourneren.

### Voorbeeld 3 - first\_month\_of\_year

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `in_year_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde jaar tot 26 juli 2021.

In dit voorbeeld stellen we maart in als de eerste maand van het fiscale jaar.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
    *,
    inyeartodate(date,'07/26/2021', 0,3) as in_year_to_date
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190, '03/27/2020', 88.27
8191, '04/16/2020', 57.42
8192, '05/21/2020', 53.80
8193, '06/14/2020', 82.06
8194, '08/07/2020', 40.39
8195, '09/05/2020', 87.21
8196, '01/22/2021', 95.93
8197, '02/03/2021', 45.89
8198, '03/17/2021', 36.23
8199, '04/23/2021', 25.66
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '07/27/2021', 25.12
8204, '06/06/2022', 46.23
8205, '07/18/2022', 84.21
8206, '11/14/2022', 96.24
8207, '12/12/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- in\_year\_to\_date

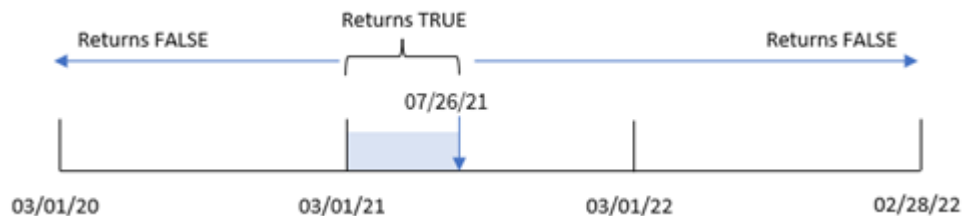
Resultatentabel

date	in_year_to_date
01/13/2020	0
02/26/2020	0
03/27/2020	0
04/16/2020	0
05/21/2020	0
06/14/2020	0
08/07/2020	0
09/05/2020	0
01/22/2021	0
02/03/2021	0
03/17/2021	-1
04/23/2021	-1
05/04/2021	-1

date	in_year_to_date
06/30/2021	-1
07/26/2021	-1
07/27/2021	0
06/06/2022	0
07/18/2022	0
11/14/2022	0
12/12/2022	0

Door 3 te gebruiken als het `first_month_of_year`-argument in de `inyeartodate()`-functie, begint de functie het jaar op 1 maart. De `base_date` van 26 juli 2021 stelt dan de einddatum in voor dat jaarsegment.

Diagram van functie `inyeartodate`, `first_month_of_year` example



Daarom zal elke transactie die plaatsvindt tussen 1 maart en 26 juli 2021 een booleaans resultaat van `TRUE` retourneren, terwijl transacties met datums buiten deze grenzen een waarde van `FALSE` zullen retourneren.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt welke transacties plaatsvonden in hetzelfde jaar tot 26 juli 2021 is gemaakt als meting in een diagramobject van de toepassing.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[  
id,date,amount  
8188,'01/13/2020',37.23  
8189,'02/26/2020',17.17  
8190,'03/27/2020',88.27  
8191,'04/16/2020',57.42  
8192,'05/21/2020',53.80  
8193,'06/14/2020',82.06  
8194,'08/07/2020',40.39  
8195,'09/05/2020',87.21  
8196,'01/22/2021',95.93  
8197,'02/03/2021',45.89  
8198,'03/17/2021',36.23  
8199,'04/23/2021',25.66  
8200,'05/04/2021',82.77  
8201,'06/30/2021',69.98  
8202,'07/26/2021',76.11  
8203,'07/27/2021',25.12  
8204,'06/06/2022',46.23  
8205,'07/18/2022',84.21  
8206,'11/14/2022',96.24  
8207,'12/12/2022',67.67  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:date.

Maak de volgende meting:

```
=inyeartodate(date,'07/26/2021',0)
```

Resultatentabel

date	=inyeartodate(date,'07/26/2021',0)
01/13/2020	0
02/26/2020	0
03/27/2020	0
04/16/2020	0
05/21/2020	0
06/14/2020	0
08/07/2020	0
09/05/2020	0
01/22/2021	-1
02/03/2021	-1
03/17/2021	-1

date	=inyeartodate(date,'07/26/2021', 0)
04/23/2021	-1
05/04/2021	-1
06/30/2021	-1
07/26/2021	-1
07/27/2021	0
06/06/2022	0
07/18/2022	0
11/14/2022	0
12/12/2022	0

De meting `in_year_to_date` wordt in het diagramobject gemaakt met behulp van de functie `inyeartodate` (). Het eerste opgegeven argument geeft aan welk veld wordt geëvalueerd. Het tweede argument is een vastgestelde datum voor 26 juli 2021, de `base_date` die de eindgrens van het vergelijkende jaarsegment aangeeft. Een `period_no` van 0 is het laatste argument, wat betekent dat de functie geen jaren voor of na het gesegmenteerde jaar vergelijkt.

*Diagram van functie inyeartodate, diagramobjectvoorbeeld*



Elke transactie die tussen 1 januari en 26 juli 2021 plaatsvindt, retourneert een booleaans resultaat van `TRUE`. Transactiedatums vóór 2021 en na 26 juli 2021 retourneren `FALSE`.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Products`.
- Informatie over product-id, producttype, productiedatum en kostprijs.

De eindgebruiker wil een diagramobject dat, per producttype, de kosten weergeeft van de producten die in 2021 tot 26 juli zijn geproduceerd.

### Load-script

```
Products:
Load
*
Inline
[
product_id,product_type,manufacture_date,cost_price
8188,product A,'01/13/2020',37.23
8189,product B,'02/26/2020',17.17
8190,product B,'03/27/2020',88.27
8191,product C,'04/16/2020',57.42
8192,product D,'05/21/2020',53.80
8193,product D,'08/14/2020',82.06
8194,product C,'10/07/2020',40.39
8195,product B,'12/05/2020',87.21
8196,product A,'01/22/2021',95.93
8197,product B,'02/03/2021',45.89
8198,product C,'03/17/2021',36.23
8199,product C,'04/23/2021',25.66
8200,product B,'05/04/2021',82.77
8201,product D,'06/30/2021',69.98
8202,product D,'07/26/2021',76.11
8203,product D,'12/27/2021',25.12
8204,product C,'06/06/2022',46.23
8205,product C,'07/18/2022',84.21
8206,product A,'11/14/2022',96.24
8207,product B,'12/12/2022',67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:product\_type.

Maak een meting die de som berekent van elk product dat vóór 27 juli in 2021 is geproduceerd:

```
=sum(if(inyeartodate(manufacture_date,makedate(2021,07,26)),0),cost_price,0))
```

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

product_type	=sum(if(inyeartodate(manufacture_date,makedate(2021,07,26)),0),cost_price,0))
product A	\$95.93
product B	\$128.66

<b>product_type</b>	<b>=sum(if(inyeartodate(manufacture_date,makedate(2021,07,26),0),cost_price,0))</b>
product C	\$61.89
product D	\$146.09

De functie `inyeartodate()` retourneert een booleaanse waarde bij het evalueren van de productiedatums van elk product. Voor elk product dat in 2021 vóór 27 juli is geproduceerd, retourneert de functie `inyeartodate()` een booleaanse waarde van `TRUE` en telt de `cost_price` op.

Product D is het enige product dat in 2021 ook na 26 juli is geproduceerd. De vermelding met `product_ID` 8203 is geproduceerd op 27 december en kost \$25.12. Daarom zijn deze kosten niet opgenomen in het totaal voor Product D in het diagramobject.

### lastworkdate

De functie **lastworkdate** retourneert de vroegste einddatum waarbij **no\_of\_workdays** (maandag-vrijdag) kan worden gehaald als wordt begonnen op **start\_date** , waarbij rekening wordt gehouden met een eventueel opgegeven **holiday**. **start\_date** en **holiday** moeten geldige datums of tijdstempels zijn.

#### Syntaxis:

```
lastworkdate (start_date, no_of_workdays {, holiday})
```



**Retourgegevenstypen:** geheel getal

*Een kalender die laat zien hoe de `lastworkdate()`-functie wordt gebruikt*

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10 start_date	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26 end_date	27
28	29	30	31			

### Beperkingen

Er is geen methode om de `lastworkdate()`-functie aan te passen voor regio's of scenario's waar sprake is van iets anders dan een werkweek die op maandag begint en op vrijdag eindigt.

De vakantieparameter moet een tekenreeksconstante zijn. Hier wordt geen uitdrukking geaccepteerd.

### Wanneer gebruiken

De `lastworkdate()`-functie wordt meestal gebruikt als onderdeel van een uitdrukking wanneer de gebruiker de voorgestelde einddatum van een project of opdracht wil berekenen op basis van wanneer het project begint en de vakantiedagen die in die periode plaatsvinden.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Argumenten

Argument	Beschrijving
<code>start_date</code>	De te evalueren begindatum.
<code>no_of_workdays</code>	Het aantal te bereiken werkdagen.
<code>holiday</code>	Vakantieperioden die moeten worden uitgesloten van werkdagen. Een vakantie wordt aangegeven als een tekenreeks met constante datum. U kunt meerdere vakantiedatums opgeven, gescheiden door komma's.  <b>Voorbeeld:</b> '12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014'

### Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met project-id's, projectbegindatums en de geschatte inspanning, in dagen, die vereist is voor de projecten. De gegevensset wordt geladen in de tabel `Projects`.
- Een voorafgaande lading met de functie `Lastworkdate()` die is ingesteld als het veld `end_date` en bepaalt wat de geplande einddatum is per project.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Projects:
  Load
    *,
    Lastworkdate(start_date,effort) as end_date
  ;
Load
id,
start_date,
effort
Inline
[
```

```
id,start_date,effort
1,01/01/2022,14
2,02/10/2022,17
3,05/17/2022,5
4,06/01/2022,12
5,08/10/2022,26
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- start\_date
- effort
- end\_date

Resultatentabel

id	start_date	inspanning	end_date
1	01/01/2022	14	01/20/2022
2	02/10/2022	17	03/04/2022
3	05/17/2022	5	05/23/2022
4	06/01/2022	12	06/16/2022
5	08/10/2022	26	09/14/2022

Omdat er geen vakantiedagen zijn gepland, telt de functie het gedefinieerde aantal werkdagen, van maandag t/m vrijdag, bij de begindatum op om de eerst mogelijke einddatum te vinden.

De volgende kalender toont de begin- en einddatum voor project 3, waarbij de werkdagen groen zijn gemarkeerd.

Een kalender die de begin- en einddatum van project 3 laat zien

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17 Start Date	18	19	20	21
22	23 End Date	24	25	26	27	28
29	30	31				

### Voorbeeld 2 - Enkele vakantiedag

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met project-id's, projectbegindatums en de geschatte inspanning, in dagen, die vereist is voor de projecten. De gegevensset wordt geladen in de tabel `Projects`.
- Een voorafgaande lading met de functie `lastworkdate()` die is ingesteld als het veld `end_date` en bepaalt wat de geplande einddatum is per project.

Maar er is één vakantiedag gepland op 18 mei 2022. De `lastworkdate()`-functie in de voorafgaande lading omvat de vakantiedag in het derde argument om te bepalen wanneer het einde van elk project gepland staat.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Projects:
  Load
    *,
    LastWorkDate(start_date,effort, '05/18/2022') as end_date
  ;
Load
id,
start_date,
effort
Inline
[
id,start_date,effort
1,01/01/2022,14
2,02/10/2022,17
3,05/17/2022,5
4,06/01/2022,12
5,08/10/2022,26
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- start\_date
- effort
- end\_date

Resultatentabel

id	start_date	inspanning	end_date
1	01/01/2022	14	01/20/2022
2	02/10/2022	17	03/04/2022
3	05/17/2022	5	05/24/2022
4	06/01/2022	12	06/16/2022
5	08/10/2022	26	09/14/2022

De enkele, geplande vakantiedag is ingevoerd als het derde argument in de `Lastworkdate()`-functie. Het resultaat is dat de einddatum voor project 3 één dag later valt omdat de vakantiedag plaatsvindt op een van de werkdagen vóór de einddatum.

De volgende kalender toont de begin- en einddatum voor project 3 en laat zien dat de vakantiedag de einddatum van het project met één dag verandert.

Een kalender die de begin- en einddatum van project 3 laat zien met een vakantiedag op 18 mei

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17 Start Date	18 Holiday	19	20	21
22	23	24 End Date	25	26	27	28
29	30	31				

### Voorbeeld 3 - Meerdere vakantiedagen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met project-id's, projectbegindatums en de geschatte inspanning, in dagen, die vereist is voor de projecten. De gegevensset wordt geladen in de tabel `Projects`.
- Een voorafgaande lading met de functie `lastworkdate()` die is ingesteld als het veld `end_date` en bepaalt wat de geplande einddatum is per project.

Maar er zijn drie vakantiedagen gepland voor 19, 20, 21 en 22 mei. De `lastworkdate()`-functie in de voorafgaande lading omvat elk van de vakantiedagen in het derde argument om te bepalen wanneer het einde van elk project gepland staat.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Projects:
  Load
    *,
    LastWorkDate(start_date,effort, '05/19/2022','05/20/2022','05/21/2022','05/22/2022') as
end_date
;
Load
id,
start_date,
effort
Inline
[
id,start_date,effort
1,01/01/2022,14
2,02/10/2022,17
3,05/17/2022,5
4,06/01/2022,12
5,08/10/2022,26
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- start\_date
- effort
- end\_date

Resultatentabel

id	start_date	inspanning	end_date
1	01/01/2022	14	01/20/2022
2	02/10/2022	17	03/04/2022
3	05/17/2022	5	05/25/2022
4	06/01/2022	12	06/16/2022
5	08/10/2022	26	09/14/2022

De vier vakantiedagen worden ingevoerd als een lijst met argumenten in de `Lastworkdate()`-functie na de begindatum en het aantal werkdagen.

De volgende kalender toont de begin- en einddatum voor project 3 en laat zien dat de vakantiedag de einddatum van het project met drie dagen verandert.

## 5 Script- en diagramfuncties

Een kalender die de begin- en einddatum van project 3 laat zien met een vakantiedag op 19 t/m 22 mei

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17 Start Date	18	19 Holiday	20 Holiday	21 Holiday
22 Holiday	23	24	25 End Date	26	27	28
29	30	31				

### Voorbeeld 4 - Enkele vakantiedag (diagram)

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de app geladen. het end\_date-veld wordt berekend als een meting in het diagram.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

Projects:

Load

id,

start\_date,

effort

Inline

[



```
id,start_date,effort
1,01/01/2022,14
2,02/10/2022,17
3,05/17/2022,5
4,06/01/2022,12
5,08/10/2022,26
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- start\_date
- effort

Maak de volgende meting om de end\_date te berekenen:

- =LastWorkDate(start\_date,effort,'05/18/2022')

Resultatentabel

id	start_date	inspanning	=LastWorkDate(start_date,effort,'05/18/2022')
1	01/01/2022	14	01/20/2022
2	02/10/2022	17	03/04/2022
3	05/17/2022	5	05/23/2022
4	06/01/2022	12	06/16/2022
5	08/10/2022	26	09/14/2022

De enkele, geplande vakantiedag wordt ingevoerd als een meting in het diagram. Het resultaat is dat de einddatum voor project 3 één dag later valt omdat de vakantiedag plaatsvindt op een van de werkdagen vóór de einddatum.

De volgende kalender toont de begin- en einddatum voor project 3 en laat zien dat de vakantiedag de einddatum van het project met één dag verandert.

## 5 Script- en diagramfuncties

Een kalender die de begin- en einddatum van project 3 laat zien met een vakantiedag op 18 mei

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17 Start Date	18 Holiday	19	20	21
22	23	24 End Date	25	26	27	28
29	30	31				

### localtime

Deze functie retourneert een tijdstempel met de huidige tijd voor een opgegeven tijdzone.

#### Syntaxis:

```
LocalTime([timezone [, ignoreDST ]])
```

Retourgegevenstypen: dual

#### Argumenten:

##### Argumenten

Argument	Beschrijving
timezone	De <b>timezone</b> wordt opgegeven als een tekenreeks die een van de geografische plaatsen bevat die worden vermeld onder <b>Time Zone</b> in de Windows Control Panel voor Date and Time of als tekenreeks in de vorm 'GMT+hh:mm'.timezone  Als geen tijdzone is opgegeven, wordt de lokale tijd geretourneerd.
ignoreDST	Als <b>ignoreDST</b> -1 (True) is, wordt de zomertijd genegeerd.

### Voorbeelden en resultaten:

De onderstaande voorbeelden zijn gebaseerd op de functie die wordt aangeroepen op 2014-10-22 12:54:47 lokale tijd, met GMT+01:00 als lokale tijdzone.

Scriptvoorbeelden

Voorbeeld	Resultaat
<code>Localtime ()</code>	Retourneert de lokale tijd 2014-10-22 12:54:47.
<code>Localtime ('London')</code>	Retourneert de lokale tijd in Londen, 2014-10-22 11:54:47.
<code>Localtime ('GMT+02:00')</code>	Retourneert de lokale tijd in de tijdzone van GMT+02:00, 2014-10-22 13:54:47.
<code>Localtime ('Paris', '-1')</code>	Retourneert de lokale tijd in Parijs waarbij de zomertijd wordt genegeerd, 2014-10-22 11:54:47.

### lunarweekend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van de maanweek die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

#### Syntaxis:

```
LunarweekEnd(date[, period_no[, first_week_day]])
```

**Retourgegevenstypen:** dual

*Voorbeelddiagram van de Lunarweekend()-functie*



De `Lunarweekend()`-functie bepaalt welke maanweek de `date` in valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van die week.

Argumenten

Argument	Beschrijving
<code>date</code>	De te evalueren datum of tijdstempel.

Argument	Beschrijving
<b>period_no</b>	<b>period_no</b> is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de maanweek aangeeft die <b>date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande maanweken aan en positieve waarden geven volgende maanweken aan.
<b>first_week_day</b>	Een verschuiving kan groter of kleiner zijn dan nul. Hiermee wordt het begin van het jaar gewijzigd met het opgegeven aantal dagen en/of fracties van een dag.

### Wanneer gebruiken

De `1unarweekend()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening een deel van de week wordt gebruikt dat nog niet heeft plaatsgevonden. Anders dan de `weekend()`-functie, eindigt de laatste maanweek van het kalenderjaar op 31 december. De `1unarweekend()`-functie zou bijvoorbeeld kunnen worden gebruikt om rente te berekenen die tijdens de week nog niet is opgebouwd.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>1unarweekend('01/12/2013')</code>	Retourneert 01/14/2013 23:59:59.
<code>1unarweekend('01/12/2013', -1)</code>	Retourneert 01/07/2013 23:59:59.
<code>1unarweekend('01/12/2013', 0, 1)</code>	Retourneert 01/15/2013 23:59:59.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, end\_of\_week, dat een tijdstempel voor het einde van de maanweek retourneert waarin de transactie plaatsvond.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *
    ,
    lunarweekend(date) as end_of_week,
    timestamp(lunarweekend(date)) as end_of_week_timestamp
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- end\_of\_week

- `end_of_week_timestamp`

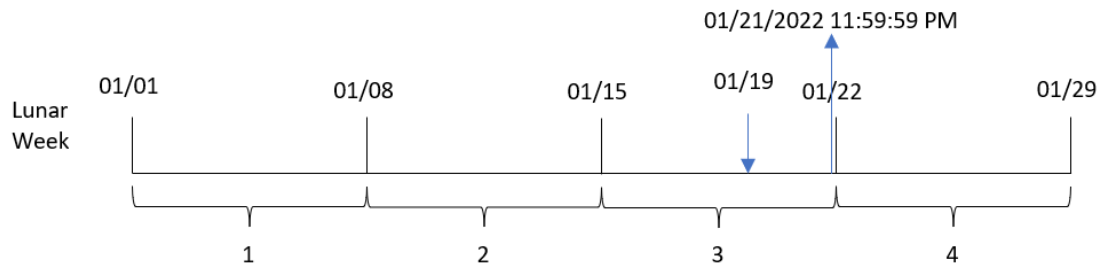
Resultatentabel

<b>date</b>	<b>end_of_week</b>	<b>end_of_week_timestamp</b>
1/7/2022	01/07/2022	1/7/2022 11:59:59 PM
1/19/2022	01/21/2022	1/21/2022 11:59:59 PM
2/5/2022	02/11/2022	2/11/2022 11:59:59 PM
2/28/2022	03/04/2022	3/4/2022 11:59:59 PM
3/16/2022	03/18/2022	3/18/2022 11:59:59 PM
4/1/2022	04/01/2022	4/1/2022 11:59:59 PM
5/7/2022	05/13/2022	5/13/2022 11:59:59 PM
5/16/2022	05/20/2022	5/20/2022 11:59:59 PM
6/15/2022	06/17/2022	6/17/2022 11:59:59 PM
6/26/2022	07/01/2022	7/1/2022 11:59:59 PM
7/9/2022	07/15/2022	7/15/2022 11:59:59 PM
7/22/2022	07/22/2022	7/22/2022 11:59:59 PM
7/23/2022	07/29/2022	7/29/2022 11:59:59 PM
7/27/2022	07/29/2022	7/29/2022 11:59:59 PM
8/2/2022	08/05/2022	8/5/2022 11:59:59 PM
8/8/2022	08/12/2022	8/12/2022 11:59:59 PM
8/19/2022	08/19/2022	8/19/2022 11:59:59 PM
9/26/2022	09/30/2022	9/30/2022 11:59:59 PM
10/14/2022	10/14/2022	10/14/2022 11:59:59 PM
10/29/2022	11/04/2022	11/4/2022 11:59:59 PM

Het `end_of_week`-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de `1unarweekend()`-functie en geeft het `date`-veld door als het argument van de functie.

De `1unarweekend()`-functie identificeert in welke maandweek de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die week.

Diagram van `lunarweekend()`-functie, voorbeeld zonder aanvullende argumenten



Transactie 8189 vond plaats op 19 januari. De `lunarweekend()`-functie identificeert dat de maanweek op 15 januari begint. De `end_of_week`-waarde voor die transactie retourneert daarom de laatste milliseconde van die dag, 21 januari om 11:59:59 PM.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_lunar_week_end`, dat de tijdstempel voor het einde van de maanweek retourneert voordat de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    lunarweekend(date,-1) as previous_lunar_week_end,
    timestamp(lunarweekend(date,-1)) as previous_lunar_week_end_timestamp
  ;
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
```

```
8195, 5/16/2022, 87.21
8196, 6/15/2022, 95.93
8197, 6/26/2022, 45.89
8198, 7/9/2022, 36.23
8199, 7/22/2022, 25.66
8200, 7/23/2022, 82.77
8201, 7/27/2022, 69.98
8202, 8/2/2022, 76.11
8203, 8/8/2022, 25.12
8204, 8/19/2022, 46.23
8205, 9/26/2022, 84.21
8206, 10/14/2022, 96.24
8207, 10/29/2022, 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_lunar\_week\_end
- previous\_lunar\_week\_end\_timestamp

Resultatentabel

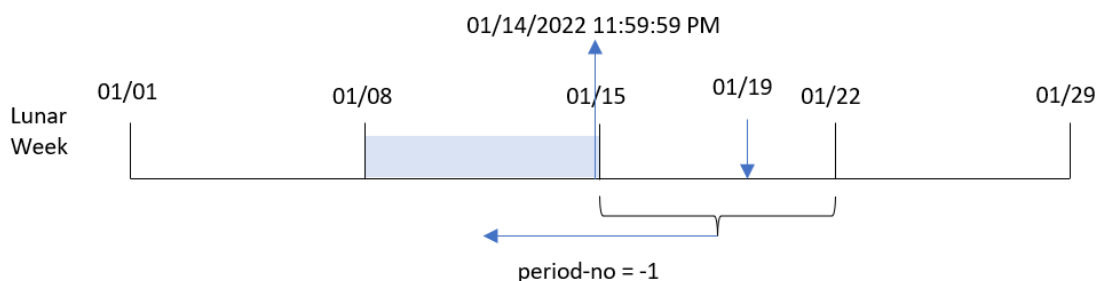
date	previous_lunar_week_end	previous_lunar_week_end_timestamp
1/7/2022	12/31/2021	12/31/2021 11:59:59 PM
1/19/2022	01/14/2022	1/14/2022 11:59:59 PM
2/5/2022	02/04/2022	2/4/2022 11:59:59 PM
2/28/2022	02/25/2022	2/25/2022 11:59:59 PM
3/16/2022	03/11/2022	3/18/2022 11:59:59 PM
4/1/2022	03/25/2022	3/25/2022 11:59:59 PM
5/7/2022	05/06/2022	5/6/2022 11:59:59 PM
5/16/2022	05/13/2022	5/13/2022 11:59:59 PM
6/15/2022	06/10/2022	6/10/2022 11:59:59 PM
6/26/2022	06/24/2022	6/24/2022 11:59:59 PM
7/9/2022	07/08/2022	7/8/2022 11:59:59 PM
7/22/2022	07/15/2022	7/15/2022 11:59:59 PM
7/23/2022	07/22/2022	7/22/2022 11:59:59 PM
7/27/2022	07/22/2022	7/22/2022 11:59:59 PM
8/2/2022	07/29/2022	7/29/2022 11:59:59 PM



date	previous_lunar_week_end	previous_lunar_week_end_timestamp
8/8/2022	08/05/2022	8/5/2022 11:59:59 PM
8/19/2022	08/12/2022	8/12/2022 11:59:59 PM
9/26/2022	09/23/2022	9/23/2022 11:59:59 PM
10/14/2022	10/07/2022	10/7/2022 11:59:59 PM
10/29/2022	10/28/2022	10/28/2022 11:59:59 PM

In dit geval, omdat een `period_no` van -1 werd gebruikt als het `offset`-argument in de `lunarweekend()`-functie, identificeert de functie eerst de maanweek waarin de transacties plaatsvonden. Het verschuift dan naar een week eerder en identificeert de laatste milliseconde van die maanweek.

Diagram van `lunarweekend()`-functie, voorbeeld `period_no`



Transactie 8189 vond plaats op 19 januari. De `lunarweekend()`-functie identificeert dat de maanweek op 15 januari begint. Daarom begon de vorige maanweek op 8 januari en eindigde hij op 14 januari om 11:59:59 PM; dit is de waarde die voor het `previous_lunar_week_end`-veld wordt geretourneerd.

### Voorbeeld 3 - first\_week\_day

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld stellen we het begin van de maanweek in op 5 januari.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    lunarweekend(date,0,4) as end_of_week,
    timestamp(lunarweekend(date,0,4)) as end_of_week_timestamp
```

```
;
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- end\_of\_week
- end\_of\_week\_timestamp

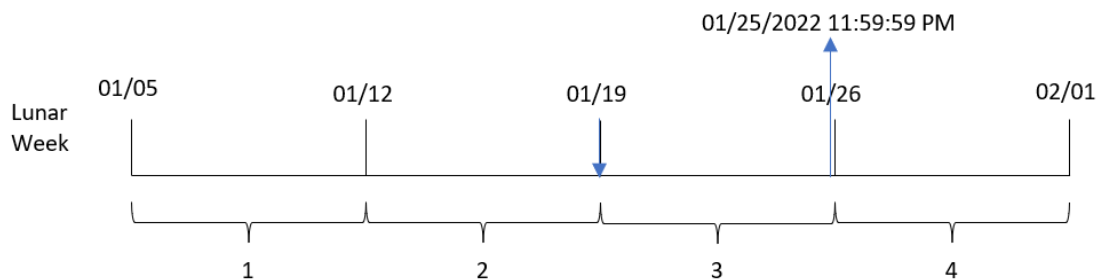
Resultatentabel

date	end_of_week	end_of_week_timestamp
1/7/2022	01/11/2022	1/11/2022 11:59:59 PM
1/19/2022	01/25/2022	1/25/2022 11:59:59 PM
2/5/2022	02/08/2022	2/8/2022 11:59:59 PM
2/28/2022	03/01/2022	3/1/2022 11:59:59 PM
3/16/2022	03/22/2022	3/22/2022 11:59:59 PM
4/1/2022	04/05/2022	4/5/2022 11:59:59 PM
5/7/2022	05/10/2022	5/10/2022 11:59:59 PM
5/16/2022	05/17/2022	5/17/2022 11:59:59 PM

date	end_of_week	end_of_week_timestamp
6/15/2022	06/21/2022	6/21/2022 11:59:59 PM
6/26/2022	06/28/2022	6/28/2022 11:59:59 PM
7/9/2022	07/12/2022	7/12/2022 11:59:59 PM
7/22/2022	07/26/2022	7/26/2022 11:59:59 PM
7/23/2022	07/26/2022	7/26/2022 11:59:59 PM
7/27/2022	08/02/2022	8/2/2022 11:59:59 PM
8/2/2022	08/02/2022	8/2/2022 11:59:59 PM
8/8/2022	08/09/2022	8/9/2022 11:59:59 PM
8/19/2022	08/23/2022	8/23/2022 11:59:59 PM
9/26/2022	09/27/2022	9/27/2022 11:59:59 PM
10/14/2022	10/18/2022	10/18/2022 11:59:59 PM
10/29/2022	11/01/2022	11/1/2022 11:59:59 PM

In dit geval verschuift het begin van het jaar van 1 januari naar 5 januari omdat het `first_week_date`-argument 4 in de `Tunarweekend()`-functie is gebruikt.

Diagram van de `Tunarweekend()`-functie, voorbeeld met `first_week_day`



Transactie 8189 vond plaats op 19 januari. Omdat maanweken op 5 januari beginnen, identificeert de `Tunarweekend()`-functie dat de maanweek waarin 19 januari valt, tevens begint met 19 januari. Daarom valt het einde van die maanweek op 25 januari om 11:59:59 PM. Dit is de waarde die is geretourneerd voor het `end_of_week`-veld.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het einde van de maanweek waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de applicatie.

### Load-script

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Voeg de volgende metingen toe:

```
=lunarweekend(date)
```

```
=timestamp(lunarweekend(date))
```

Resultatentabel

date	=lunarweekend(date)	=timestamp(lunarweekend(date))
1/7/2022	01/07/2022	1/7/2022 11:59:59 PM
1/19/2022	01/21/2022	1/21/2022 11:59:59 PM

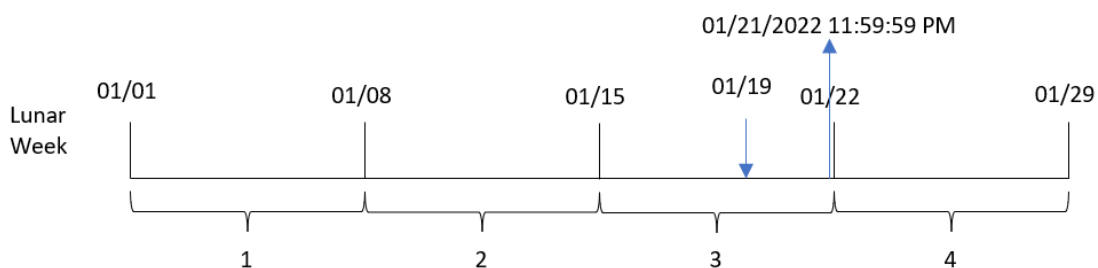
## 5 Script- en diagramfuncties

date	=lunarweekend(date)	=timestamp(lunarweekend(date))
2/5/2022	02/11/2022	2/11/2022 11:59:59 PM
2/28/2022	03/04/2022	3/4/2022 11:59:59 PM
3/16/2022	03/18/2022	3/18/2022 11:59:59 PM
4/1/2022	04/01/2022	4/1/2022 11:59:59 PM
5/7/2022	05/13/2022	5/13/2022 11:59:59 PM
5/16/2022	05/20/2022	5/20/2022 11:59:59 PM
6/15/2022	06/17/2022	6/17/2022 11:59:59 PM
6/26/2022	07/01/2022	7/1/2022 11:59:59 PM
7/9/2022	07/15/2022	7/15/2022 11:59:59 PM
7/22/2022	07/22/2022	7/22/2022 11:59:59 PM
7/23/2022	07/29/2022	7/29/2022 11:59:59 PM
7/27/2022	07/29/2022	7/29/2022 11:59:59 PM
8/2/2022	08/05/2022	8/5/2022 11:59:59 PM
8/8/2022	08/12/2022	8/12/2022 11:59:59 PM
8/19/2022	08/19/2022	8/19/2022 11:59:59 PM
9/26/2022	09/30/2022	9/30/2022 11:59:59 PM
10/14/2022	10/14/2022	10/14/2022 11:59:59 PM
10/29/2022	11/04/2022	11/4/2022 11:59:59 PM

De meting `end_of_week` wordt gemaakt in het diagramobject met behulp van de `lunarweekend()`-functie en geeft het veld `date` door als het argument van de functie.

De `lunarweekend()`-functie identificeert in welke maanweek de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die week.

*Diagram van `lunarweekend()`-functie, diagramobjectvoorbeeld*



Transactie 8189 vond plaats op 19 januari. De `1unarweekend()`-functie identificeert dat de maandag op 15 januari begint. De `end_of_week`-waarde voor die transactie retourneert daarom de laatste milliseconde van die dag, 21 januari om 11:59:59 PM.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Employee_Expenses`.
- De werknemers-id's, werknemersnaam en de gemiddelde dagelijkse onkostendeclaraties van elke werknemer.

De eindgebruiker wil graag een diagramobject dat per werknemer-id en werknemersnaam de geschatte nog te maken onkostendeclaraties voor de rest van de maandag weergeeft.

#### Load-script

```
Employee_Expenses :
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,Sydney,$27
186,Agatha,$18
];
```

#### Resultaten

Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Voeg de volgende velden als dimensies toe:
  - `employee_id`
  - `employee_name`
3. Maak nu de volgende meting om de opgebouwde rente te berekenen:  
`=(1unarweekend(today(1))-today(1))*avg_daily_claim`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

employee_id	employee_name	=(lunarweekend(today(1))-today(1))*avg_daily_claim
182	Mark	\$75.00
183	Deryck	\$62.50
184	Dexter	\$62.50
185	Sydney	\$135.00
186	Agatha	\$90.00

Door de datum van vandaag als enig argument te gebruiken, retourneert de `lunarweekend()`-functie de einddatum van de huidige maanweek. Door vervolgens de datum van vandaag af te trekken van de einddatum van de maanweek, retourneert de uitdrukking het aantal resterende dagen in deze week.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in de resterende maanweek zal indienen.

## lunarweekname

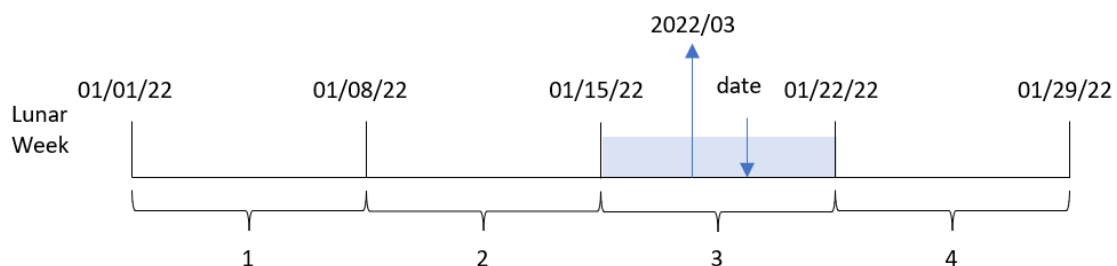
Deze functie retourneert een weergavewaarde voor het jaar en maanweeknummer dat overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de week die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

### Syntaxis:

```
LunarWeekName (date [, period_no[, first_week_day]])
```

**Retourgegevenstypen:** dual

*Voorbeelddiagram van de `lunarweekname()`-functie*



De `lunarweekname()`-functie bepaalt in welke maanweek de datum valt, waarbij de weektelling begint op 1 januari. Er wordt vervolgens een waarde geretourneerd die bestaat uit de `year/weekcount`.

### Argumenten

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de maanweek aangeeft die <b>date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande maanweken aan en positieve waarden geven volgende maanweken aan.
<b>first_week_day</b>	Een verschuiving kan groter of kleiner zijn dan nul. Hiermee wordt het begin van het jaar gewijzigd met het opgegeven aantal dagen en/of fracties van een dag.

### Wanneer gebruiken

De `Tunarweekname()`-functie is handig als u aggregaties per maanweek wilt vergelijken. Bijvoorbeeld: de functie kan worden gebruikt om de totale verkoop van producten per maanweek te bepalen. Maanweken zijn handig wanneer u er zeker van wilt zijn dat alle waarden uit de eerste week van het jaar, alleen de waarden vanaf 1 januari omvatten.

Deze dimensies kunnen in het load-script worden gemaakt door de functie te gebruiken om een veld in een masterkalendertabel te maken. De functie kan ook rechtstreeks in een diagram worden gebruikt als een berekende dimensie.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>Tunarweekname('01/12/2013')</code>	Retourneert 2006/02.
<code>Tunarweekname('01/12/2013', -1)</code>	Retourneert 2006/01.
<code>Tunarweekname('01/12/2013', 0, 1)</code>	Retourneert 2006/02.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.



### Voorbeeld 1 - Datum zonder aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.
- Het veld `lunar_week_name` wordt gemaakt en retourneert het jaar- en weeknummer voor de maanweek waarin de transacties plaatsvonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    lunarweekname(date) as lunar_week_name
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- lunar\_week\_name

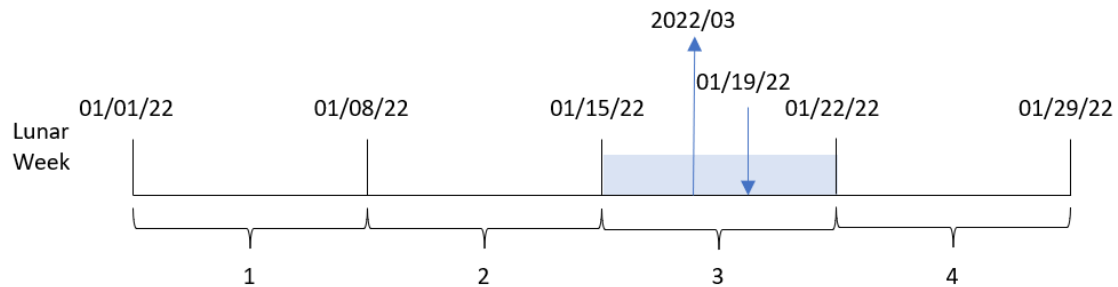
Resultatentabel

date	lunar_week_name
1/7/2022	2022/01
1/19/2022	2022/03
2/5/2022	2022/06
2/28/2022	2022/09
3/16/2022	2022/11
4/1/2022	2022/13
5/7/2022	2022/19
5/16/2022	2022/20
6/15/2022	2022/24
6/26/2022	2022/26
7/9/2022	2022/28
7/22/2022	2022/29
7/23/2022	2022/30
7/27/2022	2022/30
8/2/2022	2022/31
8/8/2022	2022/32
8/19/2022	2022/33
9/26/2022	2022/39
10/14/2022	2022/41
10/29/2022	2022/44

Het lunar\_week\_name-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de lunarweekname()-functie en geeft het date-veld door als het argument van de functie.

De lunarweekname()-functie identificeert in welke maanweek de datumwaarde valt en retourneert het jaaren weeknummer van die datum.

Diagram van `lunarweekname()`-functie, voorbeeld zonder aanvullende argumenten



Transactie 8189 vond plaats op 19 januari. De `lunarweekname()`-functie identificeert dat deze datum in de maanweek valt die op 15 januari begint. Dit is de derde maanweek van het jaar. Daarom is de `lunar_week_name`-waarde die voor deze transactie wordt geretourneerd 2022/03.

### Voorbeeld 2 - Datum met `period_no` argument

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het veld `previous_lunar_week_name` wordt gemaakt en retourneert het jaar- en weeknummer voor de maanweek voorafgaand aan de week waarin de transacties plaatsvonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    lunarweekname(date,-1) as previous_lunar_week_name
  ;

Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_lunar\_week\_name

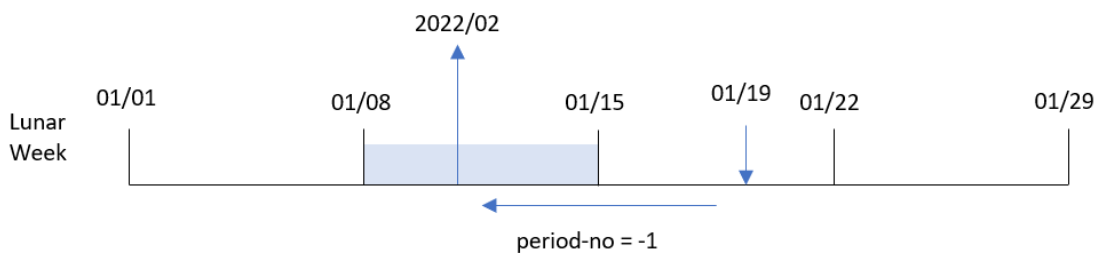
Resultatentabel

date	previous_lunar_week_name
1/7/2022	2021/52
1/19/2022	2022/02
2/5/2022	2022/05
2/28/2022	2022/08
3/16/2022	2022/10
4/1/2022	2022/12
5/7/2022	2022/18
5/16/2022	2022/19
6/15/2022	2022/23
6/26/2022	2022/25
7/9/2022	2022/27
7/22/2022	2022/28
7/23/2022	2022/29
7/27/2022	2022/29
8/2/2022	2022/30
8/8/2022	2022/31
8/19/2022	2022/32

date	previous_lunar_week_name
9/26/2022	2022/38
10/14/2022	2022/40
10/29/2022	2022/43

In dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `lunarweekname()`-functie, identificeert de functie eerst de maanweek waarin de transacties plaatsvonden. De functie retourneert vervolgens het jaar en het nummer van één week eerder.

*Diagram van `lunarweekname()`-functie, voorbeeld `period_no`*



Transactie 8189 vond plaats op 19 januari. De `lunarweekname()`-functie identificeert dat deze transactie in de derde maanweek van het jaar plaatsvond en retourneert daarom het jaar en de waarde voor één week eerder, 2022/02, voor het `previous_lunar_week_name`-veld.

### Voorbeeld 3 - Datum met `first_week_day`-argument

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld stellen we het begin van de maanweek in op 5 januari.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    lunarweekname(date,0,4) as lunar_week_name
  ;
Load
*
Inline
[
```

```
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

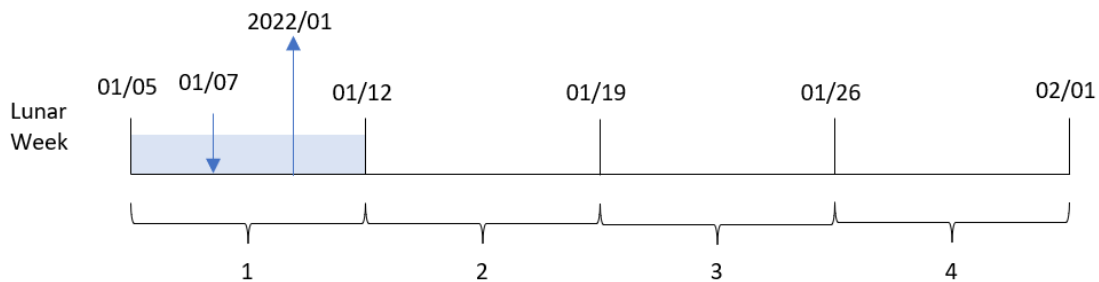
- date
- lunar\_week\_name

Resultatentabel

date	lunar_week_name
1/7/2022	2022/01
1/19/2022	2022/03
2/5/2022	2022/05
2/28/2022	2022/08
3/16/2022	2022/11
4/1/2022	2022/13
5/7/2022	2022/18
5/16/2022	2022/19
6/15/2022	2022/24
6/26/2022	2022/25
7/9/2022	2022/27

date	lunar_week_name
7/22/2022	2022/29
7/23/2022	2022/29
7/27/2022	2022/30
8/2/2022	2022/30
8/8/2022	2022/31
8/19/2022	2022/33
9/26/2022	2022/38
10/14/2022	2022/41
10/29/2022	2022/43

Diagram van de `lunarweekname()`-functie, voorbeeld met `first_week_day`



In dit geval verschuift het begin van de maanweken van 1 januari naar 5 januari omdat het `first_week_date`-argument 4 in de `lunarweekname()`-functie is gebruikt.

Transactie 8188 vond plaats op 7 januari. Omdat maanweken op 5 januari beginnen, identificeert de `lunarweekname()`-functie dat de maanweek waarin 7 januari valt, de eerste maanweek van het jaar is. Daarom is de `lunar_week_name`-waarde die voor deze transactie wordt getourneerd 2022/01.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die het maanweeknummer en het jaar retourneert waarin de transacties plaatsvonden, is gemaakt als meting in een diagramobject van de applicatie.

### Load-script

Transactions:

Load

\*

Inline

[

id,date,amount

8188,1/7/2022,17.17

8189,1/19/2022,37.23

8190,2/28/2022,88.27

8191,2/5/2022,57.42

8192,3/16/2022,53.80

8193,4/1/2022,82.06

8194,5/7/2022,40.39

8195,5/16/2022,87.21

8196,6/15/2022,95.93

8197,6/26/2022,45.89

8198,7/9/2022,36.23

8199,7/22/2022,25.66

8200,7/23/2022,82.77

8201,7/27/2022,69.98

8202,8/2/2022,76.11

8203,8/8/2022,25.12

8204,8/19/2022,46.23

8205,9/26/2022,84.21

8206,10/14/2022,96.24

8207,10/29/2022,67.67

];

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Om de begindatum van de maanweek te berekenen waarin een transactie heeft plaatsgevonden, maakt u de volgende meting:

=lunarweekname(date)

Resultatentabel

date	=lunarweekname(date)
1/7/2022	2022/01
1/19/2022	2022/03
2/5/2022	2022/06
2/28/2022	2022/09
3/16/2022	2022/11
4/1/2022	2022/13
5/7/2022	2022/19

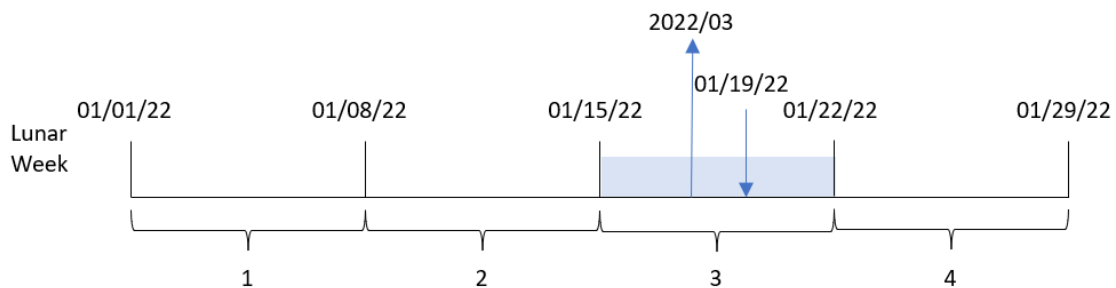


date	=lunarweekname(date)
5/16/2022	2022/20
6/15/2022	2022/24
6/26/2022	2022/26
7/9/2022	2022/28
7/22/2022	2022/29
7/23/2022	2022/30
7/27/2022	2022/30
8/2/2022	2022/31
8/8/2022	2022/32
8/19/2022	2022/33
9/26/2022	2022/39
10/14/2022	2022/41
10/29/2022	2022/44

De meting `lunar_week_name` wordt gemaakt in het diagramobject met behulp van de functie `lunarweekname()` en geeft het veld `date` door als het argument van de functie.

De `lunarweekname()`-functie identificeert in welke maanweek de datumwaarde valt en retourneert het jaar- en weeknummer van die datum.

*Diagram van `lunarweekname()`-functie, diagramobjectvoorbeeld*



Transactie 8189 vond plaats op 19 januari. De `lunarweekname()`-functie identificeert dat deze datum in de maanweek valt die op 15 januari begint. Dit is de derde maanweek van het jaar. Daarom is de `lunar_week_name`-waarde voor deze transactie 2022/03.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.

De eindgebruiker wil graag een diagramobject dat de totale verkoop per week voor het huidige jaar weergeeft. Week 1, met een lengte van zeven dagen, zou op 1 januari moeten beginnen. Dit kan zelfs worden bereikt als deze dimensie niet beschikbaar is in het gegevensmodel, door de functie Tunarweekname() te gebruiken als een berekende dimensie in het diagram.

#### Load-script

```
SET DateFormat='MM/DD/YYYY' ;
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een berekende dimensie met behulp van de volgende uitdrukking:  
`=lunarweekname(date)`
3. Bereken de totale verkoop met de volgende aggregatiemeting:  
`=sum(amount)`
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

<code>=lunarweekname(date)</code>	<code>=sum(amount)</code>
2022/01	\$17.17
2022/03	\$37.23
2022/06	\$57.42
2022/09	\$88.27
2022/11	\$53.80
2022/13	\$82.06
2022/19	\$40.39
2022/20	\$87.21
2022/24	\$95.93
2022/26	\$45.89
2022/28	\$36.23
2022/29	\$25.66
2022/30	\$152.75
2022/31	\$76.11
2022/32	\$25.12
2022/33	\$46.23
2022/39	\$84.21
2022/41	\$96.24
2022/44	\$67.67

## lunarweekstart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maanweek die **date** bevat. Maanweken in Qlik Sense worden gedefinieerd door het feit dat 1 januari wordt gezien als de eerste dag van de week en, met uitzondering van de laatste week van het jaar, bevat elke week precies zeven dagen.

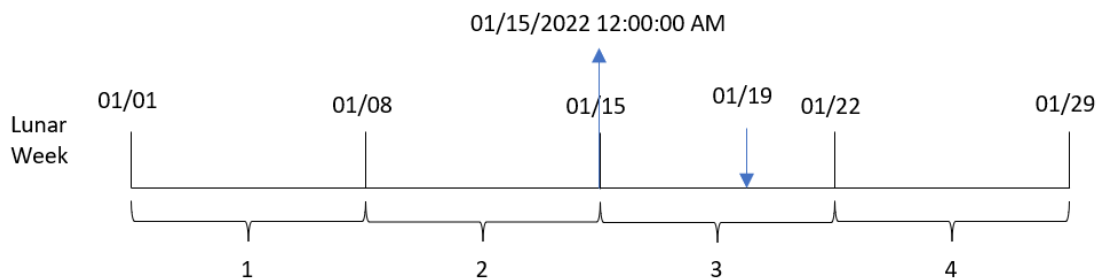
### Syntaxis:

```
LunarweekStart(date[, period_no[, first_week_day]])
```

### Retourgegevenstypen: dual

De `LunarweekStart()`-functie bepaalt welke maanweek de `date` in valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de eerste milliseconde van die week.

Voorbeelddiagram van de `LunarweekStart()`-functie



### Argumenten

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal, of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de maanweek aangeeft die <b>date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande maanweken aan en positieve waarden geven volgende maanweken aan.
<b>first_week_day</b>	Een verschuiving kan groter of kleiner zijn dan nul. Hiermee wordt het begin van het jaar gewijzigd met het opgegeven aantal dagen en/of fracties van een dag.

## Wanneer gebruiken

De `LunarweekStart()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van de week wordt gebruikt dat al is geweest. Anders dan de `weekstart()`-functie beginnen weken aan het begin van ieder nieuw kalenderjaar, op 1 januari en alle daaropvolgende weken beginnen zeven dagen later. De `LunarweekStart()`-functie wordt niet beïnvloed door de systeemvariabele `FirstWeekDay`.

De `Tunarweekstart()` kan bijvoorbeeld worden gebruikt om de rente te berekenen die in een week tot nu toe is opgebouwd.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>Tunarweekstart ('01/12/2013')</code>	Retourneert 01/08/2013.
<code>Tunarweekstart ('01/12/2013', -1)</code>	Retourneert 01/01/2013.
<code>Tunarweekstart ('01/12/2013', 0, 1 )</code>	Retourneert 01/09/2013, want als <code>first_week_day</code> wordt ingesteld op 1 wordt het begin van het jaar veranderd in 01/02/2013.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 - Geen aanvullende argumenten

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `start_of_week`, dat een tijdstempel voor het begin van de maanweek retourneert waarin de transactie plaatsvond.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
    *,
    lunarweekstart(date) as start_of_week,
    timestamp(lunarweekstart(date)) as start_of_week_timestamp
;

Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- start\_of\_week
- start\_of\_week\_timestamp

Resultatentabel

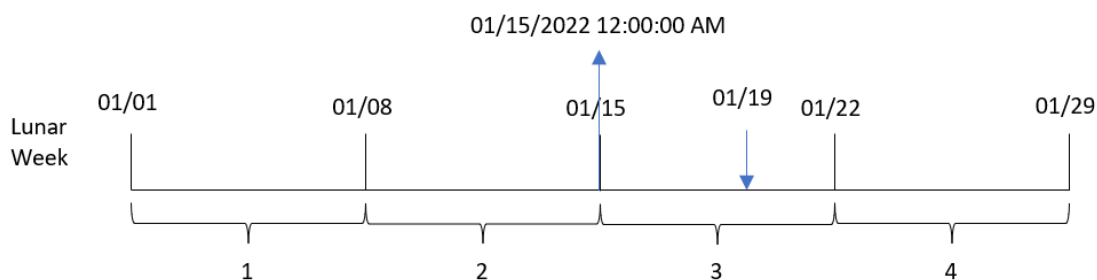
date	start_of_week	start_of_week_timestamp
1/7/2022	01/01/2022	1/1/2022 12:00:00 AM
1/19/2022	01/15/2022	1/15/2022 12:00:00 AM
2/5/2022	02/05/2022	2/5/2022 12:00:00 AM
2/28/2022	02/26/2022	2/26/2022 12:00:00 AM
3/16/2022	03/12/2022	3/12/2022 12:00:00 AM

date	start_of_week	start_of_week_timestamp
4/1/2022	03/26/2022	3/26/2022 12:00:00 AM
5/7/2022	05/07/2022	5/7/2022 12:00:00 AM
5/16/2022	05/14/2022	5/14/2022 12:00:00 AM
6/15/2022	06/11/2022	6/11/2022 12:00:00 AM
6/26/2022	06/25/2022	6/25/2022 12:00:00 AM
7/9/2022	07/09/2022	7/9/2022 12:00:00 AM
7/22/2022	07/16/2022	7/16/2022 12:00:00 AM
7/23/2022	07/23/2022	7/23/2022 12:00:00 AM
7/27/2022	07/23/2022	7/23/2022 12:00:00 AM
8/2/2022	07/30/2022	7/30/2022 12:00:00 AM
8/8/2022	08/06/2022	8/6/2022 12:00:00 AM
8/19/2022	08/13/2022	8/13/2022 12:00:00 AM
9/26/2022	09/24/2022	9/24/2022 12:00:00 AM
10/14/2022	10/08/2022	10/8/2022 12:00:00 AM
10/29/2022	10/29/2022	10/29/2022 12:00:00 AM

Het veld `start_of_week` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `Lunarweekstart()` en geeft het veld `date` door als het argument van de functie.

De `Lunarweekstart()`-functie identificeert in welke maanweek de datum valt en retourneert een tijdstempel voor de eerste milliseconde van die week.

*Diagram van `Lunarweekstart()`-functie, voorbeeld zonder aanvullende argumenten*



Transactie 8189 vond plaats op 19 januari. De `Lunarweekstart()`-functie identificeert dat de maanweek op 15 januari begint. De `start_of_week`-waarde voor die transactie retourneert daarom de eerste milliseconde van die dag, 15 januari om 12:00:00 AM.

### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_lunar_week_start`, dat de tijdstempel voor het begin van de maanweek retourneert voordat de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        lunarweekstart(date,-1) as previous_lunar_week_start,
        timestamp(lunarweekstart(date,-1)) as previous_lunar_week_start_timestamp
    ;

Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```



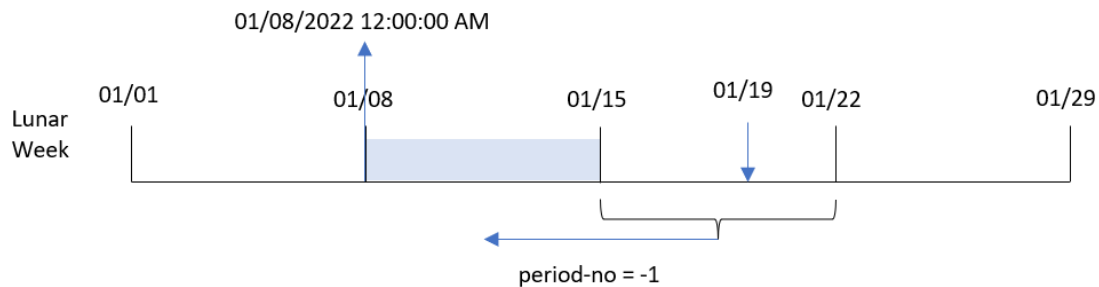
### Resultaten

Resultatentabel

date	previous_lunar_week_start	previous_lunar_week_start_timestamp
1/7/2022	12/24/2021	12/24/2021 12:00:00 AM
1/19/2022	01/08/2022	1/8/2022 12:00:00 AM
2/5/2022	01/29/2022	1/29/2022 12:00:00 AM
2/28/2022	02/19/2022	2/19/2022 12:00:00 AM
3/16/2022	03/05/2022	3/5/2022 12:00:00 AM
4/1/2022	03/19/2022	3/19/2022 12:00:00 AM
5/7/2022	04/30/2022	4/30/2022 12:00:00 AM
5/16/2022	05/07/2022	5/7/2022 12:00:00 AM
6/15/2022	06/04/2022	6/4/2022 12:00:00 AM
6/26/2022	06/18/2022	6/18/2022 12:00:00 AM
7/9/2022	07/02/2022	7/2/2022 12:00:00 AM
7/22/2022	07/09/2022	7/9/2022 12:00:00 AM
7/23/2022	07/16/2022	7/16/2022 12:00:00 AM
7/27/2022	07/16/2022	7/16/2022 12:00:00 AM
8/2/2022	07/23/2022	7/23/2022 12:00:00 AM
8/8/2022	07/30/2022	7/30/2022 12:00:00 AM
8/19/2022	08/06/2022	8/6/2022 12:00:00 AM
9/26/2022	09/17/2022	9/17/2022 12:00:00 AM
10/14/2022	10/01/2022	10/1/2022 12:00:00 AM
10/29/2022	10/22/2022	10/22/2022 12:00:00 AM

In dit geval, omdat een `period_no` van -1 werd gebruikt als het `offset`-argument in de `lunarweekstart()`-functie, identificeert de functie eerst de maanweek waarin de transacties plaatsvinden. Het verschuift dan naar een week eerder en identificeert de eerste milliseconde van die maanweek.

Diagram van `Lunarweekstart()`-functie, voorbeeld `period_no`



Transactie 8189 vond plaats op 19 januari. De `Lunarweekstart()`-functie identificeert dat de maanweek op 15 januari begint. Daarom begon de vorige maanweek op 8 januari om 12:00:00 AM; dit is de waarde die voor het `previous_lunar_week_start`-veld wordt geretourneerd.

### Voorbeeld 3 - `first_week_day`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld stellen we het begin van de maanweek in op 5 januari.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,  
lunarweekstart(date,0,4) as start_of_week,  
timestamp(lunarweekstart(date,0,4)) as start_of_week_timestamp  
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount  
8188,1/7/2022,17.17  
8189,1/19/2022,37.23  
8190,2/28/2022,88.27  
8191,2/5/2022,57.42  
8192,3/16/2022,53.80  
8193,4/1/2022,82.06  
8194,5/7/2022,40.39  
8195,5/16/2022,87.21  
8196,6/15/2022,95.93  
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- start\_of\_week
- start\_of\_week\_timestamp

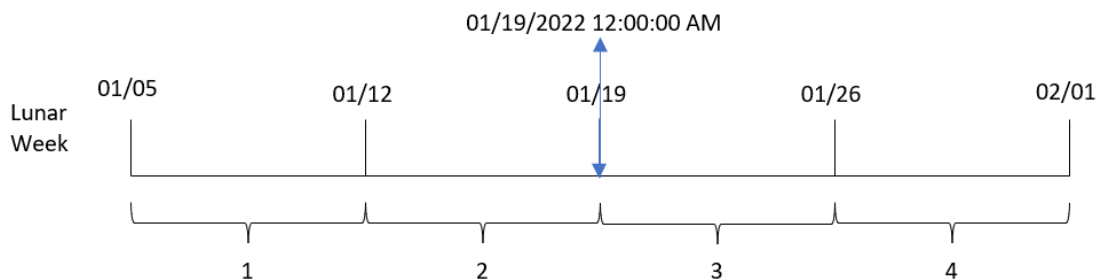
Resultatentabel

date	start_of_week	start_of_week_timestamp
1/7/2022	01/05/2022	1/5/2022 12:00:00 AM
1/19/2022	01/19/2022	1/19/2022 12:00:00 AM
2/5/2022	02/02/2022	2/2/2022 12:00:00 AM
2/28/2022	02/23/2022	2/23/2022 12:00:00 AM
3/16/2022	03/16/2022	3/16/2022 12:00:00 AM
4/1/2022	03/30/2022	3/30/2022 12:00:00 AM
5/7/2022	05/04/2022	5/4/2022 12:00:00 AM
5/16/2022	05/11/2022	5/11/2022 12:00:00 AM
6/15/2022	06/15/2022	6/15/2022 12:00:00 AM
6/26/2022	06/22/2022	6/22/2022 12:00:00 AM
7/9/2022	07/06/2022	7/6/2022 12:00:00 AM
7/22/2022	07/20/2022	7/20/2022 12:00:00 AM
7/23/2022	07/20/2022	7/20/2022 12:00:00 AM
7/27/2022	07/27/2022	7/27/2022 12:00:00 AM
8/2/2022	07/27/2022	7/27/2022 12:00:00 AM
8/8/2022	08/03/2022	8/3/2022 12:00:00 AM
8/19/2022	08/17/2022	8/17/2022 12:00:00 AM

date	start_of_week	start_of_week_timestamp
9/26/2022	09/21/2022	9/21/2022 12:00:00 AM
10/14/2022	10/12/2022	10/12/2022 12:00:00 AM
10/29/2022	10/26/2022	10/26/2022 12:00:00 AM

In dit geval verschuift het begin van het jaar van 1 januari naar 5 januari omdat het `first_week_date`-argument 4 in de `lunarweekstart()`-functie is gebruikt.

Diagram van de `lunarweekstart()`-functie, voorbeeld met `first_week_day`



Transactie 8189 vond plaats op 19 januari. Omdat maanweken op 5 januari beginnen, identificeert de `Lunarweekstart()`-functie dat de maanweek waarin 19 januari valt, ook met 19 januari om 12:00:00 AM begint. Daarom is dat de waarde die voor het `start_of_week`-veld wordt geretourneerd.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het begin van de maanweek waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de applicatie.

#### Load-script

```

Transactions:
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23

```

```

8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];

```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Voeg de volgende metingen toe:

```
=lunarweekstart(date)
```

```
=timestamp(lunarweekstart(date))
```

Resultatentabel

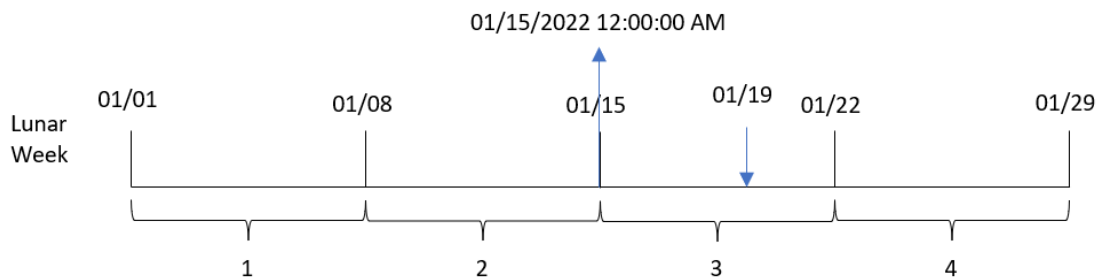
date	=lunarweekstart(date)	=timestamp(lunarweekstart(date))
1/7/2022	01/01/2022	1/1/2022 12:00:00 AM
1/19/2022	01/15/2022	1/15/2022 12:00:00 AM
2/5/2022	02/05/2022	2/5/2022 12:00:00 AM
2/28/2022	02/26/2022	2/26/2022 12:00:00 AM
3/16/2022	03/12/2022	3/12/2022 12:00:00 AM
4/1/2022	03/26/2022	3/26/2022 12:00:00 AM
5/7/2022	05/07/2022	5/7/2022 12:00:00 AM
5/16/2022	05/14/2022	5/14/2022 12:00:00 AM
6/15/2022	06/11/2022	6/11/2022 12:00:00 AM
6/26/2022	06/25/2022	6/25/2022 12:00:00 AM
7/9/2022	07/09/2022	7/9/2022 12:00:00 AM
7/22/2022	07/16/2022	7/16/2022 12:00:00 AM

date	=lunarweekstart(date)	=timestamp(lunarweekstart(date))
7/23/2022	07/23/2022	7/23/2022 12:00:00 AM
7/27/2022	07/23/2022	7/23/2022 12:00:00 AM
8/2/2022	07/30/2022	7/30/2022 12:00:00 AM
8/8/2022	08/06/2022	8/6/2022 12:00:00 AM
8/19/2022	08/13/2022	8/13/2022 12:00:00 AM
9/26/2022	09/24/2022	9/24/2022 12:00:00 AM
10/14/2022	10/08/2022	10/8/2022 12:00:00 AM
10/29/2022	10/29/2022	10/29/2022 12:00:00 AM

De meting `start_of_week` wordt gemaakt in het diagramobject met behulp van de functie `lunarweekstart()` en geeft het veld door als het argument van de functie.

De `lunarweekstart()`-functie identificeert in welke maanweek de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die week.

*Diagram van `lunarweekstart()`-functie, diagramobjectvoorbeeld*



Transactie 8189 vond plaats op 19 januari. De `lunarweekstart()`-functie identificeert dat de maanweek op 15 januari begint. De `start_of_week`-waarde voor die transactie is daarom de eerste milliseconde van die dag, en dat is 15 januari om 12:00:00 AM.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set leningsaldo's die wordt geladen in de tabel Loans.
- Gegevens bestaan uit lening-id's, het saldo aan het begin van de week en de enkelvoudige rente die op iedere lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd op elke lening in de week tot heden.

### Load-script

```
Loans:
Load
*
Inline
[
loan_id,start_balance,rate
8188,$10000.00,0.024
8189,$15000.00,0.057
8190,$17500.00,0.024
8191,$21000.00,0.034
8192,$90000.00,0.084
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Voeg de volgende velden als dimensies toe:
  - loan\_id
  - start\_balance
3. Maak nu de volgende meting om de opgebouwde rente te berekenen:  
 $=start\_balance*(rate*(today(1)-lunarweekstart(today(1)))/365)$
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

loan_id	start_balance	=start_balance*(rate*(today(1)- lunarweekstart(today(1)))/365)
8188	\$10000.00	\$15.07
8189	\$15000.00	\$128.84
8190	\$17500.00	\$63.29
8191	\$21000.00	\$107.59
8192	\$90000.00	\$1139.18

De functie `1unarweekstart()` gebruikt de datum van vandaag als enige argument en retourneert de begindatum van het huidige jaar. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking het aantal dagen dat in deze week tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor deze periode te retourneren. Het resultaat wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die deze week tot nu toe is opgebouwd.

### makedate

Deze functie retourneert een datum die is berekend voor het jaar **YYYY**, de maand **MM** en de dag **DD**.

#### Syntaxis:

```
MakeDate(YYYY [ , MM [ , DD ] ])
```

**Retourgegevenstypen:** dual

#### Argumenten

Argument	Beschrijving
YYYY	Het jaartal als een geheel getal.
MM	De maand als een geheel getal. Als geen maand is vermeld, wordt 1 (januari) verondersteld.
DD	De dag als een geheel getal. Als geen dag is vermeld, wordt 1 (de 1ste) verondersteld.

### Wanneer gebruiken

De `makedate()`-functie wordt normaal gesproken in het script voor gegevensgenerering om een kalender te genereren. Dit kan ook worden gebruikt wanneer het gegevensveld niet rechtstreeks als datum beschikbaar is, maar wat transformaties behoeft om jaar-, maand- en dagcomponenten te extraheren.

Deze voorbeelden gebruiken de datumopmaak `MM/DD/YYYY`. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>makedate(2012)</code>	Retourneert 01/01/2012.
<code>makedate(12)</code>	Retourneert 01/01/2012.
<code>makedate(2012,12)</code>	Retourneert 12/01/2012.
<code>makedate(2012,2,14)</code>	Retourneert 02/14/2012.



### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2018 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het veld `transaction_date` wordt gemaakt dat een datum retourneert met de notatie mm/dd/yyyy.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    makedate(transaction_year, transaction_month, transaction_day) as transaction_date
  ;
Load * Inline [
transaction_id, transaction_year, transaction_month, transaction_day, transaction_amount,
transaction_quantity, customer_id
3750, 2018, 08, 30, 12423.56, 23, 2038593
3751, 2018, 09, 07, 5356.31, 6, 203521
3752, 2018, 09, 16, 15.75, 1, 5646471
3753, 2018, 09, 22, 1251, 7, 3036491
3754, 2018, 09, 22, 21484.21, 1356, 049681
3756, 2018, 09, 22, -59.18, 2, 2038593
```

```
3757, 2018, 09, 23, 3177.4, 21, 203521  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- transaction\_year
- transaction\_month
- transaction\_day
- transaction\_date

Resultatentabel

transaction_year	transaction_month	transaction_day	transaction_date
2018	08	30	08/30/2018
2018	09	07	09/07/2018
2018	09	16	09/16/2018
2018	09	22	09/22/2018
2018	09	23	09/23/2018

Het veld `transaction_date` wordt gemaakt in de voorafgaande load-instructie met behulp van de `makedate` ()-functie en geeft de jaar-, maand- en dagvelden door als de argumenten van de functie.

De functie combineert en converteert deze waarden in een datumveld en retourneert de resultaten met de notatie van de `DateFormat`-systeemvariabele.

### Voorbeeld 2 - Aangepaste DateFormat

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het veld `transaction_date` wordt gemaakt met de notatie `dd/mm/jjjj` zonder de `DateFormat`-systeemvariabele aan te passen.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load  
    * ,
```

```
        date(makedate(transaction_year, transaction_month, transaction_day), 'DD/MM/YYYY') as
transaction_date
    ;
Load * Inline [
transaction_id, transaction_year, transaction_month, transaction_day, transaction_amount,
transaction_quantity, customer_id
3750, 2018, 08, 30, 12423.56, 23, 2038593
3751, 2018, 09, 07, 5356.31, 6, 203521
3752, 2018, 09, 16, 15.75, 1, 5646471
3753, 2018, 09, 22, 1251, 7, 3036491
3754, 2018, 09, 22, 21484.21, 1356, 049681
3756, 2018, 09, 22, -59.18, 2, 2038593
3757, 2018, 09, 23, 3177.4, 21, 203521
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- transaction\_year
- transaction\_month
- transaction\_day
- transaction\_date

Resultatentabel

transaction_year	transaction_month	transaction_day	transaction_date
2018	08	30	30/08/2018
2018	09	07	07/09/2018
2018	09	16	16/09/2018
2018	09	22	22/09/2018
2018	09	23	23/09/2018

In dit geval is de `makedate()`-functie in de `date()`-functie genest. Het tweede argument van de `date()`-functie stelt de datumnotatie van de `makedate()`-functieresultaten in op de vereiste `dd/mm/jjjj`.

### Voorbeeld 3 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2018 die wordt geladen in een tabel met de naam Transactions.
- De transactiedatums worden ingevoerd in twee velden: year en month.

Maak een diagramobjectmeting, transaction\_date, die een datum retourneert met de notatie mm/dd/yyyy.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
Load * Inline [
transaction_id, transaction_year, transaction_month, transaction_amount, transaction_quantity,
customer_id
3750, 2018, 08, 12423.56, 23, 2038593
3751, 2018, 09, 5356.31, 6, 203521
3752, 2018, 09, 15.75, 1, 5646471
3753, 2018, 09, 1251, 7, 3036491
3754, 2018, 09, 21484.21, 1356, 049681
3756, 2018, 09, -59.18, 2, 2038593
3757, 2018, 09, 3177.4, 21, 203521
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- year
- month

Om de transaction\_date vast te stellen, maakt u deze meting:

```
=makedate(transaction_year,transaction_month)
```

Resultatentabel

transaction_year	transaction_month	transaction_date
2018	08	08/01/2018
2018	09	09/01/2018

De meting transaction\_date wordt gemaakt in het diagramobject met behulp van de makedate()-functie en passeert de jaar- en maandvelden als functieargumenten.

De functie combineert deze waarden vervolgens, evenals de aangenomen dagwaarde van 01. Deze waarden worden geconverteerd in een datumveld en retourneren de resultaten met de indeling van de DateFormat-systeemvariabele.

### Voorbeeld 4 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Maak een kalendergegevensset voor het kalenderjaar 2022.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';

Calendar:
    load
        *
        where year(date)=2022;
    load
        date(recno()+makedate(2021,12,31)) as date
    AutoGenerate 400;
```

#### Resultaten

Resultatentabel

date
01/01/2022
01/02/2022
01/03/2022
01/04/2022
01/05/2022
01/06/2022
01/07/2022
01/08/2022
01/09/2022
01/10/2022
01/11/2022
01/12/2022
01/13/2022
01/14/2022
01/15/2022

<b>date</b>
01/16/2022
01/17/2022
01/18/2022
01/19/2022
01/20/2022
01/21/2022
01/22/2022
01/23/2022
01/24/2022
01/25/2022
Nog 340 rijen

De `makedate()`-functie maakt een datumwaarde voor 31 december 2021. De `recno()`-functie levert het recordnummer van de huidige record die in de tabel wordt geladen, beginnend bij 1. Daarom heeft de eerste record de datum 1 januari 2022. Iedere achtereenvolgende `recno()` verhoogt deze datum met 1. Deze uitdrukking is verpakt in een `date()`-functie om de waarde om te zetten in een datum. Dit proces wordt 400 keer herhaald door de `autogenerate`-functie. Als laatste kan een `where`-voorwaarde worden gebruikt om alleen datums uit het jaar 2022. Dit kan door het gebruik van een voorafgaande lading. Dit script genereert een kalender met iedere datum in 2022.

### maketime

Deze functie retourneert een tijd die is berekend voor het uur **hh**, de minuut **mm** en de seconde **ss**.

#### Syntaxis:

```
MakeTime(hh [ , mm [ , ss ] ])
```

**Retourgegevenstypen:** dual

#### Argumenten

Argument	Beschrijving
hh	Het uur als een geheel getal.
mm	De minuut als een geheel getal. Als geen minuut is aangegeven, wordt 00 verondersteld.
ss	De seconde als een geheel getal. Als geen seconde is aangegeven, wordt 00 verondersteld.

### Wanneer gebruiken

De `maketime()`-functie wordt normaal gesproken in het script voor het genereren van gegevens gebruikt om een tijdveld te genereren. Soms, als het tijdveld wordt afgeleid van invoertekst, kan deze functie worden gebruikt om de tijd op te bouwen met de verschillende componenten.

Deze voorbeelden gebruiken de tijdnotatie `h:mm:ss`. De tijdnotatie wordt opgegeven in de opdracht `SET TimeFormat` boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>maketime(22)</code>	Retourneert 22:00:00.
<code>maketime(22, 17)</code>	Retourneert 22:17:00.
<code>maketime(22,17,52 )</code>	Retourneert 22:17:52.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: `MM/DD/YYYY`. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - `maketime()`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties die wordt geladen in de tabel `Transactions`.
- De transactietijden worden ingevoerd in drie velden: `hours`, `minutes` en `seconds`.
- Het veld `transaction_time` wordt gemaakt dat de tijd retourneert met de notatie van de `TimeFormat`-systeemvariabele.

### Load-script

```
SET TimeFormat='h:mm:ss TT';

Transactions:
  Load
    *,
    maketime(transaction_hour, transaction_minute, transaction_second) as transaction_time
  ;
Load * Inline [
transaction_id, transaction_hour, transaction_minute, transaction_second, transaction_amount,
transaction_quantity, customer_id
3750, 18, 43, 30, 12423.56, 23, 2038593
3751, 6, 32, 07, 5356.31, 6, 203521
3752, 12, 09, 16, 15.75, 1, 5646471
3753, 21, 43, 41, 7, 3036491
3754, 17, 55, 22, 21484.21, 1356, 049681
3756, 2, 52, 22, -59.18, 2, 2038593
3757, 9, 25, 23, 3177.4, 21, 203521
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- transaction\_hour
- transaction\_minute
- transaction\_second
- transaction\_time

Resultatentabel

transaction_hour	transaction_minute	transaction_second	transaction_time
2	52	22	2:52:22 AM
6	32	07	6:32:07 AM
9	25	23	9:25:23 AM
12	09	16	12:09:16 PM
17	55	22	5:55:22 PM
18	43	30	6:43:30 PM
21	43	41	9:43:41 PM

Het veld `transaction_time` wordt gemaakt in de voorafgaande load-instructie met behulp van de `maketime()`-functie en geeft de uur-, minuut- en secondevelden door als de argumenten van de functie.

De functie combineert en converteert deze waarden in een tijdveld en retourneert de resultaten met de tijdnotatie van de `TimeFormat`-systeemvariabele.



### Voorbeeld 2 - time()-functie

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het veld `transaction_time` wordt gemaakt waarmee we de resultaten kunnen bekijken in een 24-uurs tijdnotatie zonder de `TimeFormat`-systeemvariabele aan te passen.

#### Load-script

```
SET TimeFormat='h:mm:ss TT';

Transactions:
  Load
    *,
    time(maketime(transaction_hour, transaction_minute, transaction_second),'h:mm:ss') as
transaction_time
  ;
Load * Inline [
transaction_id, transaction_hour, transaction_minute, transaction_second, transaction_amount,
transaction_quantity, customer_id
3750, 18, 43, 30, 12423.56, 23, 2038593
3751, 6, 32, 07, 5356.31, 6, 203521
3752, 12, 09, 16, 15.75, 1, 5646471
3753, 21, 43, 41, 7, 3036491
3754, 17, 55, 22, 21484.21, 1356, 049681
3756, 2, 52, 22, -59.18, 2, 2038593
3757, 9, 25, 23, 3177.4, 21, 203521
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `transaction_hour`
- `transaction_minute`
- `transaction_second`
- `transaction_time`

Resultatentabel

<code>transaction_hour</code>	<code>transaction_minute</code>	<code>transaction_second</code>	<code>transaction_time</code>
2	52	22	2:52:22

transaction_hour	transaction_minute	transaction_second	transaction_time
6	32	07	6:32:07
9	25	23	9:25:23
12	09	16	12:09:16
17	55	22	17:55:22
18	43	30	18:43:30
21	43	41	21:43:41

In dit geval is de `maketime()`-functie in de `time()`-functie genest. Het tweede argument van de `time()`-functie stelt de notatie van de `maketime()`-functieresultaten in op de vereiste `h:mm:ss`.

### Voorbeeld 3 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties die wordt geladen in de tabel `Transactions`.
- De transactietijden worden ingevoerd in twee velden: `hours` en `minutes`.
- Het veld `transaction_time` wordt gemaakt dat de tijd retourneert met de notatie van de `TimeFormat`-systeemvariabele.

Maak een diagramobjectmeting, `transaction_time`, die een tijd retourneert met de notatie `h:mm:ss TT`.

#### Load-script

```
SET TimeFormat='h:mm:ss TT';
```

```
Transactions:
```

```
Load * Inline [
```

```
transaction_id, transaction_hour, transaction_minute, transaction_amount, transaction_
```

```
quantity, customer_id
```

```
3750, 18, 43, 12423.56, 23, 2038593
```

```
3751, 6, 32, 5356.31, 6, 203521
```

```
3752, 12, 09, 15.75, 1, 5646471
```

```
3753, 21, 43, 7, 3036491
```

```
3754, 17, 55, 21484.21, 1356, 049681
```

```
3756, 2, 52, -59.18, 2, 2038593
```

```
3757, 9, 25, 3177.4, 21, 203521
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- transaction\_hour
- transaction\_minute

Om de transaction\_time te berekenen, maakt u deze meting:

```
=maketime(transaction_hour,transaction_minute)
```

Resultatentabel

transaction_hour	transaction_minute	=maketime(transaction_hour, transaction_minute)
2	52	2:52:00 AM
6	32	6:32:00 AM
9	25	9:25:00 AM
12	09	12:09:00 PM
17	55	5:55:00 PM
18	43	6:43:00 PM
21	43	9:43:00 PM

De meting transaction\_time wordt gemaakt in het diagramobject met behulp van de maketime()-functie en passeert de uur- en minuutvelden als functieargumenten.

De functie combineert deze waarden vervolgens en de secondeteller wordt op 00 gezet. Deze waarden worden geconverteerd in een tijdveld en retourneren de resultaten met de notatie van de TimeFormat-systeemvariabele.

### Voorbeeld 4 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Maak een kalendergegevensset voor januari 2022, opgesplitst in delen van acht uur.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';

tmpCalendar:
    load
        *
        where year(date)=2022;
load
    date(recno()+makedate(2021,12,31)) as date
```

```
AutoGenerate 31;
```

```
Left join(tmpCalendar)  
Load  
    maketime((recno()-1)*8,00,00) as time  
autogenerate 3;
```

```
Calendar:  
Load  
    timestamp(date + time) as timestamp  
resident tmpCalendar;
```

```
drop table tmpCalendar;
```

### Resultaten

Resultatentabel

<b>tijdstempel</b>
1/1/2022 12:00:00 AM
1/1/2022 8:00:00 AM
1/1/2022 4:00:00 PM
1/2/2022 12:00:00 AM
1/2/2022 8:00:00 AM
1/2/2022 4:00:00 PM
1/3/2022 12:00:00 AM
1/3/2022 8:00:00 AM
1/3/2022 4:00:00 PM
1/4/2022 12:00:00 AM
1/4/2022 8:00:00 AM
1/4/2022 4:00:00 PM
1/5/2022 12:00:00 AM
1/5/2022 8:00:00 AM
1/5/2022 4:00:00 PM
1/6/2022 12:00:00 AM
1/6/2022 8:00:00 AM
1/6/2022 4:00:00 PM
1/7/2022 12:00:00 AM
1/7/2022 8:00:00 AM

tijdstempel
1/7/2022 4:00:00 PM
1/8/2022 12:00:00 AM
1/8/2022 8:00:00 AM
1/8/2022 4:00:00 PM
1/9/2022 12:00:00 AM
Nog 68 rijen

De aanvankelijke autogenerate-functie maakt een kalender met alle datums in januari in de tabel `tmpCalendar`.

Er wordt een tweede tabel gemaakt met drie records. Voor iedere record, wordt `recno() - 1` afgetrokken (waarden 0, 1, 2) en het resultaat wordt vermenigvuldigd met 8. Het resultaat is dat dit de waarden 0, 8 16 genereert. Deze waarden worden gebruikt als de uurparameter in een `maketime()`-functie met minuut- en secondewaarden van 0. Het resultaat is dat de tabel drie tijdvakken bevat: 12:00:00 AM, 8:00:00 AM en 4:00:00 PM.

Deze tabel wordt samengevoegd met de `tmpCalendar`-tabel. Omdat de twee tabellen geen overeenkomende velden bevatten voor de samenvoeging, worden er aan iedere datumrij tijdrijen toegevoegd. Daardoor wordt iedere datumrij nu drie keer met iedere tijdwaarde herhaald.

Als laatste wordt de tabel Kalender gemaakt vanuit een resident-load van de `tmpCalendar`-tabel. De datum- en tijdvakken worden aaneengeschakeld en in de `timestamp()`-functie verpakt om het tijdstempelveld te maken.

De `tmpCalendar`-tabel wordt dan verwijderd.

### makeweekdate

Deze functie retourneert een datum die is berekend voor het jaar **YYYY**, de week **WW** en de dag van de week **D**.

#### Syntaxis:

```
MakeWeekDate (YYYY [ , WW [ , D ] ])
```

#### Retourgegevenstypen: dual

De `makeweekdate()`-functie is beschikbaar zowel als een script- als een diagramfunctie. De functie berekent de datum op basis van de parameters die in de functie worden gepasseerd. Als de dag van de week-parameter wordt weggelaten, retourneert de functie de datum van de maandag van die week.

De `makeweekdate()`-functie houdt geen rekening met de systeemvariabelen `BrokenWeek`, `ReferenceDay` of `FirstWeekDay`. Week 1 begint op de eerste maandag in januari. Zo begint week 1 in 2022 op 3 januari.

### Argumenten

Argument	Beschrijving
YYYY	Het jaartal als een geheel getal.
WW	De week als een geheel getal.  De week kan positief of negatief zijn en kan groter zijn dan 52 om datums in andere jaren te retourneren.
D	De dag van de week als een geheel getal.  Als er geen dag van de week is aangegeven, wordt 0 (maandag) verondersteld. De resterende dagen van de week worden als volgt toegewezen: 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag.

### Wanneer gebruiken

De `makeweekdate()`-functie wordt normaliter gebruikt in het script voor het genereren van gegevens om een lijst met datums te genereren of om datums op te bouwen wanneer het jaar, de week of dag van de week in de invoergegevens zijn verstrekt.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>makeweekdate(2014, 6, 6)</code>	retourneert 02/09/2014
<code>makeweekdate(2014, 6, 1)</code>	retourneert 02/04/2014
<code>makeweekdate(2014, 6)</code>	retourneert 02/03/2014 (er wordt uitgegaan van weekdag 0)

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Inclusief dag

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een totale verkoop per week voor 2022 in de tabel `sales`.
- De transactiedatums worden ingevoerd in drie velden: `year`, `week` en `sales`.
- Een voorafgaande lading die wordt gebruikt om een meting te maken, `end_of_week`, met de `makeweekdate()`-functie om de datum voor de vrijdag van die week te retourneren met de notitie `mm/dd/jjjj`.

Om aan te tonen dat de geretourneerde datum een vrijdag is, wordt de `end_of_week`-uitdrukking ook verpakt in de `weekday()`-functie om de dag van de week te laten zien.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
    *,
    makeweekdate(transaction_year, transaction_week,4) as end_of_week,
    weekday(makeweekdate(transaction_year, transaction_week,4)) as week_day
;
```

```
Load * Inline [
transaction_year, transaction_week, sales
2022, 01, 10000
2022, 02, 11250
2022, 03, 9830
2022, 04, 14010
2022, 05, 28402
2022, 06, 9992
2022, 07, 7292
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `transaction_year`
- `transaction_week`
- `end_of_week`
- `week_day`

Resultatentabel

transaction_year	transaction_week	end_of_week	week_day
2022	01	01/07/2022	Vr
2022	02	01/14/2022	Vr
2022	03	01/21/2022	Vr
2022	04	01/28/2022	Vr
2022	05	02/04/2022	Vr
2022	06	02/11/2022	Vr
2022	07	02/18/2022	Vr

Het veld `end_of_week` wordt gemaakt in de vorige load-instructie met behulp van de functie `makeweekdate()`. De velden `transaction_year` en `transaction_week` worden als jaar- en weekargumenten in de functie gepasseerd. Er wordt een waarde van 4 gebruikt voor het dagargument.

De functie combineert en converteert deze waarden in een datumveld en retourneert de resultaten met de notatie van de `DateFormat`-systeemvariabele.

De `makeweekdate()`-functie en de bijbehorende argumenten worden ook verpakt in een `weekday()`-functie om het `week_day`-veld te retourneren, en zoals in de bovenstaande tabel te zien is, toont het `week_day`-veld dat deze datums op een vrijdag vallen.

### Voorbeeld 2 - Exclusief dag

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een totale verkoop per week voor 2022 in de tabel `sales`.
- De transactiedatums worden ingevoerd in drie velden: `year`, `week` en `sales`.
- Een voorafgaande lading die wordt gebruikt om een meting te maken, `first_day_of_week`, met behulp van de `makeweekdate()`-functie. Dit retourneert de datum voor de maandag van die week met de notatie `mm/dd/yyyy`.

Om aan te tonen dat de geretourneerde datum een maandag is, wordt de `first_day_of_week`-uitdrukking ook verpakt in de `weekday()`-functie om de dag van de week te laten zien.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';  
SET FirstWeekDay=6;
```



Transactions:

```
Load
    *,
    makeweekdate(transaction_year, transaction_week) as first_day_of_week,
    day(makeweekdate(transaction_year, transaction_week)) as week_day
;
Load * Inline [
transaction_year, transaction_week, sales
2022, 01, 10000
2022, 02, 11250
2022, 03, 9830
2022, 04, 14010
2022, 05, 28402
2022, 06, 9992
2022, 07, 7292
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- transaction\_year
- transaction\_week
- first\_day\_of\_week
- week\_day

Resultatentabel

transaction_year	transaction_week	first_day_of_week	week_day
2022	01	01/03/2022	Ma
2022	02	01/10/2022	Ma
2022	03	01/17/2022	Ma
2022	04	01/24/2022	Ma
2022	05	01/31/2022	Ma
2022	06	02/07/2022	Ma
2022	07	02/14/2022	Ma

Het veld `first_day_of_week` wordt gemaakt in de vorige load-instructie met behulp van de functie `makeweekdate()`. De parameters `transaction_year` en `transaction_week` worden als functie-argumenten gepasseerd en de dagparameter blijft leeg.

De functie combineert en converteert deze waarden in een datumveld en retourneert de resultaten met de notatie van de `DateFormat`-systeemvariabele.

De `makeweekdate()`-functie en de bijbehorende argumenten worden ook verpakt in een `weekday()`-functie om het `week_day`-veld te retourneren. Zoals te zien is in de bovenstaande tabel, toont het `week_day`-veld dat deze datums op een maandag vallen (ook al is de `FirstWeekDay`-variabele ingesteld op zondag als de eerste dag van de week), omdat de dagparameter in de `makeweekdate()`-functie is leeggelaten.

### Voorbeeld 3 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een totale verkoop per week voor 2022 in de tabel `sales`.
- De transactiedatums worden ingevoerd in drie velden: `year`, `week` en `sales`.

In dit voorbeeld wordt een diagramobject gebruikt om een meting te maken die equivalent is aan de `end_of_week`-berekening uit het eerste voorbeeld. Deze meting gebruikt de `makeweekdate()`-functie om de datum te retourneren voor de vrijdag van die week met de notatie `mm/dd/jjjj`.

Om te bewijzen dat de geretourneerde datum een vrijdag is, wordt er een tweede meting gemaakt om de dag van de week te retourneren.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';

Master_Calendar:
Load * Inline [
transaction_year, transaction_week, sales
2022, 01, 10000
2022, 02, 11250
2022, 03, 9830
2022, 04, 14010
2022, 05, 28402
2022, 06, 9992
2022, 07, 7292
];
```

#### Resultaten

##### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
  - `transaction_year`
  - `transaction_week`
2. Om de berekening uit te voeren die equivalent is aan die uit het `end_of_week`-veld uit het eerste voorbeeld, maakt u de volgende meting:  
`=makeweekdate(transaction_year, transaction_week, 4)`
3. Maak de volgende meting om de dag van de week per transactie te berekenen:  
`=weekday(makeweekdate(transaction_year, transaction_week, 4))`

Resultatentabel

transaction_ year	transaction_ week	=makeweekdate (transaction_ year,transaction_ week,4)	=weekday(makeweekdate (transaction_ year,transaction_ week,4))
2022	01	01/07/2022	Vr
2022	02	01/14/2022	Vr
2022	03	01/21/2022	Vr
2022	04	01/28/2022	Vr
2022	05	02/04/2022	Vr
2022	06	02/11/2022	Vr
2022	07	02/18/2022	Vr

Er wordt een equivalent veld voor `end_of_week` in het diagramobject gemaakt als een meting met behulp van de functie `makeweekdate()`. De velden `transaction_year` en `transaction_week` worden als jaar- en weekargumenten gepasseerd. Er wordt een waarde van 4 gebruikt voor het dagargument.

De functie combineert en converteert deze waarden in een datumveld en retourneert de resultaten met de notatie van de `DateFormat`-systeemvariabele.

De `makeweekdate()`-functie en de bijbehorende argumenten worden ook verpakt in een `weekday()`-functie om een berekening te retourneren die equivalent is aan die van het `week_day`-veld uit het eerste voorbeeld. Zoals te zien in de bovenstaande tabel, toont de laatste kolom aan de rechterkant dat deze datums op een vrijdag vallen.

### Voorbeeld 4 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

In dit voorbeeld wordt een lijst met datums gemaakt met alle vrijdagen in 2022.

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';

Calendar:
  Load
    *,
    weekday(date) as weekday
  where year(date)=2022;
Load
  makeweekdate(2022,recno()-2,4) as date
AutoGenerate 60;
```

### Resultaten

Resultatentabel

<b>date</b>	<b>weekday</b>
01/07/2022	Vr
01/14/2022	Vr
01/21/2022	Vr
01/28/2022	Vr
02/04/2022	Vr
02/11/2022	Vr
02/18/2022	Vr
02/25/2022	Vr
03/04/2022	Vr
03/11/2022	Vr
03/18/2022	Vr
03/25/2022	Vr
04/01/2022	Vr
04/08/2022	Vr
04/15/2022	Vr
04/22/2022	Vr
04/29/2022	Vr
05/06/2022	Vr
05/13/2022	Vr
05/20/2022	Vr
05/27/2022	Vr
06/03/2022	Vr
06/10/2022	Vr
06/17/2022	Vr
Nog 27 rijen	

De `makeweekdate()`-functie vindt iedere vrijdag in 2022. Door gebruik van een weekparameter van -2 worden geen datums overgeslagen. Als laatste maakt een voorafgaande lading een extra `weekday`-veld voor duidelijkheid waarmee wordt aangetoond dat iedere `date`-waarde een vrijdag is.

### minute

Deze functie retourneert een geheel getal dat de minuut representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

#### Syntaxis:

```
minute (expression)
```

**Retourgegevenstypen:** geheel getal

### Wanneer gebruiken

De `minute()`-functie is handig als u aggregaties per minuut wilt vergelijken. U kunt de functie bijvoorbeeld gebruiken als u de verdeling van het aantal activiteiten per minuut wilt bekijken.

Deze dimensies kunnen in het load-script worden gemaakt door de functie te gebruiken om een veld in een masterkalendertabel te maken. Ze zouden ook rechtstreeks in een diagram kunnen worden gebruikt als een berekende dimensie.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>minute ( '09:14:36' )</code>	Retourneert 14.
<code>minute ( '0.5555' )</code>	Retourneert 19 ( omdat $0,5555 = 13:19:55$ )

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Variabele (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met transacties per tijdstempel die wordt geladen in de tabel Transactions.
- De standaard Timestamp-systeemvariabele (M/D/YYYY h:mm:ss[.fff] TT) wordt gebruikt.
- Maak een veld, minute dat berekent wanneer de transacties hebben plaatsgevonden.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
  Load
    *,
    minute(timestamp) as minute
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,timestamp,amount
9497,'2022-01-05 19:04:57',47.25,
9498,'2022-01-03 14:21:53',51.75,
9499,'2022-01-03 05:40:49',73.53,
9500,'2022-01-04 18:49:38',15.35,
9501,'2022-01-01 22:10:22',31.43,
9502,'2022-01-05 19:34:46',13.24,
9503,'2022-01-04 22:58:34',74.34,
9504,'2022-01-06 11:29:38',50.00,
9505,'2022-01-02 08:35:54',36.34,
9506,'2022-01-06 08:49:09',74.23
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- timestamp
- minute

Resultatentabel

tijdstempel	minute
2022-01-01 22:10:22	10
2022-01-02 08:35:54	35
2022-01-03 05:40:49	40
2022-01-03 14:21:53	21
2022-01-04 18:49:38	49
2022-01-04 22:58:34	58
2022-01-05 19:04:57	4
2022-01-05 19:34:46	34
2022-01-06 08:49:09	49
2022-01-06 11:29:38	29

De waarden in het `minute`-veld zijn gemaakt met de `minute()`-functie en geven de `timestamp` door als de uitdrukking in het voorafgaande load-instructie.

### Voorbeeld 2 - Diagramobject (diagram)

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- De standaard `Timestamp`-systeemvariabele (`M/D/YYYY h:mm:ss[.fff] TT`) wordt gebruikt.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De `minute`-waarden worden berekend via een meting in een diagramobject.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,timestamp,amount
```

```
9497,'2022-01-05 19:04:57',47.25,
```

```
9498,'2022-01-03 14:21:53',51.75,
```

```
9499,'2022-01-03 05:40:49',73.53,
```

```
9500, '2022-01-04 18:49:38', 15.35,  
9501, '2022-01-01 22:10:22', 31.43,  
9502, '2022-01-05 19:34:46', 13.24,  
9503, '2022-01-04 22:58:34', 74.34,  
9504, '2022-01-06 11:29:38', 50.00,  
9505, '2022-01-02 08:35:54', 36.34,  
9506, '2022-01-06 08:49:09', 74.23  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `timestamp`.

Maak de volgende meting:

```
=minute(timestamp)
```

Resultatentabel

tijdstempel	minute
2022-01-01 22:10:22	10
2022-01-02 08:35:54	35
2022-01-03 05:40:49	40
2022-01-03 14:21:53	21
2022-01-04 18:49:38	49
2022-01-04 22:58:34	58
2022-01-05 19:04:57	4
2022-01-05 19:34:46	34
2022-01-06 08:49:09	49
2022-01-06 11:29:38	29

De waarden voor `minute` worden gemaakt met behulp van de `minute()`-functie en geven de `timestamp` door als de uitdrukking in een meting voor het diagramobject.

### Voorbeeld 3 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:



- Een gegevensset met tijdstempels die wordt gegenereerd om invoeren in een ticketbarrière te vertegenwoordigen.
- Informatie bij iedere timestamp en de bijbehorende id, die wordt geladen in de tabel Ticket\_Barrier\_Tracker.
- De standaard Timestamp-systeemvariabele (M/D/YYYY h:mm:ss[.fff] TT) wordt gebruikt.

De gebruiker wil een diagramobject dat het aantal barrière-invoeren per minuut weergeeft.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';

tmpTimestampCreator:
    load
        *
        where year(date)=2022;
load
    date(recno()+makedate(2021,12,31)) as date
AutoGenerate 1;

join load
    maketime(floor(rand()*24),floor(rand()*59),floor(rand()*59)) as time
autogenerate 10000;

Ticket_Barrier_Tracker:
load
    recno() as id,
    timestamp(date + time) as timestamp
resident tmpTimestampCreator;

drop table tmpTimestampCreator;
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een berekende dimensie met behulp van de volgende uitdrukking:  
=minute(timestamp)
3. Voeg de volgende aggregatiemeting toe om het totaal aantal invoeren te berekenen:  
=count(id)
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

minute(timestamp)	=count(id)
0	174
1	171

<code>minute(timestamp)</code>	<code>=count(id)</code>
2	175
3	165
4	188
5	176
6	158
7	187
8	178
9	178
10	197
11	161
12	166
13	184
14	159
15	161
16	152
17	160
18	176
19	164
20	170
21	170
22	142
23	145
24	155
Nog 35 rijen	

### month

Deze functie retourneert een duale waarde: een maandnaam zoals vastgelegd in de omgevingsvariabele **MonthNames**, en een geheel getal tussen 1-12. De maand wordt berekend volgens de datuminterpretatie van de uitdrukking, in overeenstemming met de standaardinterpretatie van getallen.

De functie retourneert de naam van de maand in de indeling van de systeemvariabele `MonthName` voor een specifieke datum. Dit wordt vaak gebruikt om een dagveld te maken als een dimensie in een hoofdagenda.

### Syntaxis:

`month (expression)`

**Retourgegevenstypen:** geheel getal

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>month( 2012-10-12 )</code>	Retourneert Okt
<code>month( 35648 )</code>	retourneert Aug, omdat 35648 = 1997-08-06

### Voorbeeld 1 - gegevensverzameling DateFormat (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met de naam `Master_Calendar`. De `DateFormat` systeemvariabele is ingesteld op `DD/MM/JJJJ`.
- Een voorafgaande lading maakt een extra veld met de naam `month_name` met behulp van de `month()` functie.
- Een extra veld met de naam `long_date` met behulp van de `date()` functie voor het weergeven van de volledige datum.

#### Load-script

```
SET DateFormat='DD/MM/YYYY';
```

```
Master_Calendar:
```

```
Load
```

```
    date,  
    date(date, 'dd-MMMM-YYYY') as long_date,  
    month(date) as month_name
```

```
Inline
```

```
[  
date  
03/01/2022  
03/02/2022  
03/03/2022  
03/04/2022  
03/05/2022  
03/06/2022  
03/07/2022  
03/08/2022
```

```
03/09/2022
03/10/2022
03/11/2022
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- long\_date
- month\_name

Resultatentabel

date	long_date	month_name
03/01/2022	03-January- 2022	Jan
03/02/2022	03-February- 2022	Feb
03/03/2022	03-March- 2022	Mar
03/04/2022	03-April- 2022	Apr
03/05/2022	03-May- 2022	May
03/06/2022	03-June- 2022	Jun
03/07/2022	03-July- 2022	Jul
03/08/2022	03-August- 2022	Aug
03/09/2022	03-September- 2022	Sep
03/10/2022	03-October- 2022	Okt
03/11/2022	03-November- 2022	Nov

De naam van de maand wordt juist geëvalueerd door de `month()` functie in het script.

### Voorbeeld 2 - ANSI-datums (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met de naam `master_calendar`. De `DateFormat` systeemvariabele `DD/MM/JJJJ` wordt gebruikt. De datums die zijn opgenomen in de gegevensverzameling hebben echter een standaard ANSI-datumnotatie.

- Een voorafgaande lading maakt een extra veld met de naam `month_name` met behulp van de `month()` functie.
- Een extra veld met de naam `long_date` met behulp van de `date()` functie voor het weergeven van de volledige datum.

### Load-script

```
SET DateFormat='DD/MM/YYYY';
Master_Calendar:
Load
    date,
    date(date,'dd-MMMM-YYYY') as long_date,
    month(date) as month_name
```

```
Inline
[
date
2022-01-11
2022-02-12
2022-03-13
2022-04-14
2022-05-15
2022-06-16
2022-07-17
2022-08-18
2022-09-19
2022-10-20
2022-11-21
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `long_date`
- `month_name`

Resultatentabel

<code>date</code>	<code>long_date</code>	<code>month_name</code>
03/11/2022	11-March- 2022	11
03/12/2022	11-March- 2022	12
03/13/2022	13-March- 2022	13
03/14/2022	14-March- 2022	14
03/15/2022	15-March- 2022	15
03/16/2022	16-March- 2022	16

date	long_date	month_name
03/17/2022	17-March- 2022	17
03/18/2022	18-March- 2022	18
03/19/2022	19-March- 2022	19
03/20/2022	20-March- 2022	20
03/21/2022	21-March- 2022	21

De naam van de maand wordt juist geëvalueerd door de `month()` functie in het script.

### Voorbeeld 3 - Datums zonder notatie (script)

Load-script en resultaten

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met de naam `Master_Calendar`. De `DateFormat` systeemvariabele `DD/MM/JJJJ` wordt gebruikt.
- Een voorafgaande lading maakt een extra veld met de naam `month_name` met behulp van de `month()` functie.
- De originele datum zonder notatie, met de naam `unformatted_date`.
- Een extra veld met de naam `long_date` met behulp van de `date()` functie voor het weergeven van de volledige datum.

#### Load-script

```
SET DateFormat='DD/MM/YYYY';
```

```
Master_Calendar:
```

```
Load
```

```
    unformatted_date,  
    date(unformatted_date, 'dd-MMMM-YYYY') as long_date,  
    month(unformatted_date) as month_name
```

```
Inline
```

```
[
```

```
unformatted_date
```

```
44868
```

```
44898
```

```
44928
```

```
44958
```

```
44988
```

```
45018
```

```
45048
```

```
45078
45008
45038
45068
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- unformatted\_date
- long\_date
- month\_name

Resultatentabel

unformatted_date	long_date	month_name
44868	03-January- 2022	Jan
44898	03-February- 2022	Feb
44928	03-March- 2022	Mar
44958	03-April- 2022	Apr
44988	03-May- 2022	May
45018	03-June- 2022	Jun
45048	03-July- 2022	Jul
45078	03-August- 2022	Aug
45008	03-September- 2022	Sep
45038	03-October- 2022	Okt
45068	03-November- 2022	Nov

De naam van de maand wordt juist geëvalueerd door de `month()` functie in het script.

### Voorbeeld 4 - Vervalmaand berekenen

Load-script en diagramuitdrukking

#### Overzicht

Open de Editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met bestellingen die in maart zijn geplaatst, met de naam `subscriptions`. De tabel bevat drie velden.
  - `id`
  - `order_date`
  - `amount`

### Load-script

Subscriptions:

Load

```
id,  
order_date,  
amount
```

Inline

```
[  
id,order_date,amount  
1,03/01/2022,231.24  
2,03/02/2022,567.28  
3,03/03/2022,364.28  
4,03/04/2022,575.76  
5,03/05/2022,638.68  
6,03/06/2022,785.38  
7,03/07/2022,967.46  
8,03/08/2022,287.67  
9,03/09/2022,764.45  
10,03/10/2022,875.43  
11,03/11/2022,957.35  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: `order_date`.

Maak deze meting om de maand te berekenen waarin een bestelling verloopt: `=month(order_date+180)`.

Resultatentabel

<code>order_date</code>	<code>=month(order_date+180)</code>
03/01/2022	Jul
03/02/2022	Aug
03/03/2022	Aug
03/04/2022	Sep
03/05/2022	Okt
03/06/2022	Nov



<code>order_date</code>	<code>=month(order_date+180)</code>
03/07/2022	Dec
03/08/2022	Jan
03/09/2022	Mar
03/10/2022	Apr
03/11/2022	May

De `month()` functie bepaalt op correcte wijze dat een bestelling die op 11 maart is geplaatst in juli vervalt.

## monthend

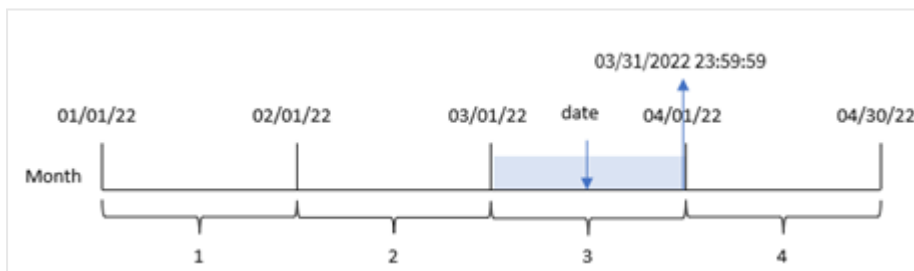
Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van de maand die `date` bevat. De standaarduitvoernotatie is de `DateFormat` die is ingesteld in het script.

### Syntaxis:

**MonthEnd**(`date`[, `period_no`])

Met andere woorden, de functie `monthend()` bepaalt in welke maand de datum valt. Hij retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van die maand.

*Diagram van monthend-functie.*



### Wanneer gebruiken

De functie `monthend()` wordt gebruikt als onderdeel van een uitdrukking als u wilt dat in de berekening een deel van de maand wordt gebruikt dat nog niet heeft plaatsgevonden. Bijvoorbeeld als u de totale rente wilt berekenen die nog niet is berekend gedurende de maand.

**Retourgegevenstypen:** dual

### Argumenten

Argument	Beschrijving
<code>date</code>	De te evalueren datum of tijdstempel.
<code>period_no</code>	<code>period_no</code> is een geheel getal dat, indien het 0 is of wordt weggelaten, de maand aangeeft die <code>date</code> bevat. Negatieve waarden in <code>period_no</code> geven voorafgaande maanden aan en positieve waarden geven volgende maanden aan.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>monthend('02/19/2012')</code>	Retourneert 02/29/2012 23:59:59.
<code>monthend('02/19/2001', -1)</code>	Retourneert 01/31/2001 23:59:59.

### Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- Een datumveld dat wordt weergegeven in de `DateFormat`-systeemvariabele met de notatie `MM/DD/YYYY`.
- Een voorgaande load-instructie met:
  - De `monthend()`-functie die is ingesteld als het veld `end_of_month`.
  - De `timestamp()`-functie die is ingesteld als het veld `end_of_month_timestamp`.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
  *,
  monthend(date) as end_of_month,
  timestamp(monthend(date)) as end_of_month_timestamp
```

```
    ;
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- end\_of\_month
- end\_of\_month\_timestamp

Resultatentabel

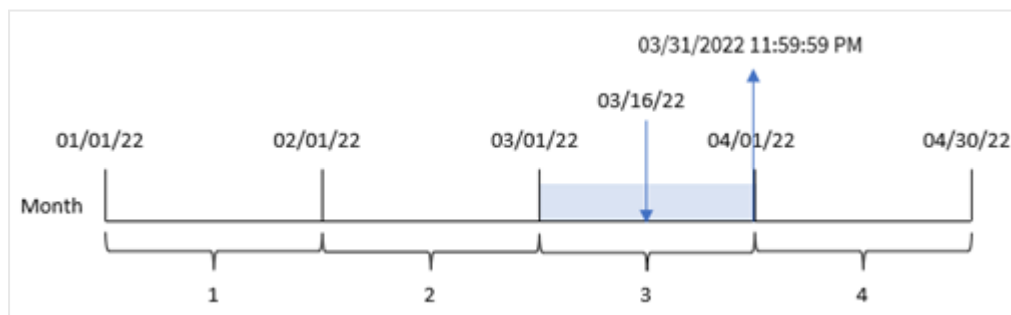
id	date	end_of_month	end_of_month_timestamp
8188	1/7/2022	01/31/2022	1/31/2022 11:59:59 PM
8189	1/19/2022	01/31/2022	1/31/2022 11:59:59 PM
8190	2/5/2022	02/28/2022	2/28/2022 11:59:59 PM
8191	2/28/2022	02/28/2022	2/28/2022 11:59:59 PM
8192	3/16/2022	03/31/2022	3/31/2022 11:59:59 PM
8193	4/1/2022	04/30/2022	4/30/2022 11:59:59 PM
8194	5/7/2022	05/31/2022	5/31/2022 11:59:59 PM

id	date	end_of_month	end_of_month_timestamp
8195	5/16/2022	05/31/2022	5/31/2022 11:59:59 PM
8196	6/15/2022	06/30/2022	6/30/2022 11:59:59 PM
8197	6/26/2022	06/30/2022	6/30/2022 11:59:59 PM
8198	7/9/2022	07/31/2022	7/31/2022 11:59:59 PM
8199	7/22/2022	07/31/2022	7/31/2022 11:59:59 PM
8200	7/23/2022	07/31/2022	7/31/2022 11:59:59 PM
8201	7/27/2022	07/31/2022	7/31/2022 11:59:59 PM
8202	8/2/2022	08/31/2022	8/31/2022 11:59:59 PM
8203	8/8/2022	08/31/2022	8/31/2022 11:59:59 PM
8204	8/19/2022	08/31/2022	8/31/2022 11:59:59 PM
8205	9/26/2022	09/30/2022	9/30/2022 11:59:59 PM
8206	10/14/2022	10/31/2022	10/31/2022 11:59:59 PM
8207	10/29/2022	10/31/2022	10/31/2022 11:59:59 PM

Het `end_of_month`-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de `monthend()`-functie en geeft het datumveld door als het argument van de functie.

De functie `monthend()` identificeert in welke maand de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die maand.

*Diagram van de `monthend`-functie waarbij maart de geselecteerde maand is.*



Transactie 8192 vond plaats op 16 maart. De functie `monthend()` retourneert de laatste milliseconde van die maand, namelijk 31 maart om 11:59:59 PM.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld wordt een veld gemaakt, `previous_month_end`, dat de tijdstempel voor het eind van de maand retourneert voordat de transactie plaatsvond.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
  *,
  monthend(date,-1) as previous_month_end,
  timestamp(monthend(date,-1)) as previous_month_end_timestamp
  ;
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

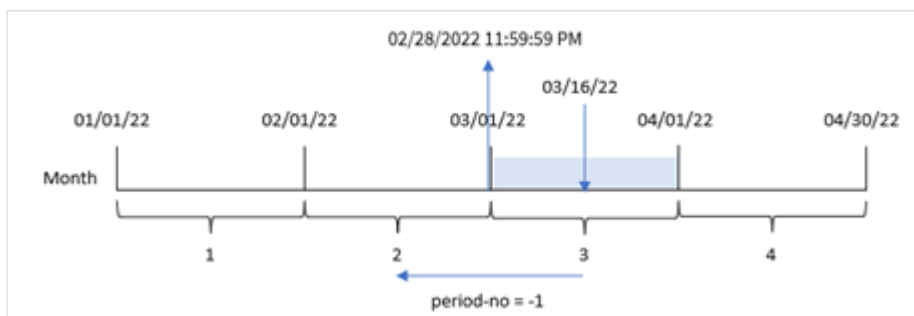
- `id`
- `date`
- `previous_month_end`
- `previous_month_end_timestamp`

Resultatentabel

id	date	previous_month_end	previous_month_end_timestamp
8188	1/7/2022	12/31/2021	12/31/2021 11:59:59 PM
8189	1/19/2022	12/31/2021	12/31/2021 11:59:59 PM
8190	2/5/2022	01/31/2022	1/31/2022 11:59:59 PM
8191	2/28/2022	01/31/2022	1/31/2022 11:59:59 PM
8192	3/16/2022	02/28/2022	2/28/2022 11:59:59 PM
8193	4/1/2022	03/31/2022	3/31/2022 11:59:59 PM
8194	5/7/2022	04/30/2022	4/30/2022 11:59:59 PM
8195	5/16/2022	04/30/2022	4/30/2022 11:59:59 PM
8196	6/15/2022	05/31/2022	5/31/2022 11:59:59 PM
8197	6/26/2022	05/31/2022	5/31/2022 11:59:59 PM
8198	7/9/2022	06/30/2022	6/30/2022 11:59:59 PM
8199	7/22/2022	06/30/2022	6/30/2022 11:59:59 PM
8200	7/23/2022	06/30/2022	6/30/2022 11:59:59 PM
8201	7/27/2022	06/30/2022	6/30/2022 11:59:59 PM
8202	8/2/2022	07/31/2022	7/31/2022 11:59:59 PM
8203	8/8/2022	07/31/2022	7/31/2022 11:59:59 PM
8204	8/19/2022	07/31/2022	7/31/2022 11:59:59 PM
8205	9/26/2022	08/31/2022	8/31/2022 11:59:59 PM
8206	10/14/2022	09/30/2022	9/30/2022 11:59:59 PM
8207	10/29/2022	09/30/2022	9/30/2022 11:59:59 PM

De `monthend()`-functie identificeert eerst de maand waarin de transacties plaatsvinden als een `period_no` van `-1` wordt gebruikt als het offset-argument. Het verschuift dan naar een maand eerder en identificeert de laatste milliseconde van die maand.

Diagram van functie `monthend` met de variabele `period_no`.



Transactie 8192 vond plaats op 16 maart. De `monthend()`-functie stelt vast dat de maand voor de transactie februari was. Vervolgens wordt de laatste milliseconde van die maand geretourneerd: 28 februari om 11:59:59 PM.

### Voorbeeld 3 - Diagramvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld wordt de gegevensverzameling ongewijzigd in de app geladen. De taak bestaat erin een berekening te maken die een tijdstempel retourneert als meting in een diagram in de app voor het eind van de maand wanneer de transacties plaatsvonden.

#### Load-script

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- id

## 5 Script- en diagramfuncties

---

Maak de volgende metingen om de einddatum van de maand waarin een transactie plaatsvindt te berekenen:

- `=monthend(date)`
- `=timestamp(monthend(date))`

Resultatentabel

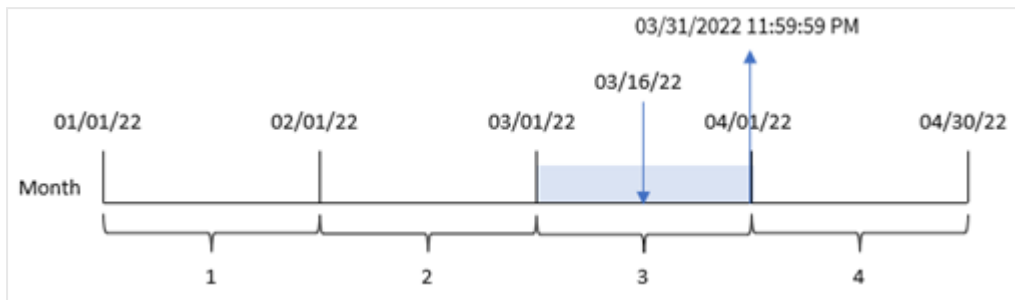
id	date	=monthend(date)	=timestamp(monthend(date))
8188	10/14/2022	10/31/2022	10/31/2022 11:59:59 PM
8189	10/29/2022	10/31/2022	10/31/2022 11:59:59 PM
8190	9/26/2022	09/30/2022	9/30/2022 11:59:59 PM
8191	8/2/2022	08/31/2022	8/31/2022 11:59:59 PM
8192	8/8/2022	08/31/2022	8/31/2022 11:59:59 PM
8193	8/19/2022	08/31/2022	8/31/2022 11:59:59 PM
8194	7/9/2022	07/31/2022	7/31/2022 11:59:59 PM
8195	7/22/2022	07/31/2022	7/31/2022 11:59:59 PM
8196	7/23/2022	07/31/2022	7/31/2022 11:59:59 PM
8197	7/27/2022	07/31/2022	7/31/2022 11:59:59 PM
8198	6/15/2022	06/30/2022	6/30/2022 11:59:59 PM
8199	6/26/2022	06/30/2022	6/30/2022 11:59:59 PM
8200	5/7/2022	05/31/2022	5/31/2022 11:59:59 PM
8201	5/16/2022	05/31/2022	5/31/2022 11:59:59 PM
8202	4/1/2022	04/30/2022	4/30/2022 11:59:59 PM
8203	3/16/2022	03/31/2022	3/31/2022 11:59:59 PM
8204	2/5/2022	02/28/2022	2/28/2022 11:59:59 PM
8205	2/28/2022	02/28/2022	2/28/2022 11:59:59 PM
8206	1/7/2022	01/31/2022	1/31/2022 11:59:59 PM
8207	1/19/2022	01/31/2022	1/31/2022 11:59:59 PM

De meting `end_of_month` wordt gemaakt in het diagram met behulp van de functie `monthend()` en geeft het veld door als het argument van de functie.

De functie `monthend()` identificeert in welke maand de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die maand.



Diagram van functie `monthend` met de variabele `period_no`.



Transactie 8192 vond plaats op 16 maart. De functie `monthend()` retourneert de laatste milliseconde van die maand, namelijk 31 maart om 11:59:59 PM.

### Voorbeeld 4 - Scenario

Load-script en resultaten

#### Overzicht

In dit voorbeeld wordt een gegevensset geladen in de tabel 'Employee\_Expenses'. De tabel bevat de volgende velden:

- Werknemers-id's
- Werknemersnamen
- De gemiddelde dagelijkse onkostendeclaraties van elke werknemer.

De eindgebruiker wil graag een diagram dat per werknemer-id en werknemersnaam de geschatte onkostendeclaraties voor de rest van de maand weergeeft.

#### Load-script

```
Employee_Expenses :
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,sydney,$27
186,Agatha,$18
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- employee\_id
- employee\_name

Maak deze meting om de opgebouwde rente te berekenen:

```
=floor(monthend(today(1),0)-today(1))*avg_daily_claim
```



*Deze meting is dynamisch en produceert verschillende tabelresultaten, afhankelijk van de datum waarop u de gegevens laadt.*

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

employee_id	employee_name	=floor(monthend(today(1),0)-today(1))*avg_daily_claim
182	Mark	\$30.00
183	Deryck	\$25.00
184	Dexter	\$25.00
185	Sydney	\$54.00
186	Agatha	\$36.00

De `monthend()`-functie retourneert de einddatum van de huidige maand door de datum van vandaag als enige argument te gebruiken. Door vervolgens de datum van vandaag af te trekken van de einddatum van de maand, retourneert de uitdrukking het aantal resterende dagen in deze maand.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in de resterende maand zal indienen.

### monthname

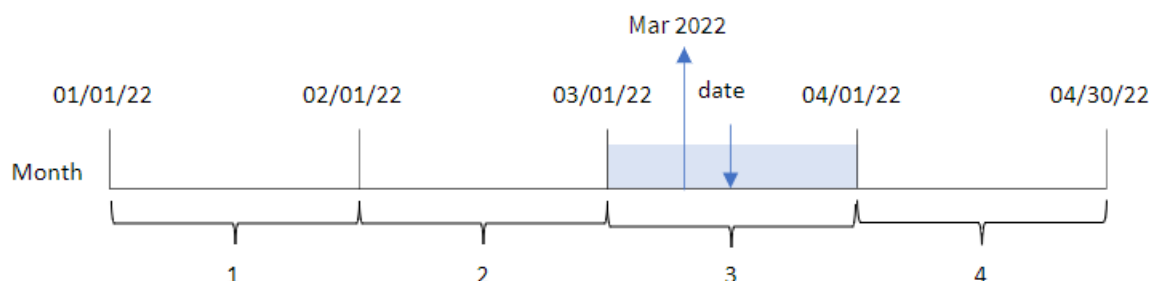
Deze functie retourneert de maand (opgemaakt volgens de scriptvariabele **MonthNames**) en het jaar met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maand.

**Syntaxis:**

```
MonthName (date[, period_no])
```

**Retourgegevenstypen:** dual

*Diagram van functie monthname*



Argumenten

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal dat, indien het 0 is of wordt weggelaten, de maand aangeeft die <b>date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande maanden aan en positieve waarden geven volgende maanden aan.

Voorbeelden van functies

Voorbeeld	Resultaat
monthname('10/19/2013')	Retourneert Oct 2013
monthname('10/19/2013', -1)	Retourneert Sep 2013

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, transaction\_month, dat de maand retourneert waarin de transacties plaatsvonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';  
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

```
Transactions:
```

```
  Load  
    *,  
    monthname(date) as transaction_month  
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

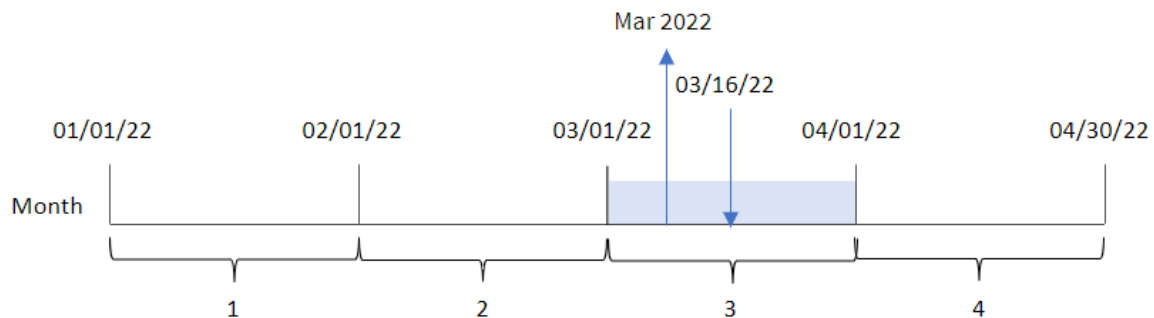
- date
- transaction\_month

Resultatentabel

date	transaction_month
1/7/2022	januari 2022
1/19/2022	januari 2022
2/5/2022	februari 2022
2/28/2022	februari 2022
3/16/2022	maart 2022
4/1/2022	april 2022
5/7/2022	mei 2022
5/16/2022	mei 2022
6/15/2022	juni 2022
6/26/2022	juni 2022
7/9/2022	juli 2022
7/22/2022	juli 2022
7/23/2022	juli 2022
7/27/2022	juli 2022
8/2/2022	augustus 2022
8/8/2022	augustus 2022
8/19/2022	augustus 2022
9/26/2022	september 2022
10/14/2022	oktober 2022
10/29/2022	oktober 2022

Het veld `transaction_month` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `monthname()` en geeft het veld `date` door als het argument van de functie.

Diagram van functie `monthname`, basisvoorbeeld



De functie `monthname()` identificeert dat transactie 8192 plaatsvond in maart 2022 en retourneert deze waarde met behulp van de systeemvariabele `MonthNames`.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde inline gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `transaction_previous_month`, dat de tijdstempel voor het einde van de maand retourneert voordat de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';  
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

Transactions:

```
Load  
*,  
monthname(date,-1) as transaction_previous_month  
;
```

Load

\*

Inline

[

```
id,date,amount  
8188,1/7/2022,17.17  
8189,1/19/2022,37.23  
8190,2/28/2022,88.27  
8191,2/5/2022,57.42  
8192,3/16/2022,53.80  
8193,4/1/2022,82.06  
8194,5/7/2022,40.39
```

```
8195, 5/16/2022, 87.21
8196, 6/15/2022, 95.93
8197, 6/26/2022, 45.89
8198, 7/9/2022, 36.23
8199, 7/22/2022, 25.66
8200, 7/23/2022, 82.77
8201, 7/27/2022, 69.98
8202, 8/2/2022, 76.11
8203, 8/8/2022, 25.12
8204, 8/19/2022, 46.23
8205, 9/26/2022, 84.21
8206, 10/14/2022, 96.24
8207, 10/29/2022, 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- transaction\_previous\_month

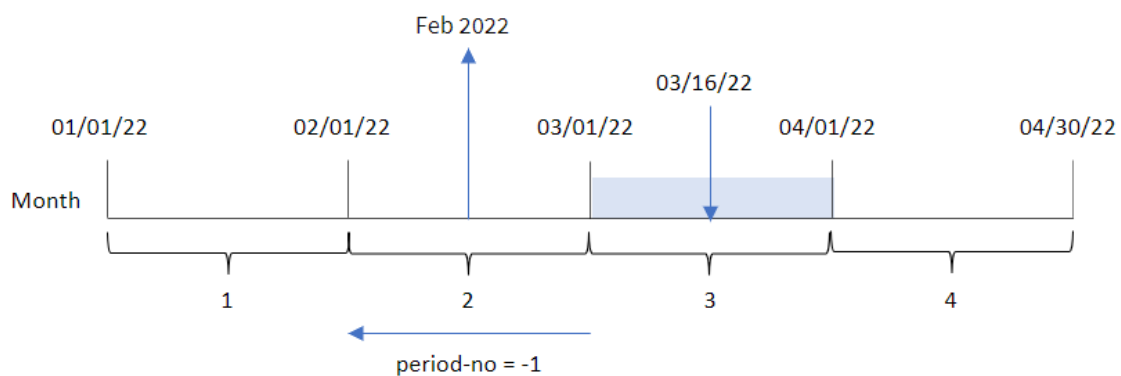
Resultatentabel

date	transaction_previous_month
1/7/2022	december 2021
1/19/2022	december 2021
2/5/2022	januari 2022
2/28/2022	januari 2022
3/16/2022	februari 2022
4/1/2022	maart 2022
5/7/2022	april 2022
5/16/2022	april 2022
6/15/2022	mei 2022
6/26/2022	mei 2022
7/9/2022	juni 2022
7/22/2022	juni 2022
7/23/2022	juni 2022
7/27/2022	juni 2022
8/2/2022	juli 2022
8/8/2022	juli 2022
8/19/2022	juli 2022

date	transaction_previous_month
9/26/2022	augustus 2022
10/14/2022	september 2022
10/29/2022	september 2022

In dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `monthname()`-functie, identificeert de functie eerst de maand waarin de transacties plaatsvinden. Het verschuift dan naar een maand eerder en retourneert de naam van de maand en het jaar.

Diagram van functie `monthname`, voorbeeld `period_no`



Transactie 8192 vond plaats op 16 maart. De functie `monthname()` identificeert dat de maand voordat de transactie plaatsvond februari was en retourneert de maand, in de systeemvariabele-indeling `MonthNames`, samen met het jaar 2022.

### Voorbeeld 3 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde inline gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het einde van de maand waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de toepassing.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

```
Transactions:
Load
```



\*

Inline

[

id,date,amount

8188,1/7/2022,17.17

8189,1/19/2022,37.23

8190,2/28/2022,88.27

8191,2/5/2022,57.42

8192,3/16/2022,53.80

8193,4/1/2022,82.06

8194,5/7/2022,40.39

8195,5/16/2022,87.21

8196,6/15/2022,95.93

8197,6/26/2022,45.89

8198,7/9/2022,36.23

8199,7/22/2022,25.66

8200,7/23/2022,82.77

8201,7/27/2022,69.98

8202,8/2/2022,76.11

8203,8/8/2022,25.12

8204,8/19/2022,46.23

8205,9/26/2022,84.21

8206,10/14/2022,96.24

8207,10/29/2022,67.67

];

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:date.

Maak de volgende meting:

=monthname(date)

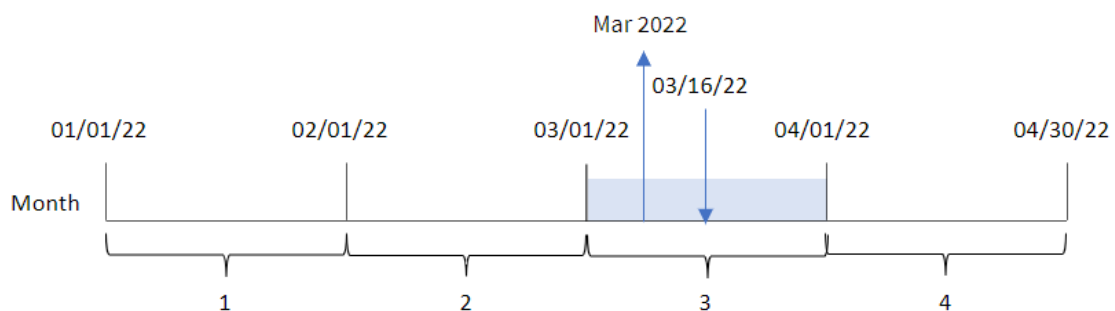
Resultatentabel

date	=monthname(date)
1/7/2022	januari 2022
1/19/2022	januari 2022
2/5/2022	februari 2022
2/28/2022	februari 2022
3/16/2022	maart 2022
4/1/2022	april 2022
5/7/2022	mei 2022
5/16/2022	mei 2022
6/15/2022	juni 2022

date	=monthname(date)
6/26/2022	juni 2022
7/9/2022	juli 2022
7/22/2022	juli 2022
7/23/2022	juli 2022
7/27/2022	juli 2022
8/2/2022	augustus 2022
8/8/2022	augustus 2022
8/19/2022	augustus 2022
9/26/2022	september 2022
10/14/2022	oktober 2022
10/29/2022	oktober 2022

De meting `month_name` wordt gemaakt in het diagramobject met behulp van de functie `monthname()` en geeft het veld `date` door als het argument van de functie.

*Diagram van functie `monthname`, diagramobjectvoorbeeld*



De functie `monthname()` identificeert dat transactie 8192 plaatsvond in maart 2022 en retourneert deze waarde met behulp van de systeemvariabele `MonthNames`.

### monthsend

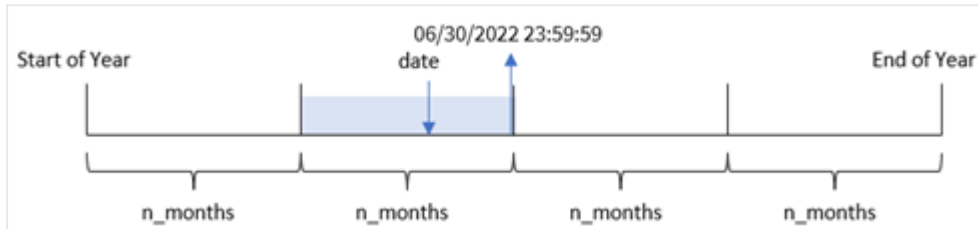
Deze functie retourneert een waarde die overeenkomt met de tijdstempel van de laatste milliseconde van de periode van een maand, twee maanden, kwartaal, vier maanden of half jaar die een basisdatum bevat. Het is tevens mogelijk om de tijdstempel voor het eind van een voorafgaande of volgende tijdsperiode te bepalen. De standaarduitvoernotatie is de `DateFormat` die is ingesteld in het script.

#### Syntaxis:

```
MonthsEnd(n_months, date[, period_no [, first_month_of_year]])
```

**Retourgegevenstypen:** dubbele waarde

Diagram van `monthsend`-functie.



### Argumenten

Argument	Beschrijving
<b>n_months</b>	Het aantal maanden dat de periode definieert. Een geheel getal of uitdrukking die wordt herleid tot een geheel getal dat een van de volgende waarden moet hebben: 1 (equivalent aan de functie <code>inmonth()</code> ), 2 (2 maanden), 3 (equivalent aan de functie <code>inquarter()</code> ), 4 (tertiaal) of 6 (half jaar).
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	De periode kan worden verschoven met <b>period_no</b> , een geheel getal of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de periode aangeeft die <b>base_date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande perioden aan en positieve waarden geven volgende perioden aan.
<b>first_month_of_year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> .

De functie `monthsend()` verdeelt het jaar in segmenten op basis van het opgegeven `n_months`-argument. De functie kijkt vervolgens in welk segment elke opgegeven datum valt en retourneert de laatste milliseconde van dat segment in datumnotitie. De functie kan de eindtijdstempel retourneren van voorgaande of volgende segmenten en kan de eerste maand van het jaar opnieuw instellen.

De volgende segmenten van het jaar zijn beschikbaar in de functie als `n_month`-argumenten:

#### n\_month-argumenten

Periode	Aantal maanden
maandelijks	1
tweemaandelijks	2
kwartaal	3
tertiaal	4
halfjaarlijks	6

### Wanneer gebruiken

De `monthsend()`-functie wordt gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van de maand wordt gebruikt dat al is geweest. De gebruiker heeft de mogelijkheid om met gebruik van een variabele, de gewenste periode te selecteren. De `monthsend()`-functie kan bijvoorbeeld een invoervariabele geven waarmee de gebruiker de totale hoeveelheid rente kan berekenen die nog niet is opgebouwd gedurende de maand, het kwartaal of het halve jaar.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>monthsend(4, '07/19/2013')</code>	Retourneert 08/31/2013.
<code>monthsend(4, '10/19/2013', -1)</code>	Retourneert 08/31/2013.
<code>monthsend(4, '10/19/2013', 0, 2)</code>	Retourneert 01/31/2014. Omdat het begin van het jaar maand 2 wordt.

### Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 wordt in de tabel `Transactions` geladen.
- Een datumveld dat wordt weergegeven in de `DateFormat`-systeemvariabele met de datumnotitie (MM/DD/YYYY).
- Een voorgaande load-instructie met:

- De `monthsend`-functie die is ingesteld als het veld `bi_monthly_end`. De transacties van deze groep in tweemaandelijke segmenten.
- De `timestamp`-functie die de begintijdstempel van het segment voor iedere transactie retourneert.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*
monthsend(2,date) as bi_monthly_end,
timestamp(monthsend(2,date)) as bi_monthly_end_timestamp
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/22/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

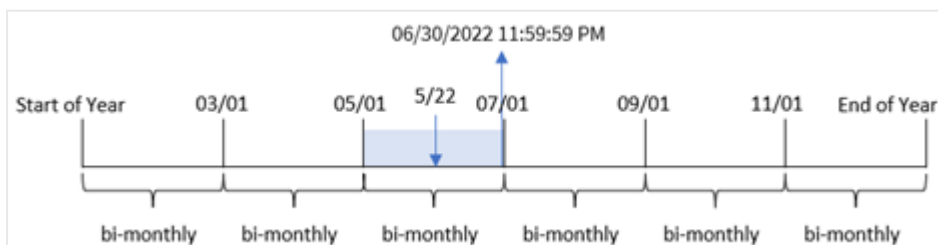
- `id`
- `date`
- `bi_monthly_end`
- `bi_monthly_end_timestamp`

Resultatentabel

id	date	bi_monthly_end	bi_monthly_end_timestamp
8188	1/7/2022	02/28/2022	2/28/2022 11:59:59 PM
8189	1/19/2022	02/28/2022	2/28/2022 11:59:59 PM
8190	2/5/2022	02/28/2022	2/28/2022 11:59:59 PM
8191	2/28/2022	02/28/2022	2/28/2022 11:59:59 PM
8192	3/16/2022	04/30/2022	4/30/2022 11:59:59 PM
8193	4/1/2022	04/30/2022	4/30/2022 11:59:59 PM
8194	5/7/2022	06/30/2022	6/30/2022 11:59:59 PM
8195	5/22/2022	06/30/2022	6/30/2022 11:59:59 PM
8196	6/15/2022	06/30/2022	6/30/2022 11:59:59 PM
8197	6/26/2022	06/30/2022	6/30/2022 11:59:59 PM
8198	7/9/2022	08/31/2022	8/31/2022 11:59:59 PM
8199	7/22/2022	08/31/2022	8/31/2022 11:59:59 PM
8200	7/23/2022	08/31/2022	8/31/2022 11:59:59 PM
8201	7/27/2022	08/31/2022	8/31/2022 11:59:59 PM
8202	8/2/2022	08/31/2022	8/31/2022 11:59:59 PM
8203	8/8/2022	08/31/2022	8/31/2022 11:59:59 PM
8204	8/19/2022	08/31/2022	8/31/2022 11:59:59 PM
8205	9/26/2022	10/31/2022	10/31/2022 11:59:59 PM
8206	10/14/2022	10/31/2022	10/31/2022 11:59:59 PM
8207	10/29/2022	10/31/2022	10/31/2022 11:59:59 PM

Het veld `bi_monthly_end` wordt gemaakt in de vorige load-instructie met behulp van de functie `monthsend` (). Het eerste opgegeven argument is 2, waarbij het jaar wordt verdeeld in tweemaandelijks segmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd.

Diagram van de `monthsend`-functie met tweemaandelijks segmenten.



Transactie 8195 vindt plaats op 22 mei. De functie `monthsend()` verdeelt het jaar in eerste instantie in tweemaandelijke segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Het resultaat is dat de functie de laatste milliseconde van dit segment retourneert: 06/30/2022 11:59:59 PM.

### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld wordt een veld gemaakt, `prev_bi_monthly_end`, dat de eerste milliseconde retourneert van het tweemaandelijke segment voordat de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
monthsend(2,date,-1) as prev_bi_monthly_end,
```

```
timestamp(monthsend(2,date,-1)) as prev_bi_monthly_end_timestamp
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/22/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- prev\_bi\_monthly\_end
- prev\_bi\_monthly\_end\_timestamp

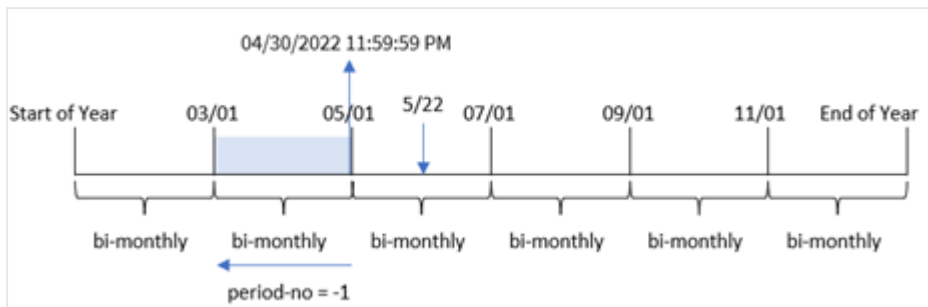
Resultatentabel

id	date	prev_bi_monthly_end	prev_bi_monthly_end_timestamp
8188	1/7/2022	12/31/2021	12/31/2021 11:59:59 PM
8189	1/19/2022	12/31/2021	12/31/2021 11:59:59 PM
8190	2/5/2022	12/31/2021	12/31/2021 11:59:59 PM
8191	2/28/2022	12/31/2021	12/31/2021 11:59:59 PM
8192	3/16/2022	02/28/2022	2/28/2022 11:59:59 PM
8193	4/1/2022	02/28/2022	2/28/2022 11:59:59 PM
8194	5/7/2022	04/30/2022	4/30/2022 11:59:59 PM
8195	5/22/2022	04/30/2022	4/30/2022 11:59:59 PM
8196	6/15/2022	04/30/2022	4/30/2022 11:59:59 PM
8197	6/26/2022	04/30/2022	4/30/2022 11:59:59 PM
8198	7/9/2022	06/30/2022	6/30/2022 11:59:59 PM
8199	7/22/2022	06/30/2022	6/30/2022 11:59:59 PM
8200	7/23/2022	06/30/2022	6/30/2022 11:59:59 PM
8201	7/27/2022	06/30/2022	6/30/2022 11:59:59 PM
8202	8/2/2022	06/30/2022	6/30/2022 11:59:59 PM
8203	8/8/2022	06/30/2022	6/30/2022 11:59:59 PM
8204	8/19/2022	06/30/2022	6/30/2022 11:59:59 PM
8205	9/26/2022	08/31/2022	8/31/2022 11:59:59 PM
8206	10/14/2022	08/31/2022	8/31/2022 11:59:59 PM
8207	10/29/2022	08/31/2022	8/31/2022 11:59:59 PM

Door -1 als het `period_no`-argument in de `monthsend()`-functie te gebruiken, nadat een jaar eerst in tweemaandelijks segmenten is verdeeld, retourneert de functie de laatste milliseconde van het vorige tweemaandelijks segment ten opzichte van wanneer een transactie plaatsvindt.



Diagram van de `monthsend`-functie die het vorige tweemaandelijke segment retourneert.



Transactie 8195 vindt plaats in het segment tussen mei en juni. Het resultaat is dat het vorige tweemaandelijke segment tussen 1 maart en 30 april ligt en de functie daarom de laatste milliseconde van dit segment retourneert: 04/30/2022 11:59:59 PM.

### Voorbeeld 3 - first\_month\_of\_year

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is april volgens het organisatiebeleid de eerste maand van het financiële boekjaar.

Maak een veld, `bi_monthly_end`, dat transacties groepeerd in tweemaandelijke segmenten en de laatste milliseconde-tijdstempel van het segment voor elke transactie retourneert.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*,
monthsend(2,date,0,4) as bi_monthly_end,
timestamp(monthsend(2,date,0,4)) as bi_monthly_end_timestamp
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/22/2022,87.21
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- bi\_monthly\_end
- bi\_monthly\_end\_timestamp

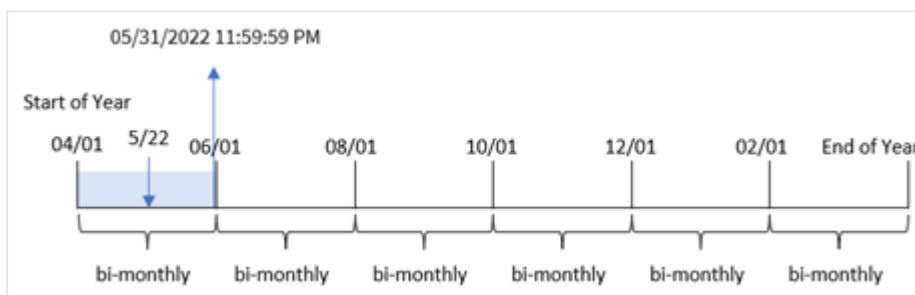
Resultatentabel

id	date	bi_monthly_end	bi_monthly_end_timestamp
8188	1/7/2022	01/31/2022	1/31/2022 11:59:59 PM
8189	1/19/2022	01/31/2022	1/31/2022 11:59:59 PM
8190	2/5/2022	03/31/2022	3/31/2022 11:59:59 PM
8191	2/28/2022	03/31/2022	3/31/2022 11:59:59 PM
8192	3/16/2022	03/31/2022	3/31/2022 11:59:59 PM
8193	4/1/2022	05/31/2022	5/31/2022 11:59:59 PM
8194	5/7/2022	05/31/2022	5/31/2022 11:59:59 PM
8195	5/22/2022	05/31/2022	5/31/2022 11:59:59 PM
8196	6/15/2022	07/31/2022	7/31/2022 11:59:59 PM
8197	6/26/2022	07/31/2022	7/31/2022 11:59:59 PM
8198	7/9/2022	07/31/2022	7/31/2022 11:59:59 PM
8199	7/22/2022	07/31/2022	7/31/2022 11:59:59 PM
8200	7/23/2022	07/31/2022	7/31/2022 11:59:59 PM
8201	7/27/2022	07/31/2022	7/31/2022 11:59:59 PM
8202	8/2/2022	09/30/2022	9/30/2022 11:59:59 PM
8203	8/8/2022	09/30/2022	9/30/2022 11:59:59 PM

id	date	bi_monthly_end	bi_monthly_end_timestamp
8204	8/19/2022	09/30/2022	9/30/2022 11:59:59 PM
8205	9/26/2022	09/30/2022	9/30/2022 11:59:59 PM
8206	10/14/2022	11/30/2022	11/30/2022 11:59:59 PM
8207	10/29/2022	11/30/2022	11/30/2022 11:59:59 PM

Door 4 te gebruiken als het `first_month_of_year`-argument in de `monthsend()`-functie, begint de functie het jaar op 1 april. Het verdeelt vervolgens het jaar in tweemaandelijks segmenten: Apr-mei, jun-jul, aug-sep, okt-nov, dec-jan, feb-maa.

Diagram van functie `monthsend` met april ingesteld als de eerste maand van het jaar.



Transactie 8195 vond plaats op 22 mei en valt in het segment tussen 1 april en 31 mei. Het resultaat is dat de functie de laatste milliseconde van dit segment retourneert: 05/31/2022 11:59:59 PM.

### Voorbeeld 4 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt. In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de app geladen.

In dit voorbeeld is de taak om een berekening te maken die transacties groepeerd in tweemaandelijks segmenten en de laatste milliseconde-tijdstempel van het segment voor iedere transactie retourneert als een meting in een diagramobject van een app.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,2/19/2022,37.23
```

```
8189,3/7/2022,17.17
8190,3/30/2022,88.27
8191,4/5/2022,57.42
8192,4/16/2022,53.80
8193,5/1/2022,82.06
8194,5/7/2022,40.39
8195,5/22/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

date

Maak de volgende metingen om de laatste milliseconde-tijdstempel van het tweemaandelijks segment waarin de transactie plaatsvond op te halen:

- =monthsEnd(2,date)
- =timestamp(monthsend(2,date))

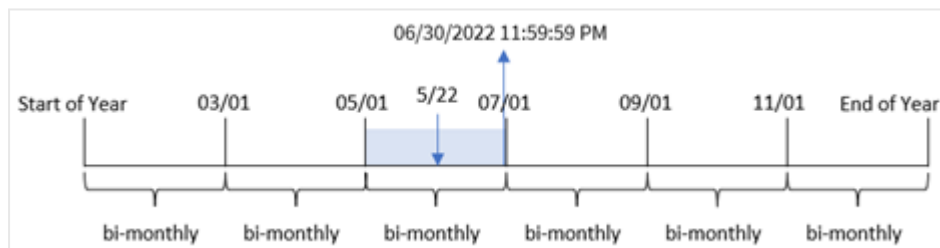
Resultatentabel

id	date	=monthsend(2,date)	=timestamp(monthsend(2,date))
8188	1/7/2022	02/28/2022	2/28/2022 11:59:59 PM
8189	1/19/2022	02/28/2022	2/28/2022 11:59:59 PM
8190	2/5/2022	02/28/2022	2/28/2022 11:59:59 PM
8191	2/28/2022	02/28/2022	2/28/2022 11:59:59 PM
8192	3/16/2022	04/30/2022	4/30/2022 11:59:59 PM
8193	4/1/2022	04/30/2022	4/30/2022 11:59:59 PM
8194	5/7/2022	06/30/2022	6/30/2022 11:59:59 PM
8195	5/22/2022	06/30/2022	6/30/2022 11:59:59 PM
8196	6/15/2022	06/30/2022	6/30/2022 11:59:59 PM
8197	6/26/2022	06/30/2022	6/30/2022 11:59:59 PM

id	date	=monthsend(2,date)	=timestamp(monthsend(2,date))
8198	7/9/2022	08/31/2022	8/31/2022 11:59:59 PM
8199	7/22/2022	08/31/2022	8/31/2022 11:59:59 PM
8200	7/23/2022	08/31/2022	8/31/2022 11:59:59 PM
8201	7/27/2022	08/31/2022	8/31/2022 11:59:59 PM
8202	8/2/2022	08/31/2022	8/31/2022 11:59:59 PM
8203	8/8/2022	08/31/2022	8/31/2022 11:59:59 PM
8204	8/19/2022	08/31/2022	8/31/2022 11:59:59 PM
8205	9/26/2022	10/31/2022	10/31/2022 11:59:59 PM
8206	10/14/2022	10/31/2022	10/31/2022 11:59:59 PM
8207	10/29/2022	10/31/2022	10/31/2022 11:59:59 PM

Het veld `bi_monthly_end` wordt in een diagramobject gemaakt als een meting met behulp van de functie `monthsend()`. Het eerste opgegeven argument is 2, dat het jaar verdeelt in tweemaandelijks segmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd.

*Diagram van de `monthsend`-functie met tweemaandelijks segmenten.*



Transactie 8195 vindt plaats op 22 mei. De functie `monthsend()` verdeelt het jaar in eerste instantie in tweemaandelijks segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Het resultaat is dat de functie de eerste milliseconde van dit segment retourneert: 06/30/2022 11:59:59 PM.

### Voorbeeld 5 - Scenario

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

In dit voorbeeld wordt een gegevensset geladen in de tabel 'Employee\_Expenses'. De tabel bevat de volgende velden:

- Werknemers-id's
- Werknemersnamen

- De gemiddelde dagelijkse onkostendeclaraties van elke werknemer.

De eindgebruiker wil graag een diagram dat per werknemer-id en werknemersnaam de geschatte onkostendeclaraties voor de rest van een periode naar keuze weergeeft. Het boekjaar begint in januari.

### Load-script

```
SET vPeriod = 1;

Employee_Expenses:
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,Sydney,$27
186,Agatha,$18
];
```

### Resultaten

Laad de gegevens en open een nieuw werkblad.

Aan het begin van het load-script is een variabele, `vPeriod`, gemaakt die wordt gekoppeld aan het besturingselement voor variabeleninput.

Doe het volgende:

1. Klik in het bedrijfsmiddelenvenster op **Aangepaste objecten**.
2. Selecteer **Qlik Dashboard-bundel** en maak een **Variabeleninput**-object.
3. Voer een titel in voor het diagramobject.
4. Selecteer onder **Variabele** de optie **vPeriod** als de naam en stel het object in om als een **Vervolgkeuzelijst** te worden weergegeven.
5. Klik onder **Waarden** op **Dynamische waarden**. Voer het volgende in:  
`= '1~month|2~bi-month|3~quarter|4~tertiaal|6~half-year'`.

Maak een nieuwe tabel en deze velden als dimensies:

- `employee_id`
- `employee_name`

Maak deze meting om de opgebouwde rente te berekenen:

```
=floor(monthsend($(vPeriod),today(1))-today(1))*avg_daily_claim
```



*Deze meting is dynamisch en produceert verschillende tabelresultaten, afhankelijk van de datum waarop u de gegevens laadt.*

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

employee_id	employee_name	=floor(monthsend(\$(vPeriod),today(1))-today(1))*avg_daily_claim
182	Mark	\$1410.00
183	Deryck	\$1175.00
184	Dexter	\$1175.00
185	Sydney	\$2538.00
186	Agatha	\$1692.00

De `monthsend()`-functie gebruikt de invoer van de gebruiker als het eerste argument en de datum van vandaag als het tweede argument. Dit retourneert de einddatum voor de door de gebruiker geselecteerde tijdperiode. De uitdrukking retourneert vervolgens het aantal dagen dat nog resteert in de geselecteerde periode door de datum van vandaag af te trekken van deze einddatum.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in de resterende dagen van deze periode zal indienen.

### monthsname

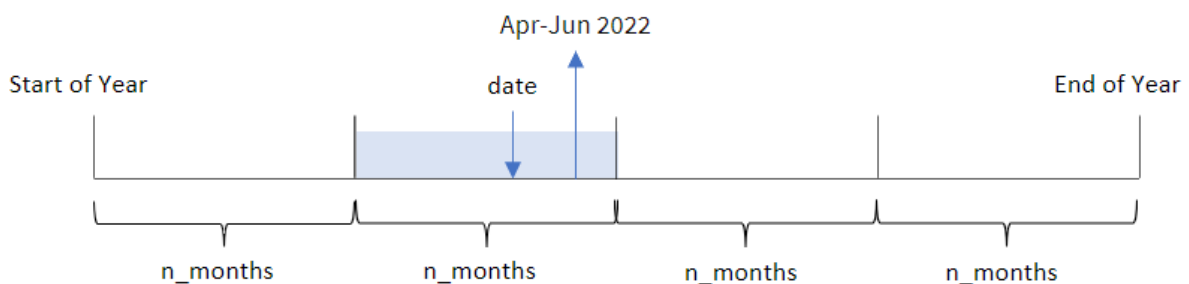
Deze functie retourneert een weergavewaarde die het bereik vertegenwoordigt van de maanden van de periode (geformatteerd volgens de scriptvariabele **MonthNames**) alsmede het jaar. De onderliggende numerieke waarde komt overeen met een tijdstempel van de eerste milliseconde van de periode van een maand, twee maanden, kwartaal, tertiaal of half jaar die een basisdatum bevat.

#### Syntaxis:

```
MonthsName (n_months, date[, period_no[, first_month_of_year]])
```

**Retourgegevenstypen:** dual

*Diagram van functie monthsname*



## 5 Script- en diagramfuncties

De functie `monthsname()` verdeelt het jaar in segmenten op basis van het opgegeven `n_months`-argument. Het evalueert vervolgens het segment waartoe elke opgegeven `date` behoort, en retourneert de namen van de begin- en eindmaanden van dat segment, evenals het jaar. De functie biedt ook de mogelijkheid om deze grenzen van voorgaande of volgende segmenten te retourneren, en om opnieuw te definiëren wat de eerste maand van het jaar is.

De volgende segmenten van het jaar zijn beschikbaar in de functie als `n_month`-argumenten:

Mogelijke `n_month`-argumenten

Perioden	Aantal maanden
maandelijks	1
tweemaandelijks	2
kwartaal	3
tertiaal	4
halfjaarlijks	6

Argumenten

Argument	Beschrijving
<code>n_months</code>	Het aantal maanden dat de periode definieert. Een geheel getal of uitdrukking die wordt herleid tot een geheel getal dat een van de volgende waarden moet hebben: 1 (equivalent aan de functie <code>inmonth()</code> ), 2 (2 maanden), 3 (equivalent aan de functie <code>inquarter()</code> ), 4 (tertiaal) of 6 (half jaar).
<code>date</code>	De te evalueren datum of tijdstempel.
<code>period_no</code>	De periode kan worden verschoven met <code>period_no</code> , een geheel getal of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de periode aangeeft die <code>base_date</code> bevat. Negatieve waarden in <code>period_no</code> geven voorafgaande perioden aan en positieve waarden geven volgende perioden aan.
<code>first_month_of_year</code>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <code>first_month_of_year</code> .

### Wanneer gebruiken

De functie `monthsname()` is handig wanneer u de gebruiker de functionaliteit wilt bieden om aggregaties te vergelijken op een periode naar keuze. U kunt bijvoorbeeld een invoervariabele opgeven om de gebruiker de totale verkoop van producten per maand, kwartaal of halfjaar te laten zien.

Deze dimensies kunnen worden gemaakt in het load-script door de functie toe te voegen als een veld in een masterkalendertabel, of door de dimensie rechtstreeks in een diagram te maken als een berekende dimensie.



### Voorbeelden van functies

Voorbeeld	Resultaat
<code>monthsname(4, '10/19/2013')</code>	Retourneert 'Sep-Dec 2013.' In dit en andere voorbeelden is de opdracht <b>SET Monthnames</b> ingesteld op Jan;Feb;Mar, enzovoort.
<code>monthsname(4, '10/19/2013', -1)</code>	Retourneert 'May-Aug 2013'.
<code>monthsname(4, '10/19/2013', 0, 2)</code>	Retourneert 'Oct-Jan 2014', aangezien het jaar is opgegeven om te beginnen in maand 2. Daarom eindigt de periode van vier maanden in de eerste maand van het volgende jaar.

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

## Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `bi_monthly_range`, dat transacties groepeerd in tweemaandelijke segmenten en de grensnamen van dat segment voor elke transactie retourneert.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
    *,
    monthsname(2,date) as bi_monthly_range
;

Load
*
Inline
[
id,date,amount
8188,2/19/2022,37.23
8189,3/7/2022,17.17
8190,3/30/2022,88.27
8191,4/5/2022,57.42
8192,4/16/2022,53.80
8193,5/1/2022,82.06
8194,5/7/2022,40.39
8195,5/22/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- bi\_monthly\_range

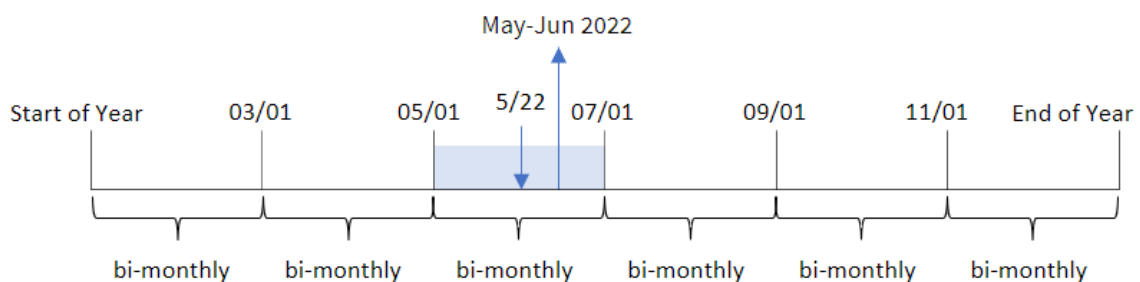
Resultatentabel

date	bi_monthly_range
2/19/2022	jan-feb 2022
3/7/2022	mrt-apr 2022
3/30/2022	mrt-apr 2022
4/5/2022	mrt-apr 2022
4/16/2022	mrt-apr 2022
5/1/2022	mei-juni 2022

date	bi_monthly_range
5/7/2022	mei-juni 2022
5/22/2022	mei-juni 2022
6/15/2022	mei-juni 2022
6/26/2022	mei-juni 2022
7/9/2022	juli-aug 2022
7/22/2022	juli-aug 2022
7/23/2022	juli-aug 2022
7/27/2022	juli-aug 2022
8/2/2022	juli-aug 2022
8/8/2022	juli-aug 2022
8/19/2022	juli-aug 2022
9/26/2022	sep-okt 2022
10/14/2022	sep-okt 2022
10/29/2022	sep-okt 2022

Het veld `bi_monthly_range` wordt gemaakt in de vorige load-instructie met behulp van de functie `monthsname()`. Het eerste opgegeven argument is 2, waarbij het jaar wordt verdeeld in tweemaandelijke segmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd.

*Diagram van functie `monthsname`, basisvoorbeeld*



Transactie 8195 vindt plaats op 22 mei. De functie `monthsname()` verdeelt het jaar in eerste instantie in tweemaandelijke segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Daarom retourneert de functie deze maanden in de systeemvariabele-indeling `MonthNames`, evenals het jaar, mei-juni 2022.

### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde inline gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `prev_bi_monthly_range`, dat transacties groepeerd in tweemaandelijks segmenten en de grensnamen van het eerdere segment voor elke transactie retourneert.

Voeg hier uw andere tekst toe, indien nodig, met lijsten enz.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        MonthsName(2,date,-1) as prev_bi_monthly_range
    ;

Load
*
Inline
[
id,date,amount
8188,2/19/2022,37.23
8189,3/7/2022,17.17
8190,3/30/2022,88.27
8191,4/5/2022,57.42
8192,4/16/2022,53.80
8193,5/1/2022,82.06
8194,5/7/2022,40.39
8195,5/22/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

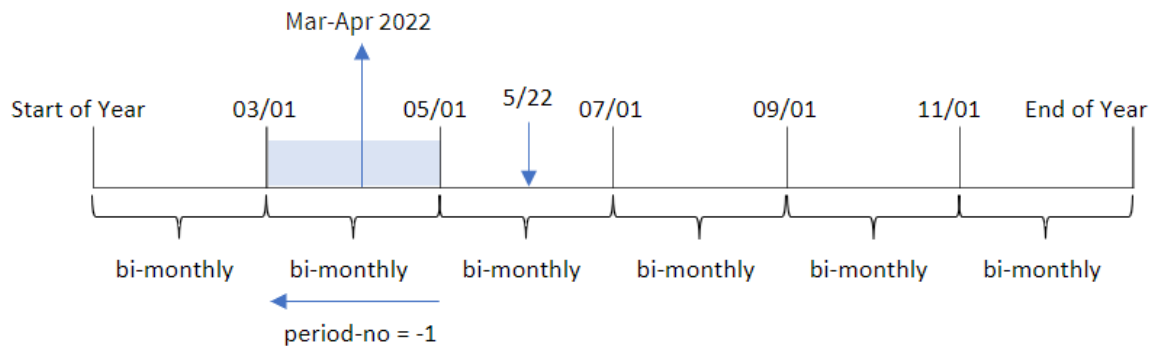
- date
- prev\_bi\_monthly\_range

Resultatentabel

date	prev_bi_monthly_range
2/19/2022	nov-dec 2021
3/7/2022	jan-feb 2022
3/30/2022	jan-feb 2022
4/5/2022	jan-feb 2022
4/16/2022	jan-feb 2022
5/1/2022	mrt-apr 2022
5/7/2022	mrt-apr 2022
5/22/2022	mrt-apr 2022
6/15/2022	mrt-apr 2022
6/26/2022	mrt-apr 2022
7/9/2022	mei-juni 2022
7/22/2022	mei-juni 2022
7/23/2022	mei-juni 2022
7/27/2022	mei-juni 2022
8/2/2022	mei-juni 2022
8/8/2022	mei-juni 2022
8/19/2022	mei-juni 2022
9/26/2022	juli-aug 2022
10/14/2022	juli-aug 2022
10/29/2022	juli-aug 2022

In dit voorbeeld wordt -1 gebruikt als het `period_no`-argument in de `monthsname()`-functie. Na aanvankelijk een jaar in tweemaandelijks segmenten te hebben verdeeld, retourneert de functie vervolgens de vorige segmentgrenzen voor wanneer een transactie plaatsvindt.

Diagram van functie `monthsname`, voorbeeld `period_no`



Transactie 8195 vindt plaats in het segment tussen mei en juni. Daarom was het vorige tweemaandelijks segment tussen 1 maart en 30 april, en daarom retourneert de functie `mrt-apr 2022`.

### Voorbeeld 3 - `first_month_of_year`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde inline gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een ander veld, `bi_monthly_range`, dat transacties groepeerd in tweemaandelijks segmenten en de segmentgrenzen voor elke transactie retourneert.

In dit voorbeeld moeten we echter ook april instellen als de eerste maand van het boekjaar.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    MonthsName(2,date,0,4) as bi_monthly_range
  ;
Load
*
Inline
[
id,date,amount
8188,2/19/2022,37.23
8189,3/7/2022,17.17
8190,3/30/2022,88.27
8191,4/5/2022,57.42
```

```
8192,4/16/2022,53.80
8193,5/1/2022,82.06
8194,5/7/2022,40.39
8195,5/22/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- bi\_monthly\_range

Resultatentabel

date	bi_monthly_range
2/19/2022	feb-mrt 2021
3/7/2022	feb-mrt 2021
3/30/2022	feb-mrt 2021
4/5/2022	apr-mei 2022
4/16/2022	apr-mei 2022
5/1/2022	apr-mei 2022
5/7/2022	apr-mei 2022
5/22/2022	apr-mei 2022
6/15/2022	jun-jul 2022
6/26/2022	jun-jul 2022
7/9/2022	jun-jul 2022
7/22/2022	jun-jul 2022
7/23/2022	jun-jul 2022
7/27/2022	jun-jul 2022

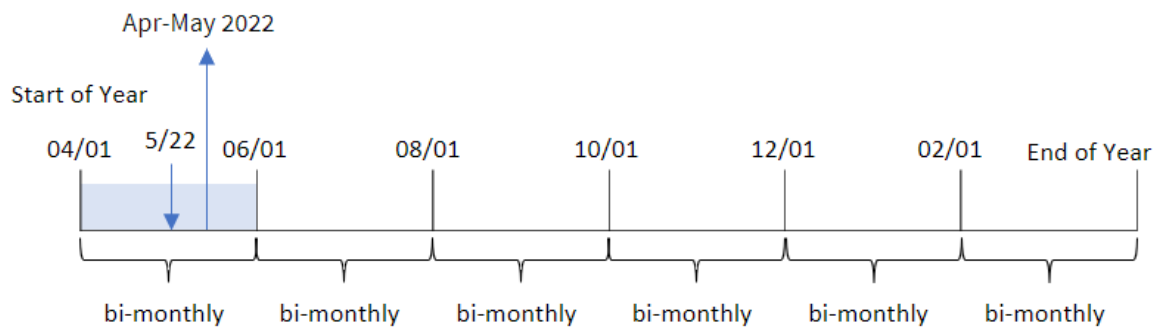
date	bi_monthly_range
8/2/2022	aug-sept 2022
8/8/2022	aug-sept 2022
8/19/2022	aug-sept 2022
9/26/2022	aug-sept 2022
10/14/2022	okt-nov 2022
10/29/2022	okt-nov 2022

Door 4 te gebruiken als het `first_month_of_year`-argument in de `monthsname()`-functie, begint de functie het jaar op 1 april. Het verdeelt vervolgens het jaar in tweemaandelijks segmenten: apr-mei, jun-jul, aug-sep, okt-nov, dec-jan, feb-mrt.

Alineatekst voor Resultaten.

Transactie 8195 vond plaats op 22 mei en valt in het segment tussen 1 april en 31 mei. Daarom retourneert de functie apr-mei 2022.

*Diagram van functie monthsname, voorbeeld first\_month\_of\_year*



### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde inline gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die transacties groepeerd in tweemaandelijks segmenten en de segmentgrenzen retourneert voor elke transactie wordt gemaakt als een meting in een diagramobject van de toepassing.



### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,2/19/2022,37.23
```

```
8189,3/7/2022,17.17
```

```
8190,3/30/2022,88.27
```

```
8191,4/5/2022,57.42
```

```
8192,4/16/2022,53.80
```

```
8193,5/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/22/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:date.

Maak de volgende meting:

```
=monthsname(2,date)
```

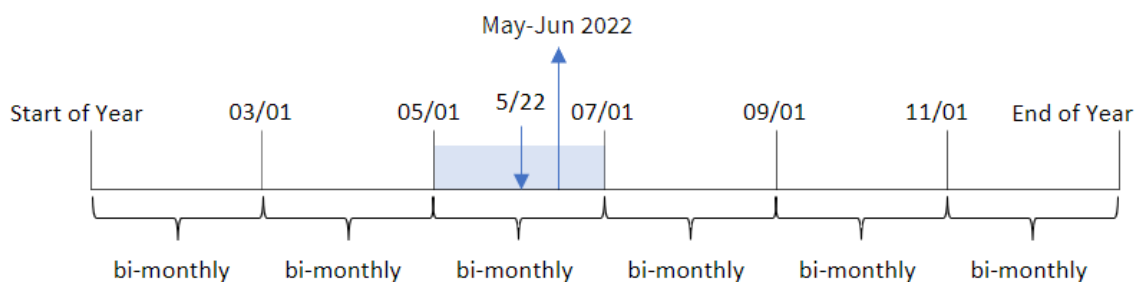
Resultatentabel

date	=monthsname(2,date)
2/19/2022	jan-feb 2022
3/7/2022	mrt-apr 2022
3/30/2022	mrt-apr 2022
4/5/2022	mrt-apr 2022
4/16/2022	mrt-apr 2022
5/1/2022	mei-juni 2022

date	=monthsname(2,date)
5/7/2022	mei-juni 2022
5/22/2022	mei-juni 2022
6/15/2022	mei-juni 2022
6/26/2022	mei-juni 2022
7/9/2022	juli-aug 2022
7/22/2022	juli-aug 2022
7/23/2022	juli-aug 2022
7/27/2022	juli-aug 2022
8/2/2022	juli-aug 2022
8/8/2022	juli-aug 2022
8/19/2022	juli-aug 2022
9/26/2022	sep-okt 2022
10/14/2022	sep-okt 2022
10/29/2022	sep-okt 2022

Het veld `bi_monthly_range` wordt in een diagramobject gemaakt als een meting met behulp van de functie `monthsname()`. Het eerste opgegeven argument is 2, waarbij het jaar wordt verdeeld in tweemaandelijke segmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd.

*Diagram van functie monthsname, diagramobjectvoorbeeld*



Transactie 8195 vindt plaats op 22 mei. De functie `monthsname()` verdeelt het jaar in eerste instantie in tweemaandelijke segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Daarom retourneert de functie deze maanden in de systeemvariabele-indeling `MonthNames`, evenals het jaar, mei-juni 2022.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.

De eindgebruiker wil graag een diagramobject dat de totale omzet per periode naar eigen keuze weergeeft. Dit kan zelfs worden bereikt als deze dimensie niet beschikbaar is in het gegevensmodel, door de functie monthsname() te gebruiken als een berekende dimensie die dynamisch wordt gewijzigd door een besturingselement voor variabeleninput.

#### Load-script

```
SET vPeriod = 1;
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/7/2022',17.17
```

```
8189,'1/19/2022',37.23
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad.

Aan het begin van het load-script is een variabele (`vPeriod`) gemaakt die wordt gekoppeld aan het besturingselement voor variabeleninput. Configureer vervolgens de variabele als een aangepast object in het werkblad.

#### Doe het volgende:

1. Klik in het bedrijfsmiddelenvenster op **Aangepaste objecten**.
2. Selecteer **Klik Dashboard-bundel** en maak een object **Variabeleninput**.
3. Voer een titel in voor het diagramobject.
4. Selecteer onder **Variabele** de optie **vPeriod** als de naam en stel het object in om als een **Vervolgkeuzelijst** te worden weergegeven.
5. Configureer onder **Waarden** het object om dynamische waarden te gebruiken. Voer het volgende in:  
`= '1~month|2~bi-month|3~quarter|4~tertiaal|6~half-year'`

Maak vervolgens de resultatentabel.

#### Doe het volgende:

1. Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:  
`=monthsname($(vPeriod),date)`
2. Voeg deze meting toe om de totale verkoop te berekenen:  
`=sum(amount)`
3. Stel de **Getalnotatie** van de meting in op **Geld**. Klik op  **Klaar met bewerken**. U kunt nu de gegevens in de tabel wijzigen door het tijdsegment in het variabele-object aan te passen.

Dit is hoe de resultatentabel eruit zal zien als de optie `tertiaal` is geselecteerd:

Resultatentabel

<code>monthsname(\$(vPeriod),date)</code>	<code>=sum(amount)</code>
jan-apr 2022	253.89
Mei-aug 2022	713.58
sep-dec 2022	248.12

### monthsstart

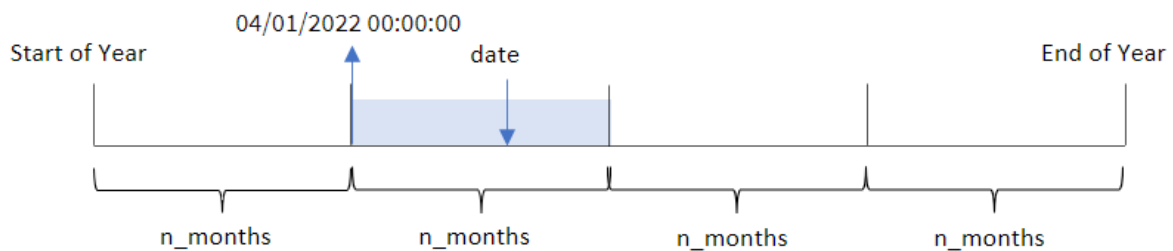
Deze functie retourneert een waarde die overeenkomt met de tijdstempel van de eerste milliseconde van de periode van een maand, twee maanden, kwartaal, tertiaal of half jaar die een basisdatum bevat. Het is tevens mogelijk om de tijdstempel voor een voorafgaande of volgende tijdsperiode te bepalen. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

### Syntaxis:

```
MonthsStart(n_months, date[, period_no [, first_month_of_year]])
```

### Retourgegevenstypen: dual

Diagram van de `monthsstart()`-functie



De functie `monthsstart()` verdeelt het jaar in segmenten op basis van het opgegeven `n_months`-argument. De functie bekijkt vervolgens in welk segment elke opgegeven datum valt en retourneert de eerste milliseconde van dat segment in datumnotitie. De functie biedt ook de mogelijkheid om de begintijdstempel van voorgaande of volgende segmenten te retourneren, en om opnieuw te definiëren wat de eerste maand van het jaar is.

De volgende segmenten van het jaar zijn beschikbaar in de functie als `n_month`-argumenten:

#### Mogelijke `n_month`-argumenten

Perioden	Aantal maanden
maandelijks	1
tweemaandelijks	2
kwartaal	3
tertiaal	4
halfjaarlijks	6

#### Argumenten

Argument	Beschrijving
<code>n_months</code>	Het aantal maanden dat de periode definieert. Een geheel getal of uitdrukking die wordt herleid tot een geheel getal dat een van de volgende waarden moet hebben: 1 (equivalent aan de functie <code>inmonth()</code> ), 2 (2 maanden), 3 (equivalent aan de functie <code>inquarter()</code> ), 4 (tertiaal) of 6 (half jaar).
<code>date</code>	De te evalueren datum of tijdstempel.

Argument	Beschrijving
<b>period_no</b>	De periode kan worden verschoven met <b>period_no</b> , een geheel getal of een uitdrukking die wordt herleid tot een geheel getal, waarbij de waarde 0 de periode aangeeft die <b>base_date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande perioden aan en positieve waarden geven volgende perioden aan.
<b>first_month_of_year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> .

### Wanneer gebruiken

De `monthsstart()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening een deel van de periode wordt gebruikt dat nog niet heeft plaatsgevonden. Dit kan bijvoorbeeld worden gebruikt om een invoervariabele te verstrekken waarmee de gebruiker de totale hoeveelheid rente kan berekenen die tot nu toe in de maand het kwartaal of het halve jaar is opgebouwd.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>monthsstart(4, '10/19/2013')</code>	Retourneert 09/01/2013.
<code>monthsstart(4, '10/19/2013', -1)</code>	Retourneert 05/01/2013.
<code>monthsstart(4, '10/19/2013', 0, 2 )</code>	Retourneert 10/01/2013, omdat het begin van het jaar maand 2 wordt.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, bi\_monthly\_start, dat transacties groepeerd in tweemaandelijks segmenten en de begintijdstempel van het segment voor elke transactie retourneert.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,
```

```
monthsstart(2,date) as bi_monthly_start,
```

```
timestamp(monthsstart(2,date)) as bi_monthly_start_timestamp
```

```
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,2/19/2022,37.23
```

```
8189,3/7/2022,17.17
```

```
8190,3/30/2022,88.27
```

```
8191,4/5/2022,57.42
```

```
8192,4/16/2022,53.80
```

```
8193,5/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/22/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- bi\_monthly\_start
- bi\_monthly\_start\_timestamp

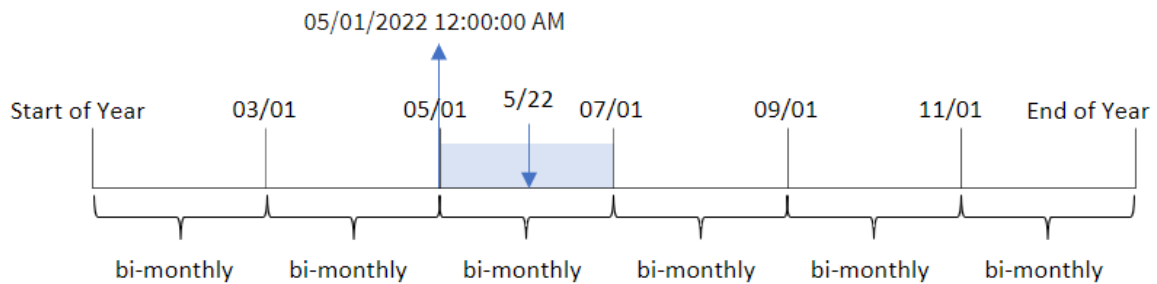
Resultatentabel

date	bi_monthly_start	bi_monthly_start_timestamp
2/19/2022	01/01/2022	1/1/2022 12:00:00 AM
3/7/2022	03/01/2022	3/1/2022 12:00:00 AM
3/30/2022	03/01/2022	3/1/2022 12:00:00 AM
4/5/2022	03/01/2022	3/1/2022 12:00:00 AM
4/16/2022	03/01/2022	3/1/2022 12:00:00 AM
5/1/2022	05/01/2022	5/1/2022 12:00:00 AM
5/7/2022	05/01/2022	5/1/2022 12:00:00 AM
5/22/2022	05/01/2022	5/1/2022 12:00:00 AM
6/15/2022	05/01/2022	5/1/2022 12:00:00 AM
6/26/2022	05/01/2022	5/1/2022 12:00:00 AM
7/9/2022	07/01/2022	7/1/2022 12:00:00 AM
7/22/2022	07/01/2022	7/1/2022 12:00:00 AM
7/23/2022	07/01/2022	7/1/2022 12:00:00 AM
7/27/2022	07/01/2022	7/1/2022 12:00:00 AM
8/2/2022	07/01/2022	7/1/2022 12:00:00 AM
8/8/2022	07/01/2022	7/1/2022 12:00:00 AM
8/19/2022	07/01/2022	7/1/2022 12:00:00 AM
9/26/2022	09/01/2022	9/1/2022 12:00:00 AM
10/14/2022	09/01/2022	9/1/2022 12:00:00 AM
10/29/2022	09/01/2022	9/1/2022 12:00:00 AM

Het veld `bi_monthly_start` wordt gemaakt in de vorige load-instructie met behulp van de functie `monthsstart()`. Het eerste opgegeven argument is 2, waarbij het jaar wordt verdeeld in tweemaandelijke segmenten. Het tweede argument geeft aan welk veld wordt geëvalueerd.



Diagram van `monthsstart()`-functie, voorbeeld zonder aanvullende argumenten



Transactie 8195 vindt plaats op 22 mei. De functie `monthsstart()` verdeelt het jaar in eerste instantie in tweemaandelijks segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Daarom retourneert de functie de eerste milliseconde van dit segment: 1 mei 2022 om 12:00:00 AM.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Er wordt een veld gemaakt, `prev_bi_monthly_start`, dat de eerste milliseconde retourneert van het tweemaandelijks segment voordat de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*,
monthsstart(2,date,-1) as prev_bi_monthly_start,
timestamp(monthsstart(2,date,-1)) as prev_bi_monthly_start_timestamp
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,2/19/2022,37.23
8189,3/7/2022,17.17
8190,3/30/2022,88.27
8191,4/5/2022,57.42
8192,4/16/2022,53.80
8193,5/1/2022,82.06
```

```
8194, 5/7/2022, 40.39
8195, 5/22/2022, 87.21
8196, 6/15/2022, 95.93
8197, 6/26/2022, 45.89
8198, 7/9/2022, 36.23
8199, 7/22/2022, 25.66
8200, 7/23/2022, 82.77
8201, 7/27/2022, 69.98
8202, 8/2/2022, 76.11
8203, 8/8/2022, 25.12
8204, 8/19/2022, 46.23
8205, 9/26/2022, 84.21
8206, 10/14/2022, 96.24
8207, 10/29/2022, 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- prev\_bi\_monthly\_start
- prev\_bi\_monthly\_start\_timestamp

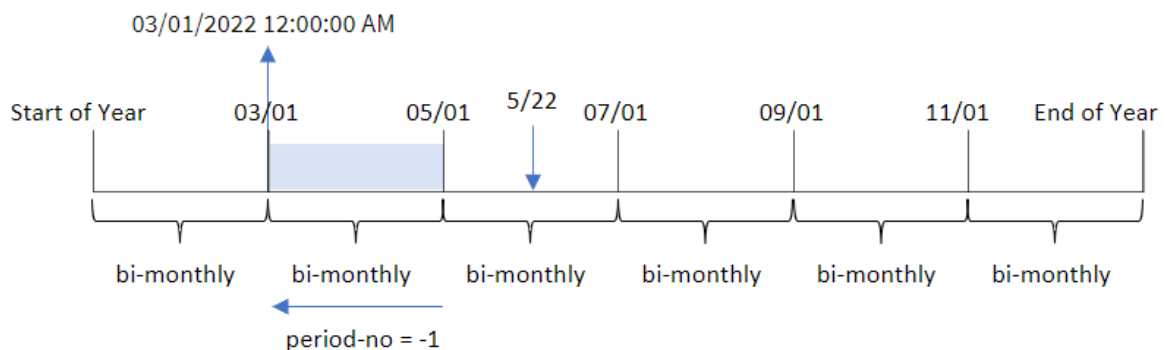
Resultatentabel

date	prev_bi_monthly_start	prev_bi_monthly_start_timestamp
2/19/2022	11/01/2021	11/1/2021 12:00:00 AM
3/7/2022	01/01/2022	1/1/2022 12:00:00 AM
3/30/2022	01/01/2022	1/1/2022 12:00:00 AM
4/5/2022	01/01/2022	1/1/2022 12:00:00 AM
4/16/2022	01/01/2022	1/1/2022 12:00:00 AM
5/1/2022	03/01/2022	3/1/2022 12:00:00 AM
5/7/2022	03/01/2022	3/1/2022 12:00:00 AM
5/22/2022	03/01/2022	3/1/2022 12:00:00 AM
6/15/2022	03/01/2022	3/1/2022 12:00:00 AM
6/26/2022	03/01/2022	3/1/2022 12:00:00 AM
7/9/2022	05/01/2022	5/1/2022 12:00:00 AM
7/22/2022	05/01/2022	5/1/2022 12:00:00 AM
7/23/2022	05/01/2022	5/1/2022 12:00:00 AM
7/27/2022	05/01/2022	5/1/2022 12:00:00 AM
8/2/2022	05/01/2022	5/1/2022 12:00:00 AM

date	prev_bi_monthly_start	prev_bi_monthly_start_timestamp
8/8/2022	05/01/2022	5/1/2022 12:00:00 AM
8/19/2022	05/01/2022	5/1/2022 12:00:00 AM
9/26/2022	07/01/2022	7/1/2022 12:00:00 AM
10/14/2022	07/01/2022	7/1/2022 12:00:00 AM
10/29/2022	07/01/2022	7/1/2022 12:00:00 AM

Door -1 als het `period_no`-argument in de `monthsstart()`-functie te gebruiken, nadat een jaar eerst in tweemaandelijke segmenten is verdeeld, retourneert de functie de eerste milliseconde van het vorige tweemaandelijke segment ten opzichte van wanneer een transactie plaatsvindt.

Diagram van `monthsstart()`-functie, voorbeeld `period_no`



Transactie 8195 vindt plaats in het segment tussen mei en juni. Daarom was het vorige tweemaandelijke segment tussen 1 maart en 30 april waardoor de functie de eerste milliseconde van dit segment retourneert: 1 maart 2022 om 12:00:00 AM.

### Voorbeeld 3 - first\_month\_of\_year

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `bi_monthly_start`, dat transacties groepeerd in tweemaandelijke segmenten en de begintijdstempel van de set voor elke transactie retourneert.

In dit voorbeeld moeten we echter ook april instellen als de eerste maand van het boekjaar.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        monthsstart(2,date,0,4) as bi_monthly_start,
        timestamp(monthsstart(2,date,0,4)) as bi_monthly_start_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- bi\_monthly\_start
- bi\_monthly\_start\_timestamp

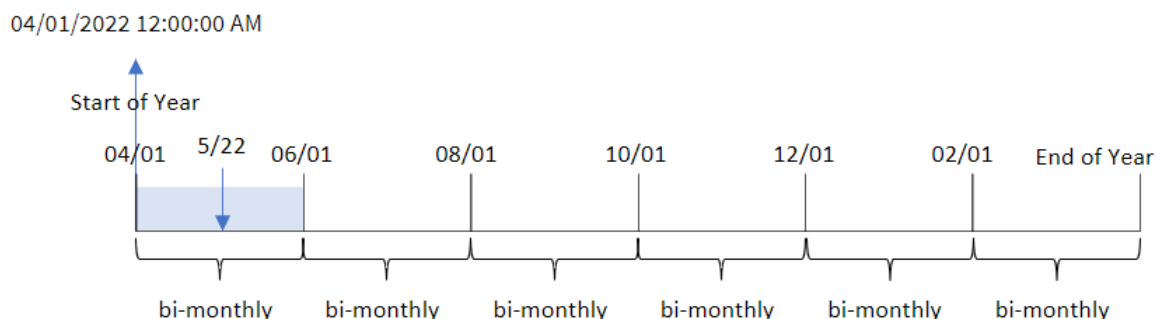
Resultatentabel

date	bi_monthly_start	bi_monthly_start_timestamp
2/19/2022	02/01/2022	2/1/2022 12:00:00 AM
3/7/2022	02/01/2022	2/1/2022 12:00:00 AM

date	bi_monthly_start	bi_monthly_start_timestamp
3/30/2022	02/01/2022	2/1/2022 12:00:00 AM
4/5/2022	04/01/2022	4/1/2022 12:00:00 AM
4/16/2022	04/01/2022	4/1/2022 12:00:00 AM
5/1/2022	04/01/2022	4/1/2022 12:00:00 AM
5/7/2022	04/01/2022	4/1/2022 12:00:00 AM
5/22/2022	04/01/2022	4/1/2022 12:00:00 AM
6/15/2022	06/01/2022	6/1/2022 12:00:00 AM
6/26/2022	06/01/2022	6/1/2022 12:00:00 AM
7/9/2022	06/01/2022	6/1/2022 12:00:00 AM
7/22/2022	06/01/2022	6/1/2022 12:00:00 AM
7/23/2022	06/01/2022	6/1/2022 12:00:00 AM
7/27/2022	06/01/2022	6/1/2022 12:00:00 AM
8/2/2022	08/01/2022	8/1/2022 12:00:00 AM
8/8/2022	08/01/2022	8/1/2022 12:00:00 AM
8/19/2022	08/01/2022	8/1/2022 12:00:00 AM
9/26/2022	08/01/2022	8/1/2022 12:00:00 AM
10/14/2022	10/01/2022	10/1/2022 12:00:00 AM
10/29/2022	10/01/2022	10/1/2022 12:00:00 AM

Door 4 te gebruiken als het `first_month_of_year`-argument in de `monthsstart()`-functie, begint de functie het jaar op 1 april. Het verdeelt vervolgens het jaar in tweemaandelijke segmenten: apr-mei, jun-jul, aug-sep, okt-nov, dec-jan, feb-mrt.

Diagram van de `monthsstart()`-functie, voorbeeld `first_month_of_year`



Transactie 8195 vond plaats op 22 mei en valt in het segment tussen 1 april en 31 mei. Daarom retourneert de functie de eerste milliseconde van dit segment: 1 april 2022 om 12:00:00 AM.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die transacties groepeerd in tweemaandelijks segmenten en de begintijdstempel van de set retourneert voor elke transactie wordt gemaakt als een meting in een diagramobject van de applicatie.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,2/19/2022,37.23
```

```
8189,3/7/2022,17.17
```

```
8190,3/30/2022,88.27
```

```
8191,4/5/2022,57.42
```

```
8192,4/16/2022,53.80
```

```
8193,5/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/22/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Maak de volgende metingen:

```
=monthsstart(2,date)
```

## 5 Script- en diagramfuncties

---

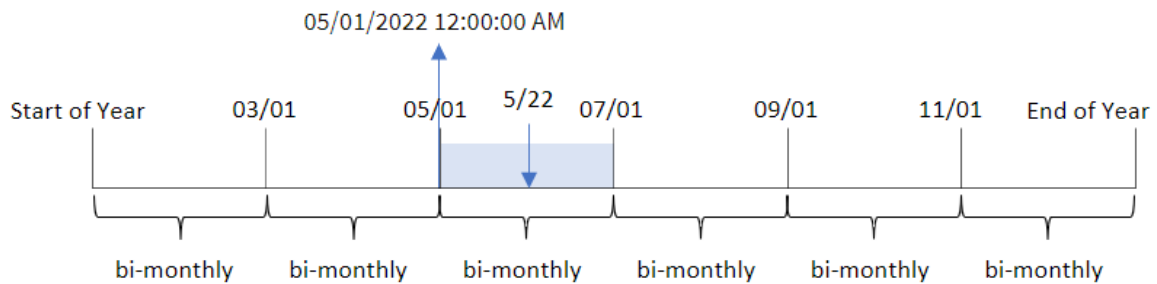
`=timestamp(monthsstart(2,date))`

Deze berekeningen halen de begintijdstempel op van het tweemaandelijks segment waarin iedere transactie plaatsvond.

Resultatentabel

<b>date</b>	<b>=monthsstart(2,date)</b>	<b>=timestamp(monthsstart(2,date))</b>
9/26/2022	09/01/2022	9/1/2022 12:00:00 AM
10/14/2022	09/01/2022	9/1/2022 12:00:00 AM
10/29/2022	09/01/2022	9/1/2022 12:00:00 AM
7/9/2022	07/01/2022	7/1/2022 12:00:00 AM
7/22/2022	07/01/2022	7/1/2022 12:00:00 AM
7/23/2022	07/01/2022	7/1/2022 12:00:00 AM
7/27/2022	07/01/2022	7/1/2022 12:00:00 AM
8/2/2022	07/01/2022	7/1/2022 12:00:00 AM
8/8/2022	07/01/2022	7/1/2022 12:00:00 AM
8/19/2022	07/01/2022	7/1/2022 12:00:00 AM
5/1/2022	05/01/2022	5/1/2022 12:00:00 AM
5/7/2022	05/01/2022	5/1/2022 12:00:00 AM
5/22/2022	05/01/2022	5/1/2022 12:00:00 AM
6/15/2022	05/01/2022	5/1/2022 12:00:00 AM
6/26/2022	05/01/2022	5/1/2022 12:00:00 AM
3/7/2022	03/01/2022	3/1/2022 12:00:00 AM
3/30/2022	03/01/2022	3/1/2022 12:00:00 AM
4/5/2022	03/01/2022	3/1/2022 12:00:00 AM
4/16/2022	03/01/2022	3/1/2022 12:00:00 AM
2/19/2022	01/01/2022	1/1/2021 12:00:00 AM

Diagram van `monthsstart()`-functie, diagramobjectvoorbeeld



Transactie 8195 vond plaats op 22 mei. De functie `monthsstart()` verdeelt het jaar in eerste instantie in tweemaandelijke segmenten. Transactie 8195 valt in het segment tussen 1 mei en 30 juni. Daarom retourneert de functie de eerste milliseconde van dit segment: 05/01/2022 om 12:00:00 AM.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set leningsaldo's die wordt geladen in de tabel `Loans`.
- Gegevens bestaan uit lening-id's, het saldo aan het begin van de maand en de enkelvoudige rente die op iedere lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd voor de door de gebruiker geselecteerde periode. Het boekjaar begint in januari.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Loans:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
loan_id,start_balance,rate
```

```
8188,$10000.00,0.024
```

```
8189,$15000.00,0.057
```

```
8190,$17500.00,0.024
```

```
8191,$21000.00,0.034
```

```
8192,$90000.00,0.084
```

```
];
```



### Resultaten

Laad de gegevens en open een werkblad.

Aan het begin van het load-script is een variabele (`vPeriod`) gemaakt die wordt gekoppeld aan het besturingselement voor variabeleninput. Configureer vervolgens de variabele als een aangepast object in het werkblad.

### Doe het volgende:

1. Klik in het bedrijfsmiddelenvenster op **Aangepaste objecten**.
2. Selecteer **Qlik Dashboard-bundel** en maak een object **Variabeleninput**.
3. Voer een titel in voor het diagramobject.
4. Selecteer onder **Variabele** de optie **vPeriod** als de naam en stel het object in om als een **Vervolgkeuzelijst** te worden weergegeven.
5. Configureer onder **Waarden** het object om dynamische waarden te gebruiken. Voer het volgende in:  
`= '1~month|2~bi-month|3~quarter|4~tertiaal|6~half-year'`

Maak vervolgens de resultatentabel.

### Doe het volgende:

1. Maak een nieuwe tabel. Voeg de volgende velden als dimensies toe:
  - `employee_id`
  - `employee_name`
2. Maak een meting om de opgebouwde rente te berekenen:  
`=start_balance*(rate*(today(1)-monthsstart($(vPeriod),today(1)))/365)`
3. Stel de **Getalnotatie** van de meting in op **Geld**. Klik op  **Klaar met bewerken**. U kunt nu de gegevens in de tabel wijzigen door het tijdsegment in het variabele-object aan te passen.

Dit is hoe de resultatentabel eruit zal zien als de periodeoptie `month` is geselecteerd:

Resultatentabel

<code>loan_id</code>	<code>start_balance</code>	<code>=start_balance*(rate*(today(1)-monthsstart(\$(vPeriod),today(1)))/365)</code>
8188	\$10000.00	\$7.95
8189	\$15000.00	\$67.93
8190	\$17500.00	\$33.37
8191	\$21000.00	\$56.73
8192	\$90000.00	\$600.66

De `monthsstart()`-functie maakt gebruik van de invoer van de gebruiker als het eerste argument en de datum van vandaag als het tweede argument om de begindatum van de door gebruiker geselecteerde periode te retourneren. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking het aantal dagen dat in deze periode tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor deze periode te retourneren. Het resultaat wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die deze periode tot nu toe is opgebouwd.

### monthstart

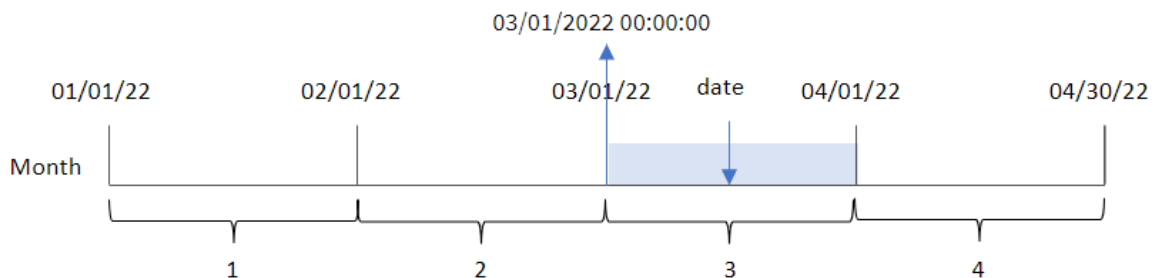
Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de maand die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

#### Syntaxis:

```
MonthStart(date[, period_no])
```

**Retourgegevenstypen:** dual

*Diagram van de monthstart()-functie*



De `monthstart()`-functie bepaalt in welke maand de datum valt. Hij retourneert vervolgens een tijdstempel, in datumnotatie, voor de eerste milliseconde van die maand.

#### Argumenten

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal dat, indien het 0 is of wordt weggelaten, de maand aangeeft die <b>date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande maanden aan en positieve waarden geven volgende maanden aan.

### Wanneer gebruiken

De functie `monthstart()` wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van de maand wordt gebruikt dat al is geweest. Hij kan bijvoorbeeld worden gebruikt om de rente te berekenen die in een maand tot een bepaalde datum is opgebouwd.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>monthstart('10/19/2001')</code>	Retourneert 10/01/2001.
<code>monthstart('10/19/2001', -1)</code>	Retourneert 09/01/2001.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, `start_of_month`, dat een tijdstempel voor het begin van de maand retourneert waarin de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load
    *,
    monthstart(date) as start_of_month,
    timestamp(monthstart(date)) as start_of_month_timestamp
;

Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- start\_of\_month
- start\_of\_month\_timestamp

Resultatentabel

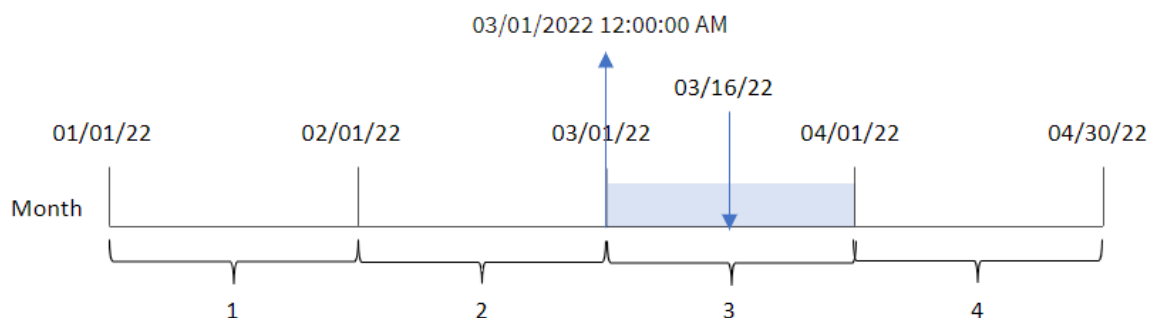
date	start_of_month	start_of_month_timestamp
1/7/2022	01/01/2022	1/1/2022 12:00:00 AM
1/19/2022	01/01/2022	1/1/2022 12:00:00 AM
2/5/2022	02/01/2022	2/1/2022 12:00:00 AM
2/28/2022	02/01/2022	2/1/2022 12:00:00 AM
3/16/2022	03/01/2022	3/1/2022 12:00:00 AM

date	start_of_month	start_of_month_timestamp
4/1/2022	04/01/2022	4/1/2022 12:00:00 AM
5/7/2022	05/01/2022	5/1/2022 12:00:00 AM
5/16/2022	05/01/2022	5/1/2022 12:00:00 AM
6/15/2022	06/01/2022	6/1/2022 12:00:00 AM
6/26/2022	07/01/2022	6/1/2022 12:00:00 AM
7/9/2022	07/01/2022	7/1/2022 12:00:00 AM
7/22/2022	07/01/2022	7/1/2022 12:00:00 AM
7/23/2022	07/01/2022	7/1/2022 12:00:00 AM
7/27/2022	07/01/2022	7/1/2022 12:00:00 AM
8/2/2022	08/01/2022	8/1/2022 12:00:00 AM
8/8/2022	08/01/2022	8/1/2022 12:00:00 AM
8/19/2022	08/01/2022	8/1/2022 12:00:00 AM
9/26/2022	09/01/2022	9/1/2022 12:00:00 AM
10/14/2022	10/01/2022	10/1/2022 12:00:00 AM
10/29/2022	10/01/2022	10/1/2022 12:00:00 AM

Het veld `start_of_month` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `monthstart()` en geeft het datumveld door als het argument van de functie.

De functie `monthstart()` identificeert in welke maand de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van die maand.

*Diagram van `monthstart()`-functie, voorbeeld zonder aanvullende argumenten*



Transactie 8192 vond plaats op 16 maart. De functie `monthstart()` retourneert de eerste milliseconde van die maand, 1 maart om 12:00:00 AM.

### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_month_start`, dat een tijdstempel voor het begin van de maand retourneert voordat de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        monthstart(date,-1) as previous_month_start,
        timestamp(monthstart(date,-1)) as previous_month_start_timestamp
    ;

Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

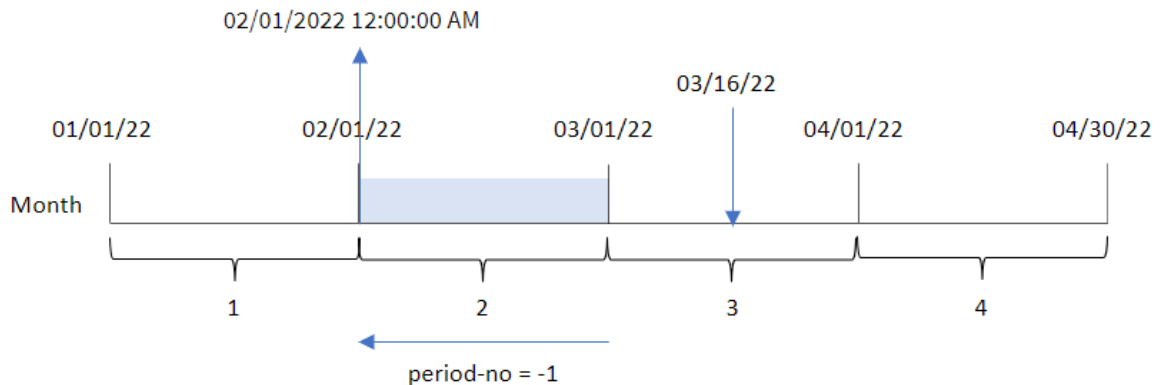
- date
- previous\_month\_start
- previous\_month\_start\_timestamp

Resultatentabel

date	previous_month_start	previous_month_start_timestamp
1/7/2022	12/01/2021	12/1/2021 12:00:00 AM
1/19/2022	12/01/2021	12/1/2021 12:00:00 AM
2/5/2022	01/01/2022	1/1/2022 12:00:00 AM
2/28/2022	01/01/2022	1/1/2022 12:00:00 AM
3/16/2022	02/01/2022	2/1/2022 12:00:00 AM
4/1/2022	03/01/2022	3/1/2022 12:00:00 AM
5/7/2022	04/01/2022	4/1/2022 12:00:00 AM
5/16/2022	04/01/2022	4/1/2022 12:00:00 AM
6/15/2022	05/01/2022	5/1/2022 12:00:00 AM
6/26/2022	05/01/2022	5/1/2022 12:00:00 AM
7/9/2022	06/01/2022	6/1/2022 12:00:00 AM
7/22/2022	06/01/2022	6/1/2022 12:00:00 AM
7/23/2022	06/01/2022	6/1/2022 12:00:00 AM
7/27/2022	06/01/2022	6/1/2022 12:00:00 AM
8/2/2022	07/01/2022	7/1/2022 12:00:00 AM
8/8/2022	07/01/2022	7/1/2022 12:00:00 AM
8/19/2022	07/01/2022	7/1/2022 12:00:00 AM
9/26/2022	08/01/2022	8/1/2022 12:00:00 AM
10/14/2022	09/01/2022	9/1/2022 12:00:00 AM
10/29/2022	09/01/2022	9/1/2022 12:00:00 AM

In dit geval, omdat een `period_no` van -1 werd gebruikt als het `offset`-argument in de `monthstart()`-functie, identificeert de functie eerst de maand waarin de transacties plaatsvinden. Het verschuift dan naar een maand eerder en identificeert de eerste milliseconde van die maand.

Diagram van `monthstart()`-functie, voorbeeld `period_no`



Transactie 8192 vond plaats op 16 maart. De `monthstart()`-functie stelt vast dat de maand voor de transactie februari was. Vervolgens wordt de eerste milliseconde van die maand geretourneerd: 1 februari om 12:00:00 AM.

### Voorbeeld 3 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het begin van de maand waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de applicatie.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```



```
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Maak de volgende metingen om de begindatum van de maand waarin een transactie plaatsvindt te berekenen:

- =monthstart(date)
- =timestamp(monthstart(date))

Resultatentabel

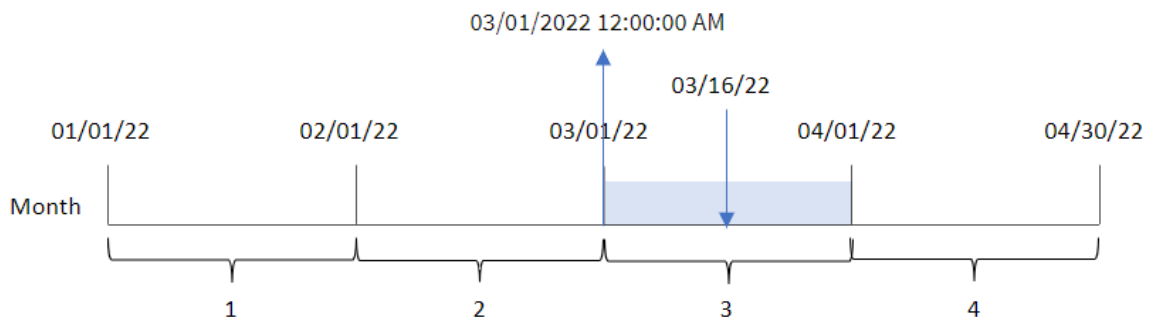
date	=monthstart(date)	=timestamp(monthstart(date))
10/14/2022	10/01/2022	10/1/2022 12:00:00 AM
10/29/2022	10/01/2022	10/1/2022 12:00:00 AM
9/26/2022	09/01/2022	9/1/2022 12:00:00 AM
8/2/2022	08/01/2022	8/1/2022 12:00:00 AM
8/8/2022	08/01/2022	8/1/2022 12:00:00 AM
8/19/2022	08/01/2022	8/1/2022 12:00:00 AM
7/9/2022	07/01/2022	7/1/2022 12:00:00 AM
7/22/2022	07/01/2022	7/1/2022 12:00:00 AM
7/23/2022	07/01/2022	7/1/2022 12:00:00 AM
7/27/2022	07/01/2022	7/1/2022 12:00:00 AM
6/15/2022	06/01/2022	6/1/2022 12:00:00 AM
6/26/2022	06/01/2022	6/1/2022 12:00:00 AM
5/7/2022	05/01/2022	5/1/2022 12:00:00 AM
5/16/2022	05/01/2022	5/1/2022 12:00:00 AM
4/1/2022	04/01/2022	4/1/2022 12:00:00 AM

date	=monthstart(date)	=timestamp(monthstart(date))
3/16/2022	03/01/2022	3/1/2022 12:00:00 AM
2/5/2022	02/01/2022	2/1/2022 12:00:00 AM
2/28/2022	02/01/2022	2/1/2022 12:00:00 AM
1/7/2022	01/01/2022	1/1/2022 12:00:00 AM
1/19/2022	01/01/2022	1/1/2022 12:00:00 AM

De meting `start_of_month` wordt gemaakt in het diagramobject met behulp van de functie `monthstart()` en geeft het veld door als het argument van de functie.

De functie `monthstart()` identificeert in welke maand de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van die maand.

*Diagram van `monthstart()`-functie, diagramobjectvoorbeeld*



Transactie 8192 vond plaats op 16 maart. De `monthstart()`-functie identificeert dat de transactie in maart heeft plaatsgevonden en retourneert de eerste milliseconde van die maand, 1 maart om 12:00:00 AM.

### Voorbeeld 4 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set leningsaldo's die wordt geladen in de tabel `Loans`.
- Gegevens bestaan uit lening-id's, het saldo aan het begin van de maand en de enkelvoudige rente die op iedere lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd op elke lening in de maand tot heden.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Loans:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
loan_id,start_balance,rate
```

```
8188,$10000.00,0.024
```

```
8189,$15000.00,0.057
```

```
8190,$17500.00,0.024
```

```
8191,$21000.00,0.034
```

```
8192,$90000.00,0.084
```

```
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
  - loan\_id
  - start\_balance
2. Maak nu een meting om de opgebouwde rente te berekenen:  
 $=start\_balance*(rate*(today(1)-monthstart(today(1)))/365)$
3. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

loan_id	start_balance	$=start\_balance*(rate*(today(1)-monthstart(today(1)))/365)$
8188	\$10000.00	\$16.44
8189	\$15000.00	\$58.56
8190	\$17500.00	\$28.77
8191	\$21000.00	\$48.90
8192	\$90000.00	\$517.81

De functie `monthstart()` gebruikt de datum van vandaag als enige argument en retourneert de begindatum van de huidige maand. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking het aantal dagen dat in deze maand tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor deze periode te retourneren. Het resultaat wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die deze maand tot nu toe is opgebouwd.

### networkdays

De functie **networkdays** retourneert het aantal werkdagen (maandag-vrijdag) tussen **start\_date** en **end\_date**, waarbij rekening wordt gehouden met een eventueel opgegeven **holiday**.

#### Syntaxis:

```
networkdays (start_date, end_date [, holiday])
```

**Retourgegevenstypen:** geheel getal

*Kalenderdiagram dat het datumbereik weergeeft dat wordt geretourneerd door de networkdays-functie*

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10 start_date	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26 end_date	27
28	29	30	31			

De networkdays-functie heeft de volgende beperkingen:

- Er is geen methode om werkdagen aan te passen. Met andere woorden: er is geen manier om de functie aan te passen voor regio's of situaties waarbij sprake is van iets anders dan werk van maandag t/m vrijdag.
- De holiday-parameter moet een tekenreeksconstante zijn. Uitdrukkingen worden niet geaccepteerd.

#### Argumenten

Argument	Beschrijving
<b>start_date</b>	De te evalueren begindatum.
<b>end_date</b>	De te evalueren einddatum.

Argument	Beschrijving
<b>holiday</b>	Vakantieperioden die moeten worden uitgesloten van werkdagen. Een vakantie wordt aangegeven als een tekenreeks met constante datum. U kunt meerdere vakantiedatums opgeven, gescheiden door komma's.  <b>Voorbeeld:</b> '12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014'

### Wanneer gebruiken

De `networkdays()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het aantal werkweekdagen gebruikt dat tussen twee datums valt. Bijvoorbeeld als een gebruiker het totale salaris wil berekenen dat een werknemer met een PAYE-contract (pay-as-you-earn) verdient.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>networkdays ('12/19/2013', '01/07/2014')</code>	Retourneert 14. In dit voorbeeld wordt geen rekening gehouden met vakanties.
<code>networkdays ('12/19/2013', '01/07/2014', '12/25/2013', '12/26/2013')</code>	Retourneert 12. In dit voorbeeld wordt rekening gehouden met de vakantie van 12/25/2013 t/m 12/26/2013.
<code>networkdays ('12/19/2013', '01/07/2014', '12/25/2013', '12/26/2013', '12/31/2013', '01/01/2014')</code>	Retourneert 10. In dit voorbeeld wordt rekening gehouden met twee vakantieperioden.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met project-id's, hun begindatums en hun einddatums. Deze informatie wordt geladen in de tabel `Projects`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Er wordt een extra veld gemaakt, `net_work_days`, om het aantal werkdagen voor elk project te berekenen.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';

Projects:
  Load
    *,
    networkdays(start_date,end_date) as net_work_days
  ;

Load
id,
start_date,
end_date
Inline
[
id,start_date,end_date
1,01/01/2022,01/18/2022
2,02/10/2022,02/17/2022
3,05/17/2022,07/05/2022
4,06/01/2022,06/12/2022
5,08/10/2022,08/26/2022
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `start_date`
- `end_date`
- `net_work_days`

## 5 Script- en diagramfuncties

Resultatentabel

id	start_date	end_date	net_work_days
1	01/01/2022	01/18/2022	12
2	02/10/2022	02/17/2022	6
3	05/17/2022	07/05/2022	36
4	06/01/2022	06/12/2022	8
5	08/10/2022	08/26/2022	13

Omdat er geen geplande vakantiedagen zijn (deze zouden aanwezig zijn geweest in het derde argument van de `networkdays()`-functie), trekt de functie de `start_date` af van de `end_date`, evenals alle weekenddagen, om het aantal werkdagen tussen de twee datums te berekenen.

*Kalenderdiagram dat werkdagen voor project 5 (geen vakantiedagen) markeert*

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

De bovenstaande kalender markeert het project visueel met de `id` 5. Project 5 begint op woensdag, 10 augustus 2022 en eindigt op 26 augustus 2022. Doordat alle zaterdagen en zondagen worden genegeerd, zitten er 13 werkdagen tussen deze twee datums.

### Voorbeeld 2 - Enkele vakantie

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het vorige voorbeeld.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Er wordt een extra veld gemaakt, `net_work_days`, om het aantal werkdagen voor elk project te berekenen.

In dit voorbeeld staat er één vakantiedag gepland op 19 augustus 2022.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Projects:
```

```
    Load
        *,
        networkdays(start_date,end_date,'08/19/2022') as net_work_days
    ;

Load
id,
start_date,
end_date
Inline
[
id,start_date,end_date
1,01/01/2022,01/18/2022
2,02/10/2022,02/17/2022
3,05/17/2022,07/05/2022
4,06/01/2022,06/12/2022
5,08/10/2022,08/26/2022
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `start_date`
- `end_date`
- `net_work_days`



## 5 Script- en diagramfuncties

Resultatentabel

id	start_date	end_date	net_work_days
1	01/01/2022	01/18/2022	12
2	02/10/2022	02/17/2022	6
3	05/17/2022	07/05/2022	36
4	06/01/2022	06/12/2022	8
5	08/10/2022	08/26/2022	12

De enkele, geplande vakantie is ingevoerd als het derde argument in de `networkdays()`-functie.

*Kalenderdiagram dat werkdagen voor project 5 (één vakantiedag) markeert*

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19 Holiday	20
21	22	23	24	25	26	27
28	29	30	31			

De bovenstaande kalender markeert project 5 visueel en toont deze aanpassing waarbij de vakantiedag is toegevoegd. Deze vakantiedag valt in project 5 op vrijdag, 19 augustus 2022. Het resultaat is dat de totale `net_work_days`-waarde voor project 5 afneemt met één dag, van 13 naar 12 dagen.

### Voorbeeld 3 - Meerdere vakantiedagen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Er wordt een extra veld gemaakt, `net_work_days`, om het aantal werkdagen voor elk project te berekenen.

Maar in dit voorbeeld staan er vier vakantiedagen gepland, van 18 augustus t/m 21 augustus 2022.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';

Projects:
  Load
    *,
    networkdays(start_date,end_date,'08/18/2022','08/19/2022','08/20/2022','08/21/2022')
  as net_work_days
  ;

Load
id,
start_date,
end_date
Inline
[
id,start_date,end_date
1,01/01/2022,01/18/2022
2,02/10/2022,02/17/2022
3,05/17/2022,07/05/2022
4,06/01/2022,06/12/2022
5,08/10/2022,08/26/2022
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `start_date`
- `end_date`
- `net_work_days`

## 5 Script- en diagramfuncties

Resultatentabel

id	start_date	end_date	net_work_days
1	01/01/2022	01/18/2022	12
2	02/10/2022	02/17/2022	6
3	05/17/2022	07/05/2022	36
4	06/01/2022	06/12/2022	8
5	08/10/2022	08/26/2022	11

De vier geplande vakantiedagen worden ingevoerd als een door komma's gescheiden lijst, vanaf het derde argument tot in de `networkdays()`-functie.

*Kalenderdiagram dat werkdagen voor project 5 (meerdere vakantiedagen) markeert*

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18 Holiday	19 Holiday	20
21	22	23	24	25	26	27
28	29	30	31			

De bovenstaande kalender markeert project 5 visueel en toont deze aanpassing waarbij deze vakantiedagen zijn toegevoegd. Deze periode van geplande vakantiedagen vindt plaats tijdens project 5, waarbij twee van de dagen op een donderdag en vrijdag vallen. Het resultaat is dat de totale `net_work_days`-waarde voor project 5 afneemt van 13 naar 11 dagen.

### Voorbeeld 4 - Enkele vakantiedag

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.

Er staat één vakantiedag gepland op 19 augustus 2022.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. Het `net_work_days`-veld wordt berekend als een meting in een diagramobject.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Projects:
```

```
Load
```

```
id,
```

```
start_date,
```

```
end_date
```

```
Inline
```

```
[
```

```
id,start_date,end_date
```

```
1,01/01/2022,01/18/2022
```

```
2,02/10/2022,02/17/2022
```

```
3,05/17/2022,07/05/2022
```

```
4,06/01/2022,06/12/2022
```

```
5,08/10/2022,08/26/2022
```

```
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `start_date`
- `end_date`

Maak de volgende meting:

```
= networkdays(start_date,end_date,'08/19/2022')
```

Resultatentabel

id	start_date	end_date	net_work_days
1	01/01/2022	01/18/2022	12
2	02/10/2022	02/17/2022	6
3	05/17/2022	07/05/2022	36
4	06/01/2022	06/12/2022	8
5	08/10/2022	08/26/2022	12

De enkele, geplande vakantie is ingevoerd als het derde argument in de `networkdays()`-functie.

*Het kalenderdiagram toont nettowerkdagen met één vakantiedag (diagramobject)*

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19 Holiday	20
21	22	23	24	25	26	27
28	29	30	31			

De bovenstaande kalender markeert project 5 visueel en toont deze aanpassing waarbij de vakantiedag is toegevoegd. Deze vakantiedag valt in project 5 op vrijdag, 19 augustus 2022. Het resultaat is dat de totale `net_work_days`-waarde voor project 5 afneemt met één dag, van 13 naar 12 dagen.

### now

Deze functie retourneert een tijdstempel van de huidige tijd. Deze functie retourneert waarden in de indeling van de systeemvariabele **TimeStamp**. De standaardwaarde voor **timer\_mode** is 1.


#### Syntaxis:

```
now([ timer_mode])
```

#### Retourgegevenstypen: dual

De `now()`-functie kan ofwel in het load-script ofwel in diagramobjecten worden gebruikt.

#### Argumenten

Argument	Beschrijving
timer_mode	<p>Kan de volgende waarden hebben:</p> <ul style="list-style-type: none"><li>0 (tijd van de als laatste voltooide laadbewerking voor gegevens)</li><li>1 (tijd van de functieaanroep)</li><li>2 (tijd wanneer de app werd geopend)</li></ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <i>Als u de functie gebruikt in een script voor het laden van gegevens, resulteert <b>timer_mode=0</b> in de tijd van de als laatste voltooide laadbewerking voor gegevens, terwijl <b>timer_mode=1</b> de tijd van de functieaanroep in de huidige laadbewerking voor gegevens oplevert.</i></div>

### Wanneer gebruiken

De `now()`-functie wordt meestal gebruikt als een component in een uitdrukking. Hij kan bijvoorbeeld worden gebruikt om de resterende tijd te berekenen in een levenscyclus van een project. De `now()`-functie zou in plaats van de `today()`-functie worden gebruikt wanneer de uitdrukking het gebruik van een deel van een dag vereist.

De volgende tabel geeft een uitleg van het resultaat dat door de `now()`-functie is geretourneerd en geeft verschillende waarden voor het `timer_mode`-argument:

### Voorbeelden van functies

timer_ mode- waarde	Resultaat bij gebruik in load-script	Resultaat bij gebruik in diagramobject
0	Retourneert een tijdstempel met de notatie van de <code>TimeStamP</code> -systeemvariabele, van de laatste keer dat de gegevens opnieuw zijn geladen voorafgaand aan de laatste keer dat de gegevens opnieuw zijn geladen.	Retourneert een tijdstempel met de notatie van de <code>TimeStamP</code> -systeemvariabele voor de laatste keer dat de gegevens opnieuw zijn geladen.
1	Retourneert een tijdstempel met de notatie van de <code>TimeStamP</code> -systeemvariabele voor de laatste keer dat de gegevens opnieuw zijn geladen.	Retourneert een tijdstempel met de notatie van de <code>TimeStamP</code> -systeemvariabele van de functieaanroep.
2	Retourneert een tijdstempel met de notatie van de <code>TimeStamP</code> -systeemvariabele voor het begin van de sessie van de gebruiker in de applicatie. Dit wordt niet bijgewerkt tenzij de gebruiker het script opnieuw laadt.	Retourneert een tijdstempel met de notatie van de <code>TimeStamP</code> -systeemvariabele voor het begin van de sessie van de gebruiker in de applicatie. Dit wordt vernieuwd zodra een nieuwe sessie begint of de gegevens weer in de applicatie zijn geladen.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Genereren van objecten met behulp van het load-script

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Dit voorbeeld creëert die variabelen met behulp van de `now()`-functie. Elke variabele gebruikt een van de `timer_mode`-opties om het effect van de variabele te demonstreren.

Laad het script opnieuw en herlaad het na een korte tijdsperiode zodat de variabelen hun doel kunnen laten zien. De variabelen `now(0)` en `now(1)` zullen nu verschillende waarden laten zien en hun doel demonstreren.

### Load-script

```
LET vPreviousDataLoad = now(0);
LET vCurrentDataLoad = now(1);
LET vApplicationOpened = now(2);
```

### Resultaten

Nadat de gegevens voor de tweede keer zijn geladen, maakt u drie tekstvakken door de onderstaande instructies te volgen.

Maak eerst een tekstvak voor de gegevens die eerder zijn geladen.

#### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:  
=`vPreviousDataLoad`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'Previous Reload Time' toe aan het object.

Maak vervolgens een tekstvak voor de gegevens die momenteel worden geladen.

#### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:  
=`vCurrentDataLoad`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'Current Reload Time' toe aan het object.

Maak een definitief tekstvak dat wordt getoond als de sessie van de gebruiker in de toepassing wordt gestart.

#### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:  
=`vApplicationOpened`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'User Session Started' toe aan het object.



*now()-load-scriptvariabelen*

<b>Previous Reload Time</b> 6/22/2022 8:54:03 AM	<b>Current Reload Time</b> 6/22/2022 9:02:08 AM	<b>User Session Began</b> 6/22/2022 8:40:40 AM
---	--	---

De bovenstaande afbeelding geeft voorbeeldwaarden weer voor elk van de gemaakte variabelen. De waarden zouden bijvoorbeeld als volgt kunnen zijn:

- Vorige herlaadtijd: 6/22/2022 8:54:03 AM
- Huidige herlaadtijd: 6/22/2022 9:02:08 AM
- Start gebruikerssessie: 6/22/2022 8:40:40 AM

### Voorbeeld 2 - Genereren van objecten zonder het load-script

Load-script en diagramuitdrukking

#### Overzicht

In dit voorbeeld maakt u drie diagramobjecten met de `now()`-functie zonder dat er variabelen of gegevens in de applicatie worden geladen. Elk diagramobject gebruikt een van de `timer_mode`-opties om het effect te demonstreren.

Er is geen load-script voor dit voorbeeld.

#### Doe het volgende:

1. Open de Editor voor laden van gegevens.
2. Klik zonder het bestaande load-script te wijzigen op **Gegevens laden**.
3. Laad het script na korte tijd nog een tweede keer.

#### Resultaten

Maak drie tekstvakken zodra de gegevens voor een tweede keer zijn geladen.

Maak eerst een tekstvak voor de laatste keer dat de gegevens opnieuw zijn geladen.

#### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:  
`=now(0)`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel Laatste herlaadtijd gegevens toe aan het object.

Maak nu een tekstvak om de actuele tijd te tonen.

### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:  
`=now(1)`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel Actuele tijd toe aan het object.

Maak een definitief tekstvak dat wordt getoond als de sessie van de gebruiker in de toepassing wordt gestart.

### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:  
`=now(2)`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel Start gebruikerssessie toe aan het object.

Voorbeelden van `now()`-diagramobject

<b>Latest Data Reload</b> 6/22/2022 9:02:08 AM	<b>Current Time</b> 6/22/2022 9:25:16 AM	<b>User Session Began</b> 6/22/2022 8:40:40 AM
---	---	---

De bovenstaande afbeelding geeft voorbeeldwaarden weer voor elk van de gemaakte objecten. De waarden zouden bijvoorbeeld als volgt kunnen zijn:

- Laatste herlaadtijd gegevens: 6/22/2022 9:02:08 AM
- Huidige tijd: 6/22/2022 9:25:16 AM
- Start gebruikerssessie: 6/22/2022 8:40:40 AM

Het diagramobject Laatste herlaadtijd gegevens maakt gebruik van een `timer_mode`-waarde van 0. Dit retourneert de tijdstempel voor de laatste keer dat de gegevens opnieuw zijn geladen.

Het diagramobject Actuele tijd gebruikt een `timer_mode`-waarde van 1. Dit retourneert de actuele tijd volgens de systeemklok. Als het werkblad of object wordt vernieuwd, wordt deze waarde bijgewerkt.

Het diagramobject Start gebruikerssessie gebruikt een `timer_mode`-waarde van 2. Dit retourneert de tijdstempel voor het moment waarop de applicatie werd geopend en waarop de gebruikerssessie is begonnen.

### Voorbeeld 3 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die bestaat uit de voorraad voor een cryptovaluta-miningactiviteit die wordt geladen in de tabel `Inventory`.
- Gegevens met de volgende velden: `id`, `purchase_date` en `wph` (watt per uur).

De gebruiker wil een tabel hebben die de totale kosten per `id` weergeeft die iedere mining-rig in de maand tot nu toe heeft gemaakt wat betreft stroomverbruik.

Deze waarde moet worden bijgewerkt wanneer het diagramobject wordt vernieuwd. De huidige prijs voor elektriciteit is USD 0,0678 per kWu.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Inventory:
Load
*
Inline
[
id,purchase_date,wph
8188,1/7/2022,1123
8189,1/19/2022,1432
8190,2/28/2022,1227
8191,2/5/2022,1322
8192,3/16/2022,1273
8193,4/1/2022,1123
8194,5/7/2022,1342
8195,5/16/2022,2342
8196,6/15/2022,1231
8197,6/26/2022,1231
8198,7/9/2022,1123
8199,7/22/2022,1212
8200,7/23/2022,1223
8201,7/27/2022,1232
8202,8/2/2022,1232
8203,8/8/2022,1211
8204,8/19/2022,1243
8205,9/26/2022,1322
8206,10/14/2022,1133
8207,10/29/2022,1231
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: id.

Maak de volgende meting:

```
=(now(1)-monthstart(now(1)))*24*wph/1000*0.0678
```

Als het diagramobject is vernieuwd op 6/22/2022 om 10:39:05 AM, dan zou het de volgende resultaten retourneren:

Resultatentabel

id	=(now(1)-monthstart(now(1)))*24*wph/1000*0.0678
8188	\$39.18
8189	\$49.97
8190	\$42.81
8191	\$46.13
8192	\$44.42
8193	\$39.18
8194	\$46.83
8195	\$81.72
8196	\$42.95
8197	\$42.95
8198	\$39.18
8199	\$42.29
8200	\$42.67
8201	\$42.99
8202	\$42.99
8203	\$42.25
8204	\$43.37
8205	\$46.13
8206	\$39.53

De gebruiker wil graag dat de objectresultaten elke keer dat het object wordt vernieuwd, ook worden vernieuwd. Daarom staat het `timer_mode`-argument in instanties van de `now()`-functie in de uitdrukking. De tijdstempel voor het begin van de maand, die is vastgesteld met behulp van de `now()`-functie als het tijdstempelargument in de `monthstart()`-functie, wordt afgetrokken van de huidige tijd die wordt vastgesteld door de `now()`-functie. Dit verstrekt de totale hoeveelheid tijd die tot nu toe in deze maand (in dagen) is verstreken.

Deze waarde wordt vermenigvuldigd met 24 (het aantal uren in een dag) en vervolgens met de waarde in het `wph`-veld.

Voor de conversie van watt per uur naar kilowatt per uur, wordt het resultaat gedeeld door 1000 voordat het uiteindelijk wordt vermenigvuldigd met het verstrekte kWu-tarief.

### quarterend

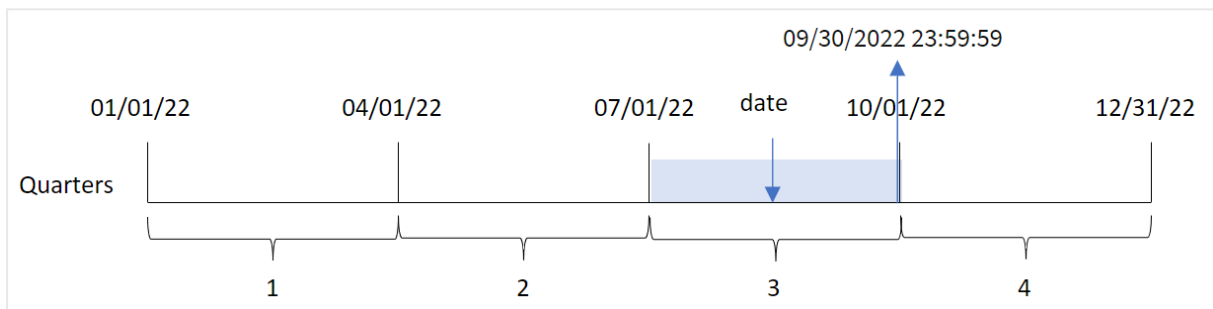
Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van het kwartaal dat `date` bevat. De standaarduitvoernotatie is de `DateFormat` die is ingesteld in het script.

#### Syntaxis:

```
QuarterEnd(date[, period_no[, first_month_of_year]])
```

#### Retourgegevenstypen: dual

Diagram van de `quarterend()`-functie



De `quarterend()`-functie bepaalt welk kwartaal de datum in valt. Hij retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van de laatste maand van dat kwartaal. De eerste maand van het jaar is standaard januari. U kunt echter ook wijzigen welke maand als eerste wordt ingesteld met behulp van het argument `first_month_of_year` in de functie `quarterend()` te gebruiken.



*De `quarterend()`-functie houdt geen rekening met de systeemvariabele `FirstMonthOfYear`. Het jaar begint op 1 januari, tenzij het argument `first_month_of_year` wordt gebruikt om het te wijzigen.*

#### Wanneer gebruiken

De `quarterend()`-functie wordt voornamelijk gebruikt als onderdeel van een uitdrukking als u wilt dat in de berekening een deel van het kwartaal wordt gebruikt dat nog niet heeft plaatsgevonden. Bijvoorbeeld als u de totale rente wilt berekenen die nog niet is berekend gedurende het kwartaal.

### Argumenten

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal, waarbij de waarde 0 het kwartaal aangeeft dat <b>date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande kwartalen aan en positieve waarden geven volgende kwartalen aan.
<b>first_month_of_year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> .

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden `first_month_of_year`

Month	Waarde
Februari	2
Maart	3
April	4
May	5
Juni	6
Juli	7
Augustus	8
September	9
Oktober	10
November	11
December	12

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze

landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>quarterend('10/29/2005')</code>	Returns 12/31/2005 23:59:59.
<code>quarterend('10/29/2005', -1)</code>	Returns 09/30/2005 23:59:59.
<code>quarterend('10/29/2005', 0, 3)</code>	Returns 11/30/2005 23:59:59.

### Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `Transactions`.
- Een voorgaande lading die het volgende bevat:
  - De `quarterend()`-functie die is ingesteld als het `end_of_quarter`-veld en een tijdstempel retourneert voor het eind van het kwartaal waarin de transacties plaatsvonden.
  - De `timestamp()`-functie die is ingesteld als het `end_of_quarter_timestamp`-veld en de exacte tijdstempel retourneert voor het eind van het geselecteerde kwartaal.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*
*,
quarterend(date) as end_of_quarter,
timestamp(quarterend(date)) as end_of_quarter_timestamp
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194, 5/7/2022, 40.39
8195, 5/16/2022, 87.21
8196, 6/15/2022, 95.93
8197, 6/26/2022, 45.89
8198, 7/9/2022, 36.23
8199, 7/22/2022, 25.66
8200, 7/23/2022, 82.77
8201, 7/27/2022, 69.98
8202, 8/2/2022, 76.11
8203, 8/8/2022, 25.12
8204, 8/19/2022, 46.23
8205, 9/26/2022, 84.21
8206, 10/14/2022, 96.24
8207, 10/29/2022, 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- end\_of\_quarter
- end\_of\_quarter\_timestamp

Resultatentabel

id	date	end_of_quarter	end_of_quarter_timestamp
8188	1/7/2022	03/31/2022	3/31/2022 11:59:59 PM
8189	1/19/2022	03/31/2022	3/31/2022 11:59:59 PM
8190	2/5/2022	03/31/2022	3/31/2022 11:59:59 PM
8191	2/28/2022	03/31/2022	3/31/2022 11:59:59 PM
8192	3/16/2022	03/31/2022	3/31/2022 11:59:59 PM
8193	4/1/2022	06/30/2022	6/30/2022 11:59:59 PM
8194	5/7/2022	06/30/2022	6/30/2022 11:59:59 PM
8195	5/16/2022	06/30/2022	6/30/2022 11:59:59 PM
8196	6/15/2022	06/30/2022	6/30/2022 11:59:59 PM
8197	6/26/2022	06/30/2022	6/30/2022 11:59:59 PM
8198	7/9/2022	09/30/2022	9/30/2022 11:59:59 PM
8199	7/22/2022	09/30/2022	9/30/2022 11:59:59 PM
8200	7/23/2022	09/30/2022	9/30/2022 11:59:59 PM
8201	7/27/2022	09/30/2022	9/30/2022 11:59:59 PM

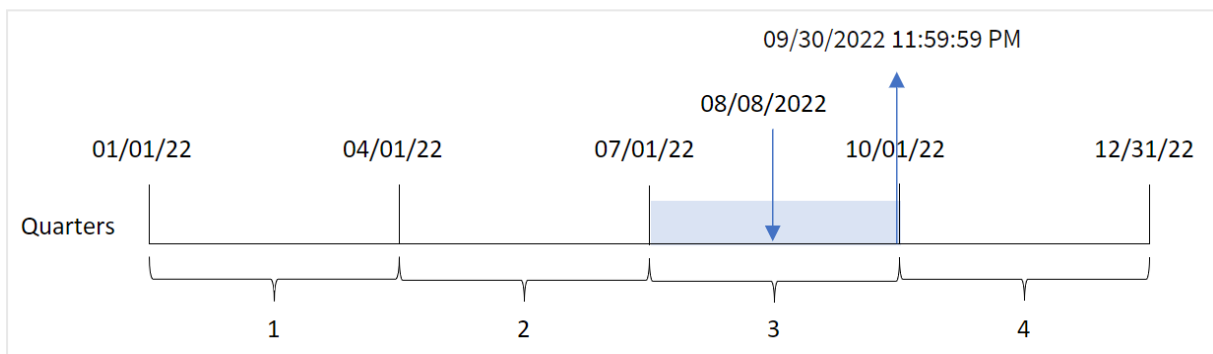


id	date	end_of_quarter	end_of_quarter_timestamp
8202	8/2/2022	09/30/2022	9/30/2022 11:59:59 PM
8203	8/8/2022	09/30/2022	9/30/2022 11:59:59 PM
8204	8/19/2022	09/30/2022	9/30/2022 11:59:59 PM
8205	9/26/2022	09/30/2022	9/30/2022 11:59:59 PM
8206	10/14/2022	12/31/2022	12/31/2022 11:59:59 PM
8207	10/29/2022	12/31/2022	12/31/2022 11:59:59 PM

Het `end_of_quarter`-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de `quarterend()`-functie en geeft het datumveld door als het argument van de functie.

De `quarterend()`-functie identificeert aanvankelijk in welk kwartaal de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van dat kwartaal.

*Diagram van de `quarterend()`-functie waarbij het kwartaaleinde van transactie 8203 is geïdentificeerd*



Transactie 8203 vond plaats op 8 augustus. De `quarterend()`-functie identificeert dat de transactie in het derde kwartaal plaatsvond en retourneert de laatste milliseconde van dat kwartaal. Dat is 30 september om 11:59:59 PM.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `transactions`.
- Een voorgaande lading die het volgende bevat:

- De `quarterend()`-functie die is ingesteld als het `previous_quarter_end`-veld en een tijdstempel retourneert voor het eind van het kwartaal vóór het kwartaal waarin de transactie plaatsvond.
- De `timestamp()`-functie die is ingesteld als het `previous_end_of_quarter_timestamp`-veld en de exacte tijdstempel retourneert voor het eind van het kwartaal vóór het kwartaal waarin de transactie plaatsvond.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        quarterend(date, -1) as previous_quarter_end,
        timestamp(quarterend(date, -1)) as previous_quarter_end_timestamp
    ;

Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`

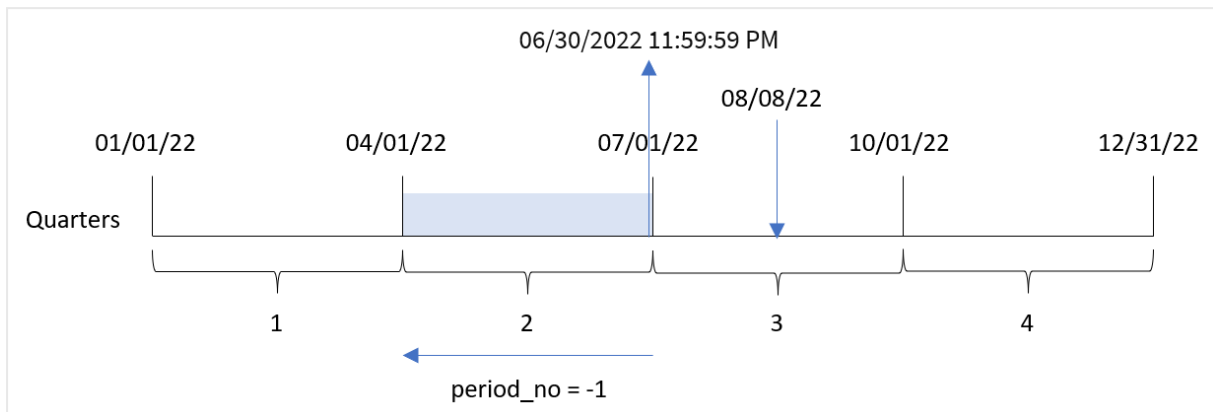
- `previous_quarter_end`
- `previous_quarter_end_timestamp`

Resultatentabel

<b>id</b>	<b>date</b>	<b>previous_quarter_end</b>	<b>previous_quarter_end_timestamp</b>
8188	1/7/2022	12/31/2021	12/31/2021 11:59:59 PM
8189	1/19/2022	12/31/2021	12/31/2021 11:59:59 PM
8190	2/5/2022	12/31/2021	12/31/2021 11:59:59 PM
8191	2/28/2022	12/31/2021	12/31/2021 11:59:59 PM
8192	3/16/2022	12/31/2021	12/31/2021 11:59:59 PM
8193	4/1/2022	03/31/2022	3/31/2022 11:59:59 PM
8194	5/7/2022	03/31/2022	3/31/2022 11:59:59 PM
8195	5/16/2022	03/31/2022	3/31/2022 11:59:59 PM
8196	6/15/2022	03/31/2022	3/31/2022 11:59:59 PM
8197	6/26/2022	03/31/2022	3/31/2022 11:59:59 PM
8198	7/9/2022	06/30/2022	6/30/2022 11:59:59 PM
8199	7/22/2022	06/30/2022	6/30/2022 11:59:59 PM
8200	7/23/2022	06/30/2022	6/30/2022 11:59:59 PM
8201	7/27/2022	06/30/2022	6/30/2022 11:59:59 PM
8202	8/2/2022	06/30/2022	6/30/2022 11:59:59 PM
8203	8/8/2022	06/30/2022	6/30/2022 11:59:59 PM
8204	8/19/2022	06/30/2022	6/30/2022 11:59:59 PM
8205	9/26/2022	06/30/2022	6/30/2022 11:59:59 PM
8206	10/14/2022	09/30/2022	9/30/2022 11:59:59 PM
8207	10/29/2022	09/30/2022	9/30/2022 11:59:59 PM

Omdat een `period_no` van `-1` wordt gebruikt als het `offset`-argument in de functie `quarterend()`, identificeert de functie eerst het kwartaal waarin de transacties plaatsvinden. Het verschuift dan naar een kwartaal eerder en identificeert de laatste milliseconde van dat kwartaal.

Diagram van de `quarterend()`-functie met een `period_no` van -1.



Transactie 8203 vond plaats op 8 augustus. De `quarterend()`-functie identificeert dat het kwartaal vóór het kwartaal waarin de transactie plaatsvond tussen 1 april en 30 juni viel. De functie retourneert vervolgens de laatste milliseconde van dat kwartaal, namelijk 30 juni om 11:59:59 PM.

### Voorbeeld 3 - `first_month_of_year`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een reeks transacties in 2022 die wordt geladen in de tabel `Transactions`.
- Een voorgaande lading die het volgende bevat:
  - De `quarterend()`-functie die is ingesteld als het `end_of_quarter`-veld en een tijdstempel retourneert voor het eind van het kwartaal waarin de transacties plaatsvonden.
  - De `timestamp()`-functie die is ingesteld als het `end_of_quarter_timestamp`-veld en de exacte tijdstempel retourneert voor het eind van het geselecteerde kwartaal.

Maar in dit voorbeeld is het bedrijfsbeleid dat het boekjaar op 1 maart begint.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    quarterend(date, 0, 3) as end_of_quarter,
    timestamp(quarterend(date, 0, 3)) as end_of_quarter_timestamp
  ;
Load
*
Inline
```

```
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

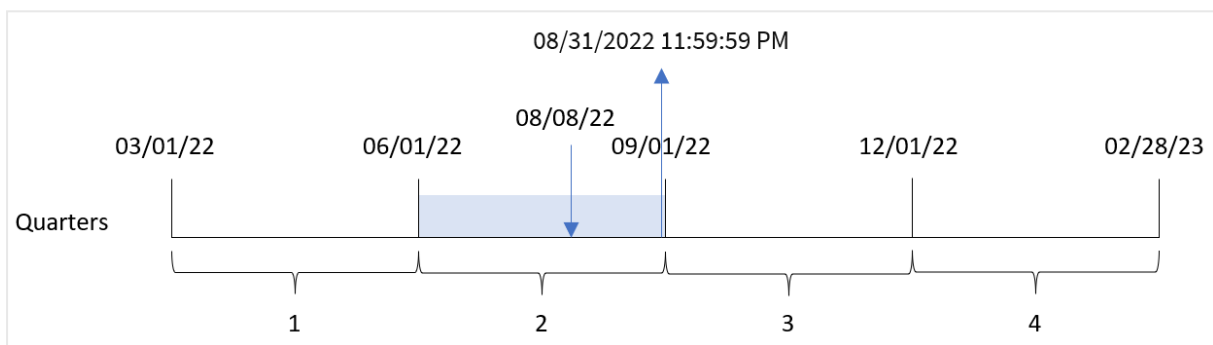
Resultatentabel

id	date	end_of_quarter	end_of_quarter_timestamp
8188	1/7/2022	02/28/2022	2/28/2022 11:59:59 PM
8189	1/19/2022	02/28/2022	2/28/2022 11:59:59 PM
8190	2/5/2022	02/28/2022	2/28/2022 11:59:59 PM
8191	2/28/2022	02/28/2022	2/28/2022 11:59:59 PM
8192	3/16/2022	05/31/2022	5/31/2022 11:59:59 PM
8193	4/1/2022	05/31/2022	5/31/2022 11:59:59 PM
8194	5/7/2022	05/31/2022	5/31/2022 11:59:59 PM
8195	5/16/2022	05/31/2022	5/31/2022 11:59:59 PM
8196	6/15/2022	08/31/2022	8/31/2022 11:59:59 PM
8197	6/26/2022	08/31/2022	8/31/2022 11:59:59 PM
8198	7/9/2022	08/31/2022	8/31/2022 11:59:59 PM
8199	7/22/2022	08/31/2022	8/31/2022 11:59:59 PM
8200	7/23/2022	08/31/2022	8/31/2022 11:59:59 PM
8201	7/27/2022	08/31/2022	8/31/2022 11:59:59 PM

id	date	end_of_quarter	end_of_quarter_timestamp
8202	8/2/2022	08/31/2022	8/31/2022 11:59:59 PM
8203	8/8/2022	08/31/2022	8/31/2022 11:59:59 PM
8204	8/19/2022	08/31/2022	8/31/2022 11:59:59 PM
8205	9/26/2022	11/30/2022	11/30/2022 11:59:59 PM
8206	10/14/2022	11/30/2022	11/30/2022 11:59:59 PM
8207	10/29/2022	11/30/2022	11/30/2022 11:59:59 PM

Omdat het `first_month_of_year`-argument van 3 is gebruikt in de `quarterend()`-functie, wordt het begin van het jaar verplaatst van 1 januari naar 1 maart.

Diagram van de `quarterend()`-functie waarbij maart is ingesteld als de eerste maand van het jaar



Transactie 8203 vond plaats op 8 augustus. Omdat het begin van het jaar 1 maart is, liggen de kwartalen van het jaar als volgt maa-mei, jun-aug, sep-nov en dec-feb.

De `quarterend()`-functie identificeert dat de transactie in het kwartaal tussen het begin van juni en augustus plaatsvond en retourneert de laatste milliseconde van dat kwartaal. Dat is 31 augustus om 11:59:59 PM.

### Voorbeeld 4 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die een tijdstempel retourneert voor het eind van het kwartaal waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject in de app.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

Transactions:

Load

\*

Inline

[

id,date,amount

8188,1/7/2022,17.17

8189,1/19/2022,37.23

8190,2/28/2022,88.27

8191,2/5/2022,57.42

8192,3/16/2022,53.80

8193,4/1/2022,82.06

8194,5/7/2022,40.39

8195,5/16/2022,87.21

8196,6/15/2022,95.93

8197,6/26/2022,45.89

8198,7/9/2022,36.23

8199,7/22/2022,25.66

8200,7/23/2022,82.77

8201,7/27/2022,69.98

8202,8/2/2022,76.11

8203,8/8/2022,25.12

8204,8/19/2022,46.23

8205,9/26/2022,84.21

8206,10/14/2022,96.24

8207,10/29/2022,67.67

];

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date

Maak de volgende metingen om de einddatum van het kwartaal waarin een transactie plaatsvindt te berekenen:

- =quarterend(date)
- =timestamp(quarterend(date))

Resultatentabel

id	date	=quarterend(date)	=timestamp(quarterend(date))
8188	1/7/2022	03/31/2022	3/31/2022 11:59:59 PM
8189	1/19/2022	03/31/2022	3/31/2022 11:59:59 PM
8190	2/5/2022	03/31/2022	3/31/2022 11:59:59 PM
8191	2/28/2022	03/31/2022	3/31/2022 11:59:59 PM
8192	3/16/2022	03/31/2022	3/31/2022 11:59:59 PM

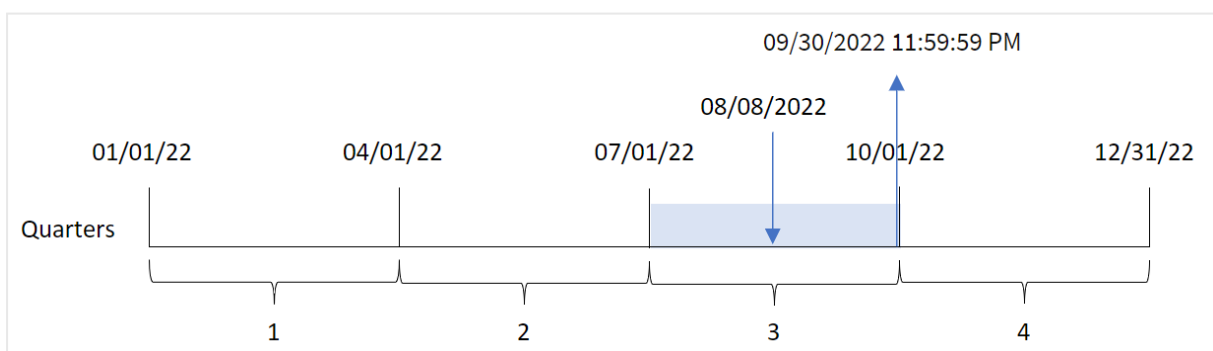
## 5 Script- en diagramfuncties

id	date	=quarterend(date)	=timestamp(quarterend(date))
8193	4/1/2022	06/30/2022	6/30/2022 11:59:59 PM
8194	5/7/2022	06/30/2022	6/30/2022 11:59:59 PM
8195	5/16/2022	06/30/2022	6/30/2022 11:59:59 PM
8196	6/15/2022	06/30/2022	6/30/2022 11:59:59 PM
8197	6/26/2022	06/30/2022	6/30/2022 11:59:59 PM
8198	7/9/2022	09/30/2022	9/30/2022 11:59:59 PM
8199	7/22/2022	09/30/2022	9/30/2022 11:59:59 PM
8200	7/23/2022	09/30/2022	9/30/2022 11:59:59 PM
8201	7/27/2022	09/30/2022	9/30/2022 11:59:59 PM
8202	8/2/2022	09/30/2022	9/30/2022 11:59:59 PM
8203	8/8/2022	09/30/2022	9/30/2022 11:59:59 PM
8204	8/19/2022	09/30/2022	9/30/2022 11:59:59 PM
8205	9/26/2022	09/30/2022	9/30/2022 11:59:59 PM
8206	10/14/2022	12/31/2022	12/31/2022 11:59:59 PM
8207	10/29/2022	12/31/2022	12/31/2022 11:59:59 PM

Het end\_of\_quarter-veld wordt gemaakt in de voorafgaande load-instructie met behulp van de quarterend()-functie en geeft het datumveld door als het argument van de functie.

De quarterend()-functie identificeert aanvankelijk in welk kwartaal de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van dat kwartaal.

*Diagram van de quarterend()-functie waarbij het kwartaaleinde van transactie 8203 is geïdentificeerd*



Transactie 8203 vond plaats op 8 augustus. De quarterend()-functie identificeert dat de transactie in het derde kwartaal plaatsvond en retourneert de laatste milliseconde van dat kwartaal. Dat is 30 september om 11:59:59 PM.



### Voorbeeld 5 - Scenario

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset wordt geladen in een tabel met de naam 'Employee\_Expenses'. De tabel bevat de volgende velden:
  - Werknemers-id's
  - Werknemersnamen
  - De gemiddelde dagelijkse onkostendeclaraties van elke werknemer.

De eindgebruiker wil graag een diagramobject dat per werknemer-id en werknemersnaam de geschatte nog te maken onkostendeclaraties voor de rest van het kwartaal weergeeft. Het boekjaar begint in januari.

#### Load-script

```
Employee_Expenses :
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,Sydney,$27
186,Agatha,$18
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- employee\_id
- employee\_name

Maak de volgende meting om de opgebouwde rente te berekenen:

- $=(\text{quarterend}(\text{today}(1))-\text{today}(1))*\text{avg\_daily\_claim}$

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

employee_id	employee_name	=(quarterend(today(1))-today(1))*avg_daily_claim
182	Mark	\$480.00
183	Deryck	\$400.00
184	Dexter	\$400.00
185	Sydney	\$864.00
186	Agatha	\$576.00

De `quarterend()`-functie gebruikt de datum van vandaag als het enige argument en retourneert de einddatum van de huidige maand. Door vervolgens de datum van vandaag af te trekken van de einddatum van het jaar, retourneert de uitdrukking het aantal resterende dagen in deze maand.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in het resterende kwartaal zal indienen.

### quartername

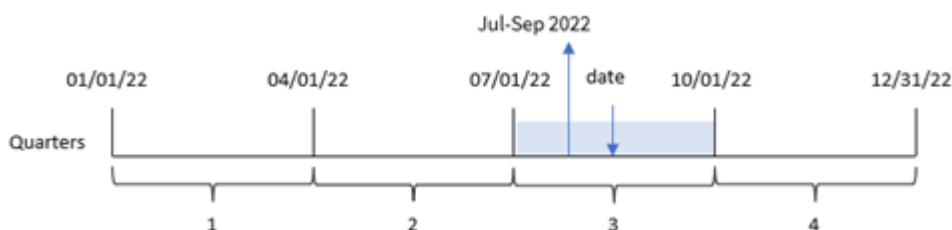
Deze functie retourneert de maanden van het kwartaal (opge maakt volgens de scriptvariabele **MonthNames**) en het jaar met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van het kwartaal.

#### Syntaxis:

```
QuarterName (date[, period_no[, first_month_of_year]])
```

**Retourgegevenstypen:** dual

*Diagram van de quartername()-functie*



De `quartername()`-functie bepaalt welk kwartaal de datum in valt. Hij retourneert vervolgens een waarde die de begin- en eindmaanden van het kwartaal en van het jaar laten zien. De achterliggende numerieke waarde van dit resultaat is de eerste milliseconde van het kwartaal.

### Argumenten

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal, waarbij de waarde 0 het kwartaal aangeeft dat <b>date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande kwartalen aan en positieve waarden geven volgende kwartalen aan.
<b>first_ month_of_ year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> .

### Wanneer gebruiken

De `quartername()`-functie is handig als u aggregaties per kwartaal wilt vergelijken. Bijvoorbeeld als u de totale verkoop van producten per kwartaal wilt zien.

Deze functie kan in het load-script worden gebruikt om een veld in een masterkalendertabel te maken. Hij zou ook rechtstreeks in een diagram kunnen worden gebruikt als een berekende dimensie.

Deze voorbeelden gebruiken de datumopmaak MM/DD/YYYY. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>quartername('10/29/2013')</code>	Retourneert Oct-Dec 2013.
<code>quartername('10/29/2013', -1)</code>	Retourneert Jul-Sep 2013.
<code>quartername('10/29/2013', 0, 3)</code>	Retourneert Sep-Nov 2013.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Datum zonder aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, transaction\_quarter, dat het kwartaal retourneert waarin de transacties plaatsvonden.

Voeg hier uw andere tekst toe, indien nodig, met lijsten enz.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';  
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

```
Transactions:
```

```
  Load  
    *,  
    quartername(date) as transaction_quarter  
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[  
id,date,amount  
8188,1/7/2022,17.17  
8189,1/19/2022,37.23  
8190,2/28/2022,88.27  
8191,2/5/2022,57.42  
8192,3/16/2022,53.80  
8193,4/1/2022,82.06  
8194,5/7/2022,40.39  
8195,5/16/2022,87.21  
8196,6/15/2022,95.93  
8197,6/26/2022,45.89  
8198,7/9/2022,36.23  
8199,7/22/2022,25.66  
8200,7/23/2022,82.77  
8201,7/27/2022,69.98  
8202,8/2/2022,76.11  
8203,8/8/2022,25.12  
8204,8/19/2022,46.23  
8205,9/26/2022,84.21  
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- transaction\_quarter

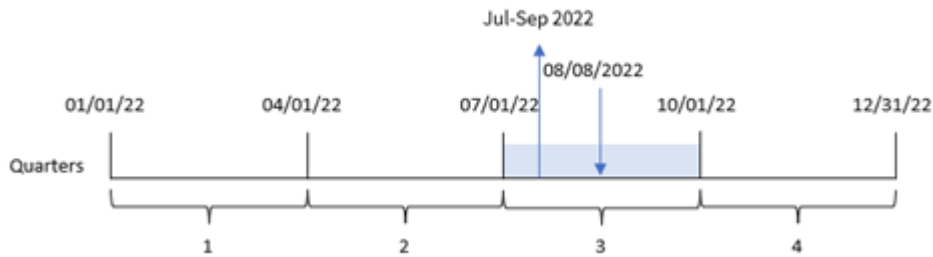
Resultatentabel

date	transaction_quarter
1/7/2022	jan-mrt 2022
1/19/2022	jan-mrt 2022
2/5/2022	jan-mrt 2022
2/28/2022	jan-mrt 2022
3/16/2022	jan-mrt 2022
4/1/2022	april-jun 2022
5/7/2022	april-jun 2022
5/16/2022	april-jun 2022
6/15/2022	april-jun 2022
6/26/2022	april-jun 2022
7/9/2022	juli-sept 2022
7/22/2022	juli-sept 2022
7/23/2022	juli-sept 2022
7/27/2022	juli-sept 2022
8/2/2022	juli-sept 2022
8/8/2022	juli-sept 2022
8/19/2022	juli-sept 2022
9/26/2022	juli-sept 2022
10/14/2022	okt-dec 2022
10/29/2022	okt-dec 2022

Het veld `transaction_quarter` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `quartername()` en geeft het datumveld door als het argument van de functie.

De `quartername()`-functie identificeert eerst het kwartaal waarin de datumwaarde valt. Hij retourneert vervolgens een waarde die de begin- en eindmaanden van het kwartaal en van het jaar laten zien.

Diagram van `quartername()`-functie, voorbeeld zonder aanvullende argumenten



Transactie 8203 vond plaats op 8 augustus 2022. De `quartername()`-functie identificeert dat de transactie plaatsvond in het derde kwartaal en retourneert daarom jul-sep 2022. De maanden worden weergegeven met dezelfde notatie als de `MonthNames`-systeemvariabele.

### Voorbeeld 2 - Datum met `period_no` argument

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het veld `previous_quarter` wordt gemaakt en retourneert het kwartaal dat voor het kwartaal valt waarin de transacties plaatsvonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';  
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

Transactions:

```
Load  
*,  
quartername(date,-1) as previous_quarter  
;
```

Load

\*

Inline

```
[  
id,date,amount  
8188,1/7/2022,17.17  
8189,1/19/2022,37.23  
8190,2/28/2022,88.27  
8191,2/5/2022,57.42  
8192,3/16/2022,53.80  
8193,4/1/2022,82.06  
8194,5/7/2022,40.39  
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_quarter

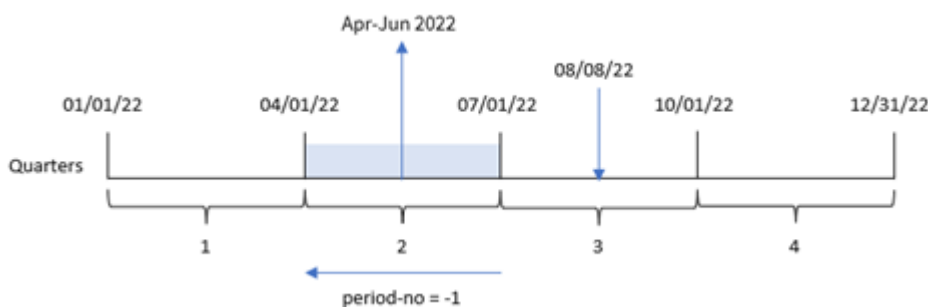
Resultatentabel

date	previous_quarter
1/7/2022	okt-dec 2021
1/19/2022	okt-dec 2021
2/5/2022	okt-dec 2021
2/28/2022	okt-dec 2021
3/16/2022	okt-dec 2021
4/1/2022	jan-mrt 2022
5/7/2022	jan-mrt 2022
5/16/2022	jan-mrt 2022
6/15/2022	jan-mrt 2022
6/26/2022	jan-mrt 2022
7/9/2022	april-jun 2022
7/22/2022	april-jun 2022
7/23/2022	april-jun 2022
7/27/2022	april-jun 2022
8/2/2022	april-jun 2022
8/8/2022	april-jun 2022
8/19/2022	april-jun 2022

date	previous_quarter
9/26/2022	april-jun 2022
10/14/2022	juli-sept 2022
10/29/2022	juli-sept 2022

In dit geval, omdat een `period_no` van -1 is gebruikt als het `offset`-argument in de `quartername()`-functie, identificeert de functie eerst dat de transacties in het derde kwartaal hebben plaatsgevonden. Vervolgens wordt dit naar één kwartaal eerder verschoven en wordt een waarde geretourneerd die de begin- en eindmaanden van het kwartaal en van het jaar laten zien.

*Diagram van `quartername()`-functie, voorbeeld `period_no`*



Transactie 8203 vond plaats op 8 augustus. De `quartername()`-functie identificeert dat het kwartaal vóór het kwartaal waarin de transactie plaatsvond tussen 1 april en 30 juni viel. Daarom wordt `apr-jun 2022` geretourneerd.

### Voorbeeld 3 - Datum met `first_week_day`-argument

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld moeten we echter 1 maart instellen als het begin van het boekjaar.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';

Transactions:
  Load
    *,
    quartername(date,0,3) as transaction_quarter
  ;
Load
```



\*

Inline

[

id,date,amount

8188,1/7/2022,17.17

8189,1/19/2022,37.23

8190,2/28/2022,88.27

8191,2/5/2022,57.42

8192,3/16/2022,53.80

8193,4/1/2022,82.06

8194,5/7/2022,40.39

8195,5/16/2022,87.21

8196,6/15/2022,95.93

8197,6/26/2022,45.89

8198,7/9/2022,36.23

8199,7/22/2022,25.66

8200,7/23/2022,82.77

8201,7/27/2022,69.98

8202,8/2/2022,76.11

8203,8/8/2022,25.12

8204,8/19/2022,46.23

8205,9/26/2022,84.21

8206,10/14/2022,96.24

8207,10/29/2022,67.67

];

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- transaction\_quarter

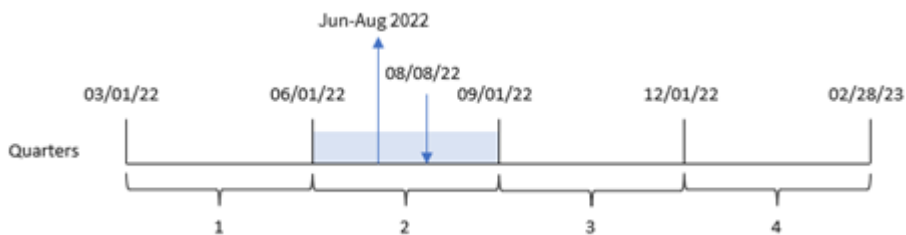
Resultatentabel

date	transaction_quarter
1/7/2022	Dec-feb 2021
1/19/2022	Dec-feb 2021
2/5/2022	Dec-feb 2021
2/28/2022	Dec-feb 2021
3/16/2022	Maa-mei 2022
4/1/2022	Maa-mei 2022
5/7/2022	Maa-mei 2022
5/16/2022	Maa-mei 2022
6/15/2022	Jun-aug 2022
6/26/2022	Jun-aug 2022

date	transaction_quarter
7/9/2022	Jun-aug 2022
7/22/2022	Jun-aug 2022
7/23/2022	Jun-aug 2022
7/27/2022	Jun-aug 2022
8/2/2022	Jun-aug 2022
8/8/2022	Jun-aug 2022
8/19/2022	Jun-aug 2022
9/26/2022	Sep-nov 2022
10/14/2022	Sep-nov 2022
10/29/2022	Sep-nov 2022

In deze instantie wordt het begin van het jaar van 1 januari naar 1 maart verplaatst omdat het `first_month_of_year`-argument van 3 is gebruikt in de `quartername()`-functie. Daarom worden de kwartalen van het jaar verdeeld in maart-mei, juni-augustus, september-november en december-februari.

*Diagram van de `quartername()`-functie, voorbeeld met `first_week_day`*



Transactie 8203 vond plaats op 8 augustus. De `quartername()`-functie identificeert dat de transactie plaatsvond in het tweede kwartaal, tussen het begin van juni en het eind van augustus. Daarom wordt jun-aug 2022 geretourneerd.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het eind van het kwartaal waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

Transactions:

Load

\*

Inline

[

id,date,amount

8188,1/7/2022,17.17

8189,1/19/2022,37.23

8190,2/28/2022,88.27

8191,2/5/2022,57.42

8192,3/16/2022,53.80

8193,4/1/2022,82.06

8194,5/7/2022,40.39

8195,5/16/2022,87.21

8196,6/15/2022,95.93

8197,6/26/2022,45.89

8198,7/9/2022,36.23

8199,7/22/2022,25.66

8200,7/23/2022,82.77

8201,7/27/2022,69.98

8202,8/2/2022,76.11

8203,8/8/2022,25.12

8204,8/19/2022,46.23

8205,9/26/2022,84.21

8206,10/14/2022,96.24

8207,10/29/2022,67.67

];

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Maak de volgende meting:

=quartername(date)

Resultatentabel

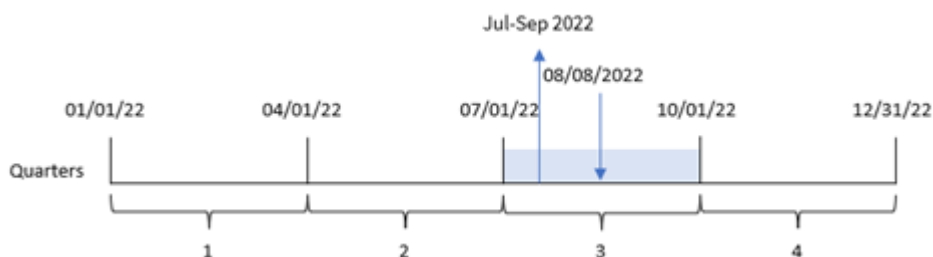
date	=quartername(date)
1/7/2022	jan-mrt 2022
1/19/2022	jan-mrt 2022
2/5/2022	jan-mrt 2022
2/28/2022	jan-mrt 2022
3/16/2022	jan-mrt 2022
4/1/2022	april-jun 2022
5/7/2022	april-jun 2022

date	=quartername(date)
5/16/2022	april-jun 2022
6/15/2022	april-jun 2022
6/26/2022	april-jun 2022
7/9/2022	juli-sept 2022
7/22/2022	juli-sept 2022
7/23/2022	juli-sept 2022
7/27/2022	juli-sept 2022
8/2/2022	juli-sept 2022
8/8/2022	juli-sept 2022
8/19/2022	juli-sept 2022
9/26/2022	juli-sept 2022
10/14/2022	okt-dec 2022
10/29/2022	okt-dec 2022

De meting `transaction_quarter` wordt gemaakt in het diagramobject met behulp van de functie `quartername()` en geeft het veld `date` door als het argument van de functie.

De `quartername()`-functie identificeert eerst het kwartaal waarin de datumwaarde valt. Hij retourneert vervolgens een waarde die de begin- en eindmaanden van het kwartaal en van het jaar laten zien.

*Diagram van `quartername()`-functie, diagramobjectvoorbeeld*



Transactie 8203 vond plaats op 8 augustus 2022. De `quartername()`-functie identificeert dat de transactie plaatsvond in het derde kwartaal en retourneert daarom `jul-sep 2022`. De maanden worden weergegeven met dezelfde notatie als de `MonthNames`-systeemvariabele.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.

De eindgebruiker wil graag een diagramobject dat de totale omzet per kwartaal voor de transacties weergeeft. Dit kan zelfs worden bereikt als deze dimensie niet beschikbaar is in het gegevensmodel, door de functie `quartername()` te gebruiken als een berekende dimensie in het diagram.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'1/7/2022',17.17
```

```
8189,'1/19/2022',37.23
```

```
8190,'2/28/2022',88.27
```

```
8191,'2/5/2022',57.42
```

```
8192,'3/16/2022',53.80
```

```
8193,'4/1/2022',82.06
```

```
8194,'5/7/2022',40.39
```

```
8195,'5/16/2022',87.21
```

```
8196,'6/15/2022',95.93
```

```
8197,'6/26/2022',45.89
```

```
8198,'7/9/2022',36.23
```

```
8199,'7/22/2022',25.66
```

```
8200,'7/23/2022',82.77
```

```
8201,'7/27/2022',69.98
```

```
8202,'8/2/2022',76.11
```

```
8203,'8/8/2022',25.12
```

```
8204,'8/19/2022',46.23
```

```
8205,'9/26/2022',84.21
```

```
8206,'10/14/2022',96.24
```

```
8207,'10/29/2022',67.67
```

```
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een berekende dimensie met behulp van de volgende uitdrukking:  
=quartername(date)
3. Bereken vervolgens de totale verkoop met de volgende aggregatiemeting:  
=sum(amount)
4. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

=quartername(date)	=sum(amount)
juli-sept 2022	\$446.31
april-jun 2022	\$351.48
jan-mrt 2022	\$253.89
okt-dec 2022	\$163.91

### quarterstart

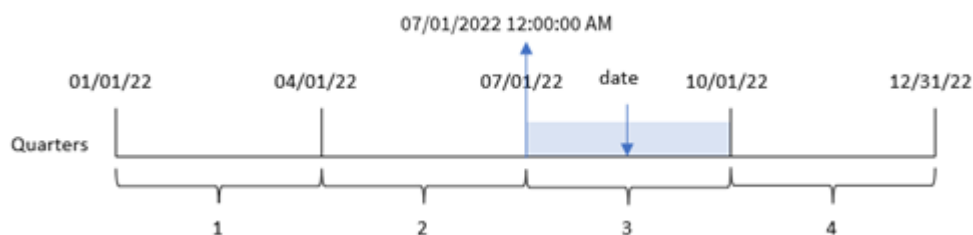
Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van het kwartaal dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

#### Syntaxis:

```
QuarterStart(date[, period_no[, first_month_of_year]])
```

#### Retourgegevenstypen: dual

Diagram van de quarterstart()-functie



De quarterstart()-functie bepaalt in welk kwartaal de datevalt. Hij retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van de laatste maand van dat kwartaal.

### Argumenten

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal, waarbij de waarde 0 het kwartaal aangeeft dat <b>date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorafgaande kwartalen aan en positieve waarden geven volgende kwartalen aan.
<b>first_ month_of_ year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> .

### Wanneer gebruiken

De `quarterstart()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van het kwartaal wordt gebruikt dat al is geweest. Hij kan bijvoorbeeld worden gebruikt als een gebruiker de rente wil berekenen die in een kwartaal tot nu toe is opgebouwd.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>quarterstart('10/29/2005')</code>	Retourneert 10/01/2005.
<code>quarterstart('10/29/2005', -1 )</code>	Retourneert 07/01/2005.
<code>quarterstart('10/29/2005', 0, 3)</code>	Retourneert 09/01/2005.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, start\_of\_quarter, dat een tijdstempel voor het begin van het kwartaal retourneert waarin de transactie plaatsvond.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *
    ,
    quarterstart(date) as start_of_quarter,
    timestamp(quarterstart(date)) as start_of_quarter_timestamp
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- start\_of\_quarter



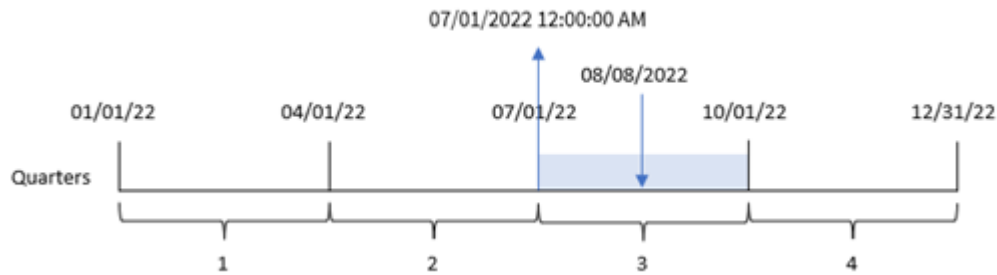
- `start_of_quarter_timestamp`

Resultatentabel

<b>date</b>	<b>start_of_quarter</b>	<b>start_of_quarter_timestamp</b>
1/7/2022	01/01/2022	1/1/2022 12:00:00 AM
1/19/2022	01/01/2022	1/1/2022 12:00:00 AM
2/5/2022	01/01/2022	1/1/2022 12:00:00 AM
2/28/2022	01/01/2022	1/1/2022 12:00:00 AM
3/16/2022	01/01/2022	1/1/2022 12:00:00 AM
4/1/2022	04/01/2022	4/1/2021 12:00:00 AM
5/7/2022	04/01/2022	4/1/2021 12:00:00 AM
5/16/2022	04/01/2022	4/1/2021 12:00:00 AM
6/15/2022	04/01/2022	4/1/2021 12:00:00 AM
6/26/2022	04/01/2022	4/1/2021 12:00:00 AM
7/9/2022	07/01/2022	7/1/2021 12:00:00 AM
7/22/2022	07/01/2022	7/1/2021 12:00:00 AM
7/23/2022	07/01/2022	7/1/2021 12:00:00 AM
7/27/2022	07/01/2022	7/1/2021 12:00:00 AM
8/2/2022	07/01/2022	7/1/2021 12:00:00 AM
8/8/2022	07/01/2022	7/1/2021 12:00:00 AM
8/19/2022	07/01/2022	7/1/2021 12:00:00 AM
9/26/2022	07/01/2022	7/1/2021 12:00:00 AM
10/14/2022	10/01/2022	10/1/2022 12:00:00 AM
10/29/2022	10/01/2022	10/1/2022 12:00:00 AM

Het veld `start_of_quarter` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `quarterstart()` en geeft het datumveld door als het argument van de functie. De `quarterstart()`-functie identificeert eerst het kwartaal waarin de datumwaarde valt. Hij retourneert vervolgens een tijdstempel voor de eerste milliseconde van dat kwartaal.

Diagram van `quarterstart()`-functie, voorbeeld zonder aanvullende argumenten



Transactie 8203 vond plaats op 8 augustus. De `quarterstart()`-functie identificeert dat de transactie in het derde kwartaal plaatsvond en retourneert de eerste milliseconde van dat kwartaal. Dat is 1 juli om 12:00:00 AM.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_quarter_start`, dat de tijdstempel voor het begin van het kwartaal retourneert voordat de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
  *,
  quarterstart(date,-1) as previous_quarter_start,
  timestamp(quarterstart(date,-1)) as previous_quarter_start_timestamp
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
```

```
8195, 5/16/2022, 87.21
8196, 6/15/2022, 95.93
8197, 6/26/2022, 45.89
8198, 7/9/2022, 36.23
8199, 7/22/2022, 25.66
8200, 7/23/2022, 82.77
8201, 7/27/2022, 69.98
8202, 8/2/2022, 76.11
8203, 8/8/2022, 25.12
8204, 8/19/2022, 46.23
8205, 9/26/2022, 84.21
8206, 10/14/2022, 96.24
8207, 10/29/2022, 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_quarter\_start
- previous\_quarter\_start\_timestamp

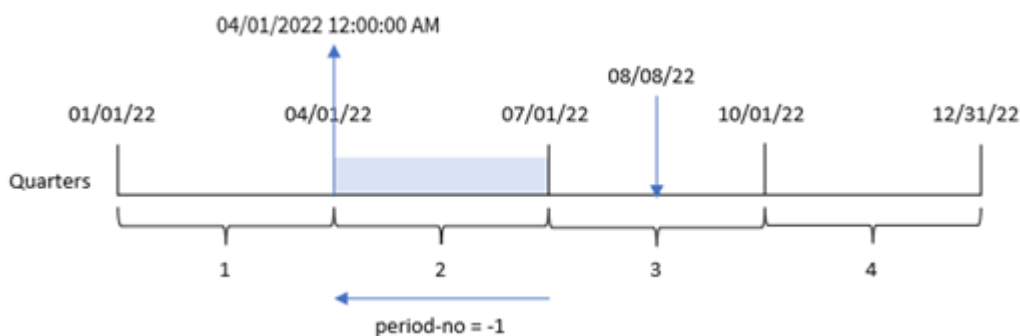
Resultatentabel

date	previous_quarter_start	previous_quarter_start_timestamp
1/7/2022	10/01/2021	10/1/2021 12:00:00 AM
1/19/2022	10/01/2021	10/1/2021 12:00:00 AM
2/5/2022	10/01/2021	10/1/2021 12:00:00 AM
2/28/2022	10/01/2021	10/1/2021 12:00:00 AM
3/16/2022	10/01/2021	10/1/2021 12:00:00 AM
4/1/2022	01/01/2022	1/1/2022 12:00:00 AM
5/7/2022	01/01/2022	1/1/2022 12:00:00 AM
5/16/2022	01/01/2022	1/1/2022 12:00:00 AM
6/15/2022	01/01/2022	1/1/2022 12:00:00 AM
6/26/2022	01/01/2022	1/1/2022 12:00:00 AM
7/9/2022	04/01/2022	4/1/2021 12:00:00 AM
7/22/2022	04/01/2022	4/1/2021 12:00:00 AM
7/23/2022	04/01/2022	4/1/2021 12:00:00 AM
7/27/2022	04/01/2022	4/1/2021 12:00:00 AM
8/2/2022	04/01/2022	4/1/2021 12:00:00 AM

date	previous_quarter_start	previous_quarter_start_timestamp
8/8/2022	04/01/2022	4/1/2021 12:00:00 AM
8/19/2022	04/01/2022	4/1/2021 12:00:00 AM
9/26/2022	04/01/2022	4/1/2021 12:00:00 AM
10/14/2022	07/01/2022	7/1/2022 12:00:00 AM
10/29/2022	07/01/2022	7/1/2022 12:00:00 AM

In dit geval, omdat een `period_no` van -1 werd gebruikt als het `offset`-argument in de `quarterstart()`-functie, identificeert de functie eerst het kwartaal waarin de transacties plaatsvinden. Het verschuift vervolgens naar een kwartaal eerder en identificeert de eerste milliseconde van dat kwartaal.

Diagram van `quarterstart()`-functie, voorbeeld `period_no`



Transactie 8203 vond plaats op 8 augustus. De `quarterstart()`-functie identificeert dat het kwartaal vóór het kwartaal waarin de transactie plaatsvond tussen 1 april en 30 juni viel. Vervolgens wordt de eerste milliseconde van dat kwartaal geretourneerd: 1 april om 12:00:00 AM.

### Voorbeeld 3 - first\_month\_of\_year

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld moeten we echter 1 maart instellen als het begin van het boekjaar.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*,
quarterstart(date,0,3) as start_of_quarter,
```

```
timestamp(quarterstart(date,0,3)) as start_of_quarter_timestamp
;
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- start\_of\_quarter
- start\_of\_quarter\_timestamp

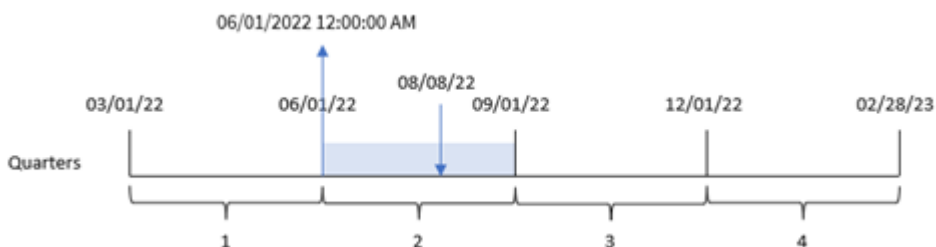
Resultatentabel

date	start_of_quarter	start_of_quarter_timestamp
1/7/2022	12/01/2021	12/1/2021 12:00:00 AM
1/19/2022	12/01/2021	12/1/2021 12:00:00 AM
2/5/2022	12/01/2021	12/1/2021 12:00:00 AM
2/28/2022	12/01/2021	12/1/2021 12:00:00 AM
3/16/2022	03/01/2022	3/1/2022 12:00:00 AM
4/1/2022	03/01/2022	3/1/2022 12:00:00 AM
5/7/2022	03/01/2022	3/1/2022 12:00:00 AM

date	start_of_quarter	start_of_quarter_timestamp
5/16/2022	03/01/2022	3/1/2022 12:00:00 AM
6/15/2022	06/01/2022	6/1/2022 12:00:00 AM
6/26/2022	06/01/2022	6/1/2022 12:00:00 AM
7/9/2022	06/01/2022	6/1/2022 12:00:00 AM
7/22/2022	06/01/2022	6/1/2022 12:00:00 AM
7/23/2022	06/01/2022	6/1/2022 12:00:00 AM
7/27/2022	06/01/2022	6/1/2022 12:00:00 AM
8/2/2022	06/01/2022	6/1/2022 12:00:00 AM
8/8/2022	06/01/2022	6/1/2022 12:00:00 AM
8/19/2022	06/01/2022	6/1/2022 12:00:00 AM
9/26/2022	09/01/2022	9/1/2022 12:00:00 AM
10/14/2022	09/01/2022	9/1/2022 12:00:00 AM
10/29/2022	09/01/2022	9/1/2022 12:00:00 AM

Omdat in dit geval het `first_month_of_year`-argument van 3 is gebruikt in de `quarterstart()`-functie, wordt het begin van het jaar verplaatst van 1 januari naar 1 maart.

Diagram van de `quarterstart()`-functie, voorbeeld `first_month_of_year`



Transactie 8203 vond plaats op 8 augustus. Omdat het begin van het jaar 1 maart is, vallen de kwartalen van het jaar tussen maart-mei, juni-augustus, september-november en december-februari. De `quarterstart()`-functie identificeert dat de transactie in het kwartaal tussen het begin van juni en augustus plaatsvond en retourneert de eerste milliseconde van dat kwartaal. Dat is 1 juni om 12:00:00 AM.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het eind van het kwartaal waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de toepassing.

### Load-script

```
Transactions:
Load
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Voeg de volgende metingen toe:

- =quarterstart(date)
- =timestamp(quarterstart(date))

Resultatentabel

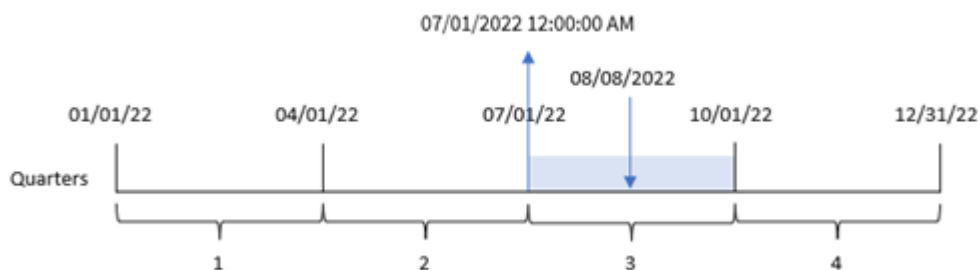
date	=quarterstart(date)	=timestamp(quarterstart(date))
10/14/2022	10/01/2022	10/1/2022 12:00:00 AM
10/29/2022	10/01/2022	10/1/2022 12:00:00 AM
7/9/2022	07/01/2022	7/1/2022 12:00:00 AM

date	=quarterstart(date)	=timestamp(quarterstart(date))
7/22/2022	07/01/2022	7/1/2022 12:00:00 AM
7/23/2022	07/01/2022	7/1/2022 12:00:00 AM
7/27/2022	07/01/2022	7/1/2022 12:00:00 AM
8/2/2022	07/01/2022	7/1/2022 12:00:00 AM
8/8/2022	07/01/2022	7/1/2022 12:00:00 AM
8/19/2022	07/01/2022	7/1/2022 12:00:00 AM
9/26/2022	07/01/2022	7/1/2022 12:00:00 AM
4/1/2022	04/01/2022	4/1/2022 12:00:00 AM
5/7/2022	04/01/2022	4/1/2022 12:00:00 AM
5/16/2022	04/01/2022	4/1/2022 12:00:00 AM
6/15/2022	04/01/2022	4/1/2022 12:00:00 AM
6/26/2022	04/01/2022	4/1/2022 12:00:00 AM
1/7/2022	01/01/2022	1/1/2022 12:00:00 AM
1/19/2022	01/01/2022	1/1/2022 12:00:00 AM
2/5/2022	01/01/2022	1/1/2022 12:00:00 AM
2/28/2022	01/01/2022	1/1/2022 12:00:00 AM
3/16/2022	01/01/2022	1/1/2022 12:00:00 AM

De meting `start_of_quarter` wordt gemaakt in het diagramobject met behulp van de functie `quarterstart()` en geeft het veld `date` door als het argument van de functie.

De `quarterstart()`-functie identificeert in welke kwartaal de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van dat kwartaal.

*Diagram van `quarterstart()`-functie, diagramobjectvoorbeeld*



Transactie 8203 vond plaats op 8 augustus. De `quarterstart()`-functie identificeert dat de transactie in het derde kwartaal plaatsvond en retourneert de eerste milliseconde van dat kwartaal. De geretourneerde waarde is 1 juli om 12:00:00 AM.



### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set leningsaldo's die wordt geladen in de tabel Loans.
- Gegevens bestaan uit lening-id's, het saldo aan het begin van het kwartaal en de enkelvoudige rente die op iedere lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd op elke lening in het kwartaal tot heden.

#### Load-script

```
Loans:
Load
*
Inline
[
loan_id,start_balance,rate
8188,$10000.00,0.024
8189,$15000.00,0.057
8190,$17500.00,0.024
8191,$21000.00,0.034
8192,$90000.00,0.084
];
```

#### Resultaten

##### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
  - loan\_id
  - start\_balance
2. Maak nu de volgende meting om de opgebouwde rente te berekenen:  
 $=start\_balance*(rate*(today(1)-quarterstart(today(1)))/365)$
3. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

loan_id	start_balance	$=start\_balance*(rate*(today(1)-quarterstart(today(1)))/365)$
8188	\$10000.00	\$15.07

loan_id	start_balance	=start_balance*(rate*(today(1)-quarterstart(today(1)))/365)
8189	\$15000.00	\$128.84
8190	\$17500.00	\$63.29
8191	\$21000.00	\$107.59
8192	\$90000.00	\$1139.18

De functie `quarterstart()` gebruikt de datum van vandaag als enige argument en retourneert de begindatum van het huidige jaar. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking het aantal dagen dat in dit kwartaal tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor deze periode te retourneren. Het resultaat wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die dit kwartaal tot nu toe is opgebouwd.

### second

Deze functie retourneert een geheel getal dat de seconde representeert als de breuk van de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

#### Syntaxis:

```
second (expression)
```

**Retourgegevenstypen:** geheel getal

### Wanneer gebruiken

De `second()`-functie is handig als u aggregaties per seconde wilt vergelijken. U kunt de functie bijvoorbeeld gebruiken als u de verdeling van het aantal activiteiten per seconde wilt bekijken.

Deze dimensies kunnen worden gemaakt in het load-script door de functie te gebruiken om een veld te maken in een hoofdagendatabel, of door de dimensie rechtstreeks in een diagram te gebruiken als een berekende dimensie.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>second( '09:14:36' )</code>	retourneert 36
<code>second( '0.5555' )</code>	retourneert 55 ( omdat $0,5555 = 13:19:55$ )

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege

landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Variabele

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensverzameling met transacties per tijdstempel die wordt geladen in de tabel Transactions.
- De standaard Timestamp-systeemvariabele (M/D/YYYY h:mm:ss[.fff] TT) wordt gebruikt.
- Maak een veld, second dat berekent wanneer de aankopen hebben plaatsgevonden.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
  Load
    *,
    second(date) as second
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
9497,'01/05/2022 7:04:57 PM',47.25
9498,'01/03/2022 2:21:53 PM',51.75
9499,'01/03/2022 5:40:49 AM',73.53
9500,'01/04/2022 6:49:38 PM',15.35
9501,'01/01/2022 10:10:22 PM',31.43
9502,'01/05/2022 7:34:46 PM',13.24
9503,'01/06/2022 10:58:34 PM',74.34
9504,'01/06/2022 11:29:38 AM',50.00
9505,'01/02/2022 8:35:54 AM',36.34
9506,'01/06/2022 8:49:09 AM',74.23
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- second

Resultatentabel

date	second
01/01/2022 10:10:22 PM	22
01/02/2022 8:35:54 AM	54
01/03/2022 5:40:49 AM	49
01/03/2022 2:21:53 PM	53
01/04/2022 6:49:38 PM	38
01/05/2022 7:04:57 PM	57
01/05/2022 7:34:46 PM	46
01/06/2022 8:49:09 AM	9
01/06/2022 11:29:38 AM	38
01/06/2022 10:58:34 PM	34

De waarden in het second-veld zijn gemaakt met de second()-functie en geven de datum door als de uitdrukking in het voorafgaande load-instructie.

### Voorbeeld 2 - Diagramobject

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De second-waarden worden berekend via een meting in een diagramobject.

#### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[  
id,date,amount  
9497,'01/05/2022 7:04:57 PM',47.25  
9498,'01/03/2022 2:21:53 PM',51.75  
9499,'01/03/2022 5:40:49 AM',73.53  
9500,'01/04/2022 6:49:38 PM',15.35  
9501,'01/01/2022 10:10:22 PM',31.43  
9502,'01/05/2022 7:34:46 PM',13.24  
9503,'01/06/2022 10:58:34 PM',74.34  
9504,'01/06/2022 11:29:38 AM',50.00  
9505,'01/02/2022 8:35:54 AM',36.34  
9506,'01/06/2022 8:49:09 AM',74.23  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:date.

Maak de volgende meting:

```
=second(date)
```

Resultatentabel

date	=second(date)
01/01/2022 10:10:22 PM	22
01/02/2022 8:35:54 AM	54
01/03/2022 5:40:49 AM	49
01/03/2022 2:21:53 PM	53
01/04/2022 6:49:38 PM	38
01/05/2022 7:04:57 PM	57
01/05/2022 7:34:46 PM	46
01/06/2022 8:49:09 AM	9
01/06/2022 11:29:38 AM	38
01/06/2022 10:58:34 PM	34

De waarden voor `second` worden gemaakt met behulp van de `second()`-functie en geven de datum door als de uitdrukking in een meting voor het diagramobject.

### Voorbeeld 3 - Scenario

Load-script en diagramuitdrukkingen

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset van tijdstempels, die wordt gegenereerd om het verkeer naar de ticketverkoopsite van een specifiek festival te vertegenwoordigen. Deze tijdstempels en een bijbehorende id worden in de tabel `web_Traffic` geladen.
- De `TimeStamp`-systeemvariabele `M/D/YYYY h:mm:ss[.fff] TT` wordt gebruikt.

In dit scenario zijn er 10.000 tickets verkocht die om 9.00 uur op 20 mei 2021 in de verkoop zijn gegaan. Eén minuut later waren de tickets uitverkocht.

De gebruiker wil een diagramobject dat per seconde het aantal bezoeken aan de website laat zien.

### Load-script

```
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';

tmpTimeStampCreator:
load
    makedate(2022,05,20) as date
AutoGenerate 1;

join load
    maketime(9+floor(rand()*2),0,floor(rand()*59)) as time
autogenerate 10000;

web_Traffic:
load
    recno() as id,
    timestamp(date + time) as timestamp
resident tmpTimeStampCreator;

drop table tmpTimeStampCreator;
```

### Resultaten

Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak nu een berekende dimensie met behulp van de volgende uitdrukking:  
`=second(timestamp)`
3. Maak een aggregatiemeting om het totaal aantal invoeren te berekenen:  
`=count(id)`

De resultatentabel ziet er vergelijkbaar uit met de onderstaande tabel, maar dan met andere waarden voor de aggregatiemeting:

<code>second(timestamp)</code>	<code>=count(id)</code>
0	150

<code>second(timestamp)</code>	<code>=count(id)</code>
1	184
2	163
3	178
4	179
5	158
6	177
7	169
8	149
9	186
10	169
11	179
12	186
13	182
14	180
15	153
16	191
17	203
18	158
19	159
20	163
Nog 39 rijen	

### setdateyear

Deze functie neemt een **timestamp** en een **year** als invoer en werkt de **timestamp** bij met het **year** dat is opgegeven in de invoer.

#### Syntaxis:

```
setdateyear (timestamp, year)
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

Argument	Beschrijving
<b>timestamp</b>	Een standaardtijdstempel in Qlik Sense (vaak gewoon een datum).
<b>year</b>	Een jaartal in vier cijfers.

Voorbeelden en resultaten:

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

Scriptvoorbeelden

Voorbeeld	Resultaat
<code>setdateyear ( '29/10/2005' , 2013)</code>	Retourneert '29/10/2013'
<code>setdateyear ( '29/10/2005 04:26:14' , 2013)</code>	Retourneert '29/10/2013 04:26:14' Als u het tijdsgedeelte van de tijdstempel wilt bekijken in een visualisatie, moet u de getalopmaak instellen op Date en een waarde kiezen voor Formatting die tijdwaarden weergeeft.

**Voorbeeld:**

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
SetYear:  
Load *,  
SetDateYear(testdates, 2013) as NewYear  
Inline [  
testdates  
1/11/2012  
10/12/2012  
1/5/2013  
2/1/2013  
19/5/2013  
15/9/2013  
11/12/2013  
2/3/2014  
14/5/2014  
13/6/2014  
7/7/2014  
4/8/2014  
];
```



De resulterende tabel bevat de oorspronkelijke datums en een kolom waarin het jaar is ingesteld op 2013.

Resultatentabel

testdates	NewYear
1/11/2012	1/11/2013
10/12/2012	10/12/2013
2/1/2012	2/1/2013
1/5/2013	1/5/2013
19/5/2013	19/5/2013
15/9/2013	15/9/2013
11/12/2013	11/12/2013
2/3/2014	2/3/2013
14/5/2014	14/5/2013
13/6/2014	13/6/2013
7/7/2014	7/7/2013
4/8/2014	4/8/2013

### setdateyearmonth

Deze functie neemt een **timestamp**, een **month** en een **year** als invoer en werkt de **timestamp** bij met het **year** en de **month** die zijn opgegeven in de invoer. .

#### Syntaxis:

```
SetDateYearMonth (timestamp, year, month)
```

Retourgegevenstypen: dual

#### Argumenten:

Argumenten

Argument	Beschrijving
<b>timestamp</b>	Een standaardtijdstempel in Qlik Sense (vaak gewoon een datum).
<b>year</b>	Een jaartal in vier cijfers.
<b>month</b>	Een maand in één of twee cijfers.

#### Voorbeelden en resultaten:

Deze voorbeelden gebruiken de datumopmaak **DD/MM/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan uw script voor het laden van gegevens. Wijzig de opmaak in de voorbeelden zodat deze in uw behoeften voorziet.

### Scriptvoorbeelden

Voorbeeld	Resultaat
<pre>setdateyearmonth ('29/10/2005', 2013, 3)</pre>	Retourneert '29/03/2013'
<pre>setdateyearmonth ('29/10/2005 04:26:14', 2013, 3)</pre>	Retourneert '29/03/2013 04:26:14' Als u het tijdsgedeelte van de tijdstempel wilt bekijken in een visualisatie, moet u de getalopmaak instellen op Date en een waarde kiezen voor Formatting die tijdwaarden weergeeft.

#### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
SetYearMonth:
Load *,
SetDateYearMonth(testdates, 2013,3) as NewYearMonth
Inline [
testdates
1/11/2012
10/12/2012
2/1/2013
19/5/2013
15/9/2013
11/12/2013
14/5/2014
13/6/2014
7/7/2014
4/8/2014
];
```

De resulterende tabel bevat de oorspronkelijke datums en een kolom waarin het jaar is ingesteld op 2013.

Resultatentabel

testdates	NewYearMonth
1/11/2012	1/3/2013
10/12/2012	10/3/2013
2/1/2012	2/3/2013
19/5/2013	19/3/2013
15/9/2013	15/3/2013
11/12/2013	11/3/2013
14/5/2014	14/3/2013
13/6/2014	13/3/2013

testdates	NewYearMonth
7/7/2014	7/3/2013
4/8/2014	4/3/2013

## timezone

De functie retourneert de tijdzone zoals gedefinieerd op de computer waarop de Qlik-engine wordt uitgevoerd.

### Syntaxis:

```
TimeZone ( )
```

**Retourgegevenstypen:** dual

### Voorbeeld:

```
timezone( )
```

Als u verschillende tijdzones in een meting in uw app wilt zien, moet u de functie `localtime()` in een meting gebruiken.

## today

Deze functie retourneert de huidige datum. Deze functie retourneert waarden in de indeling van de systeemvariabele `DateFormat`.

### Syntaxis:


```
today ([ timer_mode ])
```

**Retourgegevenstypen:** dual

De `today()`-functie kan ofwel in het load-script ofwel in diagramobjecten worden gebruikt.

De standaardwaarde voor `timer_mode` is 1.

### Argumenten

Argument	Beschrijving
timer_mode	<p>Kan de volgende waarden hebben:</p> <ul style="list-style-type: none"> <li>0 (dag van de als laatste voltooide laadbewerking voor gegevens)</li> <li>1 (dag van de functieaanroep)</li> <li>2 (dag wanneer de app werd geopend)</li> </ul> <div style="border: 1px solid gray; padding: 10px; margin-top: 10px;"> <p> Als u de functie in een load-script gebruikt, resulteert <b>timer_mode=0</b> in de dag van de als laatste voltooide laadbewerking voor gegevens, terwijl <b>timer_mode=1</b> de dag van de functieaanroep in de huidige laadbewerking voor gegevens oplevert.</p> </div>

Voorbeelden van functies

timer_ mode- waarde	Resultaat bij gebruik in load-script	Resultaat bij gebruik in diagramobject
0	Retourneert een datum met de notatie van de systeemvariabele <code>DateFormat</code> , van de laatste keer dat de gegevens opnieuw zijn geladen voorafgaand aan de laatste keer dat de gegevens opnieuw zijn geladen.	Retourneert een datum met de notatie van de systeemvariabele <code>DateFormat</code> voor de laatste keer dat de gegevens opnieuw zijn geladen.
1	Retourneert een datum met de notatie van de systeemvariabele <code>DateFormat</code> voor de laatste keer dat de gegevens opnieuw zijn geladen.	Retourneert een datum met de notatie van de systeemvariabele <code>DateFormat</code> van de functieaanroep.
2	Retourneert een datum met de notatie van de systeemvariabele <code>DateFormat</code> voor het begin van de sessie van de gebruiker in de applicatie. Dit wordt niet bijgewerkt tenzij de gebruiker het script opnieuw laadt.	Retourneert de datum met de notatie van de systeemvariabele <code>DateFormat</code> voor het begin van de sessie van de gebruiker in de applicatie. Dit wordt vernieuwd zodra een nieuwe sessie begint of de gegevens weer in de applicatie zijn geladen.

### Wanneer gebruiken

De `today()`-functie wordt meestal gebruikt als een component in een uitdrukking. Hij kan bijvoorbeeld worden gebruikt om de rente te berekenen die in een maand tot nu toe is opgebouwd.

De volgende tabel geeft een uitleg van het resultaat dat door de `today()`-functie is geretourneerd en geeft verschillende waarden voor het `timer_mode`-argument:

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Genereren van objecten met behulp van het load-script

Load-script en resultaten

#### Overzicht

Het volgende voorbeeld creëert die variabelen met behulp van de `today()`-functie. Elke variabele gebruikt een van de `timer_mode`-opties om het effect van de variabele te demonstreren.

Laad het script opnieuw en herlaad het na 24 nog een keer zodat de variabelen hun doel kunnen laten zien. De variabelen `today(0)` en `today(1)` zullen nu verschillende waarden laten zien en hun doel demonstreren.

#### Load-script

```
LET vPreviousDataLoad = today(0);  
LET vCurrentDataLoad = today(1);  
LET vApplicationOpened = today(2);
```

#### Resultaten

Nadat de gegevens voor de tweede keer zijn geladen, maakt u drie tekstvakken door de onderstaande instructies te volgen.

Maak eerst een tekstvak voor de gegevens die eerder zijn geladen.

#### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:  
`=vPreviousDataLoad`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'Previous Reload Time' toe aan het object.

Maak vervolgens een tekstvak voor de gegevens die momenteel worden geladen.

#### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:  
`=vCurrentDataLoad`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'Current Reload Time' toe aan het object.

Maak een definitief tekstvak dat wordt getoond als de sessie van de gebruiker in de toepassing wordt gestart.

### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe aan het object:  
`=vApplicationOpened`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel 'User Session Started' toe aan het object.

Diagram van variabelen die zijn gemaakt met de `today()`-functie in het load-script

<b>Previous Reload Time</b> 06/22/2022	<b>Current Reload Time</b> 06/23/2022	<b>User Session Began</b> 06/23/2022
---	--	---

De bovenstaande afbeelding geeft voorbeeldwaarden weer voor elk van de gemaakte variabelen. De waarden zouden bijvoorbeeld als volgt kunnen zijn:

- Vorige herlaadtijd: 06/22/2022
- Huidige herlaadtijd: 06/23/2022
- Start gebruikerssessie: 06/23/2022

### Voorbeeld 2 - Genereren van objecten zonder het load-script

Load-script en diagramuitdrukking

#### Overzicht

Het volgende voorbeeld creëert die variabelen met behulp van de `today()`-functie. Elk diagramobject gebruikt een van de `timer_mode`-opties om het effect te demonstreren.

Er is geen load-script voor dit voorbeeld.

#### Resultaten

Maak drie tekstvakken zodra de gegevens voor een tweede keer zijn geladen.

Maak eerst een tekstvak voor de laatste keer dat de gegevens opnieuw zijn geladen.

### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:  
`=today()`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel Laatste herlaadtijd gegevens toe aan het object.

Maak nu een tekstvak om de actuele tijd te tonen.

### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:  
`=today(1)`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel Actuele tijd toe aan het object.

Maak een definitief tekstvak dat wordt getoond als de sessie van de gebruiker in de toepassing wordt gestart.

### Doe het volgende:

1. Gebruik het diagramobject **Tekst en afbeelding** om een tekstvak te maken.
2. Voeg de volgende meting toe:  
`=today(2)`
3. Onder **Uiterlijk** selecteert u **Show titles** en voegt u de titel Start gebruikerssessie toe aan het object.

*Diagram van objecten die zijn gemaakt met de `today()`-functie zonder het load-script*

<b>Latest Data Reload</b> 06/23/2022	<b>Current Time</b> 06/23/2022	<b>User Session Began</b> 06/23/2022
---	-----------------------------------	---

De bovenstaande afbeelding geeft voorbeeldwaarden weer voor elk van de gemaakte objecten. De waarden zouden bijvoorbeeld als volgt kunnen zijn:

- Laatste herlaadtijd gegevens: 06/23/2022
- Actuele tijd: 06/23/2022
- Start gebruikerssessie: 06/23/2022

Het diagramobject Laatste herlaadtijd gegevens maakt gebruik van een `timer_mode`-waarde van 0. Dit retourneert de tijdstempel voor de laatste keer dat de gegevens opnieuw zijn geladen.

Het diagramobject Actuele tijd gebruikt een `timer_mode`-waarde van 1. Dit retourneert de actuele tijd volgens de systeemklok. Als het werkblad of object wordt vernieuwd, wordt deze waarde bijgewerkt.

Het diagramobject Start gebruikerssessie gebruikt een `timer_mode`-waarde van 2. Dit retourneert de tijdstempel voor het moment waarop de applicatie werd geopend en waarop de gebruikerssessie is begonnen.

### Voorbeeld 3 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set leningsaldo's die wordt geladen in de tabel Loans.
- Tabelgegevens met velden voor lening-id, saldo aan het begin van de maand en de enkelvoudige rente die op elke lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd op elke lening in de maand tot heden. De applicatie wordt slechts eens per week opnieuw geladen, maar de gebruiker wil graag dat de resultaten worden vernieuwd zodra het object of de applicatie wordt vernieuwd.

#### Load-script

```
Loans:
Load
*
Inline
[
loan_id,start_balance,rate
8188,$10000.00,0.024
8189,$15000.00,0.057
8190,$17500.00,0.024
8191,$21000.00,0.034
8192,$90000.00,0.084
];
```

#### Resultaten

Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Voeg de volgende velden als dimensies toe:
  - loan\_id
  - start\_balance
3. Maak nu een meting om de opgebouwde rente te berekenen:  
 $=start\_balance*(rate*(today(1)-monthstart(today(1)))/365)$
4. Stel de **Getalnotatie** van de meting in op **Geld**.



Resultatentabel

loan_id	start_balance	=start_balance*(rate*(today(1)-monthstart(today(1)))/365)
8188	\$10000.00	\$16.44
8189	\$15000.00	\$58.56
8190	\$17500.00	\$28.77
8191	\$21000.00	\$48.90
8192	\$90000.00	\$517.81

De functie `monthstart()` gebruikt de functie `today()` om de datum van vandaag als enige argument te retourneren en retourneert de begindatum van de huidige maand. Door dat resultaat af te trekken van de huidige datum en de `today()`-functie opnieuw te gebruiken, retourneert de uitdrukking het aantal dagen dat in deze maand tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor deze periode te retourneren. Het resultaat wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die deze maand tot nu toe is opgebouwd.

Omdat de waarde 1 is gebruikt als het `timer_mode`-argument in de `today()`-functies in de uitdrukking, wordt elke keer dat het diagramobject wordt vernieuwd (door de applicatie te openen, de pagina te vernieuwen, tussen werkbladen te wisselen, enzovoort) de huidige datum geretourneerd en worden de resultaten overeenkomstig vernieuwd.

### UTC

Retourneert de huidige Coordinated Universal Time.

#### Syntaxis:

```
UTC ( )
```

**Retourgegevenstypen:** dual

#### Voorbeeld:

```
utc( )
```

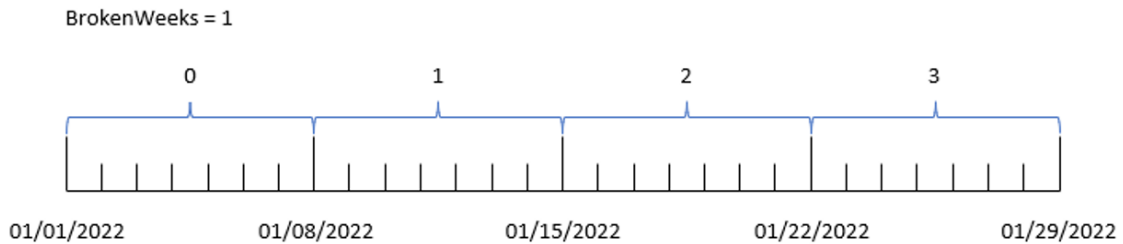
### week

Deze functie retourneert een geheel getal dat het weeknummer representeert conform ISO 8601. Het weeknummer wordt berekend volgens de datuminterpretatie van de uitdrukking, in overeenstemming met de standaardinterpretatie van getallen.

#### Syntaxis:

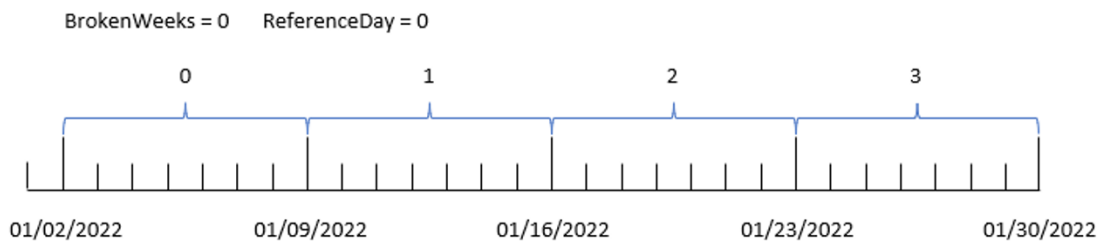
```
week (timestamp [, first_week_day [, broken_weeks [, reference_day]])
```

Voorbeelddiagram van de `week()`-functie met opgesplitste weken



De weeknummering begint op 1 januari (omdat Qlik Sense standaard is ingesteld op het gebruik van opgesplitste weken). De eerste week eindigt op de dag vóór de systeemvariabele `FirstweekDay`, ongeacht hoeveel dagen er in die week zaten. De systeemvariabele `FirstweekDay` kan in de `week()`-functie worden vervangen door het `first_week_day`-argument.

Voorbeelddiagram van de `week()`-functie met niet-opgesplitste weken en `ReferenceDay=0`



De `week()`-functie biedt tevens de mogelijkheid om via het `broken_weeks`-argument aan te geven of er al dan niet opgesplitste weken moeten worden gebruikt. Als de opgesplitste week-functie wordt toegepast, moet week 1 een bepaald aantal dagen in januari bevatten, zoals gedefinieerd door de systeemvariabele `referenceDay`. Daarom kan week 1 in december beginnen week 52 of 53 kan doorlopen tot in januari. Als laatste: het `reference_day`-argument stelt de functie in staat de systeemvariabele `referenceDay` te overschrijven.

Anders dan de `weekname()`-functie, retourneert de `week()`-functie niet ook nog de jaarwaarde. Dit maakt aggregaties mogelijk om weken in verschillende jaren te vergelijken.

Er zijn vier argumenten die in deze functie kunnen worden gebruikt.

### Argument 1: tijdstempel

Dit is de datum die moet worden geëvalueerd als tijdstempel of uitdrukking die wordt herleid tot een tijdstempel, bijvoorbeeld 2012-10-12.

### Argument 2: `first_week_day`

Als u niet `first_week_day` opgeeft, wordt de waarde van de variabele `FirstWeekDay` gebruikt als eerste dag van de week.

Als u een andere dag als eerste dag van de week wilt gebruiken, stelt u `first_week_day` in op:

- 0 voor maandag
- 1 voor dinsdag
- 2 voor woensdag
- 3 voor donderdag
- 4 voor vrijdag
- 5 voor zaterdag
- 6 voor zondag

Het gehele getal dat door de functie wordt geretourneerd gebruikt nu de eerste dag van de week die u hebt ingesteld met **first\_week\_day**.

### Argument 3: broken\_weeks

Als u **broken\_weeks** (gebroken weken) niet specificeert, dan wordt de waarde van variabele **BrokenWeeks** gebruikt om te definiëren of weken al dan niet gebroken zijn.

Standaard wordt in Qlik Sense-functies gebruikgemaakt van ongebroken weken. Dit betekent dat:

- In sommige jaren, week 1 begint in december en in andere jaren week 52 of 53 doorloopt tot in januari.
- Week 1 altijd ten minste 4 dagen heeft in januari.

Het alternatief is om gebroken weken te gebruiken.

- Week 52 of 53 lopen niet door tot in januari.
- Week 1 begint op 1 januari en is, in de meeste gevallen, geen volle week.

De volgende waarden kunnen worden gebruikt:

- 0 (= ongebroken weken gebruiken)
- 1 (= gebroken weken gebruiken)

### Argument 4: reference\_day

Als u **reference\_day** (referentiedag) niet specificeert, dan wordt de waarde van variabele **ReferenceDay** gebruikt om te definiëren welke dag in januari wordt ingesteld als referentiedag om week 1 te definiëren. Standaard wordt in Qlik Sense-functies 4 gebruikt als de referentiedag. Dit betekent dat week 1 4 januari moet bevatten of, anders gezegd, dat week 1 altijd ten minste 4 dagen in januari moet hebben.

De volgende waarden kunnen worden gebruikt voor het instellen van een andere referentiedag:

- 1 (= 1 januari)
- 2 (= 2 januari)
- 3 (= 3 januari)
- 4 (= 4 januari)
- 5 (= 5 januari)

- 6 (= 6 januari)
- 7 (= 7 januari)

### Wanneer gebruiken

De `week()`-functie is handig als u aggregaties per week wilt vergelijken. Hij kan bijvoorbeeld worden gebruikt om de totale verkoop van producten per week te bekijken. De `week()`-functie wordt verkozen boven `weekname()` wanneer de gebruiker wil dat de berekening niet per se de systeemvariabelen `BrokenWeeks`, `FirstWeekDay` of `ReferenceDay` gebruikt.

Daarnaast wordt de `week()`-functie gekozen als u een vergelijking met meerdere jaren wilt maken. Met de `week()`-functie kan de gebruiker een eigen combinatie van deze variabelen maken voor gebruik in instanties waarin de functie wordt gebruikt.

Deze dimensies kunnen worden gemaakt in het load-script door de functie te gebruiken om een veld te maken in een hoofdagendatabel, of door de dimensie rechtstreeks in een diagram te gebruiken als een berekende dimensie.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>week('10/12/2012')</code>	Retourneert 41.
<code>week('35648')</code>	Retourneert 32, omdat 35648 = 08/06/1997.
<code>week('10/12/2012', 0, 1)</code>	Retourneert 42.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Standaardsysteemvariabelen

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor de laatste week van 2021 en de eerste twee weken van 2022 wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het veld `week_number` wordt gemaakt en retourneert het jaar- en weeknummer waarin de transacties plaatsvonden.
- Het veld `week_day` wordt gemaakt om de weekdagwaarde van iedere transactiedatum weer te geven.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=6;
SET BrokenWeeks=1;
SET ReferenceDay=0;
```

Transactions:

```
Load
    *,
    weekDay(date) as week_day,
    week(date) as week_number
;
```

Load

\*

Inline

[

id,date,amount

8183,12/27/2021,58.27

8184,12/28/2021,67.42

8185,12/29/2021,23.80

8186,12/30/2021,82.06

8187,12/31/2021,40.56

8188,01/01/2022,37.23

8189,01/02/2022,17.17

8190,01/03/2022,88.27

8191,01/04/2022,57.42

8192,01/05/2022,53.80

8193,01/06/2022,82.06

8194,01/07/2022,40.56

8195,01/08/2022,53.67

8196,01/09/2022,26.63

8197,01/10/2022,72.48

8198,01/11/2022,18.37

8199,01/12/2022,45.26

8200,01/13/2022,58.23

8201,01/14/2022,18.52

];

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `week_day`
- `week_number`

Resultatentabel

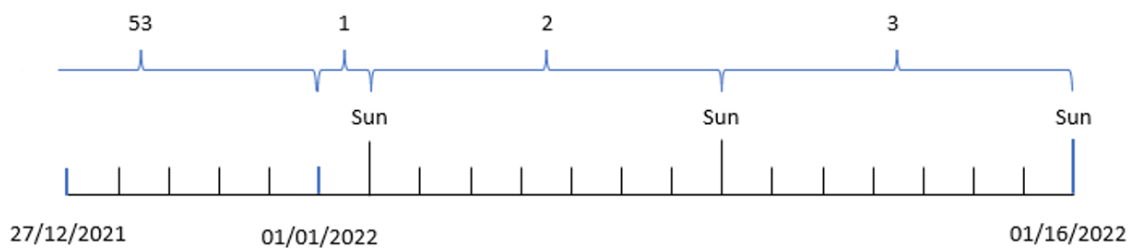
<code>id</code>	<code>date</code>	<code>week_day</code>	<code>week_number</code>
8183	12/27/2021	Ma	53
8184	12/28/2021	Di	53
8185	12/29/2021	Wo	53
8186	12/30/2021	Do	53
8187	12/31/2021	Vr	53
8188	01/01/2022	Za	1
8189	01/02/2022	Zo	2
8190	01/03/2022	Ma	2
8191	01/04/2022	Di	2
8192	01/05/2022	Wo	2
8193	01/06/2022	Do	2
8194	01/07/2022	Vr	2
8195	01/08/2022	Za	2
8196	01/09/2022	Zo	3
8197	01/10/2022	Ma	3
8198	01/11/2022	Di	3
8199	01/12/2022	Wo	3
8200	01/13/2022	Do	3
8201	01/14/2022	Vr	3

Het veld `week_number` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `week()` en geeft het veld `date` door als het argument van de functie.

Er worden geen andere parameters doorgegeven in de functie en daarom zijn de volgende standaardvariabelen die invloed hebben op de `week()`-functie van kracht:

- `brokenweeks`: De weeknummering begint op 1 januari
- `firstweekday`: De eerste dag van de week is zondag

Diagram van de week()-functie met standaardsysteemvariabelen



Omdat de applicatie de standaardsysteemvariabele `BrokenWeeks` gebruikt, begint week 1 op 1 januari, een zaterdag.

Vanwege de standaardsysteemvariabele `FirstWeekDay` begint de week op een zondag. De eerste zondag na 1 januari valt op 2 januari en daarom begint week 2 op 2 januari.

### Voorbeeld 2 - first\_week\_day

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Het veld `week_number` wordt gemaakt en retourneert het jaar- en weeknummer waarin de transacties plaatsvonden.
- Het veld `week_day` wordt gemaakt om de weekdagwaarde van iedere transactiedatum weer te geven.

In dit voorbeeld willen we het begin van de werkweek instellen op dinsdag.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=6;
SET BrokenWeeks=1;
SET ReferenceDay=0;

Transactions:
  Load
    *,
    weekDay(date) as week_day,
    week(date,1) as week_number
  ;
Load
*
Inline
```

```
[  
id,date,amount  
8183,12/27/2022,58.27  
8184,12/28/2022,67.42  
8185,12/29/2022,23.80  
8186,12/30/2022,82.06  
8187,12/31/2021,40.56  
8188,01/01/2022,37.23  
8189,01/02/2022,17.17  
8190,01/03/2022,88.27  
8191,01/04/2022,57.42  
8192,01/05/2022,53.80  
8193,01/06/2022,82.06  
8194,01/07/2022,40.56  
8195,01/08/2022,53.67  
8196,01/09/2022,26.63  
8197,01/10/2022,72.48  
8198,01/11/2022,18.37  
8199,01/12/2022,45.26  
8200,01/13/2022,58.23  
8201,01/14/2022,18.52  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- week\_day
- week\_number

Resultatentabel

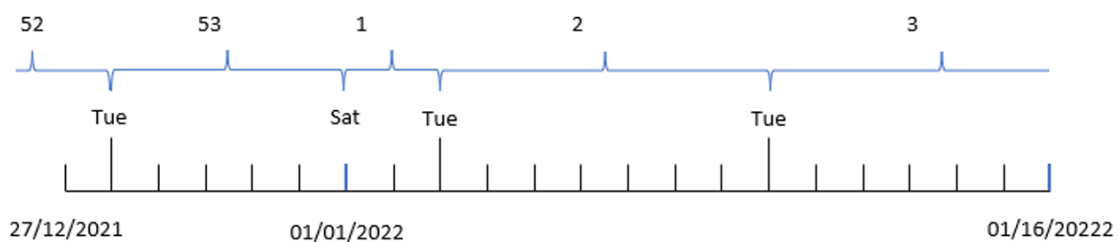
id	date	week_day	week_number
8183	12/27/2021	Ma	52
8184	12/28/2021	Di	53
8185	12/29/2021	Wo	53
8186	12/30/2021	Do	53
8187	12/31/2021	Vr	53
8188	01/01/2022	Za	1
8189	01/02/2022	Zo	1
8190	01/03/2022	Ma	1
8191	01/04/2022	Di	2
8192	01/05/2022	Wo	2



id	date	week_day	week_number
8193	01/06/2022	Do	2
8194	01/07/2022	Vr	2
8195	01/08/2022	Za	2
8196	01/09/2022	Zo	2
8197	01/10/2022	Ma	2
8198	01/11/2022	Di	3
8199	01/12/2022	Wo	3
8200	01/13/2022	Do	3
8201	01/14/2022	Vr	3

De applicatie past nog steeds opgesplitste weken toe. Maar het `first_week_day`-argument is in de `week()`-functie ingesteld op 1. Hierdoor wordt de eerste dag van de week ingesteld op dinsdag.

*Diagram van de `week()`-functie, voorbeeld met `first_week_day`*



De applicatie gebruikt de standaardstelselvariabele `brokenweeks` en daarom begint week 1 op 1 januari, een zaterdag.

Het `first_week_day`-argument van de `week()`-functie stelt de eerste weekdag in op een dinsdag. Daarom begint week 53 op 28 december 2021.

Maar omdat de functie nog steeds opgesplitste weken toepast, duurt week 1 slechts twee dagen omdat de eerste dinsdag na 1 januari op 3 januari valt.

### Voorbeeld 3 - `unbroken_weeks`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld gebruiken we niet-opgesplitste weken.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=6;
SET BrokenWeeks=1;
SET ReferenceDay=0;
```

Transactions:

```
Load
    *,
    weekDay(date) as week_day,
    week(date,6,0) as week_number
;
```

Load

\*

Inline

[

id,date,amount

8183,12/27/2022,58.27

8184,12/28/2022,67.42

8185,12/29/2022,23.80

8186,12/30/2022,82.06

8187,12/31/2021,40.56

8188,01/01/2022,37.23

8189,01/02/2022,17.17

8190,01/03/2022,88.27

8191,01/04/2022,57.42

8192,01/05/2022,53.80

8193,01/06/2022,82.06

8194,01/07/2022,40.56

8195,01/08/2022,53.67

8196,01/09/2022,26.63

8197,01/10/2022,72.48

8198,01/11/2022,18.37

8199,01/12/2022,45.26

8200,01/13/2022,58.23

8201,01/14/2022,18.52

];

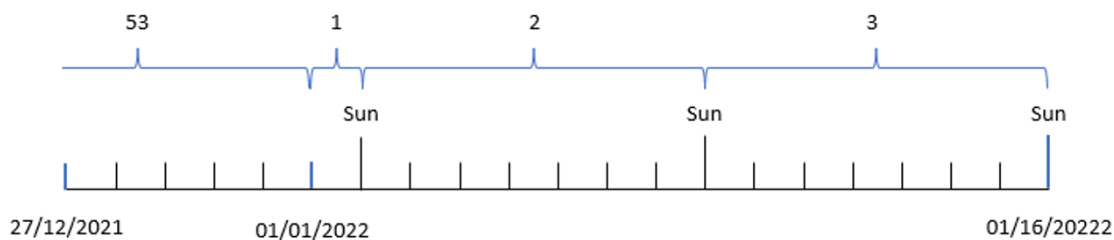
### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- week\_day
- week\_number

## 5 Script- en diagramfuncties

Diagram van week()-functie, diagramobjectvoorbeeld



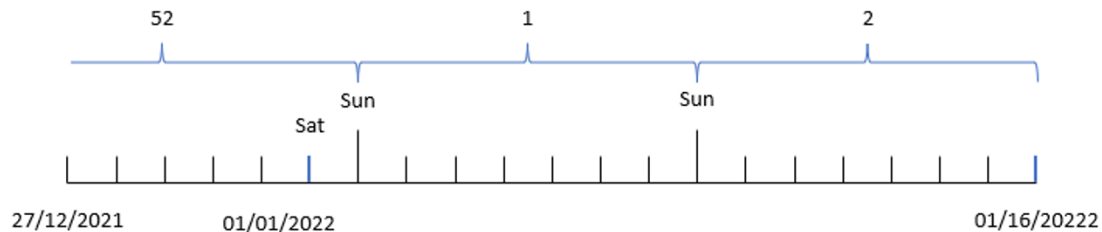
Resultatentabel

id	date	week_day	week_number
8183	12/27/2021	Ma	52
8184	12/28/2021	Di	52
8185	12/29/2021	Wo	52
8186	12/30/2021	Do	52
8187	12/31/2021	Vr	52
8188	01/01/2022	Za	52
8189	01/02/2022	Zo	1
8190	01/03/2022	Ma	1
8191	01/04/2022	Di	1
8192	01/05/2022	Wo	1
8193	01/06/2022	Do	1
8194	01/07/2022	Vr	1
8195	01/08/2022	Za	1
8196	01/09/2022	Zo	2
8197	01/10/2022	Ma	2
8198	01/11/2022	Di	2
8199	01/12/2022	Wo	2
8200	01/13/2022	Do	2
8201	01/14/2022	Vr	2

De parameter first\_week\_date is ingesteld op 1, waardoor dinsdag de eerste dag van de week is. De parameter broken\_weeks is ingesteld op 0 waardoor de functie wordt gedwongen niet-opgesplitste weken toe te passen. De derde parameter stelt de reference\_day in op 2.

De parameter `first_week_date` is ingesteld op 6, waardoor zondag de eerste dag van de week is. De parameter `broken_weeks` is ingesteld op 0 waardoor de functie wordt gedwongen niet-opgesplitste weken toe te passen.

*Diagram van de week()-functie, voorbeeld met niet-opgesplitste weken*



Door niet-opgesplitste weken te gebruiken, begint week 1 niet noodzakelijkerwijs op 1 januari. In plaats daarvan moet de week ten minste vier dagen bevatten. Daarom eindigt week 52 in de gegevensset op een zaterdag, 1 januari 2022. Week 1 begint vervolgens op de `FirstWeekDay`-systeemvariabele. Dit is zaterdag 2 januari. Deze week eindigt op de volgende zaterdag, 8 januari.

### Voorbeeld 4 - `reference_day`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het derde voorbeeld.
- Het veld `week_number` wordt gemaakt en retourneert het jaar- en weeknummer waarin de transacties plaatsvonden.
- Het veld `week_day` wordt gemaakt om de weekdagwaarde van iedere transactiedatum weer te geven.

Daarnaast moet worden voldaan aan de volgende voorwaarden:

- De werkweek begint op een dinsdag.
- Het bedrijf gebruikt niet-opgesplitste weken.
- De `reference_day`-waarde is 2. Met andere woorden: het minimumaantal dagen in januari in week 1, wordt 2.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';  
SET FirstWeekDay=6;  
SET BrokenWeeks=1;  
SET ReferenceDay=0;
```

Transactions:

```
Load
    *,
    weekDay(date) as week_day,
    week(date,1,0,2) as week_number
;

Load
*
Inline
[
id,date,amount
8183,12/27/2022,58.27
8184,12/28/2022,67.42
8185,12/29/2022,23.80
8186,12/30/2022,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- week\_day
- week\_number

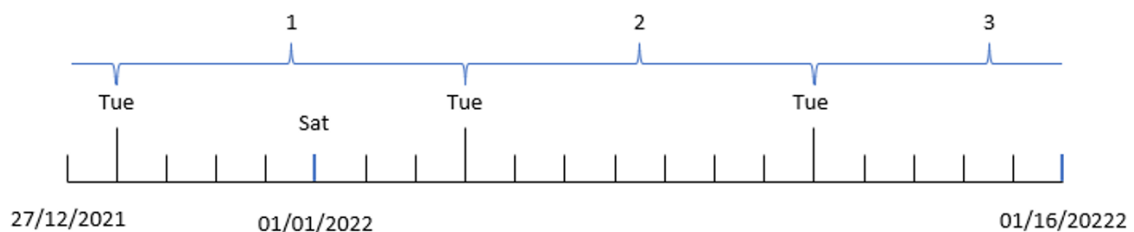
Resultatentabel

id	date	week_day	week_number
8183	12/27/2021	Ma	52
8184	12/28/2021	Di	1
8185	12/29/2021	Wo	1
8186	12/30/2021	Do	1

id	date	week_day	week_number
8187	12/31/2021	Vr	1
8188	01/01/2022	Za	1
8189	01/02/2022	Zo	1
8190	01/03/2022	Ma	1
8191	01/04/2022	Di	2
8192	01/05/2022	Wo	2
8193	01/06/2022	Do	2
8194	01/07/2022	Vr	2
8195	01/08/2022	Za	2
8196	01/09/2022	Zo	2
8197	01/10/2022	Ma	2
8198	01/11/2022	Di	3
8199	01/12/2022	Wo	3
8200	01/13/2022	Do	3
8201	01/14/2022	Vr	3

De `first_week_date`-parameter is ingesteld op 1, waardoor dinsdag de eerste dag van de week is. De `broken_weeks`-parameter is ingesteld op 0 waardoor de functie wordt gedwongen niet-opgesplitste weken toe te passen. De derde parameter stelt de `reference_day`-parameter in op 2.

*Diagram van de `week()`-functie, voorbeeld met `reference_day`*



Met de functie met niet-opgesplitste weken en een `reference_day`-waarde van 2 als een parameter, hoeft week 1 slechts twee dagen in januari te bevatten. Omdat de eerste dag van de week dinsdag is, begint week 1 op 28 december 2021 en eindigt hij op maandag 3 januari 2022.

### Voorbeeld 5 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die het weeknummer retourneert, wordt gemaakt als een meting in een diagramobject.

#### Load-script

Transactions:

Load

\*

Inline

[

id,date,amount

8183,12/27/2022,58.27

8184,12/28/2022,67.42

8185,12/29/2022,23.80

8186,12/30/2022,82.06

8187,12/31/2021,40.56

8188,01/01/2022,37.23

8189,01/02/2022,17.17

8190,01/03/2022,88.27

8191,01/04/2022,57.42

8192,01/05/2022,53.80

8193,01/06/2022,82.06

8194,01/07/2022,40.56

8195,01/08/2022,53.67

8196,01/09/2022,26.63

8197,01/10/2022,72.48

8198,01/11/2022,18.37

8199,01/12/2022,45.26

8200,01/13/2022,58.23

8201,01/14/2022,18.52

];

#### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Voeg de volgende velden als dimensies toe:
  - id
  - date

3. Maak nu de volgende meting:

=week (date)

4. Maak een , week\_day-meting om de weekdagwaarde per transactiedatum weer te geven:

=weekday(date)

Resultatentabel

id	date	=week(date)	=weekday(date)
8183	12/27/2021	53	Ma
8184	12/28/2021	53	Di
8185	12/29/2021	53	Wo
8186	12/30/2021	53	Do
8187	12/31/2021	53	Vr
8188	01/01/2022	1	Za
8189	01/02/2022	2	Zo
8190	01/03/2022	2	Ma
8191	01/04/2022	2	Di
8192	01/05/2022	2	Wo
8193	01/06/2022	2	Do
8194	01/07/2022	2	Vr
8195	01/08/2022	2	Za
8196	01/09/2022	3	Zo
8197	01/10/2022	3	Ma
8198	01/11/2022	3	Di
8199	01/12/2022	3	Wo
8200	01/13/2022	3	Do
8201	01/14/2022	3	Vr

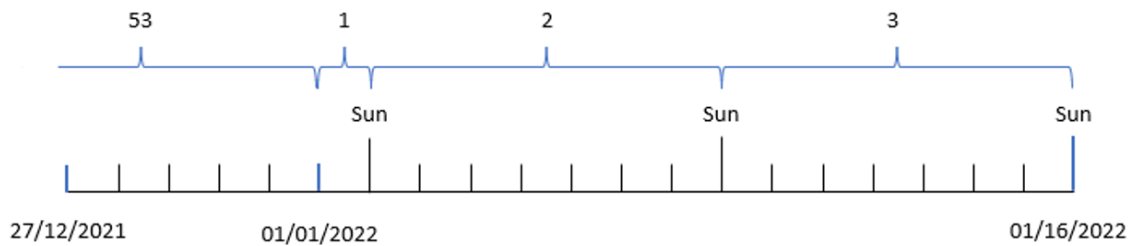
Het veld week\_number wordt gemaakt in de voorafgaande load-instructie met behulp van de functie week() en geeft het veld date door als het argument van de functie.

Er worden geen andere parameters doorgegeven in de functie en daarom zijn de volgende standaardvariabelen die invloed hebben op de week()-functie van kracht:

- BrokenWeeks: De weeknummering begint op 1 januari
- FirstWeekDay: De eerste dag van de week is zondag



Diagram van week() -functie, diagramobjectvoorbeeld



Omdat de applicatie de standaardstelsysteemvariabele `BrokenWeeks` gebruikt, begint week 1 op 1 januari, een zaterdag.

Vanwege de standaardstelsysteemvariabele `FirstWeekDay` begint de week op een zondag. De eerste zondag na 1 januari valt op 2 januari en daarom begint week 2 op 2 januari.

### Voorbeeld 6 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor de laatste week van 2019 en eerste twee weken van 2020 wordt geladen in de tabel `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-stelsysteemvariabele (MM/DD/YYYY) indeling.

De applicatie gebruikt voornamelijk opgesplitste weken op het dashboard. Maar de eindgebruiker wil een diagramobject dat de totale verkoop per week weergeeft met niet-opgesplitste weken. De referentiedag moet 2 januari zijn en de weken moeten beginnen op een dinsdag. Dit kan zelfs worden bereikt als deze dimensie niet beschikbaar is in het gegevensmodel, door de functie `week()` te gebruiken als een berekende dimensie in het diagram.

#### Load-script

```
SET BrokenWeeks=1;  
SET ReferenceDay=0;  
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load  
*  
Inline  
[  
id,date,amount  
8183,12/27/2019,58.27
```

```
8184,12/28/2019,67.42
8185,12/29/2019,23.80
8186,12/30/2019,82.06
8187,12/31/2019,40.56
8188,01/01/2020,37.23
8189,01/02/2020,17.17
8190,01/03/2020,88.27
8191,01/04/2020,57.42
8192,01/05/2020,53.80
8193,01/06/2020,82.06
8194,01/07/2020,40.56
8195,01/08/2020,53.67
8196,01/09/2020,26.63
8197,01/10/2020,72.48
8198,01/11/2020,18.37
8199,01/12/2020,45.26
8200,01/13/2020,58.23
8201,01/14/2020,18.52
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel.
2. Maak een nieuwe tabel en voeg de volgende berekende dimensie toe:  
=week(date)
3. Maak nu de volgende aggregatiemeting:  
=sum(amount)
4. Stel de **Getalnotatie** van de meting in op **Geld**.
5. Selecteer het menu **Sorteren** en verwijder Aangepaste sortering voor de berekende dimensie.
6. Hef de selecties **Numeriek sorteren** en **Alfabetisch sorteren** op.

Resultatentabel

week(date)	sum(amount)
52	\$125.69
53	\$146.42
1	\$200.09
2	\$347.57
3	\$122.01

### weekday

Deze functie retourneert een duale waarde met:

- De naam van een dag zoals gedefinieerd in de omgevingsvariabele **DayNames**.
- Een geheel getal tussen 0 en 6 dat overeenkomt met de nominale dag van de week (0-6).

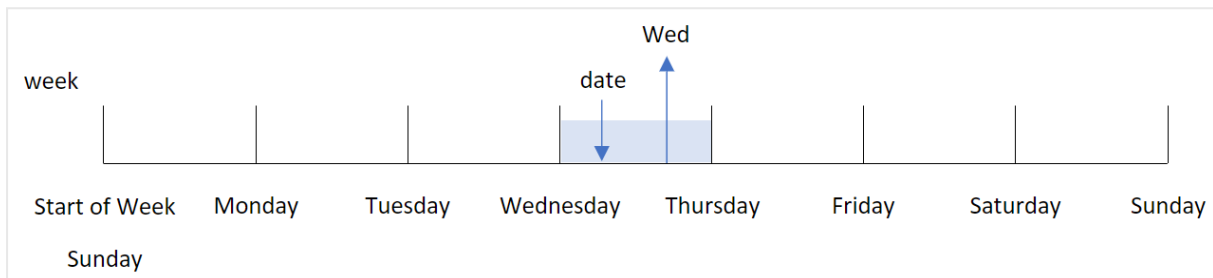
### Syntaxis:

```
weekday(date [, first_week_day=0])
```

### Retourgegevenstypen: dual

De `weekday()`-functie stelt vast op welke dag van de week een datum valt. Vervolgens wordt een tekenreekswaarde geretourneerd die die dag vertegenwoordigt.

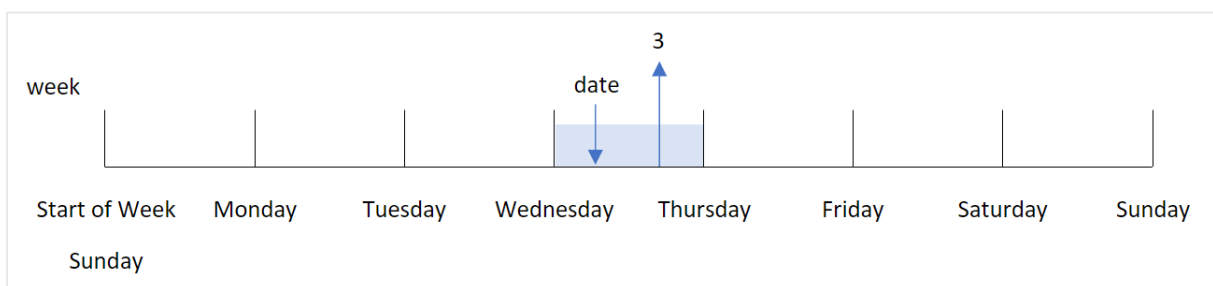
*Diagram van `weekday()`-functie die de naam van de dag retourneert waarop een datum valt*



Het resultaat retourneert de nummerwaarde die bij die dag van de week hoort (0-6), op basis van de begindag van de week. Bijvoorbeeld: als de eerste dag van de week is ingesteld op zondag, dan retourneert woensdag een nummerwaarde van 3. Deze begindag wordt bepaald door de systeemvariabele `FirstweekDay` of door de `first_week_day`-functieparameter.

U kunt deze nummerwaarde gebruiken als onderdeel van een rekenkundige uitdrukking. Bijvoorbeeld: vermenigvuldig de waarde met 1 om de waarde zelf te retourneren.

*Diagram van `weekday()`-functie waarbij de nummerwaarde van de dag wordt weergegeven in plaats van de naam van de dag*



### Wanneer gebruiken

De `weekday()`-functie is handig als u aggregaties per dag van de week wilt vergelijken. Bijvoorbeeld als u de gemiddelde verkoop van producten per weekdag wilt vergelijken.

Deze dimensies kunnen worden gemaakt in het load-script door de functie te gebruiken om een veld te maken in een **masterkalendertabel**, of rechtstreeks in een diagram worden gemaakt als een berekende dimensie.

### Gerelateerde onderwerpen

Onderwerpen	Interactie
<i>FirstWeekDay (page 219)</i>	Definieert de begindag van elke week.

### Argumenten

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>first_week_day</b>	Geeft de dag op waarop de week begint. Bij weglating wordt de waarde van de variabele <b>FirstWeekDay</b> gebruikt. <i>FirstWeekDay (page 219)</i>

U kunt de volgende waarden gebruiken om de dag in te stellen waarop de week begint in het argument `first_week_day`:

waarden `first_week_day`

Dag	Waarde
Maandag	0
Dinsdag	1
Woensdag	2
Donderdag	3
Vrijdag	4
Zaterdag	5
Zondag	6

## Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.



Tenzij anders aangegeven, is `FirstweekDay` in deze voorbeelden ingesteld op 0.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>weekday('10/12/1971')</code>	Retourneert Tue en 1.
<code>weekday('10/12/1971' , 6)</code>	Retourneert Tue en 2.  In dit voorbeeld gebruiken we zondag (6) als eerst dag van de week.
<code>SET FirstweekDay=6;</code> ... <code>weekday('10/12/1971')</code>	Retourneert Tue en 2.

### Voorbeeld 1 - Weekdagtekenreeks

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- De systeemvariabele `FirstweekDay` die is ingesteld op 6 (zondag).
- De `DayNames`-variabele die is ingesteld om de standaarddagnamen te gebruiken.
- Een voorafgaande lading die de `weekday()`-functie bevat, die is ingesteld als het `week_day`-veld en de weekdag retourneert waarin de transacties hebben plaatsgevonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
SET FirstweekDay=6;
```

Transactions:

```
Load
  *,
  weekDay(date) as week_day
;

Load
*
Inline
[
id,date,amount
8188,01/01/2022,37.23
8189,01/02/2022,17.17
```

```
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.39
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- week\_day

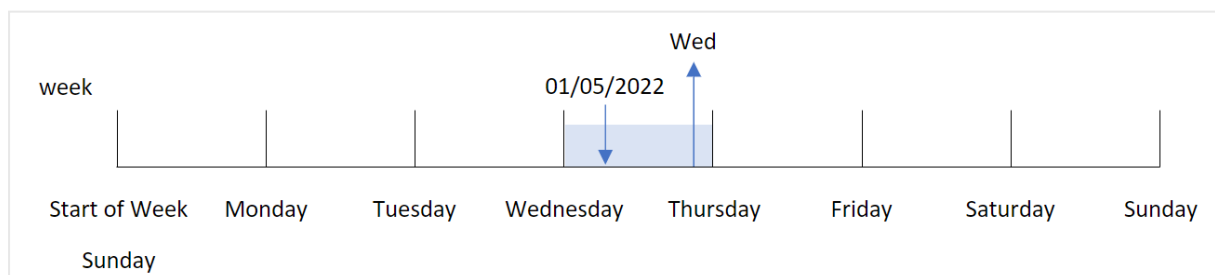
Resultatentabel

id	date	week_day
8188	01/01/2022	Za
8189	01/02/2022	Zo
8190	01/03/2022	Ma
8191	01/04/2022	Di
8192	01/05/2022	Wo
8193	01/06/2022	Do
8194	01/07/2022	Vr

Het veld week\_day wordt gemaakt in de voorafgaande load-instructie met behulp van de functie weekday() en geeft het datumveld door als het argument van de functie.

De weekday()-functie retourneert de weekdagtekenreekswaarde; dat wil zeggen, hij retourneert de naam van de weekdag die is ingesteld door de DayNames-systeemvariabele.

*Diagram van weekday()-functie die woensdag als de weekdag retourneert voor transactie 8192*



Transactie 8192 vond plaats op 5 januari. De systeemvariabele Firstweekday stelt de eerste dag van de week in op zondag. De weekday()-functie bepaalt dat de transactie op een woensdag plaatsvond en retourneert deze waarde in afgekorte vorm van de DayNames-systeemvariabele, in het week\_day-veld.

De waarden in het `week_day`-veld zijn rechts uitgelijnd in de kolom omdat er een dubbel nummer- en tekstresultaat bestaat voor het veld (woensdag 3). Om de veldwaarde om te zetten in de bijbehorende nummerequivalent, kan het veld in de `num()`-functie worden omsloten. Zo kan in transactie 8192 de woensdagwaarde worden omzet in nummer 3.

### Voorbeeld 2 - first\_week\_day

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- De systeemvariabele `FirstWeekDay` die is ingesteld op 6 (zondag).
- De `DayNames`-variabele die is ingesteld om de standaarddagnamen te gebruiken.
- Een voorafgaande lading die de `weekday()`-functie bevat, die is ingesteld als het `week_day`-veld en de weekdag retourneert waarin de transacties hebben plaatsgevonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
SET FirstWeekDay=6;
```

```
Transactions:
  Load
    *,
    weekday(date,1) as week_day
  ;
Load
*
Inline
[
id,date,amount
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.39
];
```

#### Resultaten

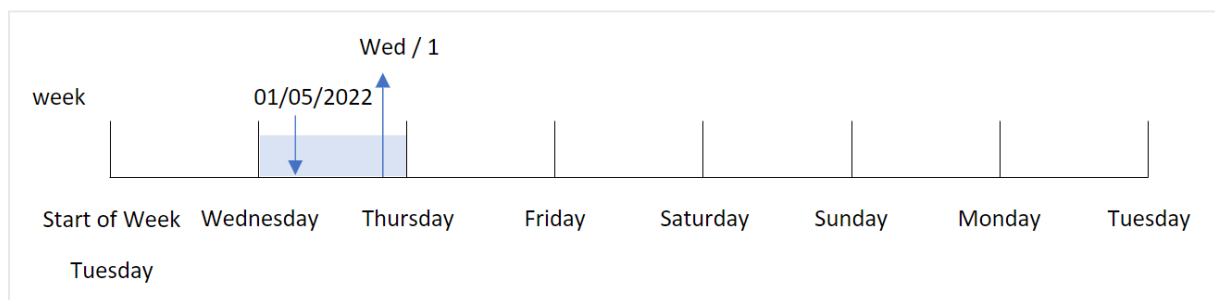
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- week\_day

Resultatentabel

id	date	week_day
8188	01/01/2022	Za
8189	01/02/2022	Zo
8190	01/03/2022	Ma
8191	01/04/2022	Di
8192	01/05/2022	Wo
8193	01/06/2022	Do
8194	01/07/2022	Vr

Diagram van de `weekday()`-functie dat laat zien dat woensdag de dubbele nummerwaarde van 1 heeft



Omdat het `first_week_day`-argument is ingesteld op 1 in de `weekday()`-functie, wordt dinsdag als eerste dag van de week gebruikt. Daarom krijgen alle transacties die op een dinsdag plaatsvinden een dubbele nummerwaarde van 0.

Transactie 8192 vond plaats op 5 januari. De `weekday()`-functie identificeert dit als een woensdag en daarom zou de uitdrukking de dubbele nummerwaarde van 1 retourneren.

### Voorbeeld 3 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- De systeemvariabele `firstweekday` die is ingesteld op 6 (zondag).



- De `DayNames`-variabele die is ingesteld om de standaarddagnamen te gebruiken.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die de weekdagwaarde identificeert, wordt gemaakt als een meting in een diagram in de app.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
SET FirstWeekDay=6;
```

Transactions:

```
Load
*
Inline
[
id,date,amount
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.39
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`

Maak de volgende meting om de weekdagwaarde te berekenen:

- `=weekday(date)`

Resultatentabel

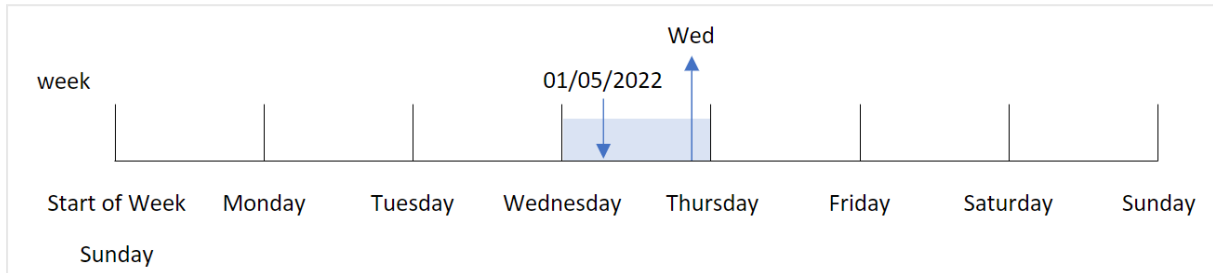
id	date	=weekday(date)
8188	01/01/2022	Za
8189	01/02/2022	Zo
8190	01/03/2022	Ma
8191	01/04/2022	Di
8192	01/05/2022	Wo
8193	01/06/2022	Do
8194	01/07/2022	Vr

## 5 Script- en diagramfuncties

Het veld `=weekday(date)` wordt gemaakt in het diagram door de functie `weekday()` te gebruiken en het datumveld door te geven als het argument van de functie.

De `weekday()`-functie retourneert de weekdagtekenreekswaarde; dat wil zeggen, hij retourneert de naam van de weekdag die is ingesteld door de `DayNames`-systeemvariabele.

*Diagram van `weekday()`-functie die woensdag als de weekdag retourneert voor transactie 8192*



Transactie 8192 vond plaats op 5 januari. De systeemvariabele `FirstWeekDay` stelt de eerste dag van de week in op zondag. De `weekday()`-functie bepaalt dat de transactie op een woensdag plaatsvond en retourneert deze waarde in afgekorte vorm van de `DayNames`-systeemvariabele, in het `=weekday(date)`-veld.

### Voorbeeld 4 - Scenario

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in de tabel `Transactions`.
- De systeemvariabele `FirstWeekDay` die is ingesteld op 6 (zondag).
- De `DayNames`-variabele die is ingesteld om de standaarddagnamen te gebruiken.

De eindgebruiker wil graag een diagram die de gemiddelde verkoop per weekdag voor de transacties weergeeft.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
SET FirstWeekDay=6;
```

`Transactions:`

```
LOAD
  RecNo() AS id,
  MakeDate(2022, 1, Ceil(Rand() * 31)) as date,
  Rand() * 1000 AS amount
```

```
Autogenerate(1000);
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- =weekday(date)
- =avg(amount)

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

weekday(date)	Avg(amount)
Zo	\$536.96
Ma	\$500.80
Di	\$515.63
Wo	\$509.21
Do	\$482.70
Vr	\$441.33
Za	\$505.22

### weekend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag (zondag) van de kalenderweek die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

#### Syntaxis:

```
WeekEnd(date [, period_no[, first_week_day]])
```

Retourgegevenstypen: dual

Voorbeelddiagram van de weekend()-functie



De weekend()-functie bepaalt welke week de datum in valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van die week. De eerste dag van de week wordt bepaald door de omgevingsvariabele `FirstweekDay`. Maar dit kan worden vervangen door het `first_week_day`-argument in de weekend()-functie.

### Argumenten

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	<b>shift</b> is een geheel getal, waarbij de waarde 0 de week aangeeft die <b>date</b> bevat. Negatieve waarden in verschuiving geven voorgaande weken aan en positieve waarden geven volgende weken aan.
<b>first_week_day</b>	Geeft de dag op waarop de week begint. Bij weglating wordt de waarde van de variabele <b>FirstWeekDay</b> gebruikt.  De mogelijke waarden voor <b>first_week_day</b> zijn 0 voor maandag, 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag.  Zie <i>FirstWeekDay</i> (page 219) voor meer informatie over de systeemvariabele
<b>broken_weeks</b>	Als u <b>broken_weeks</b> (gebroken weken) niet specificeert, dan wordt de waarde van variabele <b>BrokenWeeks</b> gebruikt om te definiëren of weken al dan niet gebroken zijn.

### Wanneer gebruiken

De `weekend()`-functie wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening de resterende dagen van de week voor de opgegeven datum wordt gebruikt. Hij zou bijvoorbeeld kunnen worden gebruikt als een gebruiker de totale rente wil berekenen die tijdens de week nog niet is opgebouwd.

Voorbeeld	Resultaat
<code>weekend('01/10/2013')</code>	Retourneert 01/12/2013 23:59:59.
<code>weekend('01/10/2013', -1)</code>	Retourneert 01/05/2013 23:59:59..
<code>weekend('01/10/2013', 0, 1)</code>	Retourneert 01/14/2013 23:59:59.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, end\_of\_week, dat een tijdstempel voor het einde van de week retourneert waarin de transactie plaatsvond.

#### Load-script

```
SET FirstWeekDay=6;
```

```
Transactions:
```

```
  Load
    *,
    weekend(date) as end_of_week,
    timestamp(weekend(date)) as end_of_week_timestamp
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- end\_of\_week
- end\_of\_week\_timestamp

Resultatentabel

date	end_of_week	end_of_week_timestamp
1/7/2022	01/08/2022	1/8/2022 11:59:59 PM
1/19/2022	01/22/2022	1/22/2022 11:59:59 PM
2/5/2022	02/05/2022	2/5/2022 11:59:59 PM
2/28/2022	03/05/2022	3/5/2022 11:59:59 PM
3/16/2022	03/19/2022	3/19/2022 11:59:59 PM
4/1/2022	04/02/2022	4/2/2022 11:59:59 PM
5/7/2022	05/07/2022	5/7/2022 11:59:59 PM
5/16/2022	05/21/2022	5/21/2022 11:59:59 PM
6/15/2022	06/18/2022	6/18/2022 11:59:59 PM
6/26/2022	07/02/2022	7/2/2022 11:59:59 PM
7/9/2022	07/09/2022	7/9/2022 11:59:59 PM
7/22/2022	07/23/2022	7/23/2022 11:59:59 PM
7/23/2022	07/23/2022	7/23/2022 11:59:59 PM
7/27/2022	07/30/2022	7/30/2022 11:59:59 PM
8/2/2022	08/06/2022	8/6/2022 11:59:59 PM
8/8/2022	08/13/2022	8/13/2022 11:59:59 PM
8/19/2022	08/20/2022	8/20/2022 11:59:59 PM
9/26/2022	10/01/2022	10/1/2022 11:59:59 PM
10/14/2022	10/15/2022	10/15/2022 11:59:59 PM
10/29/2022	10/29/2022	10/29/2022 11:59:59 PM

Het veld `end_of_week` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `weekend()` en geeft het datumveld door als het argument van de functie.

De functie `weekend()` identificeert in welke week de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die week.

### Diagram van weekend()-functie, basisvoorbeeld



Transactie 8191 vond plaats op 5 februari. De systeemvariabele `Firstweekday` stelt de eerste dag van de week in op een zondag. De `weekend()`-functie identificeert dat de eerste zaterdag na 5 februari, en daarom het einde van de week, op 5 februari viel. Daarom retourneert de `end_of_week`-waarde voor die transactie de laatste milliseconde van die dag, die op 5 februari om 11:59:59 PM was.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_week_end`, dat de tijdstempel voor het einde van de week retourneert voordat de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*,
weekend(date,-1) as previous_week_end,
timestamp(weekend(date,-1)) as previous_week_end_timestamp
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_week\_end
- previous\_week\_end\_timestamp

Resultatentabel

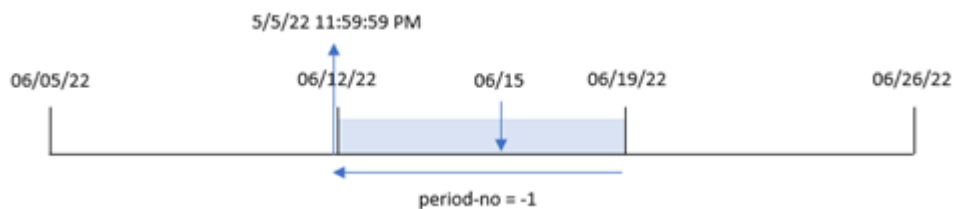
date	end_of_week	end_of_week_timestamp
1/7/2022	01/01/2022	1/1/2022 11:59:59 PM
1/19/2022	01/15/2022	1/15/2022 11:59:59 PM
2/5/2022	01/29/2022	1/29/2022 11:59:59 PM
2/28/2022	02/26/2022	2/26/2022 11:59:59 PM
3/16/2022	03/12/2022	3/12/2022 11:59:59 PM
4/1/2022	03/26/2022	3/26/2022 11:59:59 PM
5/7/2022	04/30/2022	4/30/2022 11:59:59 PM
5/16/2022	05/14/2022	5/14/2022 11:59:59 PM
6/15/2022	06/11/2022	6/11/2022 11:59:59 PM
6/26/2022	06/25/2022	6/25/2022 11:59:59 PM
7/9/2022	07/02/2022	7/2/2022 11:59:59 PM
7/22/2022	07/16/2022	7/16/2022 11:59:59 PM
7/23/2022	07/16/2022	7/16/2022 11:59:59 PM
7/27/2022	07/23/2022	7/23/2022 11:59:59 PM
8/2/2022	07/30/2022	7/30/2022 11:59:59 PM
8/8/2022	08/06/2022	8/6/2022 11:59:59 PM
8/19/2022	08/13/2022	8/13/2022 11:59:59 PM



date	end_of_week	end_of_week_timestamp
9/26/2022	09/24/2022	9/24/2022 11:59:59 PM
10/14/2022	10/08/2022	10/8/2022 11:59:59 PM
10/29/2022	10/22/2022	10/22/2022 11:59:59 PM

In dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `weekend()`-functie, identificeert de functie eerst de week waarin de transacties plaatsvinden. Het kijkt dan een week eerder en identificeert de laatste milliseconde van die week.

Diagram van `weekend()`-functie, voorbeeld `period_no`



Transactie 8196 vond plaats op 15 juni. De `weekend()`-functie identificeert dat de week op 12 juni begint. Daarom eindigt de vorige week op 11 juni om 11:59:59 PM; dit is de waarde die voor het `previous_week_end`-veld wordt geretourneerd.

### Voorbeeld 3 - first\_week\_day

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. Maar in dit voorbeeld moeten we dinsdag instellen als de eerste dag van de werkweek.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,
weekend(date,0,1) as end_of_week,
timestamp(weekend(date,0,1)) as end_of_week_timestamp,
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- end\_of\_week
- end\_of\_week\_timestamp

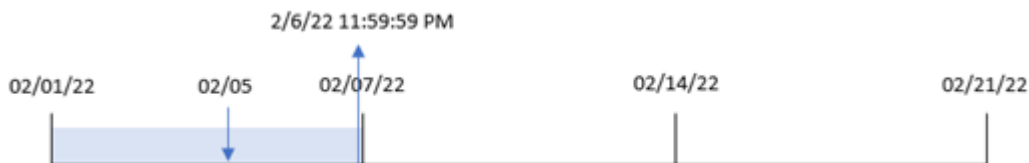
Resultatentabel

date	end_of_week	end_of_week_timestamp
1/7/2022	01/10/2022	1/10/2022 11:59:59 PM
1/19/2022	01/24/2022	1/24/2022 11:59:59 PM
2/5/2022	02/07/2022	2/7/2022 11:59:59 PM
2/28/2022	02/28/2022	2/28/2022 11:59:59 PM
3/16/2022	03/21/2022	3/21/2022 11:59:59 PM
4/1/2022	04/04/2022	4/4/2022 11:59:59 PM
5/7/2022	05/09/2022	5/9/2022 11:59:59 PM
5/16/2022	05/16/2022	5/16/2022 11:59:59 PM
6/15/2022	06/20/2022	6/20/2022 11:59:59 PM
6/26/2022	06/27/2022	6/27/2022 11:59:59 PM
7/9/2022	07/11/2022	7/11/2022 11:59:59 PM

date	end_of_week	end_of_week_timestamp
7/22/2022	07/25/2022	7/25/2022 11:59:59 PM
7/23/2022	07/25/2022	7/25/2022 11:59:59 PM
7/27/2022	08/01/2022	8/1/2022 11:59:59 PM
8/2/2022	08/08/2022	8/8/2022 11:59:59 PM
8/8/2022	08/08/2022	8/8/2022 11:59:59 PM
8/19/2022	08/22/2022	8/22/2022 11:59:59 PM
9/26/2022	09/26/2022	9/26/2022 11:59:59 PM
10/14/2022	10/17/2022	10/17/2022 11:59:59 PM
10/29/2022	10/31/2022	10/31/2022 11:59:59 PM

In dit geval geldt dat omdat het `first_week_date`-argument 1 wordt gebruikt in de `weekend()`-functie, dinsdag als eerste dag van de week wordt ingesteld.

Diagram van de `weekend()`-functie, voorbeeld met `first_week_day`



Transactie 8191 vond plaats op 5 februari. De `weekend()`-functie identificeert dat de eerste maandag na deze datum, en daarom het einde van de week en getoonde waarde, op 6 februari om 11:59:59 PM viel.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het einde van de week waarin de transacties plaatsvonden, wordt gemaakt als meting in een diagramobject van de toepassing.

#### Load-script

Transactions:

Load

\*

Inline

```
[  
id,date,amount  
8188,1/7/2022,17.17  
8189,1/19/2022,37.23  
8190,2/28/2022,88.27  
8191,2/5/2022,57.42  
8192,3/16/2022,53.80  
8193,4/1/2022,82.06  
8194,5/7/2022,40.39  
8195,5/16/2022,87.21  
8196,6/15/2022,95.93  
8197,6/26/2022,45.89  
8198,7/9/2022,36.23  
8199,7/22/2022,25.66  
8200,7/23/2022,82.77  
8201,7/27/2022,69.98  
8202,8/2/2022,76.11  
8203,8/8/2022,25.12  
8204,8/19/2022,46.23  
8205,9/26/2022,84.21  
8206,10/14/2022,96.24  
8207,10/29/2022,67.67  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Voeg de volgende metingen toe om de start van de week waarin een transactie plaatsvindt te berekenen:

- =weekend(date)
- =timestamp(weekend(date))

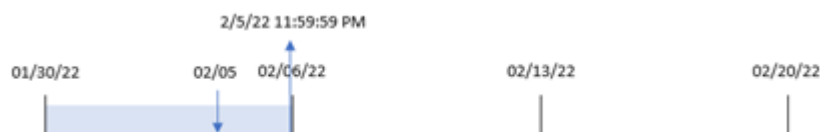
Resultatentabel

date	=weekend(date)	=timestamp(weekend(date))
1/7/2022	01/08/2022	1/8/2022 11:59:59 PM
1/19/2022	01/22/2022	1/22/2022 11:59:59 PM
2/5/2022	02/05/2022	2/5/2022 11:59:59 PM
2/28/2022	03/05/2022	3/5/2022 11:59:59 PM
3/16/2022	03/19/2022	3/19/2022 11:59:59 PM
4/1/2022	04/02/2022	4/2/2022 11:59:59 PM
5/7/2022	05/07/2022	5/7/2022 11:59:59 PM
5/16/2022	05/21/2022	5/21/2022 11:59:59 PM
6/15/2022	06/18/2022	6/18/2022 11:59:59 PM

date	=weekend(date)	=timestamp(weekend(date))
6/26/2022	07/02/2022	7/2/2022 11:59:59 PM
7/9/2022	07/09/2022	7/9/2022 11:59:59 PM
7/22/2022	07/23/2022	7/23/2022 11:59:59 PM
7/23/2022	07/23/2022	7/23/2022 11:59:59 PM
7/27/2022	07/30/2022	7/30/2022 11:59:59 PM
8/2/2022	08/06/2022	8/6/2022 11:59:59 PM
8/8/2022	08/13/2022	8/13/2022 11:59:59 PM
8/19/2022	08/20/2022	8/20/2022 11:59:59 PM
9/26/2022	10/01/2022	10/1/2022 11:59:59 PM
10/14/2022	10/15/2022	10/15/2022 11:59:59 PM
10/29/2022	10/29/2022	10/29/2022 11:59:59 PM

De meting `end_of_week` wordt gemaakt in het diagramobject met behulp van de functie `weekend()` en geeft het veld door als het argument van de functie. De `weekend()`-functie identificeert in welke week de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van die week.

*Diagram van `weekend()`-functie, diagramobjectvoorbeeld*



Transactie 8191 vond plaats op 5 februari. De systeemvariabele `FirstweekDay` stelt de eerste dag van de week in op een zondag. De `weekend()`-functie identificeert dat de eerste zaterdag na 5 februari, en daarom het einde van de week, op 5 februari viel. Daarom retourneert de `end_of_week`-waarde voor die transactie de laatste milliseconde van die dag, die op 5 februari om 11:59:59 PM was.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Employee_Expenses`.
- Gegevens omvatten werknemers-id's, werknemersnamen en de gemiddelde dagelijkse onkostendeclaraties van elke werknemer.

De eindgebruiker wil graag een diagramobject dat per werknemer-id en werknemersnaam de geschatte nog te maken onkostendeclaraties voor de rest van de week weergeeft.

### Load-script

```
Employee_Expenses :
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,Sydney,$27
186,Agatha,$18
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
  - employee\_id
  - employee\_name
2. Maak nu een meting om de opgebouwde rente te berekenen:  
 $=(\text{weekend}(\text{today}(1))-\text{today}(1))*\text{avg\_daily\_claim}$
3. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

employee_id	employee_name	$=(\text{weekend}(\text{today}(1))-\text{today}(1))*\text{avg\_daily\_claim}$
182	Mark	\$90.00
183	Deryck	\$75.00
184	Dexter	\$75.00
185	Sydney	\$162.00
186	Agatha	\$108.00

Door de datum van vandaag als het enige argument te gebruiken, retourneert de `weekend()`-functie de einddatum van de huidige week. Door vervolgens de datum van vandaag af te trekken van de einddatum van de week, retourneert de uitdrukking het aantal resterende dagen in deze week.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in de resterende week zal indienen.

## weekname

Deze functie retourneert het jaar en weeknummer met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de week die **date** bevat.

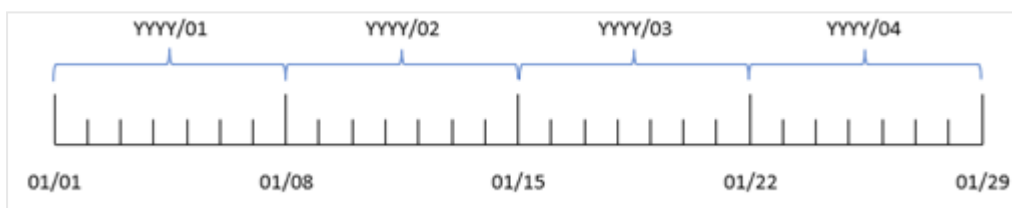
### Syntaxis:

```
WeekName (date[, period_no[, first_week_day]])
```

De functie `weekname()` bepaalt in welke week de datum valt en retourneert het weeknummer en het jaar van die week. De eerste dag van de week wordt bepaald door de systeemvariabele `FirstWeekDay`. U kunt echter ook de eerste dag van de week wijzigen door het argument `first_week_day` in de functie `weekname()` te gebruiken.

Qlik Sense-toepassingen gebruiken standaard gebroken weken (gedefinieerd door de systeemvariabele `BrokenWeeks`) en daarom begint het aantal weken op 1 januari en eindigt op de dag voorafgaand aan de systeemvariabele `FirstWeekDay`, ongeacht hoeveel dagen er zijn verstreken.

*Diagram van weekname-functie.*



Als uw toepassing echter ononderbroken weken gebruikt, kan week 1 in het voorgaande jaar of in de eerste paar dagen van januari beginnen. Dit hangt af van hoe u de systeemvariabelen `ReferenceDay` en `FirstWeekDay` gebruikt.

### Wanneer gebruiken

De functie `weekname()` is handig als u aggregaties per week wilt vergelijken.

Bijvoorbeeld als u de totale verkoop van producten per week wilt zien. Gebruik `weekname()` in plaats van `Tunarweekname()` om consistentie te behouden met de omgevingsvariabele `BrokenWeeks` in de toepassing. Als de toepassing ononderbroken weken gebruikt, kan week 1 datums van december van het voorgaande jaar bevatten of datums in januari van het lopende jaar uitsluiten. Als de toepassing gebruikmaakt van gebroken weken, kan week 1 minder dan zeven dagen bevatten.

**Retourgegevenstypen:** dual

#### Argumenten

Argument	Beschrijving
<code>date</code>	De te evalueren datum of tijdstempel.

Argument	Beschrijving
<b>period_no</b>	<b>shift</b> is een geheel getal, waarbij de waarde 0 de week aangeeft die <b>date</b> bevat. Negatieve waarden in verschuiving geven voorgaande weken aan en positieve waarden geven volgende weken aan.
<b>first_week_day</b>	Geeft de dag op waarop de week begint. Bij weglating wordt de waarde van de variabele <b>FirstWeekDay</b> gebruikt.  De mogelijke <b>first_week_day</b> -waarden zijn 0 voor maandag, 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag.  Zie <i>FirstWeekDay (page 219)</i> voor meer informatie over de systeemvariabele.

U kunt de volgende waarden gebruiken om de dag in te stellen waarop de week begint in het argument `first_week_day`:

waarden `first_week_day`

Dag	Waarde
Maandag	0
Dinsdag	1
Woensdag	2
Donderdag	3
Vrijdag	4
Zaterdag	5
Zondag	6

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.



### Voorbeelden van functies

Voorbeeld	Resultaat
weekname('01/12/2013')	Retourneert 2013/02.
weekname('01/12/2013', -1)	Returns 2013/01.
weekname('01/12/2013', 0, 1)	Retourneert 2013/02.

### Voorbeeld 1 - datum zonder aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor de laatste week van 2021 en eerste twee weken van 2022 wordt geladen in een tabel met de naam 'Transactions'.
- De systeemvariabele `DateFormat` die is ingesteld op de indeling `MM/DD/YYYY`.
- De systeemvariabele `BrokenWeeks` die is ingesteld op 1.
- De systeemvariabele `FirstWeekDay` die is ingesteld op 6.
- Een voorgaande lading die het volgende bevat:
  - De functie `weekday()` die is ingesteld als het veld, 'week\_number', dat het jaar- en weeknummer retourneert wanneer de transacties plaatsvonden.
  - De functie `weekname()` die is ingesteld als het veld met de naam 'week\_day', om de weekdagwaarde van elke transactiedatum weer te geven.

#### Load-script

```
SET BrokenWeeks=1;
SET DateFormat='MM/DD/YYYY';
SET FirstWeekDay=6;

Transactions:
  Load
    *,
    weekday(date) as week_day,
    weekname(date) as week_number
  ;
Load
*
Inline
[
id,date,amount
8183,12/27/2021,58.27
8184,12/28/2021,67.42
8185,12/29/2021,23.80
```

```
8186,12/30/2021,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- week\_day
- week\_number

Resultatentabel

id	date	week_day	week_number
8183	12/27/2021	Ma	2021/53
8184	12/28/2021	Di	2021/53
8185	12/29/2021	Wo	2021/53
8186	12/30/2021	Do	2021/53
8187	12/31/2021	Vr	2021/53
8188	01/01/2022	Za	2022/01
8189	01/02/2022	Zo	2022/02
8190	01/03/2022	Ma	2022/02
8191	01/04/2022	Di	2022/02
8192	01/05/2022	Wo	2022/02
8193	01/06/2022	Do	2022/02
8194	01/07/2022	Vr	2022/02
8195	01/08/2022	Za	2022/02

id	date	week_day	week_number
8196	01/09/2022	Zo	2022/03
8197	01/10/2022	Ma	2022/03
8198	01/11/2022	Di	2022/03
8199	01/12/2022	Wo	2022/03
8200	01/13/2022	Do	2022/03
8201	01/14/2022	Vr	2022/03

Het veld 'week\_number' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `weekname()` en geeft het datumveld door als het argument van de functie.

De functie `weekname()` identificeert aanvankelijk in welke week de datumwaarde valt en retourneert het aantal weeknummers en het jaar waarin de transactie plaatsvindt.

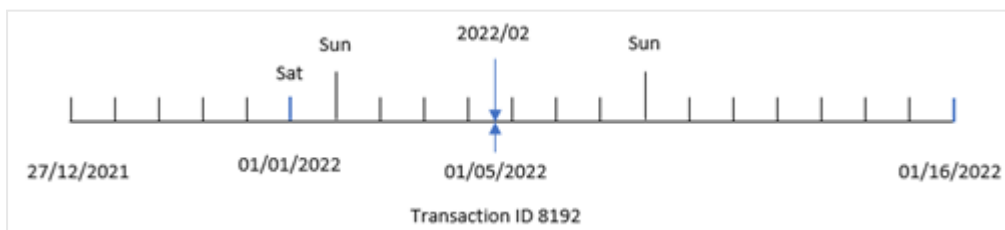
De systeemvariabele `Firstweekday` stelt zondag in als de eerste dag van de week. De systeemvariabele `Brokenweeks` stelt de toepassing in op het gebruik van gebroken weken, wat betekent dat week 1 op 1 januari begint.

*Diagram van de functie `weekname()` met de standaardvariabelen.*



Week 1 begint op 1 januari, wat een zaterdag is, en daarom retourneren transacties die op deze datum plaatsvinden de waarde 2022/01 (het jaar- en weeknummer).

*Diagram van de functie `weekname()` die het weeknummer van transactie 8192 identificeert.*



Omdat de toepassing gebroken weken gebruikt en de eerste weekdag zondag is, retourneren transacties die plaatsvinden van 2 januari tot 8 januari de waarde 2022/02 (weeknummer 2 in 2022). Een voorbeeld hiervan is transactie 8192 die plaatsvond op 5 januari en de waarde 2022/02 retourneert voor het veld 'week\_number'.

### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het echter de taak om een veld te maken, 'previous\_week\_number', dat het jaar en het weeknummer retourneert voordat de transacties plaatsvonden.

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuw tabblad.

#### Load-script

```
SET BrokenWeeks=1;
SET FirstweekDay=6;
```

Transactions:

```
Load
*,
weekname(date,-1) as previous_week_number
;
```

```
Load
*
```

Inline

```
[
id,date,amount
8183,12/27/2021,58.27
8184,12/28/2021,67.42
8185,12/29/2021,23.80
8186,12/30/2021,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

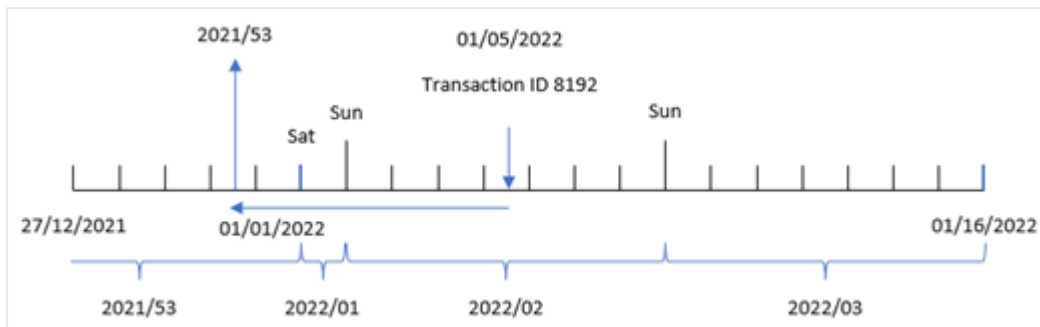
- `id`
- `date`
- `week_day`
- `week_number`

Resultatentabel

<b>id</b>	<b>date</b>	<b>week_day</b>	<b>week_number</b>
8183	12/27/2021	Ma	2021/52
8184	12/28/2021	Di	2021/52
8185	12/29/2021	Wo	2021/52
8186	12/30/2021	Do	2021/52
8187	12/31/2021	Vr	2021/52
8188	01/01/2022	Za	2021/52
8189	01/02/2022	Zo	2021/53
8190	01/03/2022	Ma	2021/53
8191	01/04/2022	Di	2021/53
8192	01/05/2022	Wo	2021/53
8193	01/06/2022	Do	2021/53
8194	01/07/2022	Vr	2021/53
8195	01/08/2022	Za	2022/01
8196	01/09/2022	Zo	2022/02
8197	01/10/2022	Ma	2022/02
8198	01/11/2022	Di	2022/02
8199	01/12/2022	Wo	2022/02
8200	01/13/2022	Do	2022/02
8201	01/14/2022	Vr	2022/02

Omdat een `period_no` van `-1` werd gebruikt als het `offset`-argument in de functie `weekname()`, identificeert de functie eerst de week waarin de transacties plaatsvinden. Het kijkt dan een week eerder en identificeert de eerste milliseconde van die week.

Diagram van functie `weekname()` met een offset `period_no` van -1.



Transactie 8192 vond plaats op 5 januari 2022. De functie `weekname()` kijkt een week eerder, 30 december 2021, en retourneert het weeknummer en het jaar voor die datum - 2021/53.

### Voorbeeld 3 - `first_week_day`

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het bedrijfsbeleid echter dat de werkweek op dinsdag begint.

Open de Editor voor laden van gegevens en voeg het volgende load-script toe aan een nieuw tabblad.

#### Load-script

```
SET BrokenWeeks=1;
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    weekday(date) as week_day,
    weekname(date,0,1) as week_number
  ;
Load
*
Inline
[
id,date,amount
8183,12/27/2021,58.27
8184,12/28/2021,67.42
8185,12/29/2021,23.80
8186,12/30/2021,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
```

```
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

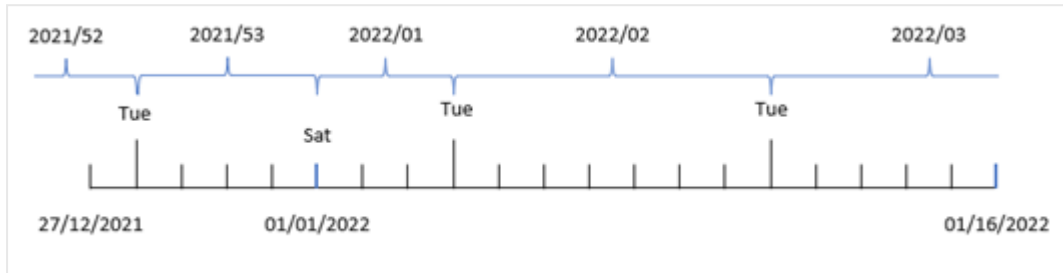
- id
- date
- week\_day
- week\_number

Resultatentabel

id	date	week_day	week_number
8183	12/27/2021	Ma	2021/52
8184	12/28/2021	Di	2021/53
8185	12/29/2021	Wo	2021/53
8186	12/30/2021	Do	2021/53
8187	12/31/2021	Vr	2021/53
8188	01/01/2022	Za	2022/01
8189	01/02/2022	Zo	2022/01
8190	01/03/2022	Ma	2022/01
8191	01/04/2022	Di	2022/02
8192	01/05/2022	Wo	2022/02
8193	01/06/2022	Do	2022/02
8194	01/07/2022	Vr	2022/02
8195	01/08/2022	Za	2022/02
8196	01/09/2022	Zo	2022/02
8197	01/10/2022	Ma	2022/02
8198	01/11/2022	Di	2022/03
8199	01/12/2022	Wo	2022/03

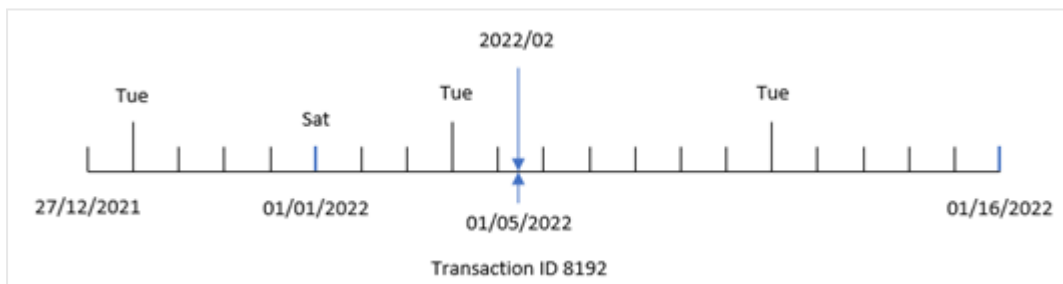
id	date	week_day	week_number
8200	01/13/2022	Do	2022/03
8201	01/14/2022	Vr	2022/03

Diagram van functie `weekname()` met dinsdag als de eerste dag van de week.



Omdat het argument `first_week_date` van 1 wordt gebruikt in de functie `weekname()`, wordt dinsdag als eerste dag van de week gebruikt. De functie bepaalt dus dat week 53 van 2021 begint op dinsdag 28 december; en vanwege de toepassing met gebroken weken begint week 1 op 1 januari 2022 en eindigt op de laatste milliseconde van maandag 3 januari 2022.

Diagram met weeknummer van transactie 8192 met dinsdag als eerste dag van de week.



Transactie 8192 vond plaats op 5 januari 2022. Daarom, met behulp van een parameter `first_week_day` van dinsdag, retourneert de functie `weekname()` de waarde 2022/02 voor het veld 'week\_number'.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die het jaarnummer van de week retourneert maand waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de toepassing.



### Load-script

```
SET BrokenWeeks=1;
Transactions:
Load
*
Inline
[
id,date,amount
8183,12/27/2021,58.27
8184,12/28/2021,67.42
8185,12/29/2021,23.80
8186,12/30/2021,82.06
8187,12/31/2021,40.56
8188,01/01/2022,37.23
8189,01/02/2022,17.17
8190,01/03/2022,88.27
8191,01/04/2022,57.42
8192,01/05/2022,53.80
8193,01/06/2022,82.06
8194,01/07/2022,40.56
8195,01/08/2022,53.67
8196,01/09/2022,26.63
8197,01/10/2022,72.48
8198,01/11/2022,18.37
8199,01/12/2022,45.26
8200,01/13/2022,58.23
8201,01/14/2022,18.52
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- =week\_day (date)

Maak de volgende meting om de start van de week waarin een transactie plaatsvindt te berekenen:

=weekname(date)

Resultatentabel

id	date	=weekday(date)	=weekname(date)
8183	12/27/2021	Ma	2021/53
8184	12/28/2021	Di	2021/53
8185	12/29/2021	Wo	2021/53
8186	12/30/2021	Do	2021/53

## 5 Script- en diagramfuncties

id	date	=weekday(date)	=weekname(date)
8187	12/31/2021	Vr	2021/53
8188	01/01/2022	Za	2022/01
8189	01/02/2022	Zo	2022/02
8190	01/03/2022	Ma	2022/02
8191	01/04/2022	Di	2022/02
8192	01/05/2022	Wo	2022/02
8193	01/06/2022	Do	2022/02
8194	01/07/2022	Vr	2022/02
8195	01/08/2022	Za	2022/02
8196	01/09/2022	Zo	2022/03
8197	01/10/2022	Ma	2022/03
8198	01/11/2022	Di	2022/03
8199	01/12/2022	Wo	2022/03
8200	01/13/2022	Do	2022/03
8201	01/14/2022	Vr	2022/03

Het veld 'week\_number' wordt gemaakt als een meting in het diagramobject door de functie weekname() te gebruiken en het datumveld door te geven als het argument van de functie.

De functie weekname() identificeert aanvankelijk in welke week de datumwaarde valt en retourneert het aantal weeknummers en het jaar waarin de transactie plaatsvindt.

De systeemvariabele Firstweekday stelt zondag in als de eerste dag van de week. De systeemvariabele Brokenweeks stelt de toepassing in op het gebruik van gebroken weken, wat betekent dat week 1 op 1 januari begint.

*Diagram met weeknummer met zondag als eerste dag van de week.*



Diagram waarop te zien is dat transactie 8192 in week nummer twee heeft plaatsgevonden.



Omdat de toepassing gebroken weken gebruikt en de eerste weekdag zondag is, retourneren transacties die plaatsvinden van 2 januari tot 8 januari de waarde 2022/02, weeknummer 2 in 2022. Transactie 8192 vond plaats op 5 januari en de waarde retourneert 2022/02 voor het veld 'week\_number'.

### Voorbeeld 5 - Scenario

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor de laatste week van 2019 en eerste twee weken van 2020 wordt geladen in een tabel met de naam 'Transactions'.
- De systeemvariabele `brokenweeks` die is ingesteld op 0.
- De systeemvariabele `referenceDay` die is ingesteld op 2.
- De systeemvariabele `dateFormat` die is ingesteld op de indeling `MM/DD/YYYY`.

#### Load-script

```
SET BrokenWeeks=0;  
SET ReferenceDay=2;  
SET DateFormat='MM/DD/YYYY';
```

Transactions:

```
Load  
*  
Inline  
[  
id,date,amount  
8183,12/27/2019,58.27  
8184,12/28/2019,67.42  
8185,12/29/2019,23.80  
8186,12/30/2019,82.06  
8187,12/31/2019,40.56  
8188,01/01/2020,37.23  
8189,01/02/2020,17.17  
8190,01/03/2020,88.27  
8191,01/04/2020,57.42  
8192,01/05/2020,53.80
```

```
8193,01/06/2020,82.06
8194,01/07/2020,40.56
8195,01/08/2020,53.67
8196,01/09/2020,26.63
8197,01/10/2020,72.48
8198,01/11/2020,18.37
8199,01/12/2020,45.26
8200,01/13/2020,58.23
8201,01/14/2020,18.52
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel.

Maak een berekende dimensie met behulp van de volgende uitdrukking:

```
=weekname(date)
```

Maak de volgende aggregatiemeting om de totale verkoop te berekenen:

```
=sum(amount)
```

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

<b>weekname(date)</b>	<b>=sum(amount)</b>
2019/52	\$125.69
2020/01	\$346.51
2020/02	\$347.57
2020/03	\$122.01

Om de resultaten van het gebruik van de functie weekname() in dit scenario te demonstreren, voegt u het volgende veld toe als een dimensie:

date

Resultatentabel met datumveld

<b>weekname(date)</b>	<b>date</b>	<b>=sum(amount)</b>
2019/52	12/27/2019	\$58.27
2019/52	12/28/2019	\$67.42
2020/01	12/29/2019	\$23.80
2020/01	12/30/2019	\$82.06
2020/01	12/31/2019	\$40.56
2020/01	01/01/2020	\$37.23

weekname(date)	date	=sum(amount)
2020/01	01/02/2020	\$17.17
2020/01	01/03/2020	\$88.27
2020/01	01/04/2020	\$57.42
2020/02	01/05/2020	\$53.80
2020/02	01/06/2020	\$82.06
2020/02	01/07/2020	\$40.56
2020/02	01/08/2020	\$53.67
2020/02	01/09/2020	\$26.63
2020/02	01/10/2020	\$72.48
2020/02	01/11/2020	\$18.37
2020/03	01/12/2020	\$45.26
2020/03	01/13/2020	\$58.23
2020/03	01/14/2020	\$18.52

Omdat de toepassing ononderbroken weken gebruikt en week 1 minimaal twee dagen in januari vereist vanwege de systeemvariabele `ReferenceDay`, bevat week 1 van 2020 transacties vanaf 29 december 2019.

### weekstart

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van de kalenderweek die **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

#### Syntaxis:

```
WeekStart(date [, period_no[, first_week_day]])
```

**Retourgegevenstypen:** dual

*Diagram van de weekstart()-functie*



De `weekstart()`-functie bepaalt welke week de datum in valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de eerste milliseconde van die week. De eerste dag van de week wordt bepaald door de omgevingsvariabele `FirstweekDay`. Maar dit kan worden vervangen door het `first_week_day`-argument in de `weekstart()`-functie.

### Argumenten

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	<b>shift</b> is een geheel getal, waarbij de waarde 0 de week aangeeft die <b>date</b> bevat. Negatieve waarden in verschuiving geven voorgaande weken aan en positieve waarden geven volgende weken aan.
<b>first_week_day</b>	Geeft de dag op waarop de week begint. Bij weglating wordt de waarde van de variabele <b>FirstWeekDay</b> gebruikt.  De mogelijke <b>first_week_day</b> -waarden zijn 0 voor maandag, 1 voor dinsdag, 2 voor woensdag, 3 voor donderdag, 4 voor vrijdag, 5 voor zaterdag en 6 voor zondag.  Zie <i>FirstWeekDay</i> (page 219) voor meer informatie over de systeemvariabele.

### Wanneer gebruiken

De functie `weekstart()` wordt doorgaans gebruikt als onderdeel van een uitdrukking als de gebruiker wil dat in de berekening het deel van de week wordt gebruikt dat al is geweest. Bijvoorbeeld: het zou kunnen worden gebruikt om het totale salaris te berekenen dat werknemers op een week tot dusver hebben verdiend.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>weekstart('01/12/2013')</code>	Retourneert 01/07/2013.
<code>weekstart('01/12/2013', -1 )</code>	Retourneert 11/31/2012.
<code>weekstart('01/12/2013', 0, 1)</code>	Retourneert 01/08/2013.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor 2022 die wordt geladen in een tabel met de naam Transactions.
- Het datumveld opgegeven in de DateFormat-systeemvariabele (MM/DD/YYYY) indeling.
- Het maken van een veld, start\_of\_week, dat een tijdstempel voor het begin van de week retourneert waarin de transactie plaatsvond.

#### Load-script

```
SET FirstWeekDay=6;
```

```
Transactions:
```

```
  Load
    *,
    weekstart(date) as start_of_week,
    timestamp(weekstart(date)) as start_of_week_timestamp
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
```

```
8189,1/19/2022,37.23
```

```
8190,2/28/2022,88.27
```

```
8191,2/5/2022,57.42
```

```
8192,3/16/2022,53.80
```

```
8193,4/1/2022,82.06
```

```
8194,5/7/2022,40.39
```

```
8195,5/16/2022,87.21
```

```
8196,6/15/2022,95.93
```

```
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
```

```
8199,7/22/2022,25.66
```

```
8200,7/23/2022,82.77
```

```
8201,7/27/2022,69.98
```

```
8202,8/2/2022,76.11
```

```
8203,8/8/2022,25.12
```

```
8204,8/19/2022,46.23
```

```
8205,9/26/2022,84.21
```

```
8206,10/14/2022,96.24
```

```
8207,10/29/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- start\_of\_week
- start\_of\_week\_timestamp

Resultatentabel

date	start_of_week	start_of_week_timestamp
1/7/2022	01/02/2022	1/2/2022 12:00:00 AM
1/19/2022	01/16/2022	1/16/2022 12:00:00 AM
2/5/2022	01/30/2022	1/30/2022 12:00:00 AM
2/28/2022	02/27/2022	2/27/2022 12:00:00 AM
3/16/2022	03/13/2022	3/13/2022 12:00:00 AM
4/1/2022	03/27/2022	3/27/2022 12:00:00 AM
5/7/2022	05/01/2022	5/1/2022 12:00:00 AM
5/16/2022	05/15/2022	5/15/2022 12:00:00 AM
6/15/2022	06/12/2022	6/12/2022 12:00:00 AM
6/26/2022	06/26/2022	6/26/2022 12:00:00 AM
7/9/2022	07/03/2022	7/3/2022 12:00:00 AM
7/22/2022	07/17/2022	7/17/2022 12:00:00 AM
7/23/2022	07/17/2022	7/17/2022 12:00:00 AM
7/27/2022	07/24/2022	7/24/2022 12:00:00 AM
8/2/2022	07/31/2022	7/31/2022 12:00:00 AM
8/8/2022	08/07/2022	8/7/2022 12:00:00 AM
8/19/2022	08/14/2022	8/14/2022 12:00:00 AM
9/26/2022	09/25/2022	9/25/2022 12:00:00 AM
10/14/2022	10/09/2022	10/9/2022 12:00:00 AM
10/29/2022	10/23/2022	10/23/2022 12:00:00 AM

Het veld `start_of_week` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `weekstart()` en geeft het datumveld door als het argument van de functie.

De `weekstart()`-functie identificeert aanvankelijk in welke week de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van die week.



Diagram van `weekstart()`-functie, voorbeeld zonder aanvullende argumenten



Transactie 8191 vond plaats op 5 februari. De systeemvariabele `Firstweekday` stelt de eerste dag van de week in op een zondag. De `weekstart()`-functie identificeert dat de eerste zaterdag vóór 5 februari, en daarom het begin van de week, op 30 februari viel. Daarom retourneert de `start_of_week`-waarde voor die transactie de eerste milliseconde van die dag, die op 30 januari om 12:00:00 AM was.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het maken van een veld, `previous_week_start`, dat de tijdstempel voor het begin van het kwartaal retourneert voordat de transactie plaatsvond.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
  *
  weekstart(date,-1) as previous_week_start,
  timestamp(weekstart(date,-1)) as previous_week_start_timestamp
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
```

```
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- previous\_week\_start
- previous\_week\_start\_timestamp

Resultatentabel

date	previous_week_start	previous_week_start_timestamp
1/7/2022	12/26/2021	12/26/2021 12:00:00 AM
1/19/2022	01/09/2022	1/9/2022 12:00:00 AM
2/5/2022	01/23/2022	1/23/2022 12:00:00 AM
2/28/2022	02/20/2022	2/20/2022 12:00:00 AM
3/16/2022	03/06/2022	3/6/2022 12:00:00 AM
4/1/2022	03/20/2022	3/20/2022 12:00:00 AM
5/7/2022	04/24/2022	4/24/2022 12:00:00 AM
5/16/2022	05/08/2022	5/8/2022 12:00:00 AM
6/15/2022	06/05/2022	6/5/2022 12:00:00 AM
6/26/2022	06/19/2022	6/19/2022 12:00:00 AM
7/9/2022	06/26/2022	6/26/2022 12:00:00 AM
7/22/2022	07/10/2022	7/10/2022 12:00:00 AM
7/23/2022	07/10/2022	7/10/2022 12:00:00 AM
7/27/2022	07/17/2022	7/17/2022 12:00:00 AM
8/2/2022	07/24/2022	7/24/2022 12:00:00 AM
8/8/2022	07/31/2022	7/31/2022 12:00:00 AM
8/19/2022	08/07/2022	8/7/2022 12:00:00 AM

date	previous_week_start	previous_week_start_timestamp
9/26/2022	09/18/2022	9/18/2022 12:00:00 AM
10/14/2022	10/02/2022	10/2/2022 12:00:00 AM
10/29/2022	10/16/2022	10/16/2022 12:00:00 AM

In dit geval, omdat een `period_no` van -1 werd gebruikt als het offset-argument in de `weekstart()`-functie, identificeert de functie eerst de week waarin de transacties plaatsvinden. Het kijkt dan een week eerder en identificeert de eerste milliseconde van die week.

Diagram van `weekstart()`-functie, voorbeeld `period_no`



Transactie 8196 vond plaats op 15 juni. De `weekstart()`-functie identificeert dat de week op 12 juni begint. Daarom begon de vorige week op 5 juni om 12:00:00 AM; dit is de waarde die voor het `previous_week_start`-veld wordt geretourneerd.

### Voorbeeld 3 - `first_week_day`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld. Maar in dit voorbeeld moeten we dinsdag instellen als de eerste dag van de werkweek.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*,  
weekstart(date,0,1) as start_of_week,  
timestamp(weekstart(date,0,1)) as start_of_week_timestamp  
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- start\_of\_week
- start\_of\_week\_timestamp

Resultatentabel

date	start_of_week	start_of_week_timestamp
1/7/2022	01/04/2022	1/4/2022 12:00:00 AM
1/19/2022	01/18/2022	1/18/2022 12:00:00 AM
2/5/2022	02/01/2022	2/1/2022 12:00:00 AM
2/28/2022	02/22/2022	2/22/2022 12:00:00 AM
3/16/2022	03/15/2022	3/15/2022 12:00:00 AM
4/1/2022	03/29/2022	3/29/2022 12:00:00 AM
5/7/2022	05/03/2022	5/3/2022 12:00:00 AM
5/16/2022	05/10/2022	5/10/2022 12:00:00 AM
6/15/2022	06/14/2022	6/14/2022 12:00:00 AM
6/26/2022	06/21/2022	6/21/2022 12:00:00 AM
7/9/2022	07/05/2022	7/5/2022 12:00:00 AM

date	start_of_week	start_of_week_timestamp
7/22/2022	07/19/2022	7/19/2022 12:00:00 AM
7/23/2022	07/19/2022	7/19/2022 12:00:00 AM
7/27/2022	07/26/2022	7/26/2022 12:00:00 AM
8/2/2022	08/02/2022	8/2/2022 12:00:00 AM
8/8/2022	08/02/2022	8/2/2022 12:00:00 AM
8/19/2022	08/16/2022	8/16/2022 12:00:00 AM
9/26/2022	09/20/2022	9/20/2022 12:00:00 AM
10/14/2022	10/11/2022	10/11/2022 12:00:00 AM
10/29/2022	10/25/2022	10/25/2022 12:00:00 AM

In dit geval geldt dat omdat het `first_week_date`-argument 1 wordt gebruikt in de `weekstart()`-functie, dinsdag als eerste dag van de week wordt ingesteld.

Diagram van de `weekstart()`-functie, voorbeeld met `first_week_day`



Transactie 8191 vond plaats op 5 februari. De `weekstart()`-functie identificeert dat de eerste dinsdag vóór deze datum, en daarom het begin van de week en geretourneerde waarde, op 1 februari om 12:00:00 AM viel.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die een tijdstempel retourneert voor het begin van de week waarin de transacties plaatsvonden, wordt gemaakt als meting in een diagramobject van de toepassing.

#### Load-script

Transactions:

Load

```
*
Inline
[
id,date,amount
8188,1/7/2022,17.17
8189,1/19/2022,37.23
8190,2/28/2022,88.27
8191,2/5/2022,57.42
8192,3/16/2022,53.80
8193,4/1/2022,82.06
8194,5/7/2022,40.39
8195,5/16/2022,87.21
8196,6/15/2022,95.93
8197,6/26/2022,45.89
8198,7/9/2022,36.23
8199,7/22/2022,25.66
8200,7/23/2022,82.77
8201,7/27/2022,69.98
8202,8/2/2022,76.11
8203,8/8/2022,25.12
8204,8/19/2022,46.23
8205,9/26/2022,84.21
8206,10/14/2022,96.24
8207,10/29/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Voeg de volgende metingen toe om het begin van de week waarin een transactie plaatsvindt te berekenen:

- =weekstart(date)
- =timestamp(weekstart(date))

Resultatentabel

date	start_of_week	start_of_week_timestamp
1/7/2022	01/02/2022	1/2/2022 12:00:00 AM
1/19/2022	01/16/2022	1/16/2022 12:00:00 AM
2/5/2022	01/30/2022	1/30/2022 12:00:00 AM
2/28/2022	02/27/2022	2/27/2022 12:00:00 AM
3/16/2022	03/13/2022	3/13/2022 12:00:00 AM
4/1/2022	03/27/2022	3/27/2022 12:00:00 AM
5/7/2022	05/01/2022	5/1/2022 12:00:00 AM
5/16/2022	05/15/2022	5/15/2022 12:00:00 AM

date	start_of_week	start_of_week_timestamp
6/15/2022	06/12/2022	6/12/2022 12:00:00 AM
6/26/2022	06/26/2022	6/26/2022 12:00:00 AM
7/9/2022	07/03/2022	7/3/2022 12:00:00 AM
7/22/2022	07/17/2022	7/17/2022 12:00:00 AM
7/23/2022	07/17/2022	7/17/2022 12:00:00 AM
7/27/2022	07/24/2022	7/24/2022 12:00:00 AM
8/2/2022	07/31/2022	7/31/2022 12:00:00 AM
8/8/2022	08/07/2022	8/7/2022 12:00:00 AM
8/19/2022	08/14/2022	8/14/2022 12:00:00 AM
9/26/2022	09/25/2022	9/25/2022 12:00:00 AM
10/14/2022	10/09/2022	10/9/2022 12:00:00 AM
10/29/2022	10/23/2022	10/23/2022 12:00:00 AM

De meting `start_of_week` wordt gemaakt in het diagramobject met behulp van de functie `weekstart()` en geeft het veld `date` door als het argument van de functie.

De `weekstart()`-functie identificeert aanvankelijk in welke week de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van die week.

*Diagram van `weekstart()`-functie, diagramobjectvoorbeeld*



Transactie 8191 vond plaats op 5 februari. De systeemvariabele `FirstweekDay` stelt de eerste dag van de week in op een zondag. De `weekstart()`-functie identificeert dat de eerste zondag vóór 5 februari, en daarom het begin van de week, op 30 januari viel. De `start_of_week`-waarde voor die transactie retourneert daarom de eerste milliseconde van die dag, 30 januari om 12:00:00 AM.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset die wordt geladen in een tabel met de naam `Payroll`.
- Gegevens omvatten werknemers-id's, werknemersnamen en het gemiddelde dagsalaris dat elke werknemer verdient.

Werknemers beginnen hun werk op maandag en werken zes dagen per week. De `Firstweekday`-systeemvariabele mag niet worden aangepast.

De eindgebruiker wil een diagramobject dat de salarissen weergeeft op werknemers-id en werknemersnaam, die in de week tot dusver zijn verdiend.

### Load-script

```
Payroll:
Load
*
Inline
[
employee_id,employee_name,day_rate
182,Mark, $150
183,Deryck, $125
184,Dexter, $125
185,Sydney,$270
186,Agatha,$128
];
```

### Resultaten

#### Doe het volgende:

1. Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:
  - `employee_id`
  - `employee_name`
2. Maak vervolgens een meting om de salarissen te berekenen die in de week tot dusver zijn verdiend:  
`=if(today(1)-weekstart(today(1),0,0)<7,(today(1)-weekstart(today(1),0,0))*day_rate,day_rate*6)`
3. Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

<code>employee_id</code>	<code>employee_name</code>	<code>=if(today(1)-weekstart(today(1),0,0)&lt;7,(today(1)-weekstart(today(1),0,0))*day_rate,day_rate*6)</code>
182	Mark	\$600.00
183	Deryck	\$500.00
184	Dexter	\$500.00
185	Sydney	\$1080.00
186	Agatha	\$512.00



De `weekstart()`-functie die door de datum van vandaag als het eerste argument en 0 als het derde argument te gebruiken, maandag instelt als de eerste dag van de week en de begindatum van de huidige week retourneert. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking vervolgens het aantal dagen dat in deze week tot nu toe is verstreken.

De voorwaarde bepaalt vervolgens of er meer dan zes dagen deze week zijn geweest. Als dat zo is dan wordt het `day_rate` van de werknemer vermenigvuldigd met 6 dagen. Anders wordt het `day_rate` vermenigvuldigd met het aantal dagen tot nu toe deze week.

### weekyear

Deze functie retourneert het jaar waarbij het weeknummer hoort conform ISO 8601. Het weeknummer varieert van 1 tot ongeveer 52.

#### Syntaxis:

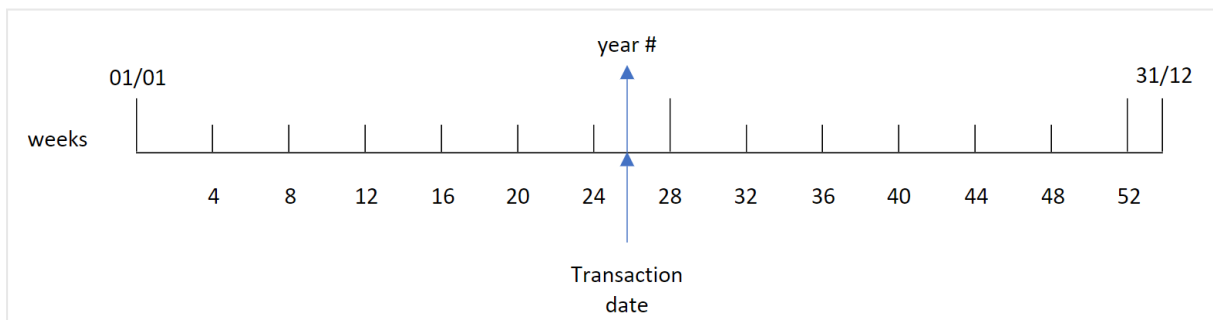
```
weekyear (expression)
```

**Retourgegevenstypen:** geheel getal

De `weekyear()`-functie bepaalt welke week van een jaar de datum in valt. Hij retourneert vervolgens het jaar dat bij dat weeknummer hoort.

Qlik-apps maken standaard gebruik van opgesplitste weken (gedefinieerd door de systeemvariabele `BrokenWeeks`) en weeknummer 1 begint op 1 januari en het jaar eindigt na week 52. Daarom retourneert de `weekyear()`-functie altijd dezelfde waarde als de `week()`-functie wanneer de applicatie opgesplitste weken gebruikt.

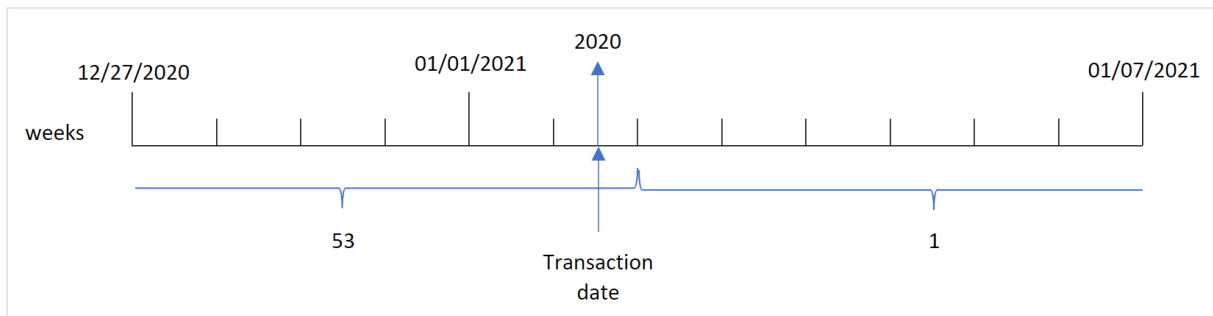
*Diagram van het bereik van de weekyear()-functie*



Maar als de `BrokenWeeks`-systeemvariabele is ingesteld op gebruik van niet-opgesplitste weken, mag week 1 alleen een bepaald aantal dagen in januari omvatten op basis van de waarde die is gespecificeerd in de `ReferenceDay`-systeemvariabele.

Als bijvoorbeeld een `ReferenceDay`-waarde van 4 wordt gebruikt, moet week 1 ten minste vier dagen in januari omvatten. Het is mogelijk dat week 1 datums in december van het voorgaande jaar bevat of dat het laatste weeknummer van een jaar datums in januari van het volgende jaar bevat. In situaties zoals deze, retourneert de `weekyear()`-functie een andere waarde dan de `year()`-functie.

Diagram van het bereik van de `weekyear()`-functie bij gebruik van niet-opgesplitste weken



### Wanneer gebruiken

De `weekyear()`-functie is handig als u aggregaties per jaar wilt vergelijken. Bijvoorbeeld als u de totale verkoop van producten per jaar wilt zien. De `weekyear()`-functie wordt verkozen boven de `year()`-functie wanneer de gebruiker consistent wil blijven met de `brokenweeks`-systeemvariabele in de app.

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>weekyear('12/30/1996')</code>	Retourneert 1997, omdat week 1 van 1997 begint op 12/30/1996
<code>weekyear('01/02/1997')</code>	retourneert 1997
<code>weekyear('12/28/1997')</code>	retourneert 1997
<code>weekyear('12/30/1997')</code>	Retourneert 1998, omdat week 1 van 1998 begint op 12/29/1997
<code>weekyear('01/02/1999')</code>	Retourneert 1998, omdat week 53 van 1998 eindigt op 01/03/1999

### Gerelateerde onderwerpen

Onderwerp	Interactie
<i>week (page 1049)</i>	Retourneert een geheel getal dat het weeknummer representeert conform ISO 8601
<i>year (page 1122)</i>	Retourneert een geheel getal dat het jaar representeert als de uitdrukking wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

### Voorbeeld 1 - Opgesplitste weken

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor de laatste week van 2020 en de eerste week van 2021 wordt geladen in de tabel `Transactions`.
- De variabele `BrokenWeeks` is ingesteld op 1.
- Een voorgaande lading die het volgende bevat:
  - De `weekyear()`-functie die is ingesteld als het veld, `week_year`, dat het jaar retourneert waarin de transacties hebben plaatsgevonden.
  - De `week()`-functie, ingesteld als het veld `week`, dat het weeknummer toont van elke transactiedatum.

#### Load-script

```
SET BrokenWeeks=1;
```

```
Transactions:
```

```
  Load
  *,
  week(date) as week,
  weekyear(date) as week_year
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8176,12/28/2020,19.42
```

```
8177,12/29/2020,23.80
```

```
8178,12/30/2020,82.06
```

```
8179,12/31/2020,40.56
```

```
8180,01/01/2021,37.23
```

```
8181,01/02/2021,17.17
```

```
8182,01/03/2021,88.27
```

```
8183,01/04/2021,57.42
8184,01/05/2021,67.42
8185,01/06/2021,23.80
8186,01/07/2021,82.06
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- week
- week\_year

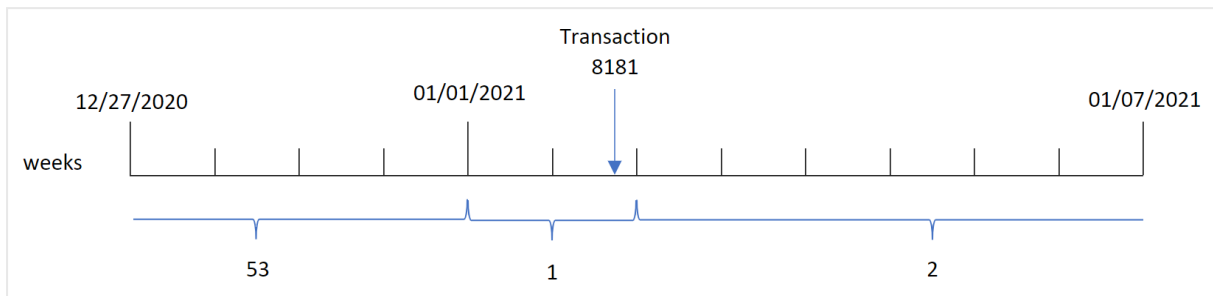
Resultatentabel

id	date	week	week_year
8176	12/28/2020	53	2020
8177	12/29/2020	53	2020
8178	12/30/2020	53	2020
8179	12/31/2020	53	2020
8180	01/01/2021	1	2021
8181	01/02/2021	1	2021
8182	01/03/2021	2	2021
8183	01/04/2021	2	2021
8184	01/05/2021	2	2021
8185	01/06/2021	2	2021
8186	01/07/2021	2	2021

Het veld `week_year` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `weekyear` () en geeft het datumveld door als het argument van de functie.

De systeemvariabele `brokenweeks` is ingesteld op 1 wat inhoudt dat de app opgesplitste weken gebruikt. Week 1 begint op 1 januari.

Diagram van het bereik van de `weekyear()`-functie bij gebruik van opgesplitste weken



Transactie 8181 vindt plaats op 2 januari. Deze datum valt in week 1. Daarom wordt de waarde 2021 geretourneerd voor het veld `week_year`.

### Voorbeeld 2 - Niet-opgesplitste weken

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor de laatste week van 2020 en de eerste week van 2021 wordt geladen in de tabel `Transactions`.
- De `brokenweeks`-variabele is ingesteld op 0.
- Een voorgaande lading die het volgende bevat:
  - De `weekyear()`-functie die is ingesteld als het veld, `week_year`, dat het jaar retourneert waarin de transacties hebben plaatsgevonden.
  - De `week()`-functie, ingesteld als het veld `week`, dat het weeknummer toont van elke transactiedatum.

Maar in dit voorbeeld is het bedrijfsbeleid om niet-opgesplitste weken te gebruiken.

#### Load-script

```
SET BrokenWeeks=0;
```

```
Transactions:
```

```
  Load
  *,
  week(date) as week,
  weekyear(date) as week_year
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8176,12/28/2020,19.42
```

```
8177,12/29/2020,23.80
8178,12/30/2020,82.06
8179,12/31/2020,40.56
8180,01/01/2021,37.23
8181,01/02/2021,17.17
8182,01/03/2021,88.27
8183,01/04/2021,57.42
8184,01/05/2021,67.42
8185,01/06/2021,23.80
8186,01/07/2021,82.06
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- week
- week\_year

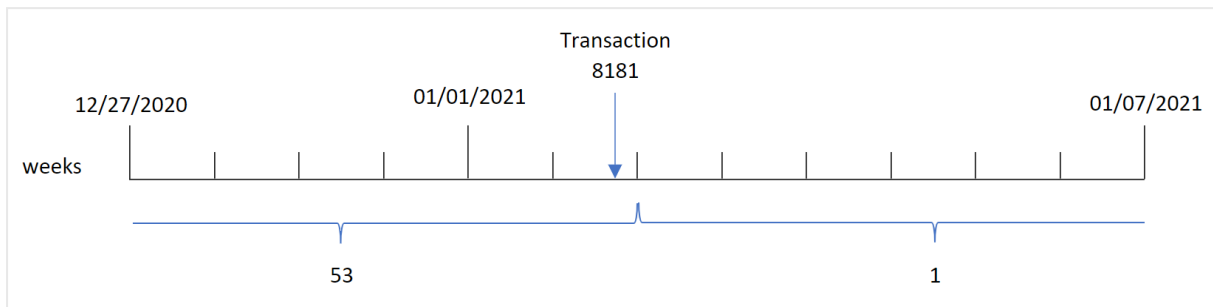
Resultatentabel

id	date	week	week_year
8176	12/28/2020	53	2020
8177	12/29/2020	53	2020
8178	12/30/2020	53	2020
8179	12/31/2020	53	2020
8180	01/01/2021	53	2020
8181	01/02/2021	53	2020
8182	01/03/2021	1	2021
8183	01/04/2021	1	2021
8184	01/05/2021	1	2021
8185	01/06/2021	1	2021
8186	01/07/2021	1	2021

De systeemvariabele `brokenweeks` is ingesteld op 0 wat inhoudt dat de applicatie niet-opgesplitste weken gebruikt. Daarom hoeft week 1 niet te beginnen op 1 januari.

Week 53 van 2020 gaat verder tot het eind van 2 januari 2021, waarbij week 1 van 2020 op zondag, 3 januari 2021 valt.

Diagram van het bereik van de `weekyear()`-functie bij gebruik van niet-opgesplitste weken



Transactie 8181 vindt plaats op 2 januari. Deze datum valt in week 1. Daarom wordt de waarde 2021 geretourneerd voor het veld `week_year`.

### Voorbeeld 3 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die het jaarnummer van de week retourneert maand waarin de transacties plaatsvonden is gemaakt als meting in een diagramobject van de toepassing.

#### Load-script

```
SET BrokenWeeks=1;
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8176,12/28/2020,19.42
```

```
8177,12/29/2020,23.80
```

```
8178,12/30/2020,82.06
```

```
8179,12/31/2020,40.56
```

```
8180,01/01/2021,37.23
```

```
8181,01/02/2021,17.17
```

```
8182,01/03/2021,88.27
```

```
8183,01/04/2021,57.42
```

```
8184,01/05/2021,67.42
```

```
8185,01/06/2021,23.80
```

```
8186,01/07/2021,82.06
```

```
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

## 5 Script- en diagramfuncties

- id
- date

Maak de volgende meting om de week waarin een transactie plaatsvindt te berekenen:

- =week(date)

Maak de volgende meting om het jaar waarin een transactie plaatsvindt te berekenen op basis van het weeknummer:

- =weekyear(date)

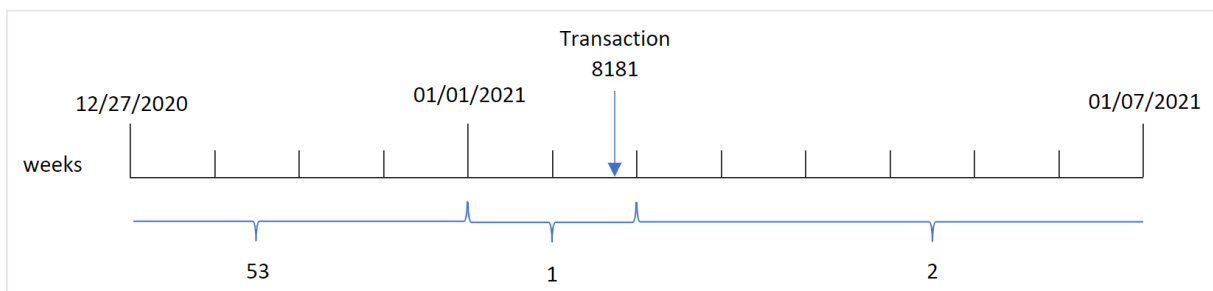
Resultatentabel

id	date	week	week_year
8176	12/28/2020	53	2020
8177	12/29/2020	53	2020
8178	12/30/2020	53	2020
8179	12/31/2020	53	2020
8180	01/01/2021	1	2021
8181	01/02/2021	1	2021
8182	01/03/2021	2	2021
8183	01/04/2021	2	2021
8184	01/05/2021	2	2021
8185	01/06/2021	2	2021
8186	01/07/2021	2	2021

Het veld week\_year wordt gemaakt in de voorafgaande load-instructie met behulp van de functie weekyear () en geeft het datumveld door als het argument van de functie.

De systeemvariabele brokenweeks is ingesteld op 1 wat inhoudt dat de app opgesplitste weken gebruikt. Week 1 begint op 1 januari.

*Diagram van het bereik van de weekyear()-functie bij gebruik van opgesplitste weken*





Transactie 8181 vindt plaats op 2 januari. Deze datum valt in week 1. Daarom wordt de waarde 2021 geretourneerd voor het veld `week_year`.

### Voorbeeld 4 - Scenario

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties voor de laatste week van 2020 en de eerste week van 2021 wordt geladen in de tabel `Transactions`.
- De `BrokenWeeks`-variabele is ingesteld op 0. Dit betekent dat de app niet-opgesplitste weken gebruikt.
- De `ReferenceDay`-variabele is ingesteld op 2. Dit betekent dat het jaar op 2 januari begint en ten minste twee dagen in januari bevat.
- De `FirstWeekDay`-variabele is ingesteld op 1. Dit betekent dat dinsdag de eerste dag van de week is.

Het bedrijfsbeleid is om opgesplitste weken te gebruiken. De eindgebruiker wil graag een diagram dat de totale verkoop per jaar weergeeft. De app gebruikt niet-opgesplitste weken waarbij week 1 ten minste twee dagen in januari bevat.

#### Load-script

```
SET BrokenWeeks=0;
SET ReferenceDay=2;
SET FirstWeekDay=1;
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8176,12/28/2020,19.42
```

```
8177,12/29/2020,23.80
```

```
8178,12/30/2020,82.06
```

```
8179,12/31/2020,40.56
```

```
8180,01/01/2021,37.23
```

```
8181,01/02/2021,17.17
```

```
8182,01/03/2021,88.27
```

```
8183,01/04/2021,57.42
```

```
8184,01/05/2021,67.42
```

```
8185,01/06/2021,23.80
```

```
8186,01/07/2021,82.06
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel.

Maak de volgende meting om het jaar waarin een transactie plaatsvindt te berekenen op basis van het weeknummer:

- `=weekyear(date)`

Maak de volgende meting om de totale verkoop te berekenen:

- `sum(amount)`

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

<code>weekyear(date)</code>	<code>=sum(amount)</code>
2020	19.42
2021	373.37

### year

Deze functie retourneert een geheel getal dat het jaar representeert als de **expression** wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

#### Syntaxis:

```
year (expression)
```

**Retourgegevenstypen:** geheel getal

De `year()`-functie is beschikbaar zowel als een script- als een diagramfunctie. De functie retourneert het jaar voor een bepaalde datum. Dit wordt vaak gebruikt om een jaarveld te maken als een dimensie in een hoofdagenda.

### Wanneer gebruiken

De `year()`-functie is handig als u aggregaties per jaar wilt vergelijken. De functie kan bijvoorbeeld worden gebruikt om de totale verkoop van producten per jaar te bekijken.

Deze dimensies kunnen in het load-script worden gemaakt door de functie te gebruiken om een veld in een masterkalendertabel te maken. Hij zou ook rechtstreeks in een diagram kunnen worden gebruikt als een berekende dimensie.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>year( '2012-10-12' )</code>	retourneert 2012
<code>year( '35648' )</code>	retourneert 1997, omdat 35648 = 1997-08-06

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - gegevensverzameling DateFormat (script)

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums die wordt geladen in de tabel `Master_Calendar`.
- De standaard `DateFormat`-systeemvariabele (`mm/dd/yyyy`) wordt gebruikt.
- Een voorafgaande lading die wordt gebruikt om een extra veld te maken met de naam `year` met behulp van de `year()`-functie.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Master_Calendar:
```

```
    Load
        date,
        year(date) as year
    ;
Load
date
inline
[
date
12/28/2020
12/29/2020
12/30/2020
12/31/2020
```

```
01/01/2021
01/02/2021
01/03/2021
01/04/2021
01/05/2021
01/06/2021
01/07/2021
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- year

Resultatentabel

date	jaar
12/28/2020	2020
12/29/2020	2020
12/30/2020	2020
12/31/2020	2020
01/01/2021	2021
01/02/2021	2021
01/03/2021	2021
01/04/2021	2021
01/05/2021	2021
01/06/2021	2021
01/07/2021	2021

### Voorbeeld 2 - ANSI-datums

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums die wordt geladen in de tabel `Master Calendar`.
- De standaard `DateFormat`-systeemvariabele (`mm/dd/yyyy`) wordt gebruikt. De datums die zijn opgenomen in de gegevensverzameling hebben echter een standaard ANSI-datumnotatie.

- Een voorafgaande lading die wordt gebruikt om een extra veld te maken met de naam year met behulp van de year()-functie.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Master_Calendar:
```

```
    Load
        date,
        year(date) as year
    ;
```

```
Load
date
InLine
[
date
2020-12-28
2020-12-29
2020-12-30
2020-12-31
2021-01-01
2021-01-02
2021-01-03
2021-01-04
2021-01-05
2021-01-06
2021-01-07
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- year

Resultatentabel

date	jaar
2020-12-28	2020
2020-12-29	2020
2020-12-30	2020
2020-12-31	2020
2021-01-01	2021
2021-01-02	2021
2021-01-03	2021

date	jaar
2021-01-04	2021
2021-01-05	2021
2021-01-06	2021
2021-01-07	2021

### Voorbeeld 3 - Datums zonder notatie

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met datums in numerieke indeling, die wordt geladen in de tabel `Master Calendar`.
- De standaard `DateFormat`-systeemvariabele (`mm/dd/yyyy`) wordt gebruikt.
- Een voorafgaande lading die wordt gebruikt om een extra veld te maken met de naam `year` met behulp van de `year()`-functie.

De oorspronkelijke, niet-opgemaakte gegevens worden geladen, met de naam `unformatted_date`. Om helderheid te verschaffen, wordt het extra veld `long_date` gebruikt om de numerieke gegevens om te zetten in een opgemaakt datumveld met behulp van de `date()`-functie.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Master_Calendar:
```

```
  Load
    unformatted_date,
    date(unformatted_date) as long_date,
    year(unformatted_date) as year
  ;
```

```
Load
unformatted_date
Inline
[
unformatted_date
44868
44898
44928
44958
44988
45018
45048
```

```
45078
45008
45038
45068
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- unformatted\_date
- long\_date
- year

Resultatentabel

unformatted_date	long_date	jaar
44868	11/03/2022	2022
44898	12/03/2022	2022
44928	01/02/2023	2023
44958	02/01/2023	2023
44988	03/03/2023	2023
45008	03/23/2023	2023
45018	04/02/2023	2023
45038	04/22/2023	2023
45048	05/02/2023	2023
45068	05/22/2023	2023
45078	06/01/2023	2023

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

In dit voorbeeld wordt een gegevensset met bestellingen geladen in de tabel Verkoop. De tabel bevat drie velden.

- id
- sales\_date
- amount

Garanties bij verkochte producten in de afgelopen twee jaar vanaf de verkoopdatum. Het is de taak om een meting in een diagram te maken om het jaar vast te stellen waarin iedere garantie zal verlopen.

### Load-script

```
sales:
Load
id,
sales_date,
amount
Inline
[
id,sales_date,amount
1,12/28/2020,231.24,
2,12/29/2020,567.28,
3,12/30/2020,364.28,
4,12/31/2020,575.76,
5,01/01/2021,638.68,
6,01/02/2021,785.38,
7,01/03/2021,967.46,
8,01/04/2021,287.67
9,01/05/2021,764.45,
10,01/06/2021,875.43,
11,01/07/2021,957.35
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: sales\_date.

Maak de volgende meting:

```
=year(sales_date+365*2)
```

Resultatentabel

sales_date	=year(sales_date+365*2)
12/28/2020	2022
12/29/2020	2022
12/30/2020	2022
12/31/2020	2022
01/01/2021	2023
01/02/2021	2023
01/03/2021	2023
01/04/2021	2023



sales_date	=year(sales_date+365*2)
01/05/2021	2023
01/06/2021	2023
01/07/2021	2023

De resultaten van deze meting zijn in de bovenstaande tabel te zien. Om twee jaar bij de datum op te tellen, vermenigvuldigt u 365 met 2 en voegt u het resultaat toe aan de verkoopdatum. De verkoop die in 2020 plaatsvond, heeft een vervvaljaar van 2022.

### yearend

Deze functie retourneert een waarde die overeenkomt met een tijdstempel van de laatste milliseconde van de laatste dag van het jaar dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

#### Syntaxis:

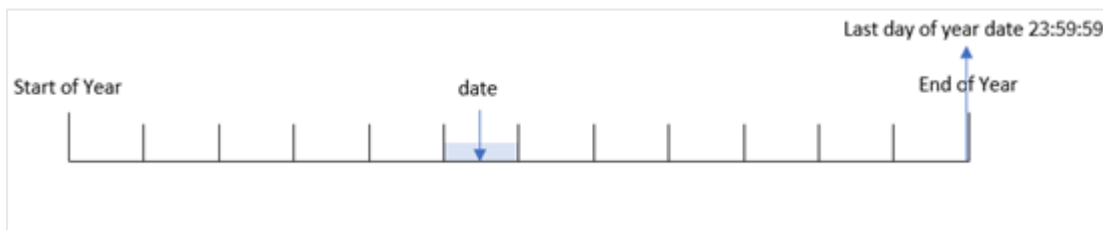
```
YearEnd( date[, period_no[, first_month_of_year = 1]])
```

Met andere woorden, de functie `yearend()` bepaalt in welk jaar de datum valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de laatste milliseconde van dat jaar. De eerste maand van het jaar is standaard januari. U kunt echter ook wijzigen welke maand als eerste wordt ingesteld met behulp van het argument `first_month_of_year` in de functie `yearend()` te gebruiken.



*De functie `yearend()` houdt geen rekening met de systeemvariabele `FirstMonthOfYear`. Het jaar begint op 1 januari, tenzij het argument `first_month_of_year` wordt gebruikt om het te wijzigen.*

Diagram van `yearend()`-functie.



#### Wanneer gebruiken

De functie `yearend()` wordt gebruikt als onderdeel van een uitdrukking als u wilt dat in de berekening een deel van het jaar wordt gebruikt dat nog niet heeft plaatsgevonden. Bijvoorbeeld als u de totale rente wilt berekenen die nog niet is berekend gedurende het jaar.

Retourgegevenstypen: dual

### Argumenten

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal, waarbij de waarde 0 het jaar aangeeft dat <b>date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorgaande jaren aan en positieve waarden geven volgende jaren aan.
<b>first_ month_of_ year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> .

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden `first_month_of_year`

Month	Waarde
Februari	2
Maart	3
April	4
May	5
Juni	6
Juli	7
Augustus	8
September	9
Oktober	10
November	11
December	12

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeelden van functies

Voorbeeld	Resultaat
<code>yearend('10/19/2001')</code>	Returns 12/31/2001 23:59:59.
<code>yearend('10/19/2001', -1)</code>	Returns 12/31/2000 23:59:59.
<code>yearend('10/19/2001', 0, 4)</code>	Returns 03/31/2002 23:59:59.

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 wordt geladen in een tabel met de naam 'Transactions'.
- Het datumveld is opgegeven in de systeemvariabele `DateFormat`, indeling (MM/DD/YYYY).
- Een voorgaande load-instructie die het volgende bevat:
  - Functie `yearend()` die is ingesteld als het veld `year_end`.
  - Functie `timestamp()` die is ingesteld als het veld `year_end_timestamp`.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    yearend(date) as year_end,
    timestamp(yearend(date)) as year_end_timestamp
  ;
Load
*
Inline
[
id,date,amount
8188,01/13/2020,37.23
8189,02/26/2020,17.17
```

```
8190,03/27/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,06/06/2022,46.23
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- year\_end
- year\_end\_timestamp

Resultatentabel

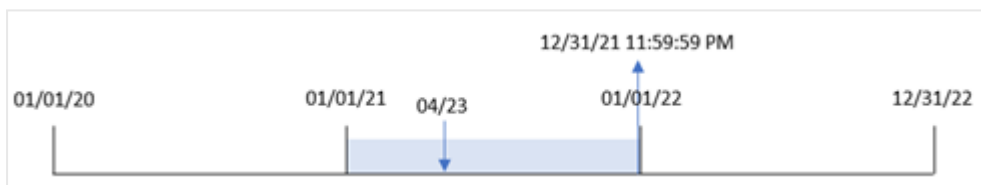
id	date	year_end	year_end_timestamp
8188	01/13/2020	12/31/2020	12/31/2020 11:59:59 P.M
8189	02/26/2020	12/31/2020	12/31/2020 11:59:59 P.M
8190	03/27/2020	12/31/2020	12/31/2020 11:59:59 P.M
8191	04/16/2020	12/31/2020	12/31/2020 11:59:59 P.M
8192	05/21/2020	12/31/2020	12/31/2020 11:59:59 P.M
8193	08/14/2020	12/31/2020	12/31/2020 11:59:59 P.M
8194	10/07/2020	12/31/2020	12/31/2020 11:59:59 P.M
8195	12/05/2020	12/31/2020	12/31/2020 11:59:59 P.M
8196	01/22/2021	12/31/2021	12/31/2021 11:59:59 PM
8197	02/03/2021	12/31/2021	12/31/2021 11:59:59 PM
8198	03/17/2021	12/31/2021	12/31/2021 11:59:59 PM
8199	04/23/2021	12/31/2021	12/31/2021 11:59:59 PM

id	date	year_end	year_end_timestamp
8200	05/04/2021	12/31/2021	12/31/2021 11:59:59 PM
8201	06/30/2021	12/31/2021	12/31/2021 11:59:59 PM
8202	07/26/2021	12/31/2021	12/31/2021 11:59:59 PM
8203	12/27/2021	12/31/2021	12/31/2021 11:59:59 PM
8204	06/06/2022	12/31/2022	12/31/2022 11:59:59 PM
8205	07/18/2022	12/31/2022	12/31/2022 11:59:59 PM
8206	11/14/2022	12/31/2022	12/31/2022 11:59:59 PM
8207	12/12/2022	12/31/2022	12/31/2022 11:59:59 PM

Het veld 'year\_end' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `yearend()` en geeft het datumveld door als het argument van de functie.

De functie `yearend()` identificeert aanvankelijk in welk jaar de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van dat jaar.

*Diagram van functie `yearend()` met transactie 8199 geselecteerd.*



Transactie 8199 vond plaats op 23 april 2021. De functie `yearend()` retourneert de laatste milliseconde van dat jaar, namelijk 31 december om 11:59:59 P.M.

### Voorbeeld 2 - `period_no`

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het echter de taak om een veld te maken, 'previous\_year\_end', dat de tijdstempel van de einddatum van het jaar voorafgaand aan het jaar waarin de transactie plaatsvond, retourneert.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*,
yearend(date,-1) as previous_year_end,
```

```
timestamp(yearend(date,-1)) as previous_year_end_timestamp
;
Load
*
Inline
[
id,date,amount
8188,01/13/2020,37.23
8189,02/26/2020,17.17
8190,03/27/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,06/06/2022,46.23
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- previous\_year\_end
- previous\_year\_end\_timestamp

Resultatentabel

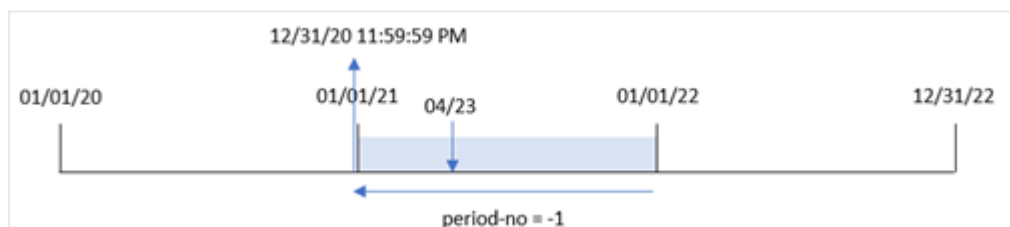
id	date	previous_year_end	previous_year_end_timestamp
8188	01/13/2020	12/31/2019	12/31/2019 11:59:59 PM
8189	02/26/2020	12/31/2019	12/31/2019 11:59:59 PM
8190	03/27/2020	12/31/2019	12/31/2019 11:59:59 PM
8191	04/16/2020	12/31/2019	12/31/2019 11:59:59 PM
8192	05/21/2020	12/31/2019	12/31/2019 11:59:59 PM
8193	08/14/2020	12/31/2019	12/31/2019 11:59:59 PM

## 5 Script- en diagramfuncties

id	date	previous_year_end	previous_year_end_timestamp
8194	10/07/2020	12/31/2019	12/31/2019 11:59:59 PM
8195	12/05/2020	12/31/2019	12/31/2019 11:59:59 PM
8196	01/22/2021	12/31/2020	12/31/2020 11:59:59 P.M
8197	02/03/2021	12/31/2020	12/31/2020 11:59:59 P.M
8198	03/17/2021	12/31/2020	12/31/2020 11:59:59 P.M
8199	04/23/2021	12/31/2020	12/31/2020 11:59:59 P.M
8200	05/04/2021	12/31/2020	12/31/2020 11:59:59 P.M
8201	06/30/2021	12/31/2020	12/31/2020 11:59:59 P.M
8202	07/26/2021	12/31/2020	12/31/2020 11:59:59 P.M
8203	12/27/2021	12/31/2020	12/31/2020 11:59:59 P.M
8204	06/06/2022	12/31/2021	12/31/2021 11:59:59 PM
8205	07/18/2022	12/31/2021	12/31/2021 11:59:59 PM
8206	11/14/2022	12/31/2021	12/31/2021 11:59:59 PM
8207	12/12/2022	12/31/2021	12/31/2021 11:59:59 PM

Omdat een `period_no` van `-1` werd gebruikt als het `offset`-argument in de functie `yearend()`, identificeert de functie eerst het jaar waarin de transacties plaatsvinden. Het kijkt dan een jaar eerder en identificeert de laatste milliseconde van dat jaar.

Diagram van functie `yearend()` met een `period_no` van `-1`.



Transactie 8199 vindt plaats op 23 april 2021. De functie `yearend()` retourneert de laatste milliseconde van het voorgaande jaar, 31 december 2020 om 11:59:59 PM, voor het veld 'previous\_year\_end'.

### Voorbeeld 3 - first\_month\_of\_year

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het bedrijfsbeleid echter dat het jaar op 1 april begint.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    yearend(date,0,4) as year_end,
    timestamp(yearend(date,0,4)) as year_end_timestamp
  ;
Load
*
Inline
[
id,date,amount
8188,01/13/2020,37.23
8189,02/26/2020,17.17
8190,03/27/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,06/06/2022,46.23
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- year\_end
- year\_end\_timestamp

Resultatentabel

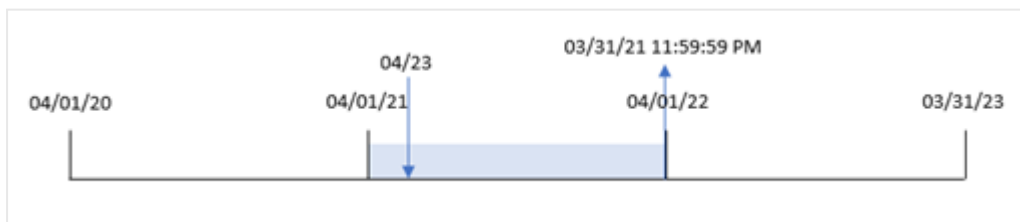
id	date	year_end	year_end_timestamp
8188	01/13/2020	03/31/2020	3/31/2020 11:59:59 PM



id	date	year_end	year_end_timestamp
8189	02/26/2020	03/31/2020	3/31/2020 11:59:59 PM
8190	03/27/2020	03/31/2020	3/31/2020 11:59:59 PM
8191	04/16/2020	03/31/2021	3/31/2021 11:59:59 PM
8192	05/21/2020	03/31/2021	3/31/2021 11:59:59 PM
8193	08/14/2020	03/31/2021	3/31/2021 11:59:59 PM
8194	10/07/2020	03/31/2021	3/31/2021 11:59:59 PM
8195	12/05/2020	03/31/2021	3/31/2021 11:59:59 PM
8196	01/22/2021	03/31/2021	3/31/2021 11:59:59 PM
8197	02/03/2021	03/31/2021	3/31/2021 11:59:59 PM
8198	03/17/2021	03/31/2021	3/31/2021 11:59:59 PM
8199	04/23/2021	03/31/2022	3/31/2022 11:59:59 PM
8200	05/04/2021	03/31/2022	3/31/2022 11:59:59 PM
8201	06/30/2021	03/31/2022	3/31/2022 11:59:59 PM
8202	07/26/2021	03/31/2022	3/31/2022 11:59:59 PM
8203	12/27/2021	03/31/2022	3/31/2022 11:59:59 PM
8204	06/06/2022	03/31/2023	3/31/2023 11:59:59 PM
8205	07/18/2022	03/31/2023	3/31/2023 11:59:59 PM
8206	11/14/2022	03/31/2023	3/31/2023 11:59:59 PM
8207	12/12/2022	03/31/2023	3/31/2023 11:59:59 PM

Omdat het argument `first_month_of_year` van 4 wordt gebruikt in de functie `yearend()`, stelt het de eerste dag van het jaar in op 1 april en de laatste dag van het jaar op 31 maart.

*Diagram van functie `yearend()` met april als de eerste maand van het jaar.*



Transactie 8199 vindt plaats op 23 april 2021. Omdat de functie `yearend()` het begin van het jaar instelt op 1 april, wordt 31 maart 2022 geretourneerd als de waarde 'year\_end' voor de transactie.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die een einddatumtijdstempel retourneert van het jaar waarin de transactie plaatsvond is gemaakt als meting in een diagramobject van de toepassing.

#### Load-script

Transactions:

Load

\*

Inline

[

id,date,amount

8188,01/13/2020,37.23

8189,02/26/2020,17.17

8190,03/27/2020,88.27

8191,04/16/2020,57.42

8192,05/21/2020,53.80

8193,08/14/2020,82.06

8194,10/07/2020,40.39

8195,12/05/2020,87.21

8196,01/22/2021,95.93

8197,02/03/2021,45.89

8198,03/17/2021,36.23

8199,04/23/2021,25.66

8200,05/04/2021,82.77

8201,06/30/2021,69.98

8202,07/26/2021,76.11

8203,12/27/2021,25.12

8204,06/06/2022,46.23

8205,07/18/2022,84.21

8206,11/14/2022,96.24

8207,12/12/2022,67.67

];

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date

Maak de volgende metingen om te berekenen in welk jaar een transactie plaatsvond:

- =yearend(date)
- =timestamp(yearend(date))

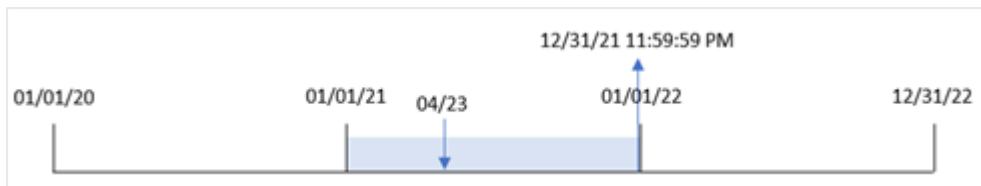
Resultatentabel

id	date	=yearend(date)	=timestamp(yearend(date))
8188	01/13/2020	12/31/2020	12/31/2020 11:59:59 P.M
8189	02/26/2020	12/31/2020	12/31/2020 11:59:59 P.M
8190	03/27/2020	12/31/2020	12/31/2020 11:59:59 P.M
8191	04/16/2020	12/31/2020	12/31/2020 11:59:59 P.M
8192	05/21/2020	12/31/2020	12/31/2020 11:59:59 P.M
8193	08/14/2020	12/31/2020	12/31/2020 11:59:59 P.M
8194	10/07/2020	12/31/2020	12/31/2020 11:59:59 P.M
8195	12/05/2020	12/31/2020	12/31/2020 11:59:59 P.M
8196	01/22/2021	12/31/2021	12/31/2021 11:59:59 PM
8197	02/03/2021	12/31/2021	12/31/2021 11:59:59 PM
8198	03/17/2021	12/31/2021	12/31/2021 11:59:59 PM
8199	04/23/2021	12/31/2021	12/31/2021 11:59:59 PM
8200	05/04/2021	12/31/2021	12/31/2021 11:59:59 PM
8201	06/30/2021	12/31/2021	12/31/2021 11:59:59 PM
8202	07/26/2021	12/31/2021	12/31/2021 11:59:59 PM
8203	12/27/2021	12/31/2021	12/31/2021 11:59:59 PM
8204	06/06/2022	12/31/2022	12/31/2022 11:59:59 PM
8205	07/18/2022	12/31/2022	12/31/2022 11:59:59 PM
8206	11/14/2022	12/31/2022	12/31/2022 11:59:59 PM
8207	12/12/2022	12/31/2022	12/31/2022 11:59:59 PM

De meting 'end\_of\_year' wordt gemaakt in het diagramobject met behulp van de functie yearend() en geeft het veld door als het argument van de functie.

De functie yearend() identificeert aanvankelijk in welk jaar de datumwaarde valt en retourneert een tijdstempel voor de laatste milliseconde van dat jaar.

Diagram van functie `yearend()` waaruit blijkt dat transactie 8199 in april heeft plaatsgevonden.



Transactie 8199 vindt plaats op 23 april 2021. De functie `yearend()` retourneert de laatste milliseconde van dat jaar, namelijk 31 december om 11:59:59 P.M.

### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset wordt geladen in een tabel met de naam 'Employee\_Expenses'. De tabel bevat de volgende velden:
  - werknemers-id's
  - naam werknemer
  - gemiddelde dagelijkse onkostendeclaraties van elke werknemer

De eindgebruiker wil graag een diagramobject dat per werknemer-id en werknemernaam de geschatte nog te maken onkostendeclaraties voor de rest van het jaar weergeeft. Het boekjaar begint in januari.

#### Load-script

```
Employee_Expenses :
Load
*
Inline
[
employee_id,employee_name,avg_daily_claim
182,Mark, $15
183,Deryck, $12.5
184,Dexter, $12.5
185,Sydney,$27
186,Agatha,$18
];
```

#### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- employee\_id
- employee\_name

Maak de volgende meting om de verwachte onkostendeclaraties te berekenen:

```
=(yearend(today(1))-today(1))*avg_daily_claim
```

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

employee_id	employee_name	=(yearend(today(1))-today(1))*avg_daily_claim
182	Mark	\$3240.00
183	Deryck	\$2700.00
184	Dexter	\$2700.00
185	Sydney	\$5832.00
186	Agatha	\$3888.00

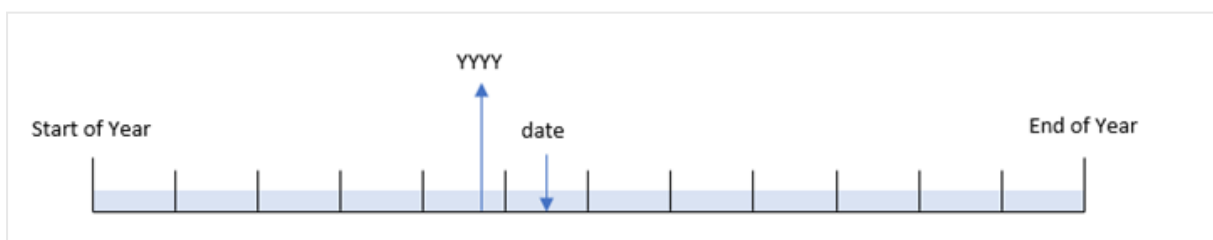
Door de datum van vandaag als enig argument te gebruiken, retourneert de functie `yearend()` de einddatum van het huidige jaar. Door vervolgens de datum van vandaag af te trekken van de einddatum van het jaar, retourneert de uitdrukking het aantal resterende dagen in dit jaar.

Deze waarde wordt vervolgens vermenigvuldigd met de gemiddelde dagelijkse onkostendeclaratie van elke werknemer om de geschatte waarde te berekenen van de declaraties die elke werknemer naar verwachting in het resterende jaar zal indienen.

### yearname

Deze functie retourneert een jaar in vier cijfers als weergavewaarde met een onderliggende numerieke waarde die overeenkomt met een tijdstempel van de eerste milliseconde van de eerste dag van het jaar dat **date** bevat.

*Schema van het tijdsbereik van de functie `yearname()`.*

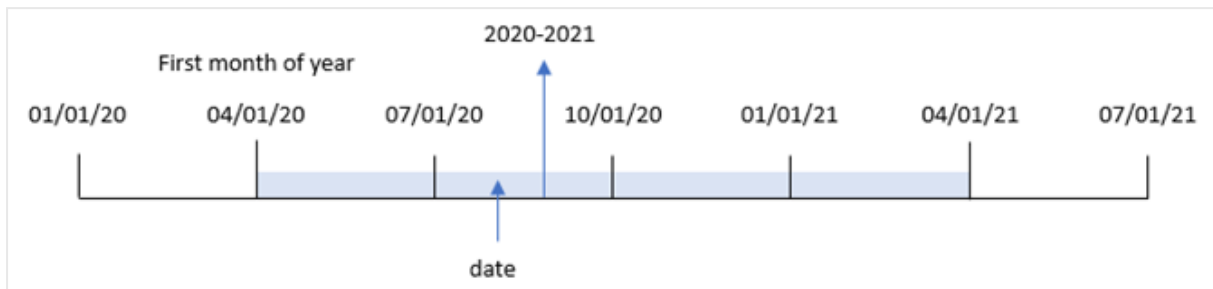


De functie `yearname()` is anders dan de functie `year()`, omdat u hiermee de datum die u wilt evalueren kunt verschuiven en de eerste maand van het jaar kunt instellen.

Als de eerste maand van het jaar niet januari is, retourneert de functie de twee viercijferige jaren over de periode van twaalf maanden die de datum bevatten. Als het begin van het jaar bijvoorbeeld april is en de datum die wordt geëvalueerd is 06/30/2020, zou het geretourneerde resultaat 2020-2021 zijn.

## 5 Script- en diagramfuncties

Diagram van functie `yearname()` met april ingesteld als de eerste maand van het jaar.



### Syntaxis:

```
YearName (date[, period_no[, first_month_of_year]] )
```

Retourgegevenstypen: dual

Argument	Beschrijving
<b>date</b>	De te evalueren datum of tijdstempel.
<b>period_no</b>	<b>period_no</b> is een geheel getal, waarbij de waarde 0 het jaar aangeeft dat <b>date</b> bevat. Negatieve waarden in <b>period_no</b> geven voorgaande jaren aan en positieve waarden geven volgende jaren aan.
<b>first_month_of_year</b>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <b>first_month_of_year</b> . De weergavewaarde is dan een tekenreeks die twee jaren toont.

U kunt de volgende waarden gebruiken om de eerste maand van het jaar in te stellen in het argument `first_month_of_year`:

waarden `first_month_of_year`

Month	Waarde
Februari	2
Maart	3
April	4
May	5
Juni	6
Juli	7
Augustus	8
September	9
Oktober	10
November	11
December	12

### Wanneer gebruiken

De functie `yearname()` is handig om aggregaties per jaar te vergelijken. Bijvoorbeeld als u de totale verkoop van producten per jaar wilt zien.

Deze dimensies kunnen in het load-script worden gemaakt door de functie te gebruiken om een veld in een masterkalendertabel te maken. Ze kunnen ook in een diagram worden gemaakt als berekende dimensies

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>yearname('10/19/2001')</code>	Returns '2001.'
<code>yearname('10/19/2001', -1)</code>	Retourneert '2000.'
<code>yearname('10/19/2001', 0, 4)</code>	Retourneert '2001-2002.'

Gerelateerde onderwerpen

Onderwerp	Beschrijving
<i>year (page 1122)</i>	Deze functie retourneert een geheel getal dat het jaar representeert als de uitdrukking wordt geïnterpreteerd als een datum volgens de standaardinterpretatie van getallen.

### Voorbeeld 1 - geen aanvullende argumenten

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 wordt geladen in een tabel met de naam 'Transactions'.
- De systeemvariabele `DateFormat` die is ingesteld op 'MM/DD/YYYY'.
- Een voorafgaande lading die de `yearname()` gebruikt en die is ingesteld als het veld `year_name`.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
  *,
  yearname(date) as year_name
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190,'03/27/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'12/27/2021',25.12
```

```
8204,'06/06/2022',46.23
```

```
8205,'07/18/2022',84.21
```

```
8206,'11/14/2022',96.24
```

```
8207,'12/12/2022',67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `date`
- `year_name`



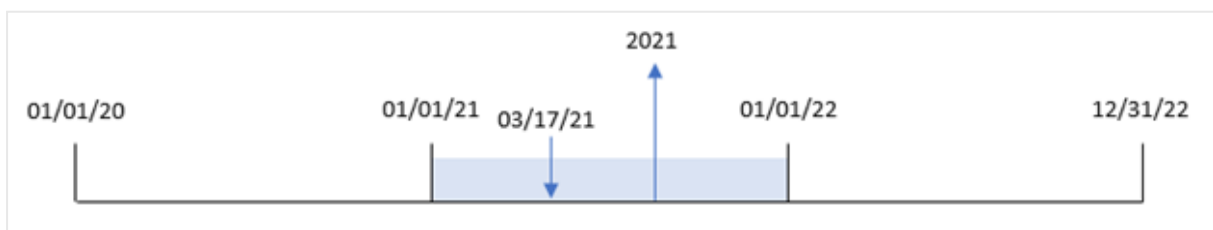
Resultatentabel

date	year_name
01/13/2020	2020
02/26/2020	2020
03/27/2020	2020
04/16/2020	2020
05/21/2020	2020
08/14/2020	2020
10/07/2020	2020
12/05/2020	2020
01/22/2021	2021
02/03/2021	2021
03/17/2021	2021
04/23/2021	2021
05/04/2021	2021
06/30/2021	2021
07/26/2021	2021
12/27/2021	2021
06/06/2022	2022
07/18/2022	2022
11/14/2022	2022
12/12/2022	2022

Het veld 'year\_name' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `yearname()` en geeft het datumveld door als het argument van de functie.

De functie `yearname()` identificeert in welk jaar de datumwaarde valt en retourneert dit als een viercijferige jaarwaarde.

*Diagram van de functie `yearname()` die 2021 als jaarwaarde weergeeft.*



### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 wordt geladen in een tabel met de naam 'Transacties'.
- De systeemvariabele DateFormat die is ingesteld op 'MM/DD/YYYY'.
- Een voorafgaande lading die de yearname() gebruikt en die is ingesteld als het veld year\_name.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    yearname(date,-1) as prior_year_name
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190,'03/27/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201,'06/30/2021',69.98
```

```
8202,'07/26/2021',76.11
```

```
8203,'12/27/2021',25.12
```

```
8204,'06/06/2022',46.23
```

```
8205,'07/18/2022',84.21
```

```
8206,'11/14/2022',96.24
```

```
8207,'12/12/2022',67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

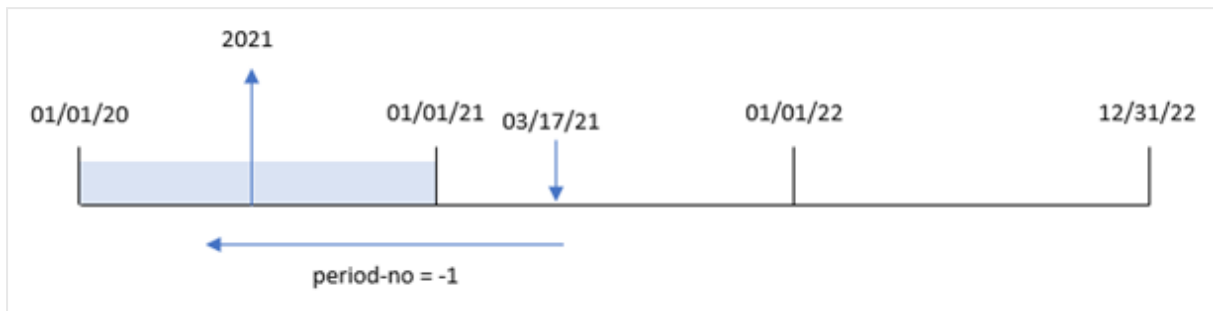
- date
- prior\_year\_name

Resultatentabel

date	prior_year_name
01/13/2020	2019
02/26/2020	2019
03/27/2020	2019
04/16/2020	2019
05/21/2020	2019
08/14/2020	2019
10/07/2020	2019
12/05/2020	2019
01/22/2021	2020
02/03/2021	2020
03/17/2021	2020
04/23/2021	2020
05/04/2021	2020
06/30/2021	2020
07/26/2021	2020
12/27/2021	2020
06/06/2022	2021
07/18/2022	2021
11/14/2022	2021
12/12/2022	2021

Omdat een `period_no` van `-1` wordt gebruikt als het `offset`-argument in de functie `yearname()`, identificeert de functie eerst het jaar waarin de transacties plaatsvinden. De functie verschuift dan een jaar eerder en retourneert het resulterende jaar.

Diagram van functie `yearname()` met de `period_no` ingesteld op `-1`.



### Voorbeeld 3 - `first_month_of_year`

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De systeemvariabele `DateFormat` die is ingesteld op `'MM/DD/YYYY'`.
- Een voorafgaande lading die de `yearname()` gebruikt en die is ingesteld als het veld `year_name`.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*,
yearname(date,0,4) as year_name
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190,'03/27/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195,'12/05/2020',87.21
```

```
8196,'01/22/2021',95.93
```

```
8197,'02/03/2021',45.89
```

```
8198,'03/17/2021',36.23
```

```
8199,'04/23/2021',25.66
```

```
8200,'05/04/2021',82.77
```

```
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '12/27/2021', 25.12
8204, '06/06/2022', 46.23
8205, '07/18/2022', 84.21
8206, '11/14/2022', 96.24
8207, '12/12/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- year\_name

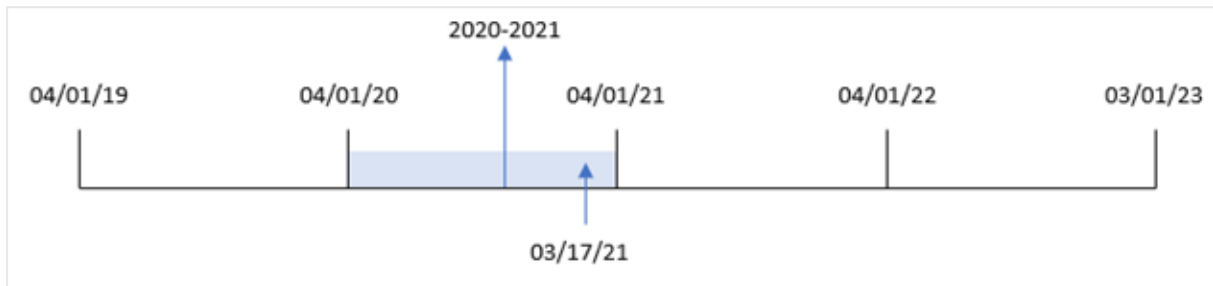
Resultatentabel

date	year_name
01/13/2020	2019-2020
02/26/2020	2019-2020
03/27/2020	2019-2020
04/16/2020	2020-2021
05/21/2020	2020-2021
08/14/2020	2020-2021
10/07/2020	2020-2021
12/05/2020	2020-2021
01/22/2021	2020-2021
02/03/2021	2020-2021
03/17/2021	2020-2021
04/23/2021	2021-2022
05/04/2021	2021-2022
06/30/2021	2021-2022
07/26/2021	2021-2022
12/27/2021	2021-2022
06/06/2022	2022-2023
07/18/2022	2022-2023
11/14/2022	2022-2023
12/12/2022	2022-2023

Omdat het argument [ `first_month_of_year` van 4 wordt gebruikt in de functie `yearname()`, verschuift het begin van het jaar van 1 januari naar 1 april. Daarom kruist elke periode van twaalf maanden twee kalenderjaren en retourneert de functie `yearname()` de twee viercijferige jaar voor geëvalueerde datums.

Transactie 8198 vindt plaats op 17 maart 2021. De functie `yearname()` stelt het begin van het jaar in op 1 april en het einde op 30 maart. Daarom vond transactie 8198 plaats in de jaarperiode van 1 april 2020 en 30 maart 2021. Als resultaat retourneert de functie `yearname()` de waarde 2020-2021.

*Diagram van functie `yearname()` met maart ingesteld als de eerste maand van het jaar.*



### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De systeemvariabele `DateFormat` die is ingesteld op `'MM/DD/YYYY'`.

Het veld dat het jaar retourneert waarin de transactie heeft plaatsgevonden, wordt echter gemaakt als een meetwaarde in een grafiekobject.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,'01/13/2020',37.23
```

```
8189,'02/26/2020',17.17
```

```
8190,'03/27/2020',88.27
```

```
8191,'04/16/2020',57.42
```

```
8192,'05/21/2020',53.80
```

```
8193,'08/14/2020',82.06
```

```
8194,'10/07/2020',40.39
```

```
8195, '12/05/2020', 87.21
8196, '01/22/2021', 95.93
8197, '02/03/2021', 45.89
8198, '03/17/2021', 36.23
8199, '04/23/2021', 25.66
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '12/27/2021', 25.12
8204, '06/06/2022', 46.23
8205, '07/18/2022', 84.21
8206, '11/14/2022', 96.24
8207, '12/12/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie:

date

Om het veld `year_name` te berekenen, maakt u deze meting:

`=yearname(date)`

Resultatentabel

date	=yearname(date)
01/13/2020	2020
02/26/2020	2020
03/27/2020	2020
04/16/2020	2020
05/21/2020	2020
08/14/2020	2020
10/07/2020	2020
12/05/2020	2020
01/22/2021	2021
02/03/2021	2021
03/17/2021	2021
04/23/2021	2021
05/04/2021	2021
06/30/2021	2021
07/26/2021	2021
12/27/2021	2021

date	=yearname(date)
06/06/2022	2022
07/18/2022	2022
11/14/2022	2022
12/12/2022	2022

De meting 'year\_name' wordt gemaakt in het diagramobject met behulp van de functie `yearname()` en geeft het veld door als het argument van de functie.

De functie `yearname()` identificeert in welk jaar de datumwaarde valt en retourneert dit als een viercijferige jaarwaarde.

Diagram van de functie `yearname()` met 2021 als jaarwaarde.



### Voorbeeld 5 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset als in het eerste voorbeeld.
- De systeemvariabele `DateFormat` die is ingesteld op 'MM/DD/YYYY'.

De eindgebruiker wil graag een diagram die de totale omzet per kwartaal voor de transacties weergeeft. Gebruik de functie `yearname()` als een berekende dimensie om dit diagram te maken wanneer de dimensie `yearname()` niet beschikbaar is in het gegevensmodel.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```



```
8188, '01/13/2020', 37.23
8189, '02/26/2020', 17.17
8190, '03/27/2020', 88.27
8191, '04/16/2020', 57.42
8192, '05/21/2020', 53.80
8193, '08/14/2020', 82.06
8194, '10/07/2020', 40.39
8195, '12/05/2020', 87.21
8196, '01/22/2021', 95.93
8197, '02/03/2021', 45.89
8198, '03/17/2021', 36.23
8199, '04/23/2021', 25.66
8200, '05/04/2021', 82.77
8201, '06/30/2021', 69.98
8202, '07/26/2021', 76.11
8203, '12/27/2021', 25.12
8204, '06/06/2022', 46.23
8205, '07/18/2022', 84.21
8206, '11/14/2022', 96.24
8207, '12/12/2022', 67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel.

Als u aggregaties per jaar wilt vergelijken, maakt u deze berekende dimensie:

```
=yearname(date)
```

Maak deze meting

```
=sum(amount)
```

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

<b>yearname(date)</b>	<b>=sum(amount)</b>
2020	\$463.55
2021	\$457.69
2022	\$294.35

### yearstart

Deze functie retourneert een tijdsstempel die overeenkomt met het begin van de eerste dag van het jaar dat **date** bevat. De standaarduitvoernotatie is de **DateFormat** die is ingesteld in het script.

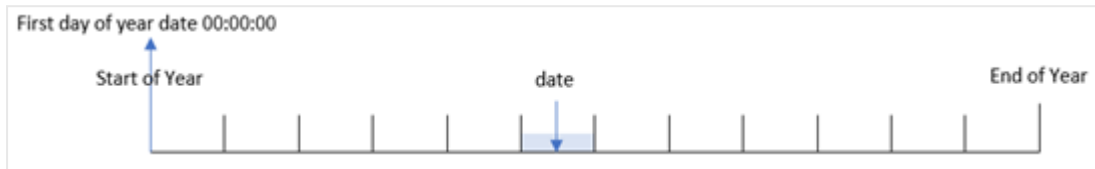
#### Syntaxis:

```
YearStart(date[, period_no[, first_month_of_year]])
```

## 5 Script- en diagramfuncties

Met andere woorden, de functie `yearstart()` bepaalt in welk jaar de datum valt. Het retourneert vervolgens een tijdstempel, in datumnotatie, voor de eerste milliseconde van dat jaar. De eerste maand van het jaar is standaard januari, u kunt echter wijzigen welke maand wordt ingesteld als eerste met behulp van het argument `first_month_of_year` in de functie `yearstart()`.

Diagram van de functie `yearstart()` dat het tijdsbereik toont dat de functie kan bestrijken.



### Wanneer gebruiken

De functie `yearstart()` wordt gebruikt als onderdeel van een uitdrukking als u wilt dat in de berekening een deel van het jaar wordt gebruikt dat tot nu toe is verstreken. Bijvoorbeeld als u de rente wilt berekenen die in een jaar tot nu toe is opgebouwd.

Retourgegevenstypen: dual

#### Argumenten

Argument	Beschrijving
<code>date</code>	De te evalueren datum of tijdstempel.
<code>period_no</code>	<code>period_no</code> is een geheel getal, waarbij de waarde 0 het jaar aangeeft dat <b>date</b> bevat. Negatieve waarden in <code>period_no</code> geven voorgaande jaren aan en positieve waarden geven volgende jaren aan.
<code>first_month_of_year</code>	Als u wilt werken met (boek)jaren die niet in januari beginnen, geeft u een waarde op tussen 2 en 12 in <code>first_month_of_year</code> .

De volgende maanden kunnen worden gebruikt in de `first_month_of_year` argument:

waarden `first_month_of_year`

Month	Waarde
Februari	2
Maart	3
April	4
May	5
Juni	6
Juli	7
Augustus	8

Month	Waarde
September	9
Oktober	10
November	11
December	12

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

#### Voorbeelden van functies

Voorbeeld	Resultaat
<code>yearstart('10/19/2001')</code>	Returns 01/01/2001 00:00:00.
<code>yearstart('10/19/2001', -1)</code>	Returns 01/01/2000 00:00:00.
<code>yearstart('10/19/2001', 0, 4)</code>	Returns 04/01/2001 00:00:00.

### Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 wordt geladen in een tabel met de naam 'Transactions'.
- Het datumveld is opgegeven in de systeemvariabele `DateFormat`, indeling (MM/DD/YYYY).
- Een voorgaande load-instructie die het volgende bevat:

- Functie `yearstart()` die is ingesteld als het veld `year_start`.
- Functie `timestamp()` die is ingesteld als het veld `year_start_timestamp`

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        yearstart(date) as year_start,
        timestamp(yearstart(date)) as year_start_timestamp
    ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,01/13/2020,37.23
```

```
8189,02/26/2020,17.17
```

```
8190,03/27/2020,88.27
```

```
8191,04/16/2020,57.42
```

```
8192,05/21/2020,53.80
```

```
8193,08/14/2020,82.06
```

```
8194,10/07/2020,40.39
```

```
8195,12/05/2020,87.21
```

```
8196,01/22/2021,95.93
```

```
8197,02/03/2021,45.89
```

```
8198,03/17/2021,36.23
```

```
8199,04/23/2021,25.66
```

```
8200,05/04/2021,82.77
```

```
8201,06/30/2021,69.98
```

```
8202,07/26/2021,76.11
```

```
8203,12/27/2021,25.12
```

```
8204,06/06/2022,46.23
```

```
8205,07/18/2022,84.21
```

```
8206,11/14/2022,96.24
```

```
8207,12/12/2022,67.67
```

```
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- `id`
- `date`
- `year_start`
- `year_start_timestamp`

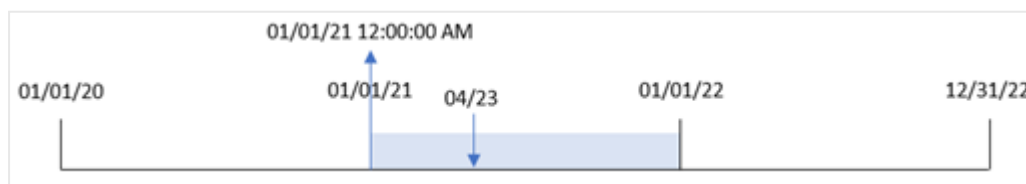
Resultatentabel

id	date	year_start	year_start_timestamp
8188	01/13/2020	01/01/2020	1/1/2020 12:00:00 AM
8189	02/26/2020	01/01/2020	1/1/2020 12:00:00 AM
8190	03/27/2020	01/01/2020	1/1/2020 12:00:00 AM
8191	04/16/2020	01/01/2020	1/1/2020 12:00:00 AM
8192	05/21/2020	01/01/2020	1/1/2020 12:00:00 AM
8193	08/14/2020	01/01/2020	1/1/2020 12:00:00 AM
8194	10/07/2020	01/01/2020	1/1/2020 12:00:00 AM
8195	12/05/2020	01/01/2020	1/1/2020 12:00:00 AM
8196	01/22/2021	01/01/2021	1/1/2021 12:00:00 AM
8197	02/03/2021	01/01/2021	1/1/2021 12:00:00 AM
8198	03/17/2021	01/01/2021	1/1/2021 12:00:00 AM
8199	04/23/2021	01/01/2021	1/1/2021 12:00:00 AM
8200	05/04/2021	01/01/2021	1/1/2021 12:00:00 AM
8201	06/30/2021	01/01/2021	1/1/2021 12:00:00 AM
8202	07/26/2021	01/01/2021	1/1/2021 12:00:00 AM
8203	12/27/2021	01/01/2021	1/1/2021 12:00:00 AM
8204	06/06/2022	01/01/2022	1/1/2022 12:00:00 AM
8205	07/18/2022	01/01/2022	1/1/2022 12:00:00 AM
8206	11/14/2022	01/01/2022	1/1/2022 12:00:00 AM
8207	12/12/2022	01/01/2022	1/1/2022 12:00:00 AM

Het veld 'year\_start' wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `yearstart()` en geeft het datumveld door als het argument van de functie.

De functie `yearstart()` identificeert aanvankelijk in welk jaar de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van dat jaar.

*Diagram van de functie `yearstart()` en transactie 8199.*



Transactie 8199 vond plaats op 23 april 2021. De functie `yearstart()` retourneert de eerste milliseconde van dat jaar, 1 januari om 12:00:00 AM.

### Voorbeeld 2 - period\_no

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het echter de taak om een veld te maken, 'previous\_year\_start', dat de tijdstempel van de startdatum van het jaar voorafgaand aan het jaar waarin de transactie plaatsvond, retourneert.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        yearstart(date,-1) as previous_year_start,
        timestamp(yearstart(date,-1)) as previous_year_start_timestamp
    ;

Load
*
Inline
[
id,date,amount
8188,01/13/2020,37.23
8189,02/26/2020,17.17
8190,03/27/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,06/06/2022,46.23
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

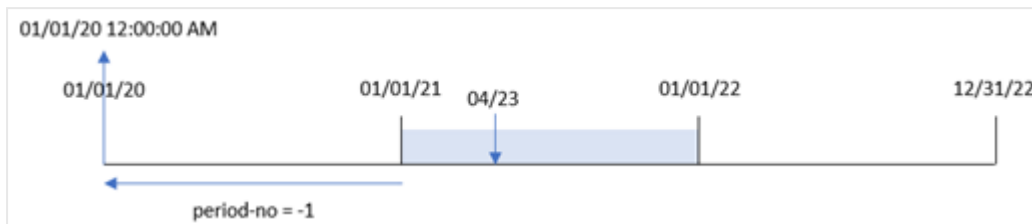
- id
- date
- previous\_year\_start
- previous\_year\_start\_timestamp

Resultatentabel

id	date	previous_year_start	previous_year_start_timestamp
8188	01/13/2020	01/01/2019	1/1/2019 12:00:00 AM
8189	02/26/2020	01/01/2019	1/1/2019 12:00:00 AM
8190	03/27/2020	01/01/2019	1/1/2019 12:00:00 AM
8191	04/16/2020	01/01/2019	1/1/2019 12:00:00 AM
8192	05/21/2020	01/01/2019	1/1/2019 12:00:00 AM
8193	08/14/2020	01/01/2019	1/1/2019 12:00:00 AM
8194	10/07/2020	01/01/2019	1/1/2019 12:00:00 AM
8195	12/05/2020	01/01/2019	1/1/2019 12:00:00 AM
8196	01/22/2021	01/01/2020	1/1/2020 12:00:00 AM
8197	02/03/2021	01/01/2020	1/1/2020 12:00:00 AM
8198	03/17/2021	01/01/2020	1/1/2020 12:00:00 AM
8199	04/23/2021	01/01/2020	1/1/2020 12:00:00 AM
8200	05/04/2021	01/01/2020	1/1/2020 12:00:00 AM
8201	06/30/2021	01/01/2020	1/1/2020 12:00:00 AM
8202	07/26/2021	01/01/2020	1/1/2020 12:00:00 AM
8203	12/27/2021	01/01/2020	1/1/2020 12:00:00 AM
8204	06/06/2022	01/01/2021	1/1/2021 12:00:00 AM
8205	07/18/2022	01/01/2021	1/1/2021 12:00:00 AM
8206	11/14/2022	01/01/2021	1/1/2021 12:00:00 AM
8207	12/12/2022	01/01/2021	1/1/2021 12:00:00 AM

Omdat in deze instantie een `period_no` van `-1` wordt gebruikt als het `offset`-argument in de functie `yearstart()`, identificeert de functie eerst het jaar waarin de transacties plaatsvinden. Het kijkt dan een jaar eerder en identificeert de eerste milliseconde van dat jaar.

Diagram van de functie `yearstart()` met een `period_no` van `-1`.



Transactie 8199 vond plaats op 23 april 2021. De functie `yearstart()` retourneert de eerste milliseconde van het voorgaande jaar, 1 januari 2020 om 12:00:00 AM, voor het veld 'previous\_year\_start'.

### Voorbeeld 3 - first\_month\_of\_year

Load-script en resultaten

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is het bedrijfsbeleid echter dat het jaar op 1 april begint.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
*,
yearstart(date,0,4) as year_start,
timestamp(yearstart(date,0,4)) as year_start_timestamp
;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
8188,01/13/2020,37.23
8189,02/26/2020,17.17
8190,03/27/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
```



```
8204,06/06/2022,46.23
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date
- year\_start
- year\_start\_timestamp

Resultatentabel

id	date	year_start	year_start_timestamp
8188	01/13/2020	04/01/2019	4/1/2019 12:00:00 AM
8189	02/26/2020	04/01/2019	4/1/2019 12:00:00 AM
8190	03/27/2020	04/01/2019	4/1/2019 12:00:00 AM
8191	04/16/2020	04/01/2020	4/1/2020 12:00:00 AM
8192	05/21/2020	04/01/2020	4/1/2020 12:00:00 AM
8193	08/14/2020	04/01/2020	4/1/2020 12:00:00 AM
8194	10/07/2020	04/01/2020	4/1/2020 12:00:00 AM
8195	12/05/2020	04/01/2020	4/1/2020 12:00:00 AM
8196	01/22/2021	04/01/2020	4/1/2020 12:00:00 AM
8197	02/03/2021	04/01/2020	4/1/2020 12:00:00 AM
8198	03/17/2021	04/01/2020	4/1/2020 12:00:00 AM
8199	04/23/2021	04/01/2021	4/1/2021 12:00:00 AM
8200	05/04/2021	04/01/2021	4/1/2021 12:00:00 AM
8201	06/30/2021	04/01/2021	4/1/2021 12:00:00 AM
8202	07/26/2021	04/01/2021	4/1/2021 12:00:00 AM
8203	12/27/2021	04/01/2021	4/1/2021 12:00:00 AM
8204	06/06/2022	04/01/2022	4/1/2022 12:00:00 AM
8205	07/18/2022	04/01/2022	4/1/2022 12:00:00 AM
8206	11/14/2022	04/01/2022	4/1/2022 12:00:00 AM
8207	12/12/2022	04/01/2022	4/1/2022 12:00:00 AM

Omdat in deze instantie het argument `first_month_of_year` van 4 wordt gebruikt in de functie `yearstart()`, stelt het de eerste dag van het jaar in op 1 april en de laatste dag van het jaar op 31 maart.

*Diagram van de functie `yearstart()` met de eerste maand ingesteld als april.*



Transactie 8199 vond plaats op 23 april 2021. Omdat de functie `yearstart()` het begin van het jaar instelt op 1 april en dit retourneert als de waarde 'year\_start' voor de transactie.

### Voorbeeld 4 - diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Dezelfde gegevensverzameling en het scenario uit het eerste voorbeeld worden gebruikt.

In dit voorbeeld is de gegevensverzameling echter niet gewijzigd en wordt in de applicatie geladen. De berekening die een startdatumtijdstempel retourneert van het jaar waarin de transactie plaatsvond is gemaakt als meting in een diagramobject van de toepassing.

#### Load-script

Transactions:

Load

\*

Inline

[

id,date,amount

8188,01/13/2020,37.23

8189,02/26/2020,17.17

8190,03/27/2020,88.27

8191,04/16/2020,57.42

8192,05/21/2020,53.80

8193,08/14/2020,82.06

8194,10/07/2020,40.39

8195,12/05/2020,87.21

8196,01/22/2021,95.93

8197,02/03/2021,45.89

8198,03/17/2021,36.23

8199,04/23/2021,25.66

8200,05/04/2021,82.77

8201,06/30/2021,69.98

8202,07/26/2021,76.11

8203,12/27/2021,25.12

8204,06/06/2022,46.23

```
8205,07/18/2022,84.21
8206,11/14/2022,96.24
8207,12/12/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- id
- date

Maak de volgende metingen om te berekenen in welk jaar een transactie plaatsvond:

- =yearstart(date)
- =timestamp(yearstart(date))

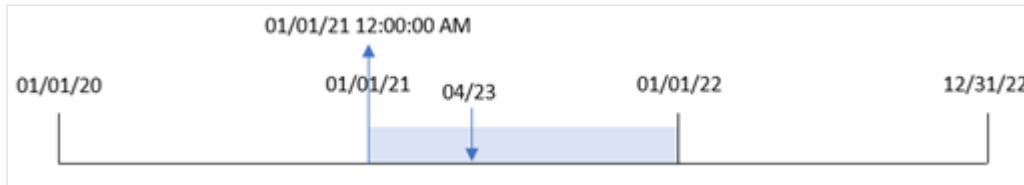
Resultatentabel

id	date	=yearstart(date)	=timestamp(yearstart(date))
8188	06/06/2022	01/01/2022	1/1/2022 12:00:00 AM
8189	07/18/2022	01/01/2022	1/1/2022 12:00:00 AM
8190	11/14/2022	01/01/2022	1/1/2022 12:00:00 AM
8191	12/12/2022	01/01/2022	1/1/2022 12:00:00 AM
8192	01/22/2021	01/01/2021	1/1/2021 12:00:00 AM
8193	02/03/2021	01/01/2021	1/1/2021 12:00:00 AM
8194	03/17/2021	01/01/2021	1/1/2021 12:00:00 AM
8195	04/23/2021	01/01/2021	1/1/2021 12:00:00 AM
8196	05/04/2021	01/01/2021	1/1/2021 12:00:00 AM
8197	06/30/2021	01/01/2021	1/1/2021 12:00:00 AM
8198	07/26/2021	01/01/2021	1/1/2021 12:00:00 AM
8199	12/27/2021	01/01/2021	1/1/2021 12:00:00 AM
8200	01/13/2020	01/01/2020	1/1/2020 12:00:00 AM
8201	02/26/2020	01/01/2020	1/1/2020 12:00:00 AM
8202	03/27/2020	01/01/2020	1/1/2020 12:00:00 AM
8203	04/16/2020	01/01/2020	1/1/2020 12:00:00 AM
8204	05/21/2020	01/01/2020	1/1/2020 12:00:00 AM
8205	08/14/2020	01/01/2020	1/1/2020 12:00:00 AM
8206	10/07/2020	01/01/2020	1/1/2020 12:00:00 AM
8207	12/05/2020	01/01/2020	1/1/2020 12:00:00 AM

De meting 'start\_of\_year' wordt gemaakt in het diagramobject met behulp van de functie `yearstart()` en geeft het veld door als het argument van de functie.

De functie `yearstart()` identificeert aanvankelijk in welk jaar de datumwaarde valt en retourneert een tijdstempel voor de eerste milliseconde van dat jaar.

*Diagram van de functie `yearstart()` en transactie 8199.*



Transactie 8199 vond plaats op 23 april 2021. De functie `yearstart()` retourneert de eerste milliseconde van dat jaar, 1 januari om 12:00:00 AM.

### Voorbeeld 5 - Scenario

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset wordt geladen in een tabel met de naam 'Loans'. De tabel bevat de volgende velden:
  - Lening-id's.
  - Het saldo aan het begin van het jaar.
  - De enkelvoudige rente die op elke lening per jaar wordt berekend.

De eindgebruiker wil een diagramobject dat per lening-id de huidige rente weergeeft die is opgebouwd op elke lening in het jaar tot heden.

#### Load-script

```
Loans:
Load
*
Inline
[
loan_id,start_balance,rate
8188,$10000.00,0.024
8189,$15000.00,0.057
8190,$17500.00,0.024
8191,$21000.00,0.034
8192,$90000.00,0.084
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- loan\_id
- start\_balance

Maak de volgende meting om de opgebouwde rente te berekenen:

`=start_balance*(rate*(today(1)-yearstart(today(1)))/365)`

Stel de **Getalnotatie** van de meting in op **Geld**.

Resultatentabel

loan_id	start_balance	=start_balance*(rate*(today(1)-yearstart(today(1)))/365)
8188	\$10000.00	\$39.73
8189	\$15000.00	\$339.66
8190	\$17500.00	\$166.85
8191	\$21000.00	\$283.64
8192	\$90000.00	\$3003.29

De functie `yearstart()` gebruikt de datum van vandaag als enige argument en retourneert de begindatum van het huidige jaar. Door dat resultaat af te trekken van de huidige datum, retourneert de uitdrukking het aantal dagen dat in dit jaar tot nu toe is verstreken.

Deze waarde wordt vervolgens vermenigvuldigd met het rentepercentage en gedeeld door 365 om het effectieve rentepercentage voor de periode te retourneren. Het effectieve rentepercentage over de periode wordt vervolgens vermenigvuldigd met het beginsaldo van de lening om de rente terug te betalen die dit jaar tot nu toe is opgebouwd.

### yeartodate

Deze functie bepaalt of de invoertijdstempel binnen het jaar valt waarin het script voor het laatst is geladen en retourneert True als dat het geval is en False als dat niet het geval is.

#### Syntaxis:

```
YearToDate(timestamp[ , yearoffset [ , firstmonth [ , todaydate] ] ])
```

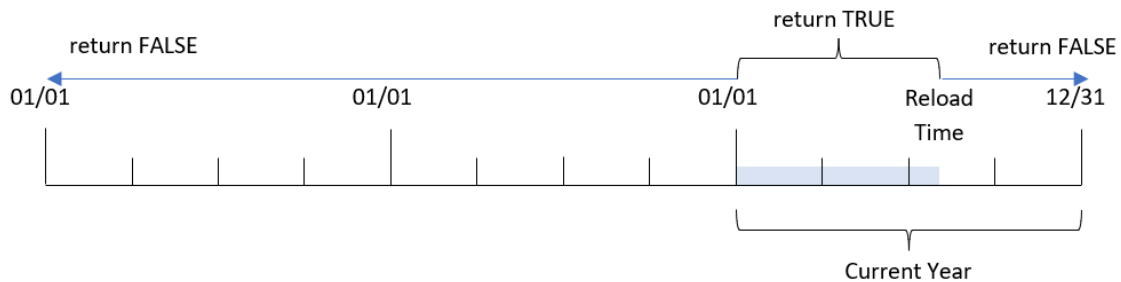
**Retourgegevenstypen:** Booleaanse waarde



*In Qlik Sense wordt de booleaanse waarde Waar vertegenwoordigd door -1 en de waarde Onwaar door 0.*

## 5 Script- en diagramfuncties

Voorbeelddiagram van de `yeartodate()`-functie



Als geen van de optionele parameters wordt gebruikt, betekent het jaar tot op heden een datum binnen één kalenderjaar vanaf 1 januari tot en met de datum waarop het script voor het laatst is uitgevoerd.

Met andere woorden: wanneer de `yeartodate()`-functie zonder extra parameters wordt getriggerd, wordt deze gebruikt om een tijdstempel te evalueren en een booleaans resultaat te retourneren op basis van of de datum in het kalenderjaar viel tot en met de datum waarop er opnieuw is geladen.

Maar het is tevens mogelijk om de begindatum van het jaar te vervangen met het `firstmonth`-argument, en om vergelijkingen te maken met voorgaande of volgende jaren met behulp van het `yearoffset`-argument.

In gevallen van historische gegevenssets, biedt de `yeartodate()`-functie een parameter om `todaydate` in te stellen, waarmee de tijdstempel met het kalenderjaar wordt vergeleken tot en met de datum die in het `todaydate`-argument is opgegeven.

### Argumenten

Argument	Beschrijving
<code>timestamp</code>	De tijdstempel die moet worden geëvalueerd, bijvoorbeeld '10/12/2012'.
<code>yearoffset</code>	Als een <b>yearoffset</b> wordt opgegeven, retourneert <b>yeartodate</b> de waarde <b>True</b> voor dezelfde periode in een ander jaar. Een negatieve <b>yearoffset</b> geeft een eerder jaar aan en een positieve verschuiving een toekomstig jaar. Het meest recente jaar tot heden wordt geretourneerd door <code>yearoffset = -1</code> op te geven. Bij weglating, wordt 0 aangenomen.
<code>firstmonth</code>	Door een <b>firstmonth</b> tussen 1 en 12 op te geven (1 indien weggelaten), kunt u het begin van het jaar naar voren verplaatsen naar de eerste dag van een andere maand. Als u bijvoorbeeld wilt werken met een fiscaal jaar dat begint op 1 mei, geeft u <b>firstmonth = 5</b> op. Een waarde van 1 zou aangeven dat een boekjaar begint op 1 januari en een waarde van 12 zou aangeven dat een boekjaar begint op 1 december.
<code>todaydate</code>	Door een <b>todaydate</b> op te geven (tijdstempel van de laatste scriptuitvoering indien weggelaten), kunt u de dag verschuiven die als bovengrens van de periode wordt gebruikt.

### Wanneer gebruiken

De functie `yeartodate()` retourneert een booleaanse waarde. Dit type functie wordt doorgaans gebruikt als voorwaarde in een if-uitdrukking. Dit retourneert een aggregatie of berekening afhankelijk van of een geëvalueerde datum plaatsvond in het jaar tot en met de datum waarop de applicatie voor het laatst opnieuw is geladen.

De `YearToDate()`-functie kan bijvoorbeeld worden gebruikt om alle materieel te identificeren dat tot dusver in het huidige jaar is geproduceerd.

Voor de volgende voorbeelden is de laatste datum waarop het script werd geladen = 11/18/2011.

Voorbeelden van functies

Voorbeeld	Resultaat
<code>yeartodate( '11/18/2010')</code>	retourneert False
<code>yeartodate( '02/01/2011')</code>	retourneert True
<code>yeartodate( '11/18/2011')</code>	retourneert True
<code>yeartodate( '11/19/2011')</code>	retourneert False
<code>yeartodate( '11/19/2011', 0, 1, '12/31/2011')</code>	retourneert True
<code>yeartodate( '11/18/2010', -1)</code>	retourneert True
<code>yeartodate( '11/18/2011', -1)</code>	retourneert False
<code>yeartodate( '04/30/2011', 0, 5)</code>	retourneert False
<code>yeartodate( '05/01/2011', 0, 5)</code>	retourneert True

### Landinstellingen

Tenzij anders aangegeven, wordt in de voorbeelden in dit onderwerp de volgende datumindeling gebruikt: MM/DD/JJJJ. De datumopmaak wordt opgegeven in de opdracht `SET DateFormat` in uw script voor het laden van gegevens. De standaard datumindeling is mogelijk anders op uw systeem vanwege landinstellingen en andere factoren. U kunt de indelingen in de voorbeelden wijzigen zodat deze aansluiten op uw behoeften. Of u kunt de indelingen in uw load-script wijzigen zodat ze overeenkomen met deze voorbeelden.

Standaard landinstellingen in apps zijn gebaseerd op de regionale systeeminstellingen van de computer of server waarop Qlik Sense is geïnstalleerd. Als de Qlik Sense-server die u gebruikt is ingesteld op Zweden, gebruikt de editor voor laden van gegevens de Zweedse landinstellingen voor datums, tijd en valuta. Deze landinstellingen voor indelingen zijn niet gerelateerd aan de taal die wordt weergegeven in de gebruikersinterface van Qlik Sense. Qlik Sense wordt weergegeven in dezelfde taal als de browser die u gebruikt.

### Voorbeeld 1 - Basisvoorbeeld

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (MM/DD/YYYY) indeling.
- Het aanmaken van een veld, `year_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde kalenderjaar als de laatste keer dat er is geladen.

Op dit moment is de datum 26 april 2022.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    yeartodate(date) as year_to_date
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,01/10/2020,37.23
```

```
8189,02/28/2020,17.17
```

```
8190,04/09/2020,88.27
```

```
8191,04/16/2020,57.42
```

```
8192,05/21/2020,53.80
```

```
8193,08/14/2020,82.06
```

```
8194,10/07/2020,40.39
```

```
8195,12/05/2020,87.21
```

```
8196,01/22/2021,95.93
```

```
8197,02/03/2021,45.89
```

```
8198,03/17/2021,36.23
```

```
8199,04/23/2021,25.66
```

```
8200,05/04/2021,82.77
```

```
8201,06/30/2021,69.98
```

```
8202,07/26/2021,76.11
```

```
8203,12/27/2021,25.12
```

```
8204,02/02/2022,46.23
```

```
8205,02/26/2022,84.21
```

```
8206,03/07/2022,96.24
```



```
8207,03/11/2022,67.67  
];
```

### Resultaten

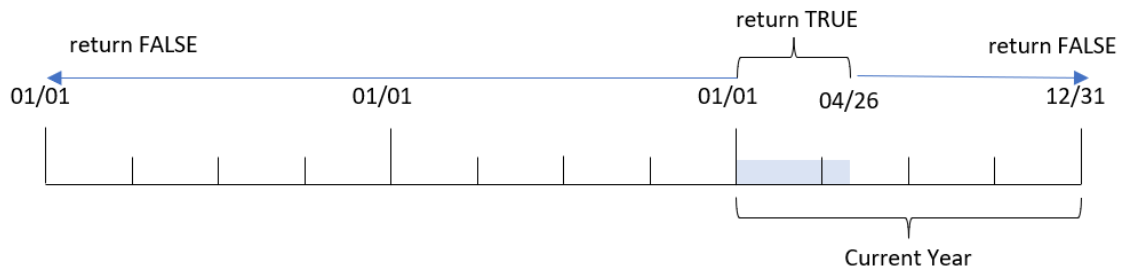
Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- year\_to\_date

Resultatentabel

date	year_to_date
01/10/2020	0
02/28/2020	0
04/09/2020	0
04/16/2020	0
05/21/2020	0
08/14/2020	0
10/07/2020	0
12/05/2020	0
01/22/2021	0
02/03/2021	0
03/17/2021	0
04/23/2021	0
05/04/2021	0
06/30/2021	0
07/26/2021	0
12/27/2021	0
02/02/2022	-1
02/26/2022	-1
03/07/2022	-1
03/11/2022	-1

Diagram van `yeartodate()`-functie, basisvoorbeeld



Het veld `year_to_date` wordt gemaakt in de voorafgaande load-instructie met behulp van de functie `yeartodate()` en geeft het veld `date` door als het argument van de functie.

Omdat er geen andere parameters worden doorgegeven in de functie, identificeert de `yeartodate()`-functie aanvankelijk de herlaaddatum en daarom de grenswaarden voor het huidige kalenderjaar (vanaf 1 januari) die een booleaanse waarde van `TRUE` retourneren.

Daarom zal elke transactie die plaatsvindt tussen 1 januari en 26 april, de datum van opnieuw laden, een booleaans resultaat van `TRUE` retourneren. Een transactie die vóór het begin van 2022 plaatsvindt, retourneert een booleaans resultaat voor `FALSE`.

### Voorbeeld 2 - yearoffset

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `two_years_prior`, dat bepaalt welke transacties twee jaar voor het kalenderjaar tot heden hebben plaatsgevonden.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
  Load
    *,
    yeartodate(date,-2) as two_years_prior
  ;
```

```
Load
```

```
*
```

```
Inline
```

```
[
id,date,amount
```

```
8188,01/10/2020,37.23
8189,02/28/2020,17.17
8190,04/09/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,02/02/2022,46.23
8205,02/26/2022,84.21
8206,03/07/2022,96.24
8207,03/11/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- two\_years\_prior

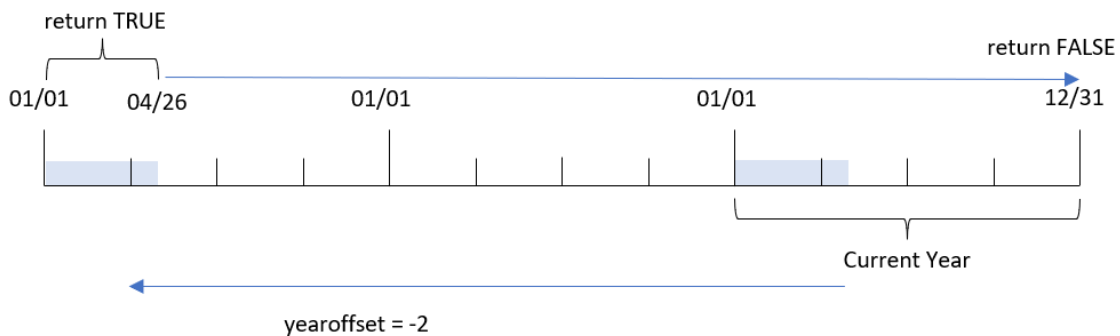
Resultatentabel

date	two_years_prior
01/10/2020	-1
02/28/2020	-1
04/09/2020	-1
04/16/2020	-1
05/21/2020	0
08/14/2020	0
10/07/2020	0
12/05/2020	0
01/22/2021	0
02/03/2021	0
03/17/2021	0
04/23/2021	0

date	two_years_prior
05/04/2021	0
06/30/2021	0
07/26/2021	0
12/27/2021	0
02/02/2022	0
02/26/2022	0
03/07/2022	0
03/11/2022	0

Door -2 als het `yearoffset`-argument te gebruiken in de `yeartodate()`-functie, verschuift de functie de grenswaarden van het kalenderjaarsegment voor vergelijking met twee hele jaren. Aanvankelijk komt het jaarsegment overeen met de periode tussen 1 januari en 26 april 2022. Het `yearoffset`-argument verschuift dit segment vervolgens naar twee jaar eerder. De datumgrenzen vallen tussen 1 januari en 26 april.

*Diagram van `yeartodate()`-functie, voorbeeld `yearoffset`*



Daarom zal elke transactie die plaatsvindt tussen 1 januari en 26 april 2020 een booleaans resultaat van TRUE retourneren. Iedere transactie die vóór of na dit segment valt, retourneert FALSE.

### Voorbeeld 3 - firstmonth

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `year_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde kalenderjaar als de laatste keer dat er is geladen.

In dit voorbeeld stellen de het begin van het boekjaar in op 1 juli.

### Load-script

```
SET DateFormat='MM/DD/YYYY';

Transactions:
  Load
    *,
    yeartodate(date,0,7) as year_to_date
  ;
Load
*
Inline
[
id,date,amount
8188,01/10/2020,37.23
8189,02/28/2020,17.17
8190,04/09/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,02/02/2022,46.23
8205,02/26/2022,84.21
8206,03/07/2022,96.24
8207,03/11/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

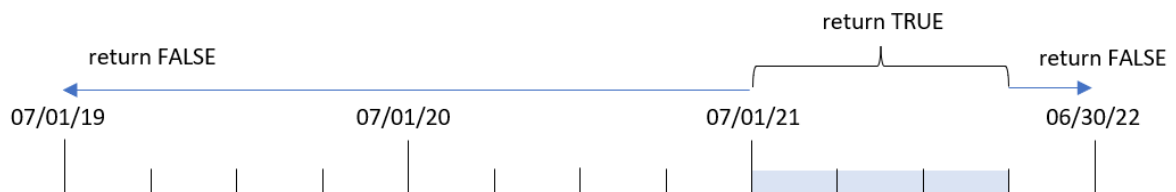
- `date`
- `year_to_date`

Resultatentabel

date	year_to_date
01/10/2020	0
02/28/2020	0
04/09/2020	0
04/16/2020	0
05/21/2020	0
08/14/2020	0
10/07/2020	0
12/05/2020	0
01/22/2021	0
02/03/2021	0
03/17/2021	0
04/23/2021	0
05/04/2021	0
06/30/2021	0
07/26/2021	-1
12/27/2021	-1
02/02/2022	-1
02/26/2022	-1
03/07/2022	-1
03/11/2022	-1

Omdat in deze instantie het `firstmonth`-argument 7 wordt gebruikt in de `yeartodate()`-functie, stelt het de eerste dag van het jaar in op 1 juli en de laatste dag van het jaar op 30 juni.

*Diagram van de `yeartodate()`-functie, voorbeeld `firstmonth`*



Daarom zal elke transactie die plaatsvindt tussen 1 juli 2021 en 26 april 2022, de datum van opnieuw laden, een booleaans resultaat voor TRUE retourneren. Een transactie die vóór 1 juli 2021 plaatsvindt, retourneert een booleaans resultaat voor FALSE.

### Voorbeeld 4 - todaydate

Load-script en resultaten

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Dezelfde gegevensset en het scenario uit het eerste voorbeeld.
- Het aanmaken van een veld, `year_to_date`, dat bepaalt welke transacties hebben plaatsgevonden in hetzelfde kalenderjaar als de laatste keer dat er is geladen.

Maar in dit voorbeeld moeten we alle transacties identificeren die plaatsvonden in het kalenderjaar tot en met 1 maart 2022.

#### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
    Load
        *,
        yeartodate(date, 0, 1, '03/01/2022') as year_to_date
;
Load
*
Inline
[
id,date,amount
8188,01/10/2020,37.23
8189,02/28/2020,17.17
8190,04/09/2020,88.27
8191,04/16/2020,57.42
8192,05/21/2020,53.80
8193,08/14/2020,82.06
8194,10/07/2020,40.39
8195,12/05/2020,87.21
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,02/02/2022,46.23
```

```
8205,02/26/2022,84.21  
8206,03/07/2022,96.24  
8207,03/11/2022,67.67  
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg deze velden toe als dimensies:

- date
- year\_to\_date

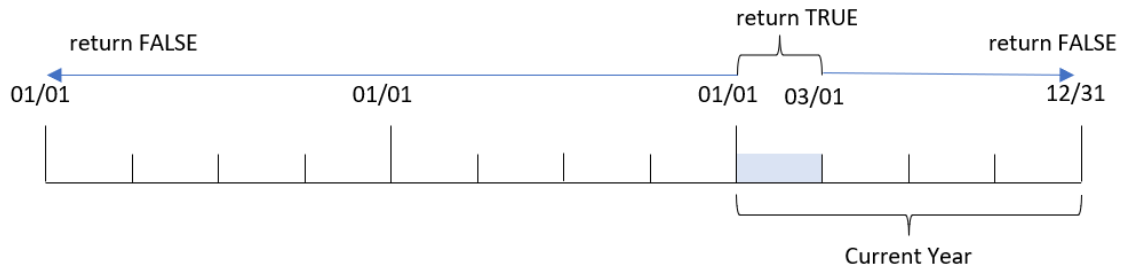
Resultatentabel

date	year_to_date
01/10/2020	0
02/28/2020	0
04/09/2020	0
04/16/2020	0
05/21/2020	0
08/14/2020	0
10/07/2020	0
12/05/2020	0
01/22/2021	0
02/03/2021	0
03/17/2021	0
04/23/2021	0
05/04/2021	0
06/30/2021	0
07/26/2021	0
12/27/2021	0
02/02/2022	-1
02/26/2022	-1
03/07/2022	0
03/11/2022	0



Omdat het `todaydate`-argument van 03/01/2022 in dit geval wordt gebruikt in de `yeartodate()`-functie, stelt het de eindgrens van het kalenderjaarsegment waarmee wordt vergeleken in op 1 maart 2022. Het is cruciaal om de `firstmonth`-parameter (tussen 1 en 12) in te voeren omdat de functie anders geen resultaten oplevert.

*Diagram van de `yeartodate()`-functie, voorbeeld met het `todaydate`-argument*



Daarom zal elke transactie die plaatsvindt tussen 1 januari 2022 en 1 maart 2022, de parameter `todaydate`, een booleaans resultaat voor `TRUE` retourneren. Een transactie die vóór 1 januari 2022 of na 1 maart 2022 plaatsvindt, retourneert een booleaans resultaat van `FALSE`.

### Voorbeeld 5 - Diagramobjectvoorbeeld

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat dezelfde gegevensset en scenario als het eerste voorbeeld.

In dit voorbeeld wordt de ongewijzigde gegevensset echter in de applicatie geladen. De berekening die bepaalt welke transacties plaatsvonden in het kalenderjaar tot de datum van de laatste keer dat er is geladen, wordt gemaakt als een meting in een diagramobject van de applicatie.

#### Load-script

Transactions:

Load

\*

Inline

[

id,date,amount

8188,01/10/2020,37.23

8189,02/28/2020,17.17

8190,04/09/2020,88.27

8191,04/16/2020,57.42

8192,05/21/2020,53.80

8193,08/14/2020,82.06

8194,10/07/2020,40.39

8195,12/05/2020,87.21

```
8196,01/22/2021,95.93
8197,02/03/2021,45.89
8198,03/17/2021,36.23
8199,04/23/2021,25.66
8200,05/04/2021,82.77
8201,06/30/2021,69.98
8202,07/26/2021,76.11
8203,12/27/2021,25.12
8204,02/02/2022,46.23
8205,02/26/2022,84.21
8206,03/07/2022,96.24
8207,03/11/2022,67.67
];
```

### Resultaten

Laad de gegevens en open een werkblad. Maak een nieuwe tabel en voeg dit veld toe als dimensie: date.

Voeg de volgende meting toe:

```
=yeartodate(date)
```

Resultatentabel

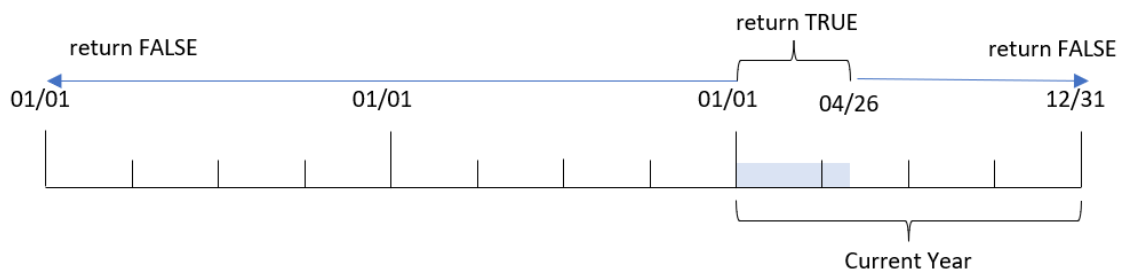
date	=yeartodate(date)
01/10/2020	0
02/28/2020	0
04/09/2020	0
04/16/2020	0
05/21/2020	0
08/14/2020	0
10/07/2020	0
12/05/2020	0
01/22/2021	0
02/03/2021	0
03/17/2021	0
04/23/2021	0
05/04/2021	0
06/30/2021	0
07/26/2021	0
12/27/2021	0
02/02/2022	-1

date	=yeartodate(date)
02/26/2022	-1
03/07/2022	-1
03/11/2022	-1

De meting `year_to_date` wordt gemaakt in het diagramobject met behulp van de `yeartodate()`-functie en geeft het veld `date` door als het argument van de functie.

Omdat er geen andere parameters worden doorgegeven in de functie, identificeert de `yeartodate()`-functie aanvankelijk de herlaaddatum en daarom de grenswaarden voor het huidige kalenderjaar (vanaf 1 januari) die een booleaanse waarde van `TRUE` retourneren.

*Diagram van `yeartodate()`-functie, voorbeeld met gebruik van een diagramobject*



Elke transactie die plaatsvindt tussen 1 januari en 26 april, de datum van opnieuw laden, zal een booleaans resultaat voor `TRUE` retourneren. Een transactie die vóór het begin van 2022 plaatsvindt, retourneert een booleaans resultaat voor `FALSE`.

### Voorbeeld 6 - Scenario

Load-script en diagramuitdrukking

#### Overzicht

Open de editor voor laden van gegevens en voeg het onderstaande load-script toe aan een nieuw tabblad.

Het load-script bevat:

- Een gegevensset met een set transacties tussen 2020 en 2022 die wordt geladen in een tabel met de naam `Transactions`.
- Het datumveld opgegeven in de `DateFormat`-systeemvariabele (`MM/DD/YYYY`) indeling.

De eindgebruiker wil een KPI-object dat de totale verkoop voor de overeenkomstige periode in 2021 weergeeft als het huidige jaar tot heden als de datum van de laatste keer dat er is geladen.

Op dit moment is de datum 16 juni 2022.

### Load-script

```
SET DateFormat='MM/DD/YYYY';
```

```
Transactions:
```

```
Load
```

```
*
```

```
Inline
```

```
[
```

```
id,date,amount
```

```
8188,01/10/2020,37.23
```

```
8189,02/28/2020,17.17
```

```
8190,04/09/2020,88.27
```

```
8191,04/16/2020,57.42
```

```
8192,05/21/2020,53.80
```

```
8193,08/14/2020,82.06
```

```
8194,10/07/2020,40.39
```

```
8195,12/05/2020,87.21
```

```
8196,01/22/2021,95.93
```

```
8197,02/03/2021,45.89
```

```
8198,03/17/2021,36.23
```

```
8199,04/23/2021,25.66
```

```
8200,05/04/2021,82.77
```

```
8201,06/30/2021,69.98
```

```
8202,07/26/2021,76.11
```

```
8203,12/27/2021,25.12
```

```
8204,02/02/2022,46.23
```

```
8205,02/26/2022,84.21
```

```
8206,03/07/2022,96.24
```

```
8207,03/11/2022,67.67
```

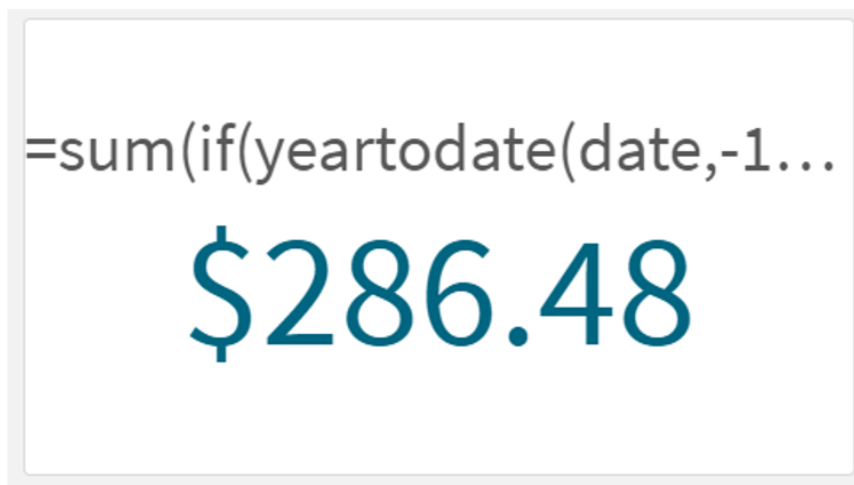
```
];
```

### Resultaten

#### Doe het volgende:

1. Maak een KPI-object.
2. Maak de volgende aggregatiemeting om de totale verkoop te berekenen:  
=sum(if(yeartodate(date,-1),amount,0))
3. Stel de **Getalnotatie** van de meting in op **Geld**.

KPI `yeartodate()`-diagram voor 2021



De `yeartodate()`-functie retourneert een booleaanse waarde bij het evalueren van de productiedatums van elke transactie-id. Omdat er op 16 juni 2022 opnieuw is geladen, segmenteert de `yeartodate`-functie de periode van het jaar tussen 01/01/2022 en 06/16/2022. Maar omdat er een `period_no`-waarde van -1 in de functie is gebruikt, worden deze grenzen naar het vorige jaar verschoven. Daarom retourneert de `yeartodate()`-functie voor iedere transactie die tussen 01/01/2021 en 06/16/2021 plaatsvindt, een booleaanse waarde van `TRUE` en telt hij het bedrag op.

### 5.8 Exponentiële en logaritmische functies

In dit hoofdstuk worden functies beschreven die verband houden met exponentiële en logaritmische berekeningen. Alle functies kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

In de onderstaande functies zijn de parameters uitdrukkingen waarbij **x** en **y** moeten worden geïnterpreteerd als reële getallen.

#### **exp**

De natuurlijk exponentiële functie,  $e^x$ , die de natuurlijke logaritme **e** als grondtal gebruikt. Het resultaat is een positief getal.

```
exp ( x )
```

#### **Voorbeelden en resultaten:**

`exp(3)` retourneert 20,085.

#### **log**

De natuurlijke logaritme van **x**. De functie is alleen gedefinieerd als  $x > 0$ . Het resultaat is een getal.

```
log ( x )
```

### Voorbeelden en resultaten:

`log(3)` retourneert 1,0986

### **log10**

De algemene logaritme (grondtal 10) van **x**. De functie is alleen gedefinieerd als  $x > 0$ . Het resultaat is een getal.

```
log10(x )
```

### Voorbeelden en resultaten:

`log10(3)` retourneert 0,4771

### **pow**

Retourneert **x** tot de macht **y**. Het resultaat is een getal.

```
pow(x, y )
```

### Voorbeelden en resultaten:

`pow(3, 3)` retourneert 27

### **sqr**

**x** kwadraat (x tot de macht 2). Het resultaat is een getal.

```
sqr(x )
```

### Voorbeelden en resultaten:

`sqr(3)` retourneert 9

### **sqrt**

Vierkantswortel van **x**. De functie is alleen gedefinieerd als  $x \geq 0$ . Het resultaat is een positief getal.

```
sqrt(x )
```

### Voorbeelden en resultaten:

`sqrt(3)` retourneert 1,732

## 5.9 Veldfuncties

Deze functies kunnen alleen worden gebruikt in diagramuitdrukkingen.

Veldfuncties moeten gehele getallen of tekenreeksen retourneren waarin verschillende aspecten van veldselecties worden geïdentificeerd.

### Tellerfuncties

GetAlternativeCount

**GetAlternativeCount()** wordt gebruikt om het aantal alternatieve (lichtgrijze) waarden in het vastgestelde veld te zoeken.

```
GetAlternativeCount - diagramfunctie (field_name)
```

GetExcludedCount

**GetExcludedCount()** vindt het aantal uitgesloten distinctieve waarden in het vastgestelde veld. Tot de uitgesloten waarden behoren alternatieve (lichtgrijze), uitgesloten (donkergrijze) en geselecteerde uitgesloten (donkergrijze met vinkje) velden.

```
GetExcludedCount - diagramfunctie (page 1187) (field_name)
```

GetNotSelectedCount

Deze diagramfunctie retourneert het aantal niet-geselecteerde waarden in het veld met de naam **fieldname**. Deze functie is alleen zinvol als de EN-modus is geactiveerd voor het veld.

```
GetNotSelectedCount - diagramfunctie(fieldname [, includeexcluded=false])
```

GetPossibleCount

**GetPossibleCount()** wordt gebruikt om het aantal mogelijke waarden in het vastgestelde veld te zoeken. Als het vastgestelde veld selecties bevat, worden de geselecteerde (groene) velden geteld. Anders worden de geassocieerde (witte) waarden geteld.

```
GetPossibleCount - diagramfunctie(field_name)
```

GetSelectedCount

**GetSelectedCount()** retourneert het aantal geselecteerde (groene) waarden in een veld.

```
GetSelectedCount - diagramfunctie (field_name [, include_excluded])
```

### Veld- en selectiefuncties

GetCurrentSelections

**GetCurrentSelections()** retourneert een lijst van de huidige selecties in de app. Als de selecties in plaats daarvan zijn gemaakt met een zoekreeks in een zoekvak, retourneert **GetCurrentSelections()** de zoekreeks.

```
GetCurrentSelections - diagramfunctie([record_sep [,tag_sep [,value_sep  
[,max_values]]]])
```

GetFieldSelections

**GetFieldSelections()** retourneert een **tekenreeks** met de huidige selecties in een veld.

```
GetFieldSelections - diagramfunctie ( field_name [, value_sep [, max_  
values]])
```

GetObjectDimension

**GetObjectDimension()** retourneert de naam van de dimensie. **Index** is een optioneel geheel getal dat de dimensie aangeeft die moet worden geretourneerd.

```
GetObjectDimension - diagramfunctie ([index])
```

GetObjectField

**GetObjectField()** retourneert de naam van de dimensie. **Index** is een optioneel geheel getal dat de dimensie aangeeft die moet worden geretourneerd.

```
GetObjectField - diagramfunctie ([index])
```

GetObjectMeasure

**GetObjectMeasure()** retourneert de naam van de meting. **Index** is een optioneel geheel getal dat de meting aangeeft die moet worden geretourneerd.

```
GetObjectMeasure - diagramfunctie ([index])
```

### GetAlternativeCount - diagramfunctie

**GetAlternativeCount()** wordt gebruikt om het aantal alternatieve (lichtgrijze) waarden in het vastgestelde veld te zoeken.

**Syntaxis:**

```
GetAlternativeCount (field_name)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

Argument	Beschrijving
field_name	Het veld met het bereik van gegevens die moeten worden gemeten.

**Voorbeelden en resultaten:**

In het volgende voorbeeld wordt het veld **First name** gebruikt, geladen in een filtervak.

Voorbeelden en resultaten

Voorbeelden	Resultaten
Stel dat <b>John</b> wordt geselecteerd in <b>First name</b> . <code>GetAlternativeCount ([First name])</code>	4 aangezien er 4 unieke en uitgesloten (grijze) waarden zijn in <b>First name</b> .
Stel dat <b>John</b> en <b>Peter</b> worden geselecteerd. <code>GetAlternativeCount ([First name])</code>	3 aangezien er 3 unieke en uitgesloten (grijze) waarden zijn in <b>First name</b> .



Voorbeelden	Resultaten
<p>Stel dat er geen waarden worden geselecteerd in <b>First name</b>.</p> <p><code>GetAlternativeCount ([First name])</code></p>	0 aangezien er geen selecties zijn.

Gebruikte gegevens in voorbeeld:

```
Names:
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

### GetCurrentSelections - diagramfunctie

**GetCurrentSelections()** retourneert een lijst van de huidige selecties in de app. Als de selecties in plaats daarvan zijn gemaakt met een zoekreeks in een zoekvak, retourneert **GetCurrentSelections()** de zoekreeks.

Als opties worden gebruikt, moet u `record_sep` opgeven. U kunt een nieuw regel opgeven door **record\_sep** in te stellen op `chr(13)&chr(10)`.

Als alle waarden op één of twee na worden geselecteerd, wordt respectievelijk de indeling 'NOT x,y' of 'NOT y' gebruikt. Als u alle waarden selecteert en het aantal waarden is groter dan `max_values`, wordt de tekst ALL geretourneerd.

#### Syntaxis:

```
GetCurrentSelections ([record_sep [, tag_sep [, value_sep [, max_values [, state_name]]]])
```

**Retourgegevenstypen:** tekenreeks

#### Argumenten:

##### Argumenten

Argumenten	Beschrijving
<code>record_sep</code>	Het scheidingsteken dat tussen veldrecords moet worden geplaatst. De standaardinstelling is <CR><LF>, oftewel een nieuwe regel.
<code>tag_sep</code>	Het scheidingsteken dat tussen de tag van de veldnaam en de veldwaarden moet worden geplaatst. De standaardwaarde is ' '.
<code>value_sep</code>	Het scheidingsteken dat tussen twee veldwaarden moet worden geplaatst. De standaardwaarde is ','.

Argumenten	Beschrijving
max_values	Het maximale aantal veldwaarden dat afzonderlijk moet worden vermeld. Als een groter aantal waarden wordt geselecteerd, wordt de notatie 'x van y waarden' gebruikt. De standaardwaarde is 6.
state_name	De naam van een alternatieve state die voor de specifieke visualisatie is gekozen. Als het argument <b>state_name</b> wordt gebruikt, wordt alleen rekening gehouden met de selecties die zijn gekoppeld aan de opgegeven statenaam.

### Voorbeelden en resultaten:

In het volgende voorbeeld wordt gebruikgemaakt van twee velden die in verschillende filtervakken worden geladen, één voor **First name** en één voor **Initials**.

#### Voorbeelden en resultaten

Voorbeelden	Resultaten
Stel dat <b>John</b> wordt geselecteerd in <b>First name</b> . <code>GetCurrentSelections ()</code>	'First name: John'
Stel dat <b>John</b> en <b>Peter</b> worden geselecteerd in <b>First name</b> . <code>GetCurrentSelections ()</code>	'First name: John, Peter'
Stel dat <b>John</b> en <b>Peter</b> worden geselecteerd in <b>First name</b> en <b>JA</b> in <b>Initials</b> wordt geselecteerd. <code>GetCurrentSelections ()</code>	'First name: John, Peter Initials: JA'
Stel dat <b>John</b> wordt geselecteerd in <b>First name</b> en <b>JA</b> in <b>Initials</b> . <code>GetCurrentSelections ( chr(13)&amp;chr(10) , ' = ' )</code>	'First name = John Initials = JA'
Stel dat u alle namen behalve Sue hebt geselecteerd in <b>First name</b> en in <b>Initials</b> niets is geselecteerd. <code>GetCurrentSelections (chr(13)&amp;chr(10), '=', ', ', 3)</code>	'First name=NOT Sue'

### Gebruikte gegevens in voorbeeld:

```
Names:
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

## GetExcludedCount - diagramfunctie

**GetExcludedCount()** vindt het aantal uitgesloten distinctieve waarden in het vastgestelde veld. Tot de uitgesloten waarden behoren alternatieve (lichtgrijze), uitgesloten (donkergrijze) en geselecteerde uitgesloten (donkergrijze met vinkje) velden.

### Syntaxis:

```
GetExcludedCount (field_name)
```

**Retourgegevenstypen:** tekenreeks

### Argumenten:

#### Argumenten

Argumenten	Beschrijving
field_name	Het veld met het bereik van gegevens die moeten worden gemeten.

### Voorbeelden en resultaten:

In het volgende voorbeeld wordt gebruikgemaakt van drie velden die in verschillende filtervakken worden geladen, één voor **First name**, één voor **Last name** en één voor **Initials**.

#### Voorbeelden en resultaten

Voorbeelden	Resultaten
Als er geen waarden worden geselecteerd in <b>First name</b> .	GetExcludedCount (Initials) = 0 Er zijn geen selecties.
Als <b>John</b> wordt geselecteerd in <b>First name</b> .	GetExcludedCount (Initials) = 5 Er zijn 5 uitgesloten waarden met donkergrijze kleur in <b>Initials</b> . De zesde cel (JA) is wit, aangezien deze is verbonden met de selectie John in <b>First name</b> .
Als <b>John</b> en <b>Peter</b> worden geselecteerd.	GetExcludedCount (Initials) = 3 John is verbonden met 1 waarde en Peter is verbonden met 2 waarden, in <b>Initials</b> .
Als <b>John</b> en <b>Peter</b> worden geselecteerd in <b>First name</b> en vervolgens <b>Franc</b> wordt geselecteerd in <b>Last name</b> .	GetExcludedCount ([First name]) = 4 Er zijn 4 uitgesloten waarden met donkergrijze kleur in <b>First name</b> . <b>GetExcludedCount()</b> is een evaluatie van velden met uitgesloten waarden, waaronder alternatieve en geselecteerde uitgesloten velden.
Als <b>John</b> en <b>Peter</b> worden geselecteerd in <b>First name</b> en vervolgens <b>Franc</b> en <b>Anderson</b> worden geselecteerd in <b>Last name</b> .	GetExcludedCount (Initials) = 4 Er zijn 4 uitgesloten waarden met donkergrijze kleur in <b>Initials</b> . De overige twee cellen (JA en PF) zijn wit, aangezien ze zijn verbonden met de selecties John en Peter in <b>First name</b> .

Voorbeelden	Resultaten
Als <b>John</b> en <b>Peter</b> worden geselecteerd in <b>First name</b> en vervolgens <b>Franc</b> en <b>Anderson</b> worden geselecteerd in <b>Last name</b> .	GetExcludedCount ([Last name]) = 4 Er zijn 4 uitgesloten waarden in <b>Initials</b> . Devonshire heeft een lichtgrijze kleur, terwijl Brown, Carr en Elliot een donkergrijze kleur hebben.

Gebruikte gegevens in voorbeeld:

```
Names:
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

### GetFieldSelections - diagramfunctie

**GetFieldSelections()** retourneert een **tekenreeks** met de huidige selecties in een veld.

Als alle waarden op één of twee na worden geselecteerd, wordt respectievelijk de indeling 'NOT x,y' of 'NOT y' gebruikt. Als u alle waarden selecteert en het aantal waarden is groter dan max\_values, wordt de tekst ALL geretourneerd.

**Syntaxis:**

```
GetFieldSelections ( field_name [, value_sep [, max_values [, state_name]])
```

**Retourgegevenstypen:** tekenreeks

#### Retourtekenreeksindelingen

Notatie	Beschrijving
'a, b, c'	Als het aantal geselecteerde waarden max_values of minder is, is de tekenreeks die wordt geretourneerd, een lijst met de geselecteerde waarden.  De waarden worden gescheiden met value_sep als scheidingstekens.
'NOT a, b, c'	Als het aantal niet-geselecteerde waarden max_values of minder is, is de tekenreeks die wordt geretourneerd, een lijst met niet-geselecteerde waarden met NOT als een voorvoegsel.  De waarden worden gescheiden met value_sep als scheidingstekens.
'x of y'	x = het aantal geselecteerde waarden  y = het totale aantal waarden  Dit wordt geretourneerd wanneer max_values < x < (y - max_values).

Notatie	Beschrijving
'ALL'	Geretourneerd als alle waarden zijn geselecteerd.
'.'	Geretourneerd als geen waarde is geselecteerd.
<search string>	Als u een selectie hebt gemaakt met behulp van een zoekopdracht, wordt de zoekreeks geretourneerd.

### Argumenten:

#### Argumenten

Argumenten	Beschrijving
field_name	Het veld met het bereik van gegevens die moeten worden gemeten.
value_sep	Het scheidingsteken dat tussen twee veldwaarden moet worden geplaatst. De standaardwaarde is ','.
max_values	Het maximale aantal veldwaarden dat afzonderlijk moet worden vermeld. Als een groter aantal waarden wordt geselecteerd, wordt de notatie 'x van y waarden' gebruikt. De standaardwaarde is 6.
state_name	De naam van een alternatieve state die voor de specifieke visualisatie is gekozen. Als het argument <b>state_name</b> wordt gebruikt, wordt alleen rekening gehouden met de selecties die zijn gekoppeld aan de opgegeven statenaam.

### Voorbeelden en resultaten:

In het volgende voorbeeld wordt het veld **First name** gebruikt, geladen in een filtervak.

#### Voorbeelden en resultaten

Voorbeelden	Resultaten
Stel dat <b>John</b> wordt geselecteerd in <b>First name</b> .  GetFieldSelections ([First name])	'John'
Stel dat <b>John</b> en <b>Peter</b> worden geselecteerd.  GetFieldSelections ([First name])	'John,Peter'
Stel dat <b>John</b> en <b>Peter</b> worden geselecteerd.  GetFieldSelections ([First name],';')	'John; Peter'

Voorbeelden	Resultaten
<p>Stel dat <b>John, Sue, Mark</b> worden geselecteerd in <b>First name</b>.</p> <pre>GetFieldSelections ([First name],',';2)</pre>	<p>'NOT Jane;Peter', omdat de waarde 2 is aangegeven als de waarde van het argument <code>max_values</code>. Anders zou het resultaat John; Sue; Mark. zijn geweest.</p>

Gebruikte gegevens in voorbeeld:

```
Names:
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

### GetNotSelectedCount - diagramfunctie

Deze diagramfunctie retourneert het aantal niet-geselecteerde waarden in het veld met de naam **fieldname**. Deze functie is alleen zinvol als de EN-modus is geactiveerd voor het veld.

#### Syntaxis:

```
GetNotSelectedCount(fieldname [, includeexcluded=false])
```

#### Argumenten:

##### Argumenten

Argument	Beschrijving
fieldname	De naam van het veld dat moet worden geëvalueerd.
includeexcluded	Als <b>includeexcluded</b> de waarde True heeft, worden ook de geselecteerde waarden die door selecties in een ander veld zijn uitgesloten meegeteld.

#### Voorbeelden:

```
GetNotSelectedCount( Country )
GetNotSelectedCount( Country, true )
```

### GetObjectDimension - diagramfunctie

**GetObjectDimension()** retourneert de naam van de dimensie. **Index** is een optioneel geheel getal dat de dimensie aangeeft die moet worden geretourneerd.



*U kunt deze functie niet gebruiken in een diagram in de volgende locaties: titel, subtitel, voettekst, uitdrukking referentielijn.*



*U kunt niet naar de naam van een dimensie of meting in een ander object verwijzen met behulp van de Object ID.*

### Syntaxis:

```
GetObjectDimension ([index])
```

### Voorbeeld:

```
GetObjectDimension(1)
```

Voorbeeld: Diagramuitdrukking

*Qlik Sense-tabel met voorbeelden van de functie GetObjectDimension in een diagramuitdrukking*

transactio n_date	custome r_id	transactio n_quantity	=GetObjectDimen sion ()	=GetObjectDimen sion (0)	=GetObjectDimen sion (1)
2018/08/3 0	049681	13	transaction_date	transaction_date	customer_id
2018/08/3 0	203521	6	transaction_date	transaction_date	customer_id
2018/08/3 0	203521	21	transaction_date	transaction_date	customer_id

Als u de naam van een meting wilt retourneren, gebruikt u de functie **GetObjectMeasure**.

## GetObjectField - diagramfunctie

**GetObjectField()** retourneert de naam van de dimensie. **Index** is een optioneel geheel getal dat de dimensie aangeeft die moet worden geretourneerd.



*U kunt deze functie niet gebruiken in een diagram in de volgende locaties: titel, subtitel, voettekst, uitdrukking referentielijn.*



*U kunt niet naar de naam van een dimensie of meting in een ander object verwijzen met behulp van de Object ID.*

### Syntaxis:

```
GetObjectField ([index])
```

### Voorbeeld:

GetObjectField(1)

Voorbeeld: Diagramuitdrukking

Klik Sense-tabel met voorbeelden van de functie GetObjectField in een diagramuitdrukking.

transaction_date	customer_id	transaction_quantity	=GetObjectField ()	=GetObjectField (0)	=GetObjectField (1)
2018/08/30	049681	13	transaction_date	transaction_date	customer_id
2018/08/30	203521	6	transaction_date	transaction_date	customer_id
2018/08/30	203521	21	transaction_date	transaction_date	customer_id

Als u de naam van een meting wilt retourneren, gebruikt u de functie **GetObjectMeasure**.

### GetObjectMeasure - diagramfunctie

**GetObjectMeasure()** retourneert de naam van de meting. **Index** is een optioneel geheel getal dat de meting aangeeft die moet worden geretourneerd.



*U kunt deze functie niet gebruiken in een diagram in de volgende locaties: titel, subtitel, voettekst, uitdrukking referentielijn.*



*U kunt niet naar de naam van een dimensie of meting in een ander object verwijzen met behulp van de Object ID.*

### Syntaxis:

```
GetObjectMeasure ([index])
```

### Voorbeeld:

GetObjectMeasure(1)

Voorbeeld: Diagramuitdrukking

Klik Sense-tabel met voorbeelden van de functie GetObjectMeasure in een diagramuitdrukking

customer_id	sum (transaction_quantity)	Avg (transaction_quantity)	=GetObjectMeasure ()	=GetObjectMeasure(0)	=GetObjectMeasure(1)
49681	13	13	sum(transaction_quantity)	sum(transaction_quantity)	Avg(transaction_quantity)
203521	27	13.5	sum(transaction_quantity)	sum(transaction_quantity)	Avg(transaction_quantity)



Als u de naam van een dimensie wilt retourneren, gebruikt u de functie **GetObjectField**.

### GetPossibleCount - diagramfunctie

**GetPossibleCount()** wordt gebruikt om het aantal mogelijke waarden in het vastgestelde veld te zoeken. Als het vastgestelde veld selecties bevat, worden de geselecteerde (groene) velden geteld. Anders worden de geassocieerde (witte) waarden geteld. .

Voor velden met selecties retourneert **GetPossibleCount()** het aantal geselecteerde (groene) velden.

**Retourgegevenstypen:** geheel getal

#### Syntaxis:

```
GetPossibleCount (field_name)
```

#### Argumenten:

Argumenten

Argumenten	Beschrijving
field_name	Het veld met het bereik van gegevens die moeten worden gemeten.

#### Voorbeelden en resultaten:

In het volgende voorbeeld wordt gebruikgemaakt van twee velden die in verschillende filtervakken worden geladen, één voor **First name** en één voor **Initials**.

Voorbeelden en resultaten

Voorbeelden	Resultaten
Stel dat <b>John</b> wordt geselecteerd in <b>First name</b> .  GetPossibleCount ([Initials])	1 aangezien er 1 waarde in Initials is gekoppeld aan de selectie, <b>John</b> , in <b>First name</b> .
Stel dat <b>John</b> wordt geselecteerd in <b>First name</b> .  GetPossibleCount ([First name])	1 aangezien er 1 selectie, <b>John</b> , is in <b>First name</b> .
Stel dat <b>Peter</b> wordt geselecteerd in <b>First name</b> .  GetPossibleCount ([Initials])	2 aangezien Peter is verbonden met 2 waarden in <b>Initials</b> .
Stel dat er geen waarden worden geselecteerd in <b>First name</b> .  GetPossibleCount ([First name])	5 aangezien er geen selecties zijn en er 5 unieke waarden zijn in <b>First name</b> .
Stel dat er geen waarden worden geselecteerd in <b>First name</b> .  GetPossibleCount ([Initials])	6 aangezien er geen selecties zijn en er 6 unieke waarden zijn in <b>Initials</b> .

Gebruikte gegevens in voorbeeld:

```
Names:
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

### GetSelectedCount - diagramfunctie

**GetSelectedCount()** retourneert het aantal geselecteerde (groene) waarden in een veld.

**Syntaxis:**

```
GetSelectedCount (field_name [, include_excluded [, state_name]])
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

#### Argumenten

Argumenten	Beschrijving
field_name	Het veld met het bereik van gegevens die moeten worden gemeten.
include_excluded	Als de parameter is ingesteld op <b>True()</b> , worden ook geselecteerde waarden meegeteld die momenteel worden uitgesloten door selecties in andere velden. Als de parameter is ingesteld op <b>False</b> of wordt weggelaten, worden deze waarden niet opgenomen.
state_name	De naam van een alternatieve state die voor de specifieke visualisatie is gekozen. Als het argument <b>state_name</b> wordt gebruikt, wordt alleen rekening gehouden met de selecties die zijn gekoppeld aan de opgegeven statenaam.

**Voorbeelden en resultaten:**

In het volgende voorbeeld wordt gebruikgemaakt van drie velden die in verschillende filtervakken worden geladen, één voor **First name**, één voor **Initials** en één voor **Has cellphone**.

#### Voorbeelden en resultaten

Voorbeelden	Resultaten
Stel dat <b>John</b> wordt geselecteerd in <b>First name</b> .  <code>GetSelectedCount ([First name])</code>	1 aangezien één waarde is geselecteerd in <b>First name</b> .

Voorbeelden	Resultaten
<p>Stel dat <b>John</b> wordt geselecteerd in <b>First name</b>.</p> <pre>getSelectedCount ([Initials])</pre>	<p>0 aangezien er geen waarden zijn geselecteerd in <b>Initials</b>.</p>
<p>Met geen selecties in <b>.First name</b>, selecteert u alle waarden in <b>Initials</b> en selecteert u vervolgens de waarde <b>Yes</b> in <b>Has cellphone</b>.</p> <pre>getSelectedCount ([Initials], True())</pre>	<p>Hoewel bij selecties met <b>InitialsMC</b> en <b>PD Has cellphone</b> is ingesteld op <b>No</b>, is het resultaat nog steeds 6, omdat het argument <code>include_excluded</code> is ingesteld op <code>True()</code>.</p>

Gebruikte gegevens in voorbeeld:

```
Names:
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

### 5.10 Bestandsfuncties

De bestandsfuncties (alleen beschikbaar in scriptuitdrukkingen) retourneren informatie over het tabelbestand dat op dat moment wordt gelezen. Deze functies retourneren een voor alle gegevensbronnen, behalve tabelbestanden (uitzondering: `.NULL ConnectString`).

#### Overzicht van bestandsfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

##### Attribute

Deze scriptfunctie retourneert de waarde van de metatags van verschillende mediabestanden als tekst. De volgende bestandsindelingen worden ondersteund: MP3, WMA, WMV, PNG en JPG. Als het bestand **filename** niet bestaat, niet een ondersteunde bestandsindeling heeft of niet een metatag bevat met de naam **attributename**, wordt NULL geretourneerd.

```
Attribute (filename, attributename)
```

##### ConnectString

De functie **ConnectString()** retourneert de naam van de actieve gegevensverbinding voor ODBC- of OLE DB-verbindingen. De functie retourneert een lege tekenreeks als geen **connect**-opdracht is uitgevoerd of na een **disconnect**-opdracht.

**ConnectionString** ()

### **FileBaseName**

De functie **FileBaseName** retourneert een tekenreeks met de naam van het tabelbestand dat op dat moment wordt gelezen, zonder pad of extensie.

**FileBaseName** ()

### **FileDir**

De functie **FileDir** retourneert een tekenreeks met het pad naar de map van het tabelbestand dat op dat moment wordt gelezen.

**FileDir** ()

### **FileExtension**

De functie **FileExtension** retourneert een tekenreeks met de extensie van het tabelbestand dat op dat moment wordt gelezen.

**FileExtension** ()

### **FileName**

De functie **FileName** retourneert een tekenreeks met de naam van het tabelbestand dat op dat moment wordt gelezen, zonder pad maar met de extensie.

**FileName** ()

### **FilePath**

De functie **FilePath** retourneert een tekenreeks met het volledige pad naar het tabelbestand dat op dat moment wordt gelezen.

**FilePath** ()

### **FileSize**

De functie retourneert een geheel getal met de grootte in bytes van het bestand of, als geen is opgegeven, van het tabelbestand dat op dat moment wordt gelezen.  
filenamefilename

**FileSize** ()

### **FileTime**

De functie **FileTime** retourneert een tijdstempel in UTC voor de datum en tijd van de laatste wijziging van het bestand filename. Als geen filename is opgegeven, wordt het tabelbestand gebruikt dat op dat moment wordt gelezen.

**FileTime** ([ filename ])

### **GetFolderPath**

De functie **GetFolderPath** retourneert de waarde van de functie Microsoft Windows *SHGetFolderPath*. Deze functie neemt de naam van een map in Microsoft Windows als invoer en retourneert het volledige pad van de map.

**GetFolderPath** ()

### QvdCreateTime

Deze scriptfunctie retourneert de tijdstempel voor de XML-koptekst uit een QVD-bestand, indien aanwezig. Anders wordt NULL geretourneerd. In de tijdstempel wordt tijd weergegeven in UTC.

```
QvdCreateTime (filename)
```

### QvdFieldName

Deze script-functie retourneert de naam van veldnummer **fieldno** in een QVD-bestand. Als het veld niet bestaat, wordt NULL geretourneerd.

```
QvdFieldName (filename , fieldno)
```

### QvdNoOfFields

Deze scriptfunctie retourneert het aantal velden in een QVD-bestand.

```
QvdNoOfFields (filename)
```

### QvdNoOfRecords

Deze scriptfunctie retourneert het huidige aantal records in een QVD-bestand.

```
QvdNoOfRecords (filename)
```

### QvdTableName

Deze scriptfunctie retourneert de naam van de tabel die is opgeslagen in een QVD-bestand.

```
QvdTableName (filename)
```

## Attribute

Deze scriptfunctie retourneert de waarde van de metatags van verschillende mediabestanden als tekst. De volgende bestandsindelingen worden ondersteund: MP3, WMA, WMV, PNG en JPG. Als het bestand **filename** niet bestaat, niet een ondersteunde bestandsindeling heeft of niet een metatag bevat met de naam **attributename**, wordt NULL geretourneerd.

### Syntaxis:

```
Attribute(filename, attributename)
```

Een groot aantal metatags kan worden gelezen. De voorbeelden in dit onderwerp laten zien welke tags kunnen worden gelezen voor de respectievelijke ondersteunde bestandstypen.



*U kunt alleen metatags lezen die volgens de relevante specificatie in het bestand zijn opgeslagen, bijvoorbeeld ID2v3 voor MP3-bestanden of EXIF voor JPG-bestanden. U kunt geen meta-informatie die is opgeslagen in **Windows Bestandsbeheer** lezen.*

### Argumenten:

#### Argumenten

Argument	Beschrijving
filename	<p>De naam van een mediabestand, inclusief pad, indien nodig, als mapgegevensverbinding.</p> <p><b>Voorbeeld: 'lib://Table Files'</b></p> <p>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:</p> <ul style="list-style-type: none"> <li>• absoluut</li> </ul> <p><b>Voorbeeld: c:\data\</b></p> <ul style="list-style-type: none"> <li>• relatief ten opzichte van de werkmap van de Qlik Sense-app.</li> </ul> <p><b>Voorbeeld: data\</b></p>
attributenam	De naam van een metatag.

In de voorbeelden wordt de functie **GetFolderPath** gebruikt om de paden naar mediabestanden te vinden. Aangezien **GetFolderPath** alleen wordt ondersteund in de bestaande modus, moet u de verwijzingen naar **GetFolderPath** vervangen door een lib:// gegevensverbindingspad als u deze functie gebruikt in de standaardmodus of in Qlik Sense SaaS.

*Beperking van toegang tot bestandssysteem (page 1464)*

### Example 1: MP3-bestanden

Dit script leest alle mogelijke MP3-metatags in de map *MyMusic*.

```
// Script to read MP3 meta tags for each vExt in 'mp3' for each vFoundFile in filelist(
GetFolderPath('MyMusic') & '\*.' & vExt ) FileList: LOAD FileLongName, subfield
(FileLongName,'\',-1) as FileShortName, num(FileSize(FileLongName),'# ### ### ##',',',',',
') as FileSize, FileTime(FileLongName) as FileTime, // ID3v1.0 and ID3v1.1 tags
Attribute(FileLongName, 'Title') as Title, Attribute(FileLongName, 'Artist') as Artist,
Attribute(FileLongName, 'Album') as Album, Attribute(FileLongName, 'Year') as Year,
Attribute(FileLongName, 'Comment') as Comment, Attribute(FileLongName, 'Track') as Track,
Attribute(FileLongName, 'Genre') as Genre,

// ID3v2.3 tags Attribute(FileLongName, 'AENC') as AENC, // Audio encryption
Attribute(FileLongName, 'APIC') as APIC, // Attached picture Attribute(FileLongName,
'COMM') as COMM, // Comments Attribute(FileLongName, 'COMR') as COMR, // Commercial frame
Attribute(FileLongName, 'ENCR') as ENCR, // Encryption method registration Attribute
(FileLongName, 'EQUA') as EQUA, // Equalization Attribute(FileLongName, 'ETCO') as ETCO,
// Event timing codes Attribute(FileLongName, 'GEOB') as GEOB, // General encapsulated
object Attribute(FileLongName, 'GRID') as GRID, // Group identification registration
Attribute(FileLongName, 'IPLS') as IPLS, // Involved people list Attribute(FileLongName,
'LINK') as LINK, // Linked information Attribute(FileLongName, 'MCDI') as MCDI, // Music
```

## 5 Script- en diagramfuncties

---

CD identifier      Attribute(FileLongName, 'MLLT') as MLLT, // MPEG location lookup table  
Attribute(FileLongName, 'OWNE') as OWNE, // Ownership frame      Attribute(FileLongName,  
'PRIV') as PRIV, // Private frame      Attribute(FileLongName, 'PCNT') as PCNT, // Play counter  
Attribute(FileLongName, 'POPM') as POPM, // Popularimeter

Attribute(FileLongName, 'POSS') as POSS, // Position synchronisation frame      Attribute  
(FileLongName, 'RBUF') as RBUF, // Recommended buffer size      Attribute(FileLongName, 'RVAD')  
as RVAD, // Relative volume adjustment      Attribute(FileLongName, 'RVRB') as RVRB, // Reverb  
Attribute(FileLongName, 'SYLT') as SYLT, // Synchronized lyric/text      Attribute  
(FileLongName, 'SYTC') as SYTC, // Synchronized tempo codes      Attribute(FileLongName,  
'TALB') as TALB, // Album/Movie/Show title      Attribute(FileLongName, 'TBPM') as TBPM, // BPM  
(beats per minute)      Attribute(FileLongName, 'TCOM') as TCOM, // Composer      Attribute  
(FileLongName, 'TCON') as TCON, // Content type      Attribute(FileLongName, 'TCOP') as TCOP,  
// Copyright message      Attribute(FileLongName, 'TDAT') as TDAT, // Date      Attribute  
(FileLongName, 'TDLY') as TDLY, // Playlist delay

Attribute(FileLongName, 'TENC') as TENC, // Encoded by      Attribute(FileLongName,  
'TEXT') as TEXT, // Lyricist/Text writer      Attribute(FileLongName, 'TFLT') as TFLT, // File  
type      Attribute(FileLongName, 'TIME') as TIME, // Time      Attribute(FileLongName, 'TIT1')  
as TIT1, // Content group description      Attribute(FileLongName, 'TIT2') as TIT2, //  
Title/songname/content description      Attribute(FileLongName, 'TIT3') as TIT3, //  
Subtitle/Description refinement      Attribute(FileLongName, 'TKEY') as TKEY, // Initial key  
Attribute(FileLongName, 'TLAN') as TLAN, // Language(s)      Attribute(FileLongName, 'TLEN')  
as TLEN, // Length      Attribute(FileLongName, 'TMED') as TMED, // Media type

Attribute(FileLongName, 'TOAL') as TOAL, // Original album/movie/show title      Attribute  
(FileLongName, 'TOFN') as TOFN, // Original filename      Attribute(FileLongName, 'TOLY') as  
TOLY, // Original lyricist(s)/text writer(s)      Attribute(FileLongName, 'TOPE') as TOPE, //  
Original artist(s)/performer(s)      Attribute(FileLongName, 'TORY') as TORY, // Original  
release year      Attribute(FileLongName, 'TOWN') as TOWN, // File owner/licensee      Attribute  
(FileLongName, 'TPE1') as TPE1, // Lead performer(s)/Soloist(s)      Attribute(FileLongName,  
'TPE2') as TPE2, // Band/orchestra/accompaniment

Attribute(FileLongName, 'TPE3') as TPE3, // Conductor/performer refinement      Attribute  
(FileLongName, 'TPE4') as TPE4, // Interpreted, remixed, or otherwise modified by  
Attribute(FileLongName, 'TPOS') as TPOS, // Part of a set      Attribute(FileLongName, 'TPUB')  
as TPUB, // Publisher      Attribute(FileLongName, 'TRCK') as TRCK, // Track number/Position in  
set      Attribute(FileLongName, 'TRDA') as TRDA, // Recording dates      Attribute  
(FileLongName, 'TRSN') as TRSN, // Internet radio station name      Attribute(FileLongName,  
'TRSO') as TRSO, // Internet radio station owner

Attribute(FileLongName, 'TSIZ') as TSIZ, // Size      Attribute(FileLongName, 'TSRC') as  
TSRC, // ISRC (international standard recording code)      Attribute(FileLongName, 'TSSE') as  
TSSE, // Software/Hardware and settings used for encoding      Attribute(FileLongName, 'TYER')  
as TYER, // Year      Attribute(FileLongName, 'TXXX') as TXXX, // User defined text information  
frame      Attribute(FileLongName, 'UFID') as UFID, // Unique file identifier      Attribute  
(FileLongName, 'USER') as USER, // Terms of use      Attribute(FileLongName, 'USLT') as USLT,  
// Unsynchronized lyric/text transcription      Attribute(FileLongName, 'WCOP') as WCOP, //  
Commercial information      Attribute(FileLongName, 'WCOP') as WCOP, // Copyright/Legal  
information

Attribute(FileLongName, 'WOAF') as WOAF, // Official audio file webpage      Attribute  
(FileLongName, 'WOAR') as WOAR, // Official artist/performer webpage      Attribute  
(FileLongName, 'WOAS') as WOAS, // Official audio source webpage      Attribute(FileLongName,  
'WORS') as WORS, // Official internet radio station homepage      Attribute(FileLongName,

```
'WPAY') as WPAY, // Payment      Attribute(FileLongName, 'WPUB') as WPUB, // Publishers
official webpage      Attribute(FileLongName, 'WXXX') as WXXX; // User defined URL link frame
LOAD @1:n as FileLongName Inline "$(vFoundFile)" (fix, no labels); Next vFoundFile Next vExt
```

### Example 2: JPEG

Dit script leest alle mogelijke EXIF-metatags uit JPG-bestanden in de map *MyPictures*.

```
// Script to read Jpeg Exif meta tags for each vExt in 'jpg', 'jpeg', 'jpe', 'jfif', 'jif',
'jfi' for each vFoundFile in fileList( GetFolderPath('MyPictures') & '\*.' & vExt )

FileList: LOAD FileLongName,      subfield(FileLongName,'\",-1) as FileShortName,      num
(FileSize(FileLongName),'# ### ### ##',',',' ') as FileSize,      FileTime(FileLongName) as
FileTime,      // ***** Exif Main (IFD0) Attributes *****      Attribute
(FileLongName, 'Imagewidth') as Imagewidth,      Attribute(FileLongName, 'ImageLength') as
ImageLength,      Attribute(FileLongName, 'BitsPerSample') as BitsPerSample,      Attribute
(FileLongName, 'Compression') as Compression,

// examples: 1=uncompressed, 2=CCITT, 3=CCITT 3, 4=CCITT 4,

//5=LZW, 6=JPEG (old style), 7=JPEG, 8=Deflate, 32773=PackBits RLE,      Attribute
(FileLongName, 'PhotometricInterpretation') as PhotometricInterpretation,

// examples: 0=whiteIsZero, 1=BlackIsZero, 2=RGB, 3=Palette, 5=CMYK, 6=YCbCr,
Attribute(FileLongName, 'ImageDescription') as ImageDescription,      Attribute(FileLongName,
'Make') as Make,      Attribute(FileLongName, 'Model') as Model,      Attribute(FileLongName,
'StripOffsets') as StripOffsets,      Attribute(FileLongName, 'Orientation') as Orientation,

// examples: 1=TopLeft, 2=TopRight, 3=BottomRight, 4=BottomLeft,

// 5=LeftTop, 6=RightTop, 7=RightBottom, 8=LeftBottom,      Attribute(FileLongName,
'SamplesPerPixel') as SamplesPerPixel,      Attribute(FileLongName, 'RowsPerStrip') as
RowsPerStrip,      Attribute(FileLongName, 'StripByteCounts') as StripByteCounts,      Attribute
(FileLongName, 'XResolution') as XResolution,      Attribute(FileLongName, 'YResolution') as
YResolution,      Attribute(FileLongName, 'PlanarConfiguration') as PlanarConfiguration,

// examples: 1=chunky format, 2=planar format,      Attribute(FileLongName,
'ResolutionUnit') as ResolutionUnit,

// examples: 1=none, 2=inches, 3=centimeters,      Attribute(FileLongName,
'TransferFunction') as TransferFunction,      Attribute(FileLongName, 'Software') as Software,
Attribute(FileLongName, 'DateTime') as DateTime,      Attribute(FileLongName, 'Artist') as
Artist,      Attribute(FileLongName, 'HostComputer') as HostComputer,      Attribute
(FileLongName, 'WhitePoint') as WhitePoint,      Attribute(FileLongName,
'PrimaryChromaticities') as PrimaryChromaticities,      Attribute(FileLongName,
'YCbCrCoefficients') as YCbCrCoefficients,      Attribute(FileLongName, 'YCbCrSubSampling') as
YCbCrSubSampling,      Attribute(FileLongName, 'YCbCrPositioning') as YCbCrPositioning,

// examples: 1=centered, 2=co-sited,      Attribute(FileLongName, 'ReferenceBlackwhite')
as ReferenceBlackwhite,      Attribute(FileLongName, 'Rating') as Rating,      Attribute
(FileLongName, 'RatingPercent') as RatingPercent,      Attribute(FileLongName,
'ThumbnailFormat') as ThumbnailFormat,
```



## 5 Script- en diagramfuncties

---

```
// examples: 0=Raw Rgb, 1=Jpeg,      Attribute(FileLongName, 'Copyright') as Copyright,
Attribute(FileLongName, 'ExposureTime') as ExposureTime,      Attribute(FileLongName,
'FNumber') as FNumber,      Attribute(FileLongName, 'ExposureProgram') as ExposureProgram,

// examples: 0=Not defined, 1=Manual, 2=Normal program, 3=Aperture priority, 4=Shutter
priority,

// 5=Creative program, 6=Action program, 7=Portrait mode, 8=Landscape mode, 9=Bulb,
Attribute(FileLongName, 'ISOSpeedRatings') as ISOSpeedRatings,      Attribute(FileLongName,
'TimeZoneOffset') as TimeZoneOffset,      Attribute(FileLongName, 'SensitivityType') as
SensitivityType,

// examples: 0=Unknown, 1=Standard output sensitivity (SOS), 2=Recommended exposure index
(REI),

// 3=ISO speed, 4=Standard output sensitivity (SOS) and Recommended exposure index (REI),

//5=Standard output sensitivity (SOS) and ISO Speed, 6=Recommended exposure index (REI)
and ISO Speed,

// 7=Standard output sensitivity (SOS) and Recommended exposure index (REI) and ISO speed,
Attribute(FileLongName, 'ExifVersion') as ExifVersion,      Attribute(FileLongName,
'DateTimeOriginal') as DateTimeOriginal,      Attribute(FileLongName, 'DateTimeDigitized') as
DateTimeDigitized,      Attribute(FileLongName, 'ComponentsConfiguration') as
ComponentsConfiguration,

// examples: 1=Y, 2=Cb, 3=Cr, 4=R, 5=G, 6=B,      Attribute(FileLongName,
'CompressedBitsPerPixel') as CompressedBitsPerPixel,      Attribute(FileLongName,
'ShutterSpeedValue') as ShutterSpeedValue,      Attribute(FileLongName, 'ApertureValue') as
ApertureValue,      Attribute(FileLongName, 'BrightnessValue') as BrightnessValue, //
examples: -1=Unknown,      Attribute(FileLongName, 'ExposureBiasValue') as ExposureBiasValue,
Attribute(FileLongName, 'MaxApertureValue') as MaxApertureValue,      Attribute
(FileLongName, 'SubjectDistance') as SubjectDistance,

// examples: 0=Unknown, -1=Infinity,      Attribute(FileLongName, 'MeteringMode') as
MeteringMode,

// examples: 0=Unknown, 1=Average, 2=CenterWeightedAverage, 3=Spot,

// 4=MultiSpot, 5=Pattern, 6=Partial, 255=Other,      Attribute(FileLongName,
'LightSource') as LightSource,

// examples: 0=Unknown, 1=Daylight, 2=Fluorescent, 3=Tungsten, 4=Flash, 9=Fine weather,

// 10=Cloudy weather, 11=Shade, 12=Daylight fluorescent,

// 13=Day white fluorescent, 14=Cool white fluorescent,

// 15=White fluorescent, 17=Standard light A, 18=Standard light B, 19=Standard light C,

// 20=D55, 21=D65, 22=D75, 23=D50, 24=ISO studio tungsten, 255=other light source,
Attribute(FileLongName, 'Flash') as Flash,      Attribute(FileLongName, 'FocalLength') as
FocalLength,      Attribute(FileLongName, 'SubjectArea') as SubjectArea,      Attribute
(FileLongName, 'MakerNote') as MakerNote,      Attribute(FileLongName, 'UserComment') as
UserComment,      Attribute(FileLongName, 'SubSecTime') as SubSecTime,
```

## 5 Script- en diagramfuncties

---

```
Attribute(FileLongName, 'SubsecTimeOriginal') as SubsecTimeOriginal, Attribute
(FileLongName, 'SubsecTimeDigitized') as SubsecTimeDigitized, Attribute(FileLongName,
'XPTitle') as XPTitle, Attribute(FileLongName, 'XPComment') as XPComment,

Attribute(FileLongName, 'XPAuthor') as XPAuthor, Attribute(FileLongName,
'XPKeywords') as XPKeywords, Attribute(FileLongName, 'XPSubject') as XPSubject,
Attribute(FileLongName, 'FlashpixVersion') as FlashpixVersion, Attribute(FileLongName,
'ColorSpace') as ColorSpace, // examples: 1=sRGB, 65535=Uncalibrated, Attribute
(FileLongName, 'PixelXDimension') as PixelXDimension, Attribute(FileLongName,
'PixelYDimension') as PixelYDimension, Attribute(FileLongName, 'RelatedSoundFile') as
RelatedSoundFile,

Attribute(FileLongName, 'FocalPlaneXResolution') as FocalPlaneXResolution, Attribute
(FileLongName, 'FocalPlaneYResolution') as FocalPlaneYResolution, Attribute(FileLongName,
'FocalPlaneResolutionUnit') as FocalPlaneResolutionUnit,

// examples: 1=None, 2=Inch, 3=Centimeter, Attribute(FileLongName, 'ExposureIndex')
as ExposureIndex, Attribute(FileLongName, 'SensingMethod') as SensingMethod,

// examples: 1=Not defined, 2=One-chip color area sensor, 3=Two-chip color area sensor,

// 4=Three-chip color area sensor, 5=Color sequential area sensor,

// 7=Trilinear sensor, 8=Color sequential linear sensor, Attribute(FileLongName,
'FileSource') as FileSource,

// examples: 0=Other, 1=Scanner of transparent type,

// 2=Scanner of reflex type, 3=Digital still camera, Attribute(FileLongName,
'SceneType') as SceneType,

// examples: 1=A directly photographed image, Attribute(FileLongName, 'CFAPattern')
as CFAPattern, Attribute(FileLongName, 'CustomRendered') as CustomRendered,

// examples: 0=Normal process, 1=Custom process, Attribute(FileLongName,
'ExposureMode') as ExposureMode,

// examples: 0=Auto exposure, 1=Manual exposure, 2=Auto bracket, Attribute
(FileLongName, 'whiteBalance') as whiteBalance,

// examples: 0=Auto white balance, 1=Manual white balance, Attribute(FileLongName,
'DigitalZoomRatio') as DigitalZoomRatio, Attribute(FileLongName, 'FocalLengthIn35mmFilm')
as FocalLengthIn35mmFilm, Attribute(FileLongName, 'SceneCaptureType') as SceneCaptureType,

// examples: 0=Standard, 1=Landscape, 2=Portrait, 3=Night scene, Attribute
(FileLongName, 'GainControl') as GainControl,

// examples: 0=None, 1=Low gain up, 2=High gain up, 3=Low gain down, 4=High gain down,
Attribute(FileLongName, 'Contrast') as Contrast,

// examples: 0=Normal, 1=Soft, 2=Hard, Attribute(FileLongName, 'Saturation') as
Saturation,

// examples: 0=Normal, 1=Low saturation, 2=High saturation, Attribute(FileLongName,
'Sharpness') as Sharpness,
```

## 5 Script- en diagramfuncties

---

```
// examples: 0=Normal, 1=Soft, 2=Hard,      Attribute(FileLongName,
'SubjectDistanceRange') as SubjectDistanceRange,

// examples: 0=Unknown, 1=Macro, 2=Close view, 3=Distant view,      Attribute
(FileLongName, 'ImageUniqueID') as ImageUniqueID,      Attribute(FileLongName,
'BodySerialNumber') as BodySerialNumber,      Attribute(FileLongName, 'CMNT_GAMMA') as CMNT_
GAMMA,      Attribute(FileLongName, 'PrintImageMatching') as PrintImageMatching,      Attribute
(FileLongName, 'OffsetSchema') as OffsetSchema,

// ***** Interoperability Attributes *****      Attribute(FileLongName,
'InteroperabilityIndex') as InteroperabilityIndex,      Attribute(FileLongName,
'InteroperabilityVersion') as InteroperabilityVersion,      Attribute(FileLongName,
'InteroperabilityRelatedImageFileFormat') as InteroperabilityRelatedImageFileFormat,
Attribute(FileLongName, 'InteroperabilityRelatedImageWidth') as
InteroperabilityRelatedImageWidth,      Attribute(FileLongName,
'InteroperabilityRelatedImageLength') as InteroperabilityRelatedImageLength,      Attribute
(FileLongName, 'InteroperabilityColorSpace') as InteroperabilityColorSpace,

// examples: 1=sRGB, 65535=Uncalibrated,      Attribute(FileLongName,
'InteroperabilityPrintImageMatching') as InteroperabilityPrintImageMatching,      //
***** GPS Attributes *****      Attribute(FileLongName, 'GPSVersionID') as
GPSVersionID,      Attribute(FileLongName, 'GPSLatitudeRef') as GPSLatitudeRef,      Attribute
(FileLongName, 'GPSLatitude') as GPSLatitude,      Attribute(FileLongName, 'GPSLongitudeRef')
as GPSLongitudeRef,      Attribute(FileLongName, 'GPSLongitude') as GPSLongitude,      Attribute
(FileLongName, 'GPSAltitudeRef') as GPSAltitudeRef,

// examples: 0=Above sea level, 1=Below sea level,      Attribute(FileLongName,
'GPSAltitude') as GPSAltitude,      Attribute(FileLongName, 'GPSTimeStamp') as GPSTimeStamp,
Attribute(FileLongName, 'GPSSatellites') as GPSSatellites,      Attribute(FileLongName,
'GPSStatus') as GPSStatus,      Attribute(FileLongName, 'GPSMeasureMode') as GPSMeasureMode,
Attribute(FileLongName, 'GPSDOP') as GPSDOP,      Attribute(FileLongName, 'GPSSpeedRef') as
GPSSpeedRef,

Attribute(FileLongName, 'GPSSpeed') as GPSSpeed,      Attribute(FileLongName,
'GPSTrackRef') as GPSTrackRef,      Attribute(FileLongName, 'GPSTrack') as GPSTrack,
Attribute(FileLongName, 'GPSImgDirectionRef') as GPSImgDirectionRef,      Attribute
(FileLongName, 'GPSImgDirection') as GPSImgDirection,      Attribute(FileLongName,
'GPSMapDatum') as GPSMapDatum,      Attribute(FileLongName, 'GPSDestLatitudeRef') as
GPSDestLatitudeRef,

Attribute(FileLongName, 'GPSDestLatitude') as GPSDestLatitude,      Attribute
(FileLongName, 'GPSDestLongitudeRef') as GPSDestLongitudeRef,      Attribute(FileLongName,
'GPSDestLongitude') as GPSDestLongitude,      Attribute(FileLongName, 'GPSDestBearingRef') as
GPSDestBearingRef,      Attribute(FileLongName, 'GPSDestBearing') as GPSDestBearing,
Attribute(FileLongName, 'GPSDestDistanceRef') as GPSDestDistanceRef,

Attribute(FileLongName, 'GPSDestDistance') as GPSDestDistance,      Attribute
(FileLongName, 'GPSProcessingMethod') as GPSProcessingMethod,      Attribute(FileLongName,
'GPSAreaInformation') as GPSAreaInformation,      Attribute(FileLongName, 'GPSDateStamp') as
GPSDateStamp,      Attribute(FileLongName, 'GPSDifferential') as GPSDifferential;

// examples: 0=No correction, 1=Differential correction, LOAD @1:n as FileLongName
Inline "$(vFoundFile)" (fix, no labels); Next vFoundFile Next vExt
```

### Example 3: Windows-mediabestanden

Dit script leest alle mogelijke WMA/WMV ASF-metatags in de map *MyMusic*.

```
/ Script to read WMA/WMV ASF meta tags for each vExt in 'asf', 'wma', 'wmv' for each
vFoundFile in fileList( GetFolderPath('MyMusic') & '\*.*' & vExt )

FileList: LOAD FileLongName,      subfield(FileLongName,'\",-1) as FileShortName,      num
(FileSize(FileLongName),'# ### ### ###',' ',' ') as FileSize,      FileTime(FileLongName) as
FileTime,      Attribute(FileLongName, 'Title') as Title,      Attribute(FileLongName,
'Author') as Author,      Attribute(FileLongName, 'Copyright') as Copyright,      Attribute
(FileLongName, 'Description') as Description,

      Attribute(FileLongName, 'Rating') as Rating,      Attribute(FileLongName, 'PlayDuration')
as PlayDuration,      Attribute(FileLongName, 'MaximumBitrate') as MaximumBitrate,
Attribute(FileLongName, 'WMFSDKVersion') as WMFSDKVersion,      Attribute(FileLongName,
'WMFSDKNeeded') as WMFSDKNeeded,      Attribute(FileLongName, 'ISVBR') as ISVBR,      Attribute
(FileLongName, 'ASFLeakyBucketPairs') as ASFLeakyBucketPairs,

      Attribute(FileLongName, 'PeakValue') as PeakValue,      Attribute(FileLongName,
'AverageLevel') as AverageLevel; LOAD @1:n as FileLongName Inline "$(vFoundFile)" (fix, no
labels); Next vFoundFile Next vExt
```

### Example 4: PNG

Dit script leest alle mogelijke PNG-metatags in de map *MyPictures*.

```
// Script to read PNG meta tags for each vExt in 'png' for each vFoundFile in fileList(
GetFolderPath('MyPictures') & '\*.*' & vExt )

FileList: LOAD FileLongName,      subfield(FileLongName,'\",-1) as FileShortName,      num
(FileSize(FileLongName),'# ### ### ###',' ',' ') as FileSize,      FileTime(FileLongName) as
FileTime,      Attribute(FileLongName, 'Comment') as Comment,

      Attribute(FileLongName, 'Creation Time') as Creation_Time,      Attribute(FileLongName,
'Source') as Source,      Attribute(FileLongName, 'Title') as Title,      Attribute
(FileLongName, 'Software') as Software,      Attribute(FileLongName, 'Author') as Author,
Attribute(FileLongName, 'Description') as Description,

      Attribute(FileLongName, 'Copyright') as Copyright; LOAD @1:n as FileLongName Inline
"${vFoundFile}" (fix, no labels); Next vFoundFile Next vExt
```

## ConnectionString

De functie **ConnectionString()** retourneert de naam van de actieve gegevensverbinding voor ODBC- of OLE DB-verbindingen. De functie retourneert een lege tekenreeks als geen **connect**-opdracht is uitgevoerd of na een **disconnect**-opdracht.

### Syntaxis:

```
ConnectionString()
```

Voorbeelden en resultaten:

Scriptvoorbeelden

Voorbeeld	Resultaat
<pre>LIB CONNECT TO 'Tutorial ODBC'; ConnectionString: Load ConnectString() as ConnectionString AutoGenerate 1;</pre>	<p>Retourneert 'Tutorial ODBC' in het veld ConnectString.</p> <p>In dit voorbeeld wordt ervan uitgegaan dat u over een gegevensverbinding beschikt met de naam Tutorial ODBC.</p>

### FileName

De functie **FileName** retourneert een tekenreeks met de naam van het tabelbestand dat op dat moment wordt gelezen, zonder pad of extensie.

**Syntaxis:**

**FileName ( )**

Voorbeelden en resultaten:

Scriptvoorbeelden

Voorbeeld	Resultaat
<pre>LOAD *, filename( ) as X from C:\UserFiles\abc.txt</pre>	<p>Retourneert 'abc' in veld X in elke gelezen record.</p>

### FileDir

De functie **FileDir** retourneert een tekenreeks met het pad naar de map van het tabelbestand dat op dat moment wordt gelezen.

**Syntaxis:**

**FileDir ( )**



*Deze functie ondersteunt alleen mapgegevensverbindingen in de standaardmodus.*

Voorbeelden en resultaten:

Scriptvoorbeelden

Voorbeeld	Resultaat
<pre>Load *, filedir( ) as X from C:\UserFiles\abc.txt</pre>	<p>Retourneert 'C:\UserFiles' in veld X in elke gelezen record.</p>

### FileExtension

De functie **FileExtension** retourneert een tekenreeks met de extensie van het tabelbestand dat op dat moment wordt gelezen.

### Syntaxis:

**FileExtension()**

Voorbeelden en resultaten:

Scriptvoorbeelden

Voorbeeld	Resultaat
LOAD *, FileExtension( ) as X from C:\UserFiles\abc.txt	Retourneert 'txt' in veld X in elke gelezen record.

### FileName

De functie **FileName** retourneert een tekenreeks met de naam van het tabelbestand dat op dat moment wordt gelezen, zonder pad maar met de extensie.

### Syntaxis:

**FileName()**

Voorbeelden en resultaten:

Scriptvoorbeelden

Voorbeeld	Resultaat
LOAD *, FileName( ) as X from C:\UserFiles\abc.txt	Retourneert 'abc.txt' in veld X in elke gelezen record.

### FilePath

De functie **FilePath** retourneert een tekenreeks met het volledige pad naar het tabelbestand dat op dat moment wordt gelezen.

### Syntaxis:

**FilePath()**



*Deze functie ondersteunt alleen mapgegevensverbindingen in de standaardmodus.*

Voorbeelden en resultaten:

Scriptvoorbeelden

Voorbeeld	Resultaat
Load *, FilePath( ) as X from C:\UserFiles\abc.txt	Retourneert 'C:\UserFiles\abc.txt' in veld X in elke gelezen record.

### FileSize

De functie retourneert een geheel getal met de grootte in bytes van het bestand of, als geen is opgegeven, van het tabelbestand dat op dat moment wordt gelezen. `filename`

#### Syntaxis:

**FileSize** ([filename])

#### Argumenten:

##### Argumenten

Argument	Beschrijving
filename	<p>De naam van een bestand, indien nodig met pad, als map- of webgegevensverbinding. Als u geen bestandsnaam opgeeft, wordt het tabelbestand gebruikt dat momenteel wordt gelezen.</p> <p><b>Voorbeeld: 'lib://Table Files'</b></p> <p>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:</p> <ul style="list-style-type: none"> <li>• absoluut</li> </ul> <p><b>Voorbeeld: c:\data</b></p> <ul style="list-style-type: none"> <li>• relatief ten opzichte van de werkmap van de Qlik Sense-app.</li> </ul> <p><b>Voorbeeld: data</b></p> <ul style="list-style-type: none"> <li>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.</li> </ul> <p><b>Voorbeeld: http://www.qlik.com</b></p>

#### Voorbeelden en resultaten:

##### Scriptvoorbeelden

Voorbeeld	Resultaat
LOAD *, FileSize( ) as X from abc.txt;	Retourneert de grootte van het opgegeven bestand (abc.txt) als een geheel getal in veld X in elke gelezen record.
FileSize( 'lib://DataFiles/xyz.xls' )	Retourneert de grootte van het bestand xyz.xls.

### FileTime

De functie **FileTime** retourneert een tijdstempel in UTC voor de datum en tijd van de laatste wijziging van het bestand filename. Als geen filename is opgegeven, wordt het tabelbestand gebruikt dat op dat moment wordt gelezen.

### Syntaxis:

```
FileTime( [ filename ] )
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
filename	<p>De naam van een bestand, indien nodig met pad, als map- of webgegevensverbinding.</p> <p><b>Voorbeeld: 'lib://Table Files'</b></p> <p>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:</p> <ul style="list-style-type: none"> <li>absoluut</li> </ul> <p><b>Voorbeeld: c:\data1</b></p> <ul style="list-style-type: none"> <li>relatief ten opzichte van de werkmap van de Qlik Sense-app.</li> </ul> <p><b>Voorbeeld: data1</b></p> <ul style="list-style-type: none"> <li>URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.</li> </ul> <p><b>Voorbeeld: http://www.qlik.com</b></p>

### Voorbeelden en resultaten:

#### Scriptvoorbeelden

Voorbeeld	Resultaat
LOAD *, FileTime( ) as X from abc.txt;	Retourneert de datum en tijd van de laatste wijziging van het bestand (abc.txt) als een tijdstempel in veld X in elke gelezen record.
FileTime( 'xyz.xls' )	Retourneert de tijdstempel van de laatste wijziging van het bestand xyz.xls.

## GetFolderPath

De functie **GetFolderPath** retourneert de waarde van de functie Microsoft Windows *SHGetFolderPath*. Deze functie neemt de naam van een map in Microsoft Windows als invoer en retourneert het volledige pad van de map.



*Deze functie wordt niet ondersteund in de standaardmodus.*

### Syntaxis:

```
GetFolderPath( foldername )
```



### Argumenten:

#### Argumenten

Argument	Beschrijving
<b>foldername</b>	<p>Naam van de Microsoft Windows-map.</p> <p>De mapnaam mag geen spatie bevatten. Alle spaties in de mapnaam in Windows Explorer moeten worden verwijderd uit de mapnaam.</p> <p>Voorbeelden:</p> <p><i>MyMusic</i></p> <p><i>MyDocuments</i></p>

### Voorbeelden en resultaten:

Het doel van dit voorbeeld is het ophalen van de paden van de volgende Microsoft Windows-mappen: *MyMusic*, *MyPictures* en *Windows*. Voeg het voorbeeldscript toe aan uw app en laad het opnieuw.

```
LOAD GetFolderPath('MyMusic') as MyMusic, GetFolderPath('MyPictures') as MyPictures,
GetFolderPath('Windows') as windows AutoGenerate 1;
```

Nadat de app opnieuw is geladen, worden de velden *MyMusic*, *MyPictures* en *Windows* toegevoegd aan het gegevensmodel. Elk veld bevat het pad naar de map die is gedefinieerd in de invoer. Bijvoorbeeld:

- *C:\Users\smu\Music* for the folder *MyMusic*
- *C:\Users\smu\Pictures* for the folder *MyPictures*
- *C:\Windows* for the folder *Windows*

## QvdCreateTime

Deze scriptfunctie retourneert de tijdstempel voor de XML-koptekst uit een QVD-bestand, indien aanwezig. Anders wordt NULL geretourneerd. In de tijdstempel wordt tijd weergegeven in UTC.

### Syntaxis:

```
QvdCreateTime (filename)
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
filename	<p>De naam van een QVD-bestand, indien nodig met pad, als map- of webgegevensverbinding.</p> <p><b>Voorbeeld: 'lib://Table Files'</b></p> <p>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:</p> <ul style="list-style-type: none"><li>• absoluut</li></ul> <p><b>Voorbeeld: c:\data</b></p> <ul style="list-style-type: none"><li>• relatief ten opzichte van de werkmap van de Qlik Sense-app.</li></ul> <p><b>Voorbeeld: data</b></p> <ul style="list-style-type: none"><li>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.</li></ul> <p><b>Voorbeeld: http://www.qlik.com</b></p>

### Voorbeeld:

```
QvdCreateTime('MyFile.qvd')
```

```
QvdCreateTime('C:\MyDir\MyFile.qvd')
```

```
QvdCreateTime('lib://DataFiles/MyFile.qvd')
```

## QvdFieldName

Deze script-functie retourneert de naam van veldnummer **fieldno** in een QVD-bestand. Als het veld niet bestaat, wordt NULL geretourneerd.

### Syntaxis:

```
QvdFieldName(filename , fieldno)
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
filename	<p>De naam van een QVD-bestand, indien nodig met pad, als map- of webgegevensverbinding.</p> <p><b>Voorbeeld: 'lib://Table Files'</b></p> <p>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:</p> <ul style="list-style-type: none"><li>• absoluut</li></ul> <p><b>Voorbeeld: c:\data</b></p> <ul style="list-style-type: none"><li>• relatief ten opzichte van de werkmap van de Qlik Sense-app.</li></ul> <p><b>Voorbeeld: data</b></p> <ul style="list-style-type: none"><li>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.</li></ul> <p><b>Voorbeeld: http://www.qlik.com</b></p>
fieldno	Het nummer van het veld in de tabel binnen het QVD-bestand.

### Voorbeelden:

```
QvdFieldName ('MyFile.qvd', 5)
```

```
QvdFieldName ('C:\MyDir\MyFile.qvd', 5)
```

```
QvdFieldName ('lib://DataFiles/MyFile.qvd', 5)
```

Alle drie de exemplaren retourneren de naam van het vijfde veld van de tabel binnen het QVD-bestand.

## QvdNoOfFields

Deze scriptfunctie retourneert het aantal velden in een QVD-bestand.

### Syntaxis:

```
QvdNoOfFields (filename)
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
filename	<p>De naam van een QVD-bestand, indien nodig met pad, als map- of webgegevensverbinding.</p> <p><b>Voorbeeld: 'lib://Table Files'</b></p> <p>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:</p> <ul style="list-style-type: none"><li>• absoluut</li></ul> <p><b>Voorbeeld: c:\data</b></p> <ul style="list-style-type: none"><li>• relatief ten opzichte van de werkmap van de Qlik Sense-app.</li></ul> <p><b>Voorbeeld: data</b></p> <ul style="list-style-type: none"><li>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.</li></ul> <p><b>Voorbeeld: http://www.qlik.com</b></p>

### Voorbeelden:

```
QvdNoOfFields ('MyFile.qvd')
```

```
QvdNoOfFields ('C:\MyDir\MyFile.qvd')
```

```
QvdNoOfFields ('lib://DataFiles/MyFile.qvd')
```

## QvdNoOfRecords

**Voorbeeld:** Deze scriptfunctie retourneert het huidige aantal records in een QVD-bestand.

### Syntaxis:

```
QvdNoOfRecords (filename)
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
filename	<p>De naam van een QVD-bestand, indien nodig met pad, als map- of webgegevensverbinding.</p> <p><b>Voorbeeld: 'lib://Table Files'</b></p> <p>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:</p> <ul style="list-style-type: none"><li>• absoluut</li></ul> <p><b>Voorbeeld: c:\data</b></p> <ul style="list-style-type: none"><li>• relatief ten opzichte van de werkmap van de Qlik Sense-app.</li></ul> <p><b>Voorbeeld: data</b></p> <ul style="list-style-type: none"><li>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.</li></ul> <p><b>Voorbeeld: http://www.qlik.com</b></p>

### Voorbeelden:

```
QvdNoOfRecords ('MyFile.qvd')
```

```
QvdNoOfRecords ('C:\MyDir\MyFile.qvd')
```

```
QvdNoOfRecords ('lib://DataFiles/MyFile.qvd')
```

## QvdTableName

Deze scriptfunctie retourneert de naam van de tabel die is opgeslagen in een QVD-bestand.

### Syntaxis:

```
QvdTableName (filename)
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
filename	<p>De naam van een QVD-bestand, indien nodig met pad, als map- of webgegevensverbinding.</p> <p><b>Voorbeeld: 'lib://Table Files'</b></p> <p>In de bestaande scriptmodus, worden tevens de volgende padindelingen ondersteund:</p> <ul style="list-style-type: none"> <li>• absoluut</li> </ul> <p><b>Voorbeeld: c:\data</b></p> <ul style="list-style-type: none"> <li>• relatief ten opzichte van de werkmap van de Qlik Sense-app.</li> </ul> <p><b>Voorbeeld: data</b></p> <ul style="list-style-type: none"> <li>• URL-adres (HTTP of FTP) dat verwijst naar een locatie op internet of intranet.</li> </ul> <p><b>Voorbeeld: http://www.qlik.com</b></p>

### Voorbeelden:

```

QvdTableName ('MyFile.qvd')
QvdTableName ('C:\MyDir\MyFile.qvd')
QvdTableName ('lib://data\MyFile.qvd')
    
```

## 5.11 Financiële functies

Financiële functies kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen voor het berekenen van betalingen en rentepercentages.

Voor alle argumenten worden contante betalingen als negatieve getallen weergegeven. Contant geld dat u ontvangt, wordt als positief getal weergegeven.

Hierna volgt een lijst met de argumenten die u kunt gebruiken in de financiële functies (uitgezonderd de argumenten die beginnen met **range-**).



*Voor alle financiële functies is het belangrijk dat u consequent bent bij het opgeven van eenheden voor **rate** en **nper**. Als er maandelijkse betalingen worden gedaan voor een lening van vijf jaar tegen een rente van 6% per jaar, gebruikt u 0,005 (6%/12) voor **rate** en 60 (5\*12) voor **nper**. Als er jaarlijkse betalingen worden gedaan voor dezelfde lening, gebruikt u 6% voor **rate** en 5 voor **nper**.*

### Overzicht van financiële functies

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### FV

Deze functie retourneert de toekomstige waarde van een investering op basis van periodieke, constante betalingen en een enkelvoudige jaarlijkse rente.

```
FV (rate, nper, pmt [ ,pv [ , type ] ])
```

#### nPer

Deze functie retourneert het aantal perioden voor een investering op basis van periodieke, constante betalingen en een constant rentepercentage.

```
nPer (rate, pmt, pv [ ,fv [ , type ] ])
```

#### Pmt

Deze functie retourneert de betaling voor een lening op basis van periodieke, constante betalingen en een constant rentepercentage. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20.

```
Pmt (rate, nper, pv [ ,fv [ , type ] ])
```

#### PV

Deze functie retourneert de huidige waarde van een investering.

```
PV (rate, nper, pmt [ ,fv [ , type ] ])
```

#### Rate

Deze functie retourneert het rentepercentage per periode op annuïteit. Het resultaat heeft de standaardgetalnotatie van **Fix** twee decimalen en %.

```
Rate (nper, pmt , pv [ ,fv [ , type ] ])
```

### BlackAndSchole

Het Black and Scholes-model is een wiskundig model voor afgeleide financiële instrumenten. De formule berekent de theoretische waarde van een optie. In Qlik Sense retourneert de functie **BlackAndSchole** de waarde volgens de ongewijzigde Black and Scholes-formule (opties Europese stijl).

```
BlackAndSchole (strike , time_left , underlying_price , vol , risk_free_rate , type)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
strike	De toekomstige aankoopprijs van het aandeel.
time_left	Het resterende aantal tijdsperioden.
underlying_price	De huidige waarde van het aandeel.
vol	Volatiliteit (van de aandelenprijs) uitgedrukt als percentage in decimale vorm, per tijdsperiode.
risk_free_rate	De risicovrije rente uitgedrukt als percentage in decimale vorm, per tijdsperiode.
call_or_put	Het type optie:  'c', 'call' of enige numerieke waarde die niet gelijk is aan nul voor callopties  'p', 'put' of 0 voor putopties.

**Beperkingen:**

De waarde van strike, time\_left en underlying\_price moet >0 zijn.

De waarde van vol en risk\_free\_rate moet <) of >0 zijn.

Voorbeelden en resultaten:

Scriptvoorbeelden

Voorbeeld	Resultaat
<code>BlackAndSchole(130, 4, 68.5, 0.4, 0.04, 'call')</code>  Hiermee wordt de theoretische prijs berekend van een optie voor het aankopen van een aandeel dat vandaag 68,5 waard is, bij een waarde van 130 over 4 jaar. In de formule wordt gebruikgemaakt van een volatiliteit van 0,4 (40%) per jaar en een risicovrij rentetarief van 0,04 (4%).	retourneert 11,245

## FV

Deze functie retourneert de toekomstige waarde van een investering op basis van periodieke, constante betalingen en een enkelvoudige jaarlijkse rente.

**Syntaxis:**

```
FV(rate, nper, pmt [ ,pv [ , type ] ])
```



**Retourgegevenstypen:** numeriek. Het resultaat wordt standaard geformatteerd als een valuta..

**Argumenten:**

Argumenten

Argument	Beschrijving
rate	Het rentepercentage per periode.
nper	Het totale aantal betalingsperioden in een annuïteit.
pmt	De betaling die elke periode wordt gedaan. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20.
pv	De huidige waarde die of het totale bedrag dat een reeks toekomstige betalingen op dit moment waard is. Als <b>pv</b> wordt weggelaten, wordt dit verondersteld 0 (nul) te zijn.
type	Moet 0 zijn als de betalingen aan het eind van de periode verschuldigd zijn en 1 als de betalingen aan het begin van de periode verschuldigd zijn. Als <b>type</b> niet wordt aangegeven, wordt dit verondersteld 0 te zijn.

Voorbeelden en resultaten:

Scriptvoorbeeld

Voorbeeld	Resultaat
U betaalt een nieuw huishoudelijk apparaat af in 36 maandelijkse termijnen van \$20. De rente is 6% op jaarbasis. De rekening komt aan het einde van elke maand. Wat is het geïnvesteerde totaal wanneer de laatste rekening is betaald?  <code>FV(0.005, 36, -20)</code>	Retourneert \$786.72

### nPer

Deze functie retourneert het aantal perioden voor een investering op basis van periodieke, constante betalingen en een constant rentepercentage.

**Syntaxis:**

```
nPer(rate, pmt, pv [ ,fv [ , type ] ])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
rate	Het rentepercentage per periode.

Argument	Beschrijving
nper	Het totale aantal betalingsperioden in een annuïteit.
pmt	De betaling die elke periode wordt gedaan. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20.
pv	De huidige waarde die of het totale bedrag dat een reeks toekomstige betalingen op dit moment waard is. Als <b>pv</b> wordt weggelaten, wordt dit verondersteld 0 (nul) te zijn.
fv	De toekomstige waarde die of het kassaldo dat u wilt bereiken nadat de laatste betaling is gedaan. Als <b>fv</b> niet wordt aangegeven, wordt dit verondersteld 0 te zijn.
type	Moet 0 zijn als de betalingen aan het eind van de periode verschuldigd zijn en 1 als de betalingen aan het begin van de periode verschuldigd zijn. Als <b>type</b> niet wordt aangegeven, wordt dit verondersteld 0 te zijn.

Voorbeelden en resultaten:

### Scriptvoorbeeld

Voorbeeld	Resultaat
<p>U wilt een huishoudelijk apparaat verkopen en dat er in maandelijks termijnen van \$ 20 wordt betaald. De rente is 6% op jaarbasis. De rekening komt aan het einde van elke maand. Hoeveel perioden zijn vereist als de waarde van het geld dat u hebt ontvangen na de laatste rekening gelijk moet zijn aan \$ 800?</p> <p><code>nPer(0.005, -20, 0, 800)</code></p>	retourneert 36,56

## Pmt

Deze functie retourneert de betaling voor een lening op basis van periodieke, constante betalingen en een constant rentepercentage. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20.

```
Pmt(rate, nper, pv [ ,fv [ , type ] ] )
```

**Retourgegevenstypen:** numeriek. Het resultaat wordt standaard geformatteerd als een valuta..

Het totaalbedrag dat is betaald tijdens de looptijd van de lening is gelijk aan de waarde van **pmt** vermenigvuldigd met **nper**.

**Argumenten:**

### Argumenten

Argument	Beschrijving
rate	Het rentepercentage per periode.

## 5 Script- en diagramfuncties

Argument	Beschrijving
nper	Het totale aantal betalingsperioden in een annuïteit.
pv	De huidige waarde die of het totale bedrag dat een reeks toekomstige betalingen op dit moment waard is. Als <b>pv</b> wordt weggelaten, wordt dit verondersteld 0 (nul) te zijn.
fv	De toekomstige waarde die of het kassaldo dat u wilt bereiken nadat de laatste betaling is gedaan. Als <b>fv</b> niet wordt aangegeven, wordt dit verondersteld 0 te zijn.
type	Moet 0 zijn als de betalingen aan het eind van de periode verschuldigd zijn en 1 als de betalingen aan het begin van de periode verschuldigd zijn. Als <b>type</b> niet wordt aangegeven, wordt dit verondersteld 0 te zijn.

Voorbeelden en resultaten:

### Scriptvoorbeelden

Voorbeeld	Resultaat
De volgende formule retourneert de maandelijkse betaling op een lening van € 20.000 tegen een jaarrente van 10 procent, die in 8 maanden moet worden afbetaald:  <code>Pmt(0.1/12,8,20000)</code>	Retourneert - \$2,594.66
Als voor dezelfde lening betalingen aan het begin van de maand moeten plaatsvinden, is de betaling als volgt:  <code>Pmt(0.1/12,8,20000,0,1)</code>	Retourneert - \$2,573.21

## PV

Deze functie retourneert de huidige waarde van een investering.

```
PV(rate, nper, pmt [ ,fv [ , type ] ])
```

**Retourgegevenstypen:** numeriek. Het resultaat wordt standaard geformatteerd als een valuta..

De huidige waarde is het totale bedrag dat een reeks toekomstige betalingen op dit moment waard is. Als u bijvoorbeeld geld leent, is het geleende bedrag de huidige waarde voor de verstrekker van de lening.

**Argumenten:**

### Argumenten

Argument	Beschrijving
rate	Het rentepercentage per periode.
nper	Het totale aantal betalingsperioden in een annuïteit.
pmt	De betaling die elke periode wordt gedaan. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20.

Argument	Beschrijving
fv	De toekomstige waarde die of het kassaldo dat u wilt bereiken nadat de laatste betaling is gedaan. Als <b>fv</b> niet wordt aangegeven, wordt dit verondersteld 0 te zijn.
type	Moet 0 zijn als de betalingen aan het eind van de periode verschuldigd zijn en 1 als de betalingen aan het begin van de periode verschuldigd zijn. Als <b>type</b> niet wordt aangegeven, wordt dit verondersteld 0 te zijn.

Voorbeelden en resultaten:

### Scriptvoorbeeld

Voorbeeld	Resultaat
<p>Wat is de huidige waarde van een schuld, wanneer u \$100 moet betalen aan het eind van elke maand gedurende een periode van vijf jaar, met een rentepercentage van 7%?</p> <p>PV(0.07/12,12*5,-100,0,0)</p>	Retourneert \$5,050.20

## Rate

Deze functie retourneert het rentepercentage per periode op annuïteit. Het resultaat heeft de standaardgetalnotatie van **Fix** twee decimalen en %.

**Syntaxis:**

```
Rate(nper, pmt, pv [,fv [, type ] ])
```

**Retourgegevenstypen:** numeriek.

De **rate** wordt per iteratie berekend en kan nul of meer oplossingen hebben. Als de opeenvolgende resultaten van **rate** niet samenvallen, wordt een NULL-waarde geretourneerd.

**Argumenten:**

### Argumenten

Argument	Beschrijving
nper	Het totale aantal betalingsperioden in een annuïteit.
pmt	De betaling die elke periode wordt gedaan. Dit bedrag moet gedurende de hele duur van de annuïteit gelijk blijven. Een betaling wordt weergegeven als een negatief getal, bijvoorbeeld, -20.
pv	De huidige waarde die of het totale bedrag dat een reeks toekomstige betalingen op dit moment waard is. Als <b>pv</b> wordt weggelaten, wordt dit verondersteld 0 (nul) te zijn.
fv	De toekomstige waarde die of het kassaldo dat u wilt bereiken nadat de laatste betaling is gedaan. Als <b>fv</b> niet wordt aangegeven, wordt dit verondersteld 0 te zijn.

Argument	Beschrijving
type	Moet 0 zijn als de betalingen aan het eind van de periode verschuldigd zijn en 1 als de betalingen aan het begin van de periode verschuldigd zijn. Als <b>type</b> niet wordt aangegeven, wordt dit verondersteld 0 te zijn.

Voorbeelden en resultaten:

Scriptvoorbeeld

Voorbeeld	Resultaat
Wat is het rentepercentage van een annuïteitenlening van € 10.000 met een looptijd van vijf jaar en maandelijkse betalingen van € 300?  <code>Rate(60, -300, 10000)</code>	Retourneert 2.00%

### 5.12 Opmaakfuncties

De opmaakfuncties leggen de opmaakindeling op voor de ingevoerd numerieke velden of uitdrukkingen. Afhankelijk van het gegevenstype, kunt u de tekens opgeven voor het scheidingsteken voor decimalen, scheidingsteken voor duizendtallen enzovoort.

Alle functies retourneren een dubbele waarde met zowel de tekenreeks als de getalwaarde, maar kunnen worden beschouwd als een conversie van getal naar tekenreeks. **Dual()** is een speciaal geval, maar de andere opmaakfuncties nemen de numerieke waarde van de invoeruitdrukking en genereren een tekenreeks die het getal vertegenwoordigt.

De interpretatiefuncties, daarentegen, doen het omgekeerde: zij nemen tekenreeksuitdrukkingen en evalueren deze als getallen, waarbij de notatie van het resulterende getal wordt opgegeven.

De functies kunnen zowel worden gebruikt in load-scripts voor gegevens als in diagramuitdrukkingen.



*Alle getallen moeten worden weergegeven met een decimale punt als scheidingsteken voor decimalen.*

### Overzicht van opmaakfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### ApplyCodepage

**ApplyCodepage()** hiermee past u een andere codepagina-tekenset toe op het veld of de tekst die wordt aangegeven in de uitdrukking. Gebruik een getalnotatie voor het **codepage**-argument.

**ApplyCodepage** (text, codepage)

### Date

**Date()** maakt een uitdrukking op als een datum met de opmaak die in de systeemvariabelen in het load-script voor gegevens, in het besturingssysteem of in een opmaakreeks is ingesteld, indien opgegeven.

```
Date (number[, format])
```

### Dual

**Dual()** combineert een getal en een tekenreeks tot één record, zodat de nummerweergave van de record kan worden gebruikt voor sorterings- en berekeningsdoeleinden, terwijl de tekenreekswaarde kan worden gebruikt voor weergavedoeleinden.

```
Dual (text, number)
```

### Interval

**Interval()** maakt een getal op als een tijdsinterval met de opmaak die in de systeemvariabelen in het load-script voor gegevens, in het besturingssysteem of in een opmaakreeks is ingesteld, indien opgegeven.

```
Interval (number[, format])
```

### Money

**Money()** maakt een uitdrukking op numerieke wijze op als een geldwaarde in de opmaak die is ingesteld in de systeemvariabelen die zijn ingesteld in het load-script voor gegevens of in het besturingssysteem, tenzij een opmaakreeks is opgegeven, alsmede optionele scheidingstekens voor decimalen en duizendtallen.

```
Money (number[, format[, dec_sep [, thou_sep]])
```

### Num

**Num()** geeft de opmaak van een getal, wat wil zeggen dat deze functie de numerieke waarde van de invoer converteert naar tekst met gebruikmaking van de opmaak die is opgegeven in de tweede parameter. Als de tweede parameter wordt weggelaten, worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

```
Num (number[, format[, dec_sep [, thou_sep]])
```

### Time

**Time()** maakt een uitdrukking op als een tijdwaarde in de tijdnotatie die in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

```
Time (number[, format])
```

### Timestamp

**TimeStamp()** maakt een uitdrukking op als een datum- en tijdwaarde in de notatie voor tijdstempels die in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

```
Timestamp (number[, format])
```

**Zie ook:**

p *Interpretatiefuncties (page 1256)*

## ApplyCodepage

**ApplyCodepage()** hiermee past u een andere codepagina-tekenset toe op het veld of de tekst die wordt aangegeven in de uitdrukking. Gebruik een getalnotatie voor het **codepage**-argument.



*Hoewel ApplyCodepage in diagramuitdrukkingen kan worden gebruikt, dient het vaker als scriptfunctie in de editor voor laden van gegevens. Terwijl u bijvoorbeeld bestanden laadt die mogelijk zijn opgeslagen in andere tekensets waarover u geen controle hebt, kunt u de codepagina toepassen die de tekenset vertegenwoordigt die u nodig hebt.*

**Syntaxis:**

```
ApplyCodepage (text, codepage)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

## Argumenten

Argument	Beschrijving
text	Veld of tekst waarop u een andere codepagina wilt toepassen, opgegeven via het argument <b>codepage</b> .
codepage	Nummer dat de codepagina aangeeft die moet worden toegepast op het veld of de uitdrukking die is opgegeven via <b>text</b> .

**Voorbeelden en resultaten:**

## Scriptvoorbeelden

Voorbeeld	Resultaat
<pre>LOAD ApplyCodepage (ROWX,1253) as GreekProduct, ApplyCodepage (ROWY, 1255) as HebrewProduct, ApplyCodepage (ROWZ, 65001) as EnglishProduct; SQL SELECT ROWX, ROWY, ROWZ From Products;</pre>	<p>Bij het laden vanuit SQL bevat de bron mogelijk een aantal verschillende tekensets: Cyrillisch, Hebreeuws, enzovoort, die afwijken van de indeling UTF-8. Deze zouden rij voor rij moeten worden geladen waarbij voor elke rij een andere codepagina zou moeten worden toegepast.</p> <p>De <b>codepage</b>-waarde 1253 staat voor de Griekse tekenset van Windows, de waarde 1255 voor Hebreeuws en de waarde 65001 voor standaard Latijnse UTF-8-tekens.</p>

Zie ook: *Tekenset (page 161)*

### Date

**Date()** maakt een uitdrukking op als een datum met de opmaak die in de systeemvariabelen in het load-script voor gegevens, in het besturingssysteem of in een opmaakreeks is ingesteld, indien opgegeven.

#### Syntaxis:

```
Date (number [, format ])
```

**Retourgegevenstypen:** dual

#### Argumenten:

Argumenten

Argument	Beschrijving
number	Het getal dat moet worden opgemaakt.
format	Tekenreeks waarmee de notatie van de resulterende tekenreeks wordt beschreven. Als er geen opmaakreeks wordt opgegeven, wordt de datumopmaak gebruikt die is ingesteld in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem.

Voorbeelden en resultaten:

In de onderstaande voorbeelden worden de volgende standaardinstellingen verondersteld:

- Datuminstelling 1: YY-MM-DD
- Datuminstelling 2: M/D/YY

#### Voorbeeld:

```
Date( A )  
waarbij A=35648
```

Resultatentabel

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	97-08-06	8/6/97
Getal:	35648	35648

#### Voorbeeld:

```
Date( A, 'YY.MM.DD' )  
waarbij A=35648
```



Resultatentabel

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	97.08.06	97.08.06
Getal:	35648	35648

**Voorbeeld:**

Date( A, 'DD.MM.YYYY' )  
 waarbij A=35648.375

Resultatentabel

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	06.08.1997	06.08.1997
Getal:	35648.375	35648.375

**Voorbeeld:**

Date( A, 'YY.MM.DD' )  
 waarbij A=8/6/97

Resultatentabel

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	NULL (niets)	97.08.06
Getal:	NULL	35648

## Dual

**Dual()** combineert een getal en een tekenreeks tot één record, zodat de nummerweergave van de record kan worden gebruikt voor sorterings- en berekeningsdoeleinden, terwijl de tekenreekswaarde kan worden gebruikt voor weergavedoeleinden.

**Syntaxis:**

**Dual** (text, number)

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

Argument	Beschrijving
text	De tekenreekswaarde die moet worden gebruikt in combinatie met het getalargument.
number	Het getal dat moet worden gebruikt in combinatie met het tekenreeksargument.

## 5 Script- en diagramfuncties

In Qlik Sense zijn alle veldwaarden mogelijk dubbele waarden. Dit betekent dat de veldwaarden zowel een numerieke waarde als een tekstuele waarde kunnen hebben. Een voorbeeld is een datum die een numerieke waarde 40908 en de tekstuele representatie '2011-12-31' zou kunnen hebben.



*Als diverse gegevens die in één veld worden gelezen, verschillende tekenreeksweergaven hebben maar wel dezelfde geldige getalswaarde, delen ze allemaal de eerst aangetroffen tekenreeksweergave.*



*De functie **dual** wordt meestal vroeg in het script gebruikt, nog voordat andere gegevens in het desbetreffende veld worden gelezen, zodat die eerste tekenreeksweergave wordt gemaakt en dan in filtervakken wordt getoond.*

Voorbeelden en resultaten:

### Scriptvoorbeelden

Voorbeeld	Beschrijving
<p>Voeg de volgende voorbeelden toe aan uw script en voer het uit.</p> <pre>Load dual ( NameDay, NumDay ) as DayOfWeek inline [ NameDay, NumDay Monday, 0 Tuesday, 1 Wednesday, 2 Thursday, 3 Friday, 4 Saturday, 5 Sunday, 6 ];</pre>	<p>Het veld DayOfWeek kan worden gebruikt in een visualisatie, als een dimensie, bijvoorbeeld. In een tabel met de weekdays worden de dagen automatisch gesorteerd op juiste getalvolgorde in plaats van op alfabetische volgorde.</p>
<pre>Load Dual('Q' &amp; Ceil(Month(Now ())/3), Ceil(Month (Now())/3)) as Quarter AutoGenerate 1;</pre>	<p>Dit voorbeeld retourneert het huidige kwartaal. Het wordt weergegeven als als de functie wordt uitgevoerd in de eerste drie maanden van het , voor de tweede reeks van drie maanden enzovoort. Q1Q2 Als het veld Quarter echter wordt gebruikt bij sorteren, gedraagt het zich als een numerieke waarde: 1 t/m 4.</p>
<pre>Dual('Q' &amp; Ceil (Month(Date)/3), Ceil(Month (Date)/3)) as Quarter</pre>	<p>Net als in het voorafgaande voorbeeld, wordt het veld Quarter gemaakt met de tekstwaarden 'Q1' t/m 'Q4' en worden hieraan de numerieke waarden 1 t/m 4 toegewezen. Als u dit in het script wilt gebruiken, moeten de waarden voor Date worden geladen.</p>

Voorbeeld	Beschrijving
Dual(WeekYear(Date) & '-w' & Week(Date), weekStart(Date)) as YearWeek	In dit voorbeeld wordt een veld gemaakt met tekstwaarden van het formulier en wordt tegelijkertijd een numerieke waarde toegewezen die overeenkomt met het datumnummer van de eerste dag van de week, bijvoorbeeld .YearWeek'2012-W22' 41057. Als u dit in het script wilt gebruiken, moeten de waarden voor Date worden geladen.

### Interval

**Interval()** maakt een getal op als een tijdsinterval met de opmaak die in de systeemvariabelen in het load-script voor gegevens, in het besturingssysteem of in een opmaakreeks is ingesteld, indien opgegeven.

Intervallen kunnen worden opgemaakt als een tijd, als dagen of als een combinatie van dagen, uren, minuten, seconden en fracties van seconden.

#### Syntaxis:

**Interval** (number[, format])

**Retourgegevenstypen:** dual

#### Argumenten:

##### Argumenten

Argument	Beschrijving
number	Het getal dat moet worden opgemaakt.
format	Tekenreeks die beschrijft hoe de resulterende intervalreeks moet worden opgemaakt. Bij weglating wordt gebruikgemaakt van de korte datumnotatie, tijdsnotatie en decimaal scheidingsteken die zijn ingesteld in het besturingssysteem.

#### Voorbeelden en resultaten:

In de onderstaande voorbeelden worden de volgende standaardinstellingen verondersteld:

- Instelling voor datumnotatie 1: YY-MM-DD
- Instelling voor datumnotatie 2: hh:mm:ss
- Scheidingsteken voor decimalen: .

##### Resultatentabel

Voorbeeld	Tekenreeks	Getal
Interval( A ) waarbij A=0,375	09:00:00	0.375

Voorbeeld	Tekenreeks	Getal
Interval( A ) waarbij A=1.375	33:00:00	1.375
Interval( A, 'D hh:mm' ) waarbij A=1.375	1 09:00	1.375
Interval( A-B, 'D hh:mm' ) waarbij A=97-08-06 09:00:00 en B=96-08-06 00:00:00	365 09:00	365.375

### Money

**Money()** maakt een uitdrukking op numerieke wijze op als een geldwaarde in de opmaak die is ingesteld in de systeemvariabelen die zijn ingesteld in het load-script voor gegevens of in het besturingssysteem, tenzij een opmaakreeks is opgegeven, alsmede optionele scheidingstekens voor decimalen en duizendtallen.

#### Syntaxis:

```
Money( number[, format[, dec_sep[, thou_sep]])
```

**Retourgegevenstypen:** dual

#### Argumenten:

##### Argumenten

Argument	Beschrijving
number	Het getal dat moet worden opgemaakt.
format	Tekenreeks die beschrijft hoe de resulterende valutareeks moet worden opgemaakt.
dec_sep	Tekenreeks die het scheidingsteken voor decimalen opgeeft.
thou_sep	Tekenreeks die het scheidingsteken voor duizendtallen opgeeft.

Als de argumenten 2-4 zijn weggelaten, wordt de getalnotatie van het besturingssysteem gebruikt.

Voorbeelden en resultaten:

In de onderstaande voorbeelden worden de volgende standaardinstellingen verondersteld:

- Instelling 1 voor geldopmaak: kr ##0,00, MoneyThousandSep''
- Instelling 2 voor geldopmaak: \$ #,##0.00, MoneyThousandSep','

#### Voorbeeld:

```
Money( A )  
waarbij A=35648
```

Resultatentabel

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	kr 35 648,00	\$ 35,648.00
Getal:	35648.00	35648.00

### Voorbeeld:

```
Money( A, '#,##0 ¥', '.' , ',' )
waarbij A=3564800
```

Resultatentabel

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	3,564,800 ¥	3,564,800 ¥
Getal:	3564800	3564800

## Num

**Num()** geeft de opmaak van een getal, wat wil zeggen dat deze functie de numerieke waarde van de invoer converteert naar tekst met gebruikmaking van de opmaak die is opgegeven in de tweede parameter. Als de tweede parameter wordt weggelaten, worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

### Syntaxis:

```
Num(number[, format[, dec_sep [, thou_sep]])
```

### Retourgegevenstypen: dual

De functie Num retourneert een dubbele waarde met zowel de tekenreeks als de numerieke waarde. De functie neemt de numerieke waarde van de invoeruitdrukking en genereert een tekenreeks die het getal vertegenwoordigt.

### Argumenten:

Argumenten

Argument	Beschrijving
number	Het getal dat moet worden opgemaakt.
format	Tekenreeks die specificeert hoe de resulterende tekenreeks moet worden opgemaakt. Bij weglating worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens.

Argument	Beschrijving
dec_sep	Tekenreeks die het scheidingsteken voor decimalen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor de variabele DecimalSep die is ingesteld in het load-script voor gegevens.
thou_sep	Tekenreeks die het scheidingsteken voor duizendtallen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor de variabele ThousandSep die is ingesteld in het load-script voor gegevens.

Voorbeeld: Diagramuitdrukking

### Voorbeeld:

De volgende tabel toont de resultaten wanneer veld A gelijk is aan 35648.312.

Resultaten	
Een	Resultaat
Num(A)	35648.312 (afhankelijk van omgevingsvariabelen in script)
Num(A, '0.0', ',')	35648.3
Num(A, '0,00', ',')	35648,31
Num(A, '#,##0.0', ',', ',')	35,648.3
Num(A, '# ##0', ',', ',')	35 648

Voorbeeld: Load-script

### Load-script

Zelfs als de scheidingstekens voor duizendtallen en decimalen al in het script zijn ingesteld, kan *Num* worden gebruikt voor de notatie van een getal in load-script. Het onderstaande load-script bevat specifieke scheidingstekens voor duizendtallen en decimalen, maar vervolgens is *Num* gebruikt om de gegevens op andere manieren te noteren.

Maak een nieuwe sectie aan in de **Editor voor laden van gegevens**, voeg vervolgens het voorbeeldscript toe en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in uw app toe om het resultaat te bekijken.

```
SET ThousandSep=','; SET DecimalSep='.'; Transactions: Load *, Num(transaction_amount) as [No
formatting], Num(transaction_amount,'0') as [0], Num(transaction_amount,'#,#0') as [#,#0],
Num(transaction_amount,'# ###,00') as [# ###,00], Num(transaction_amount,'# ###,00',',',',')
as [# ###,00 , ', ' , ' '], Num(transaction_amount,'#,###.00',',',',') as [#,###.00 , '.',
','], Num(transaction_amount,'$,###.00') as [$$,###.00], ; Load * Inline [ transaction_id,
transaction_date, transaction_amount, transaction_quantity, discount, customer_id, size,
color_code 3750, 20180830, 12423.56, 23, 0,2038593, L, Red 3751, 20180907, 5356.31, 6, 0.1,
203521, m, orange 3752, 20180916, 15.75, 1, 0.22, 5646471, s, blue 3753, 20180922, 1251, 7, 0,
3036491, l, black 3754, 20180922, 21484.21, 1356, 75, 049681, xs, Red 3756, 20180922, -59.18,
2, 0.3333333333333333, 2038593, m, blue 3757, 20180923, 3177.4, 21, .14, 203521, XL, black ];
```

## 5 Script- en diagramfuncties

Klik Sense-tabel met de resultaten van verschillende manieren waarop de functie *Num* in het load-script wordt gebruikt. Ter illustratie bevat de vierde kolom van de tabel het gebruik van incorrecte notatie.

Geen opmaak	0	#,##0	# ###,00	# ###,00 , ',' , ''	#,###.00 , ',' , ''	\$\$,###.00
-59.18	-59	-59	-59###,00	-59,18	-59.18	\$-59,18
15.75	16	16	16###,00	15,75	15.75	\$15,75
1251	1251	1,251	1251###,00	1 251,00	1,251.00	\$1,251.00
3177.4	3177	3,177	3177###,00	3 177,40	3,177.40	\$3,177.40
5356.31	5356	5,356	5356###,00	5 356,31	5,356.31	\$5,356.31
12423.56	12424	12,424	12424###,00	12 423,56	12,423.56	\$12,423.56
21484.21	21484	21,484	21484###,00	21 484,21	21,484.21	\$21,484.21

Voorbeeld: Load-script

### Load-script

*Num* kan worden gebruikt in een load-script om getallen als percentage te noteren.

Maak een nieuwe sectie aan in de **Editor voor laden van gegevens**, voeg vervolgens het voorbeeldscript toe en voer het uit. Voeg vervolgens ten minste de velden die in de resultatenkolom staan aan een werkblad in uw app toe om het resultaat te bekijken.

```
SET ThousandSep=','; SET DecimalSep='.'; Transactions: Load *, Num(discount,'#,#0%') as [Discount #,#0%] ; Load * Inline [ transaction_id, transaction_date, transaction_amount, transaction_quantity, discount, customer_id, size, color_code 3750, 20180830, 12423.56, 23, 0,2038593, L, Red 3751, 20180907, 5356.31, 6, 0.1, 203521, m, orange 3752, 20180916, 15.75, 1, 0.22, 5646471, s, blue 3753, 20180922, 1251, 7, 0, 3036491, l, Black 3754, 20180922, 21484.21, 1356, 75, 049681, xs, Red 3756, 20180922, -59.18, 2, 0.3333333333333333, 2038593, M, Blue 3757, 20180923, 3177.4, 21, .14, 203521, xL, Black ];
```

Klik Sense-tabel met de resultaten van de functie *Num* die in het load-script wordt gebruikt voor het noteren van percentages.

Korting	Korting #,#0%
0.3333333333333333	33%
0.22	22%
0	0%
.14	14%
0.1	10%
0	0%
75	7,500%

### Time

**Time()** maakt een uitdrukking op als een tijdwaarde in de tijdnotatie die in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

#### Syntaxis:

```
Time(number[, format])
```

**Retourgegevenstypen:** dual

#### Argumenten:

##### Argumenten

Argument	Beschrijving
number	Het getal dat moet worden opgemaakt.
format	Tekenreeks die beschrijft hoe de resulterende tijdreeks moet worden opgemaakt. Bij weglating wordt gebruikgemaakt van de korte datumnotatie, tijdnotatie en decimaal scheidingstekens die zijn ingesteld in het besturingssysteem.

Voorbeelden en resultaten:

In de onderstaande voorbeelden worden de volgende standaardinstellingen verondersteld:

- Instelling voor tijdnotatie 1: hh:mm:ss
- Instelling voor tijdnotatie 2: hh.mm.ss

#### Voorbeeld:

Time( A )  
waarbij A=0,375

##### Resultatentabel

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	09:00:00	09.00.00
Getal:	0.375	0.375

#### Voorbeeld:

Time( A )  
waarbij A=35648,375



Resultatentabel

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	09:00:00	09.00.00
Getal:	35648.375	35648.375

### Voorbeeld:

Time( A, 'hh-mm' )  
 waarbij A=0,99999

Resultatentabel

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	23-59	23-59
Getal:	0.99999	0.99999

## Timestamp

**TimeStamp()** maakt een uitdrukking op als een datum- en tijdwaarde in de notatie voor tijdstempels die in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

### Syntaxis:

**TimeStamp**(number[, format])

**Retourgegevenstypen:** dual

### Argumenten:

Argumenten

Argument	Beschrijving
number	Het getal dat moet worden opgemaakt.
format	Tekenreeks die beschrijft hoe de resulterende tekenreeks voor de tijdstempel moet worden opgemaakt. Bij weglating wordt gebruikgemaakt van de korte datumnotatie, tijdnotatie en decimaal scheidingsteken die zijn ingesteld in het besturingssysteem.

Voorbeelden en resultaten:

In de onderstaande voorbeelden worden de volgende standaardinstellingen verondersteld:

- Instelling voor TimeStampFormat 1: YY-MM-DD hh:mm:ss
- Instelling voor TimeStampFormat 2: M/D/YY hh:mm:ss

### Voorbeeld:

Timestamp( A )  
waarbij A=35648,375

Resultatentabel

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	97-08-06 09:00:00	8/6/97 09:00:00
Getal:	35648.375	35648.375

### Voorbeeld:

Timestamp( A, 'YYYY-MM-DD hh.mm' )  
waarbij A=35648

Resultaattabel

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	1997-08-06 00.00	1997-08-06 00.00
Getal:	35648	35648

## 5.13 Algemene numerieke functies

In deze algemene numerieke functies zijn de argumenten uitdrukkingen waarbij **x** moet worden geïnterpreteerd als een reëel getal. Alle functies kunnen zowel worden gebruikt in scripts voor het laden van gegevens als in diagramuitdrukkingen.

### Overzicht van algemene numerieke functies

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

bitcount

**BitCount()** retourneert hoeveel bits in het binaire equivalent van een getal zijn ingesteld op 1. De functie retourneert dus het aantal setbits in **integer\_number**, waarbij **integer\_number** wordt geïnterpreteerd als een ondertekend 32-bits geheel getal.

```
BitCount(integer_number)
```

div

**Div()** retourneert het gehele deel van de rekenkundige deling van het eerste argument door het tweede argument. Beide parameters worden geïnterpreteerd als reële getallen. Ze hoeven dus geen gehele getallen te zijn.

```
Div (integer_number1, integer_number2)
```

`fabs`

**Fabs()** retourneert de absolute waarde van `x`. Het resultaat is een positief getal.

```
Fabs (x)
```

`fact`

**Fact()** retourneert de faculteit van een positief geheel getal `x`.

```
Fact (x)
```

`frac`

**Frac()** retourneert het breukgedeelte van `x`.

```
Frac (x)
```

`sign`

**Sign()** retourneert 1, 0 of -1, afhankelijk van het feit of `x` een positief getal, 0 of een negatief getal is.

```
Sign (x)
```

### Combinatie- en permutatiefuncties

`combin`

**Combin()** retourneert het aantal combinaties van `q` elementen dat kan worden gekozen uit een groep van `p` elementen. Zoals weergegeven door de formule:  $\text{combin}(p, q) = \frac{p!}{q!(p-q)!}$ . De volgorde waarin de items worden geselecteerd is niet van belang.

```
Combin (p, q)
```

`permut`

**Permut()** retourneert het aantal permutaties van `q` elementen dat kan worden geselecteerd uit een groep van `p` elementen. Zoals weergegeven door de formule:  $\text{permut}(p, q) = \frac{(p)!}{(p - q)!}$ . De volgorde waarin de items worden geselecteerd is van belang.

```
Permut (p, q)
```

### Modulofuncties

`fmod`

**fmod()** is een gegeneraliseerde modulo-functie die het gehele deel van de rekenkundige deling van het eerste argument (het deeltal) op het tweede argument (de deler) retourneert. Het resultaat is een reëel getal. Beide argumenten worden geïnterpreteerd als reële getallen. Ze hoeven dus geen gehele getallen te zijn.

```
Fmod (a, b)
```

`mod`

**Mod()** is een wiskundige modulo-functie die de niet-negatieve restwaarde van een deling van gehele getallen retourneert. Het eerste argument is het deeltal en het tweede argument de deler. Beide argumenten moeten gehele getallen zijn.

```
Mod (integer_number1, integer_number2)
```

### Pariteitsfuncties

even

**Even()** retourneert True (-1), als **integer\_number** een even geheel getal of nul is. Retourneert False (0), als **integer\_number** een oneven geheel getal is en NULL als **integer\_number** niet een geheel getal is.

```
Even (integer_number)
```

odd

**Odd()** retourneert True (-1), als **integer\_number** een oneven geheel getal of nul is. Retourneert False (0), als **integer\_number** een even geheel getal is en NULL als **integer\_number** niet een geheel getal is.

```
Odd (integer_number)
```

### Afrondingsfuncties

ceil

**Ceil()** rondt een getal naar boven af op het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset**-getal.

```
Ceil (x[, step[, offset]])
```

floor

**Floor()** rondt een getal naar beneden af op het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset**-getal.

```
Floor (x[, step[, offset]])
```

round

**Round()** retourneert het resultaat van de afronding omhoog of omlaag naar het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset**-getal.

```
Round ( x [ , step [ , offset ] ] )
```

### BitCount

**BitCount()** retourneert hoeveel bits in het binaire equivalent van een getal zijn ingesteld op 1. De functie retourneert dus het aantal setbits in **integer\_number**, waarbij **integer\_number** wordt geïnterpreteerd als een ondertekend 32-bits geheel getal.

**Syntaxis:**

```
BitCount(integer_number)
```

**Retourgegevenstypen:** geheel getal

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

Voorbeelden	Resultaten
<code>BitCount ( 3 )</code>	3 is binair 101, dus wordt 2 geretourneerd
<code>BitCount ( -1 )</code>	-1 is 64 enen in binaire code, dus wordt hierbij 64 geretourneerd

### Ceil

**Ceil()** rondt een getal naar boven af op het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset** -getal.

Vergelijk deze functie met de functie **floor**, waarbij ingevoerde cijfers naar beneden worden afgerond.

**Syntaxis:**

```
Ceil(x[, step[, offset]])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
<b>x</b>	Invoergetal.
<b>step</b>	Increment van interval. De standaardwaarde is 1.
<b>offset</b>	Definieert de basis van de stapinterval. De standaardwaarde is 0.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

Voorbeelden	Resultaten
<code>ceil(2.4 )</code>	retourneert 3  In dit voorbeeld is de grootte van de stap 1 en is de basis van de stapinterval 0.  De intervallen zijn ... $0 < x \leq 1$ , $1 < x \leq 2$ , <b><math>2 &lt; x \leq 3</math></b> , $3 < x \leq 4$ ...
<code>ceil(4.2 )</code>	retourneert 5

Voorbeelden	Resultaten
<code>ceil(3.88 ,0.1)</code>	retourneert 3,9  In dit voorbeeld is de grootte van de interval 0,1 en is de basis van de interval 0.  De intervallen zijn ... $3.7 < x \leq 3.8$ , <b><math>3.8 &lt; x \leq 3.9</math></b> , $3.9 < x \leq 4.0$ ...
<code>ceil(3.88 ,5)</code>	retourneert 5
<code>ceil(1.1 ,1)</code>	retourneert 2
<code>ceil(1.1 ,1,0.5)</code>	retourneert 1,5  In dit voorbeeld is de grootte van de stap 1 en is de beginwaarde 0,5. Dit betekent dat de basis van de stapinterval 0,5 is en niet 0.  De intervallen zijn ... <b><math>0.5 &lt; x \leq 1.5</math></b> , $1.5 < x \leq 2.5$ , $2.5 < x \leq 3.5$ , $3.5 < x \leq 4.5$ ...
<code>ceil(1.1 ,1,-0.01)</code>	retourneert 1,99  De intervallen zijn ... $-0.01 < x \leq 0.99$ , <b><math>0.99 &lt; x \leq 1.99</math></b> , $1.99 < x \leq 2.99$ ...

## Combin

**Combin()** retourneert het aantal combinaties van **q** elementen dat kan worden gekozen uit een groep van **p** elementen. Zoals weergegeven door de formule:  $\text{combin}(p, q) = \frac{p!}{q!(p-q)!}$  De volgorde waarin de items worden geselecteerd is niet van belang.

### Syntaxis:

`Combin(p, q)`

**Retourgegevenstypen:** geheel getal

### Beperkingen:

Elementen die niet een geheel getal zijn, worden afgekapt.

### Voorbeelden en resultaten:

Voorbeelden en resultaten

Voorbeelden	Resultaten
Hoeveel combinaties van 7 getallen kunnen worden gekozen uit een totaal van 35 lottogetallen?  <code>combin( 35,7 )</code>	retourneert 6.724.520

### Div

**Div()** retourneert het gehele deel van de rekenkundige deling van het eerste argument door het tweede argument. Beide parameters worden geïnterpreteerd als reële getallen. Ze hoeven dus geen gehele getallen te zijn.

**Syntaxis:**

```
Div(integer_number1, integer_number2)
```

**Retourgegevenstypen:** geheel getal

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

Voorbeelden	Resultaten
Div( 7,2 )	retourneert 3
Div( 7.1,2.3 )	retourneert 3
Div( 9,3 )	retourneert 3
Div( -4,3 )	retourneert -1
Div( 4,-3 )	retourneert -1
Div( -4,-3 )	retourneert 1

### Even

**Even()** retourneert True (-1), als **integer\_number** een even geheel getal of nul is. Retourneert False (0), als **integer\_number** een oneven geheel getal is en NULL als **integer\_number** niet een geheel getal is.

**Syntaxis:**

```
Even(integer_number)
```

**Retourgegevenstypen:** Booleaanse waarde

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

Voorbeelden	Resultaten
Even( 3 )	Retourneert 0, False
Even( 2 * 10 )	Retourneert -1, True
Even( 3.14 )	Retourneert NULL

### Fabs

**Fabs()** retourneert de absolute waarde van **x**. Het resultaat is een positief getal.

### Syntaxis:

```
fabs (x)
```

**Retourgegevenstypen:** numeriek

### Voorbeelden en resultaten:

Voorbeelden en resultaten

Voorbeelden	Resultaten
<code>fabs( 2.4 )</code>	retourneert 2,4
<code>fabs( -3.8 )</code>	retourneert 3,8

## Fact

**Fact()** retourneert de faculteit van een positief geheel getal **x**.

### Syntaxis:

```
Fact (x)
```

**Retourgegevenstypen:** geheel getal

### Beperkingen:

Als het getal **x** geen geheel getal is, wordt het afgekapt. Niet-positieve getallen retourneren NULL.

### Voorbeelden en resultaten:

Voorbeelden en resultaten

Voorbeelden	Resultaten
<code>Fact( 1 )</code>	retourneert 1
<code>Fact( 5 )</code>	Retourneert 120 (1 * 2 * 3 * 4 * 5 = 120)
<code>Fact( -5 )</code>	Retourneert NULL

## Floor

**Floor()** rondt een getal naar beneden af op het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset** -getal.

Vergelijk deze functie met de functie **ceil**, waarbij ingevoerde getallen naar boven worden afgerond.

### Syntaxis:

```
Floor (x[, step[, offset]])
```



**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
<b>x</b>	Invoergetal.
<b>step</b>	Increment van interval. De standaardwaarde is 1.
<b>offset</b>	Definieert de basis van de stapinterval. De standaardwaarde is 0.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

Voorbeelden	Resultaten
Floor(2.4)	retourneert 2  In this example, the size of the step is 1 and the base of the step interval is 0.  The intervals are ... $0 \leq x < 1$ , $1 \leq x < 2$ , <b><math>2 \leq x &lt; 3</math></b> , $3 \leq x < 4$ ...
Floor(4.2)	retourneert 4
Floor(3.88 ,0.1)	retourneert 3,8  In dit voorbeeld is de grootte van de interval 0,1 en is de basis van de interval 0.  De intervallen zijn ... $3.7 \leq x < 3.8$ , <b><math>3.8 \leq x &lt; 3.9</math></b> , $3.9 \leq x < 4.0$ ...
Floor(3.88 ,5)	Retourneert 0
Floor(1.1 ,1)	retourneert 1
Floor(1.1 ,1,0.5)	retourneert 0,5  In dit voorbeeld is de grootte van de stap 1 en is de beginwaarde 0,5. Dit betekent dat de basis van de stapinterval 0,5 is en niet 0.  De intervallen zijn ... <b><math>0.5 \leq x &lt; 1.5</math></b> , $1.5 \leq x < 2.5$ , $2.5 \leq x < 3.5$ ,...

## Fmod

**fmod()** is een gegeneraliseerde modulo-functie die het gehele deel van de rekenkundige deling van het eerste argument (het deeltal) op het tweede argument (de deler) retourneert. Het resultaat is een reëel getal. Beide argumenten worden geïnterpreteerd als reële getallen. Ze hoeven dus geen gehele getallen te zijn.

**Syntaxis:**

**fmod**(a, b)

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
<b>a</b>	Deeltal
<b>b</b>	Deler

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

Voorbeelden	Resultaten
<code>fmod( 7, 2 )</code>	retourneert 1
<code>fmod( 7.5, 2 )</code>	retourneert 1,5
<code>fmod( 9, 3 )</code>	retourneert 0
<code>fmod( -4, 3 )</code>	retourneert -1
<code>fmod( 4, -3 )</code>	retourneert 1
<code>fmod( -4, -3 )</code>	retourneert -1

## Frac

**Frac()** retourneert het breukgedeelte van **x**.

De fractie is zodanig geformuleerd dat  $\text{Frac}(x) + \text{Floor}(x) = x$ . In eenvoudige termen betekent dit dat het fractionele deel van een positief getal het verschil is tussen het getal ( $x$ ) en het gehele getal dat aan het fractionele deel voorafgaat.

Bijvoorbeeld: Het fractionele deel van 11,43 =  $11,43 - 11 = 0,43$

Voor een negatief getal, bijvoorbeeld -1,4,  $\text{Floor}(-1.4) = -2$ , hetgeen het volgende resultaat oplevert:

Het fractionele deel van -1,4 =  $1,4 - (-2) = -1,4 + 2 = 0,6$

**Syntaxis:**

```
Frac(x)
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
x	Getal waarvoor fractie wordt geretourneerd.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

Voorbeelden	Resultaten
Frac( 11.43 )	retourneert 0,43
Frac( -1.4 )	retourneert 0,6
Extraheer het tijdelement uit de numerieke representatie van een tijdstempel waardoor de datum wordt weggelaten.  Time(Frac(44518.663888889))	retourneert 3:56:00 PM

### Mod

**Mod()** is een wiskundige modulo-functie die de niet-negatieve restwaarde van een deling van gehele getallen retourneert. Het eerste argument is het deeltal en het tweede argument de deler. Beide argumenten moeten gehele getallen zijn.

**Syntaxis:**

```
Mod(integer_number1, integer_number2)
```

**Retourgegevenstypen:** geheel getal

**Beperkingen:**

`integer_number2` moet groter zijn dan 0.

**Voorbeelden en resultaten:**

Voorbeelden en resultaten

Voorbeelden	Resultaten
Mod( 7,2 )	retourneert 1
Mod( 7.5,2 )	Retourneert NULL

Voorbeelden	Resultaten
Mod( 9, 3 )	retourneert 0
Mod( -4, 3 )	retourneert 2
Mod( 4, -3 )	Retourneert NULL
Mod( -4, -3 )	Retourneert NULL

### Odd

**Odd()** retourneert True (-1), als **integer\_number** een oneven geheel getal of nul is. Retourneert False (0), als **integer\_number** een even geheel getal is en NULL als **integer\_number** niet een geheel getal is.

#### Syntaxis:

```
Odd(integer_number)
```

**Retourgegevenstypen:** Booleaanse waarde

#### Voorbeelden en resultaten:

Voorbeelden en resultaten

Voorbeelden	Resultaten
odd( 3 )	Retourneert -1, True
odd( 2 * 10 )	Retourneert 0, False
odd( 3.14 )	Retourneert NULL

### Permut

**Permut()** retourneert het aantal permutaties van **q** elementen dat kan worden geselecteerd uit een groep van **p** elementen. Zoals weergegeven door de formule:  $Permut(p, q) = \frac{p!}{(p - q)!}$  De volgorde waarin de items worden geselecteerd is van belang.

#### Syntaxis:

```
Permut(p, q)
```

**Retourgegevenstypen:** geheel getal

#### Beperkingen:

Argumenten die niet een geheel getal zijn, worden afgekapt.

## Voorbeelden en resultaten:

## Voorbeelden en resultaten

Voorbeelden	Resultaten
Op hoeveel manieren kunnen de gouden, zilveren en bronzen medailles worden verdeeld na een finale op de 100 m met 8 deelnemers?  Permut( 8,3 )	retourneert 336

## Round

**Round()** retourneert het resultaat van de afronding omhoog of omlaag naar het dichtstbijzijnde veelvoud van de **step** die is verschoven met het **offset** -getal.

Als het getal dat moet worden afgerond zich precies in het midden van een interval bevindt, wordt er naar boven afgerond.

## Syntaxis:

```
Round(x[, step[, offset]])
```

**Retourgegevenstypen:** numeriek



*Als u een getal met zwevend decimaalteken afrondt, krijgt u mogelijk onjuiste resultaten. Deze afrondingsfouten worden veroorzaakt doordat getallen met zwevend decimaalteken worden weergegeven door een eindig aantal binaire cijfers. Dit heeft tot gevolg dat er een berekening wordt uitgevoerd met een getal dat al is afgerond. Als deze afrondingsfouten van invloed zijn op uw werk, vermenigvuldigt u de getallen om ze om te zetten in gehele getallen voordat u afrondt.*

## Argumenten:

## Argumenten

Argument	Beschrijving
<b>x</b>	Invoergetal.
<b>step</b>	Increment van interval. De standaardwaarde is 1.
<b>offset</b>	Definieert de basis van de stapinterval. De standaardwaarde is 0.

### Voorbeelden en resultaten:

Voorbeelden en resultaten

Voorbeelden	Resultaten
Round(3.8 )	retourneert 4  In dit voorbeeld is de grootte van de stap 1 en is de basis van de stapinterval 0.  De intervallen zijn ... $0 \leq x < 1$ , $1 \leq x < 2$ , $2 \leq x < 3$ , $3 \leq x < 4$ ...
Round(3.8,4 )	retourneert 4
Round(2.5 )	Retourneert 3.  In dit voorbeeld is de grootte van de stap 1 en is de basis van de stapinterval 0.  De intervallen zijn ... $0 \leq x < 1$ , $1 \leq x < 2$ , $2 \leq x < 3$ ...
Round(2,4 )	Retourneert 4. Wordt naar boven afgerond omdat 2 precies de helft van de stapinterval van 4 is.  In dit voorbeeld is de grootte van de stap 4 en is de basis van de stapinterval 0.  De intervallen zijn ... $0 \leq x < 4$ , $4 \leq x < 8$ , $8 \leq x < 12$ ...
Round(2,6 )	Retourneert 0. Wordt naar beneden afgerond omdat 2 minder dan de helft van de stapinterval van 6 is.  In dit voorbeeld is de grootte van de stap 6 en is de basis van de stapinterval 0.  De intervallen zijn ... $0 \leq x < 6$ , $6 \leq x < 12$ , $12 \leq x < 18$ ...
Round(3.88 ,0.1)	retourneert 3,9  In dit voorbeeld is de grootte van de stap 0,1 en is de basis van de stapinterval 0.  De intervallen zijn ... $3.7 \leq x < 3.8$ , $3.8 \leq x < 3.9$ , $3.9 \leq x < 4.0$ ...
Round(3.88875,1/1000)	Retourneert 3,889  In dit voorbeeld is de stapgrootte 0,001, waardoor het getal naar boven wordt afgerond op drie decimalen achter de komma.
Round(3.88 ,5)	retourneert 5
Round(1.1 ,1,0.5)	retourneert 1,5  In dit voorbeeld is de grootte van de stap 1 en is de basis van de stapinterval 0,5.  De intervallen zijn ... $0.5 \leq x < 1.5$ , $1.5 \leq x < 2.5$ , $2.5 \leq x < 3.5$ ...

## Sign

**Sign()** retourneert 1, 0 of -1, afhankelijk van het feit of x een positief getal, 0 of een negatief getal is.

### Syntaxis:

**Sign** (x)

**Retourgegevenstypen:** numeriek

### Beperkingen:

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

### Voorbeelden en resultaten:

Voorbeelden en resultaten

Voorbeelden	Resultaten
sign( 66 )	retourneert 1
sign( 0 )	retourneert 0
sign( - 234 )	retourneert -1

## 5.14 Geospatiale functies

Deze functies worden gebruikt voor het verwerken van geospatiale gegevens in kaartvisualisaties. Qlik Sense volgt de GeoJSON-specificaties voor geospatiale gegevens en ondersteunt het volgende:

- Punt
- Linestring
- Polygon
- Multipolygon

Voor meer informatie over GeoJSON-specificaties raadpleegt u:

≤ [GeoJSON.org](https://geojson.org/)

### Overzicht van geospatiale functies

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

Er zijn twee categorieën geospatiale functies: aggregatie- en niet-aggregatiefuncties.

Aggregatiefuncties nemen een geometrieset (punten of gebieden) als invoer en geven een enkele geometrie als resultaat. Zo kunnen bijvoorbeeld meerdere gebieden worden samengevoegd, waarbij een enkele grens voor de aggregatie op de kaart kan worden getekend.

Niet-aggregatiefuncties nemen een enkele geometrie en retourneren één geometrie. Als bijvoorbeeld voor de functie `\GeoGetPolygonCenter()`, de grensgeometrie van het ene gebied is ingesteld als invoer, wordt de puntgeometrie (lengtegraad en breedtegraad) voor het midden van dat gebied geretourneerd.

De volgende functies zijn aggregatiefuncties:

### **GeoAggrGeometry**

**GeoAggrGeometry()** wordt gebruikt voor het aggregeren van een aantal gebieden tot een groter gebied, bijvoorbeeld het aggregeren van een aantal subregio's tot een regio.

```
GeoAggrGeometry (field_name)
```

### **GeoBoundingBox**

**GeoBoundingBox()** wordt gebruikt in scripts voor aggregatie van een geometrie in een gebied en voor berekening van het kleinste kader dat alle coördinaten bevat.

```
GeoBoundingBox (field_name)
```

### **GeoCountVertex**

**GeoCountVertex()** wordt gebruikt voor het zoeken van het aantal hoekpunten dat een polygoongeometrie bevat.

```
GeoCountVertex (field_name)
```

### **GeoInvProjectGeometry**

**GeoInvProjectGeometry()** wordt voor aggregatie van een geometrie in een gebied en voor toepassing van de inverse van een projectie.

```
GeoInvProjectGeometry (type, field_name)
```

### **GeoProjectGeometry**

**GeoProjectGeometry()** wordt voor aggregatie van een geometrie in een gebied en voor toepassing van een projectie.

```
GeoProjectGeometry (type, field_name)
```

### **GeoReduceGeometry**

**GeoReduceGeometry()** wordt gebruikt om het aantal hoekpunten van een geometrie te verminderen, en om meerdere gebieden te aggregeren tot één gebied waarbij wel de grenslijnen van de afzonderlijke gebieden worden getoond.

```
GeoReduceGeometry (geometry)
```

De volgende functies zijn niet-aggregatiefuncties:

### **GeoGetBoundingBox**

**GeoGetBoundingBox()** wordt gebruikt in scripts en diagramuitdrukkingen om het kleinste geospatiale kader te berekenen dat alle coördinaten van een geometrie bevat.

```
GeoGetBoundingBox (geometry)
```

### **GeoGetPolygonCenter**

**GeoGetPolygonCenter()** wordt gebruikt in scripts en diagramuitdrukkingen voor het berekenen en retourneren van het middelpunt van een geometrie.



```
GeoGetPolygonCenter (geometry)
```

### GeoMakePoint

**GeoMakePoint()** wordt gebruikt in scripts en diagramuitdrukkingen om een punt met lengtegraad en breedtegraad te maken en aan te duiden.

```
GeoMakePoint (lat_field_name, lon_field_name)
```

### GeoProject

**GeoProject()** wordt gebruikt in scripts en diagramuitdrukkingen voor toepassing van een projectie op een geometrie.

```
GeoProject (type, field_name)
```

## GeoAggrGeometry

**GeoAggrGeometry()** wordt gebruikt voor het aggregeren van een aantal gebieden tot een groter gebied, bijvoorbeeld het aggregeren van een aantal subregio's tot een regio.

### Syntaxis:

```
GeoAggrGeometry (field_name)
```

**Retourgegevenstypen:** tekenreeks

### Argumenten:

#### Argumenten

Argument	Beschrijving
field_name	Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied.

Gewoonlijk kan **GeoAggrGeometry()** worden gebruikt voor het combineren van geospatiale grensgegevens. U beschikt bijvoorbeeld over postcodegebieden voor randgemeenten in een stad en de verkoopomzet voor elk gebied. Als het rayon van een verkoper verschillende postcodegebieden omvat, kan het handig zijn de totale omzet per verkooprayon te presenteren in plaats van afzonderlijke gebieden en de resultaten weer te geven op een kaart met kleurarceringen.

Met **GeoAggrGeometry()** kan de aggregatie van de geometrieën van de afzonderlijke randgemeenten worden berekend en de geometrie voor het samengevoegde rayon worden gegenereerd in het gegevensmodel. Als vervolgens de grenzen van het verkooprayon worden aangepast, worden de nieuwe samengevoegde grenzen en omzetgegevens weergegeven op de kaart als de gegevens opnieuw worden geladen.

**GeoAggrGeometry()** is een aggregatiefunctie. Wanneer u het in een script gebruikt, is een **LOAD**-instructie met de clausule **Sorteren op** vereist.



De grenslijnen van kaarten die zijn gemaakt met **GeoAggrGeometry()** zijn de grenslijnen van de samengevoegde gebieden. Als u de afzonderlijke grenslijnen van de vooraf geaggregeerde gebieden wilt weergeven, gebruikt u **GeoReduceGeometry()**.

Voorbeelden:

In dit voorbeeld wordt een KML-bestand met regiogegevens geladen. Daarna wordt een tabel geladen met de samengevoegde gegevens.

```
[MapSource]: LOAD [world.Name], [world.Point], [world.Area] FROM [lib://Downloads/world.kml]
(kml, Tabel is [world.shp/Features]); Map: LOAD world.Name, GeoAggrGeometry(world.Area) as
[AggrArea] resident MapSource Group By world.Name;
```

```
Drop Table MapSource;
```

### GeoBoundingBox

**GeoBoundingBox()** wordt gebruikt in scripts voor aggregatie van een geometrie in een gebied en voor berekening van het kleinste kader dat alle coördinaten bevat.

Een GeoBoundingBox wordt weergegeven als een lijst met vier waarden: links, rechts, boven, beneden.

**Syntaxis:**

```
GeoBoundingBox (field_name)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

Argument	Beschrijving
field_name	Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied.

GeoBoundingBox() aggregeert een reeks van geometrieën en retourneert vier coördinaten voor de kleinste rechthoek die alle coördinaten van die geaggregeerde geometrie bevat.

Als u het resultaat wilt visualiseren op een kaart, brengt u de resulterende reeks van vier coördinaten over naar een polygoonvorm, tagt u het overgedragen veld met een polygoonvorm en sleept u dat veld in het kaartobject. De rechthoekige vakken worden vervolgens weergegeven in de kaartvisualisatie.

### GeoCountVertex

**GeoCountVertex()** wordt gebruikt voor het zoeken van het aantal hoekpunten dat een polygoongeometrie bevat.

**Syntaxis:**

```
GeoCountVertex (field_name)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

Argument	Beschrijving
field_name	Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied.

### GeoGetBoundingBox

**GeoGetBoundingBox()** wordt gebruikt in scripts en diagramuitdrukkingen om het kleinste geospatiale kader te berekenen dat alle coördinaten van een geometrie bevat.

Een geospatiaal kader, dat is gemaakt met de functie `GeoBoundingBox()`, wordt weergegeven als een lijst met vier waarden: links, rechts, boven, beneden.

**Syntaxis:**

```
GeoGetBoundingBox (field_name)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

Argument	Beschrijving
field_name	Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied.



*Maak geen gebruik van de **Group by**-clausule in de editor voor het laden van gegevens bij deze en andere niet-aggregerende geospatiale functies, omdat dit tot een fout bij het laden leidt.*

### GeoGetPolygonCenter

**GeoGetPolygonCenter()** wordt gebruikt in scripts en diagramuitdrukkingen voor het berekenen en retourneren van het middelpunt van een geometrie.

In sommige gevallen moet een punt worden getekend in plaats van een kleurarcering op een kaart. Als de bestaande geospatiale gegevens niet beschikbaar zijn in de vorm van gebiedsgeometrie (bijvoorbeeld een grens), gebruikt u **GeoGetPolygonCenter()** om een set lengte- en breedtegraden voor op te halen voor het midden van het gebied.

### Syntaxis:

```
GeoGetPolygonCenter(field_name)
```

**Retourgegevenstypen:** tekenreeks

### Argumenten:

#### Argumenten

Argument	Beschrijving
field_name	Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied.



*Maak geen gebruik van de **Group by**-clausule in de editor voor het laden van gegevens bij deze en andere niet-aggregerende geospatiale functies, omdat dit tot een fout bij het laden leidt.*

## GeoInvProjectGeometry

**GeoInvProjectGeometry()** wordt voor aggregatie van een geometrie in een gebied en voor toepassing van de inverse van een projectie.

### Syntaxis:

```
GeoInvProjectGeometry(type, field_name)
```

**Retourgegevenstypen:** tekenreeks

### Argumenten:

#### Argumenten

Argument	Beschrijving
type	Projectietype dat wordt gebruikt bij het transformeren van de geometrie van de kaart. Hierbij zijn twee waarden mogelijk: 'eenheid' (standaard), hetgeen in een 1:1-projectie, of 'mercator', waarbij de standaard Mercator-projectie wordt gebruikt.
field_name	Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied.

Voorbeeld:

Scriptvoorbeeld

Voorbeeld	Resultaat
In een Load-opdracht: GeoInvProjectGeometry ( 'mercator', AreaPolygon) as InvProjectGeometry	De geometrie die is geladen als <b>AreaPolygon</b> wordt getransformeerd met de inverse transformatie van de Mercator-projectie en opgeslagen als <b>InvProjectGeometry</b> voor gebruik in visualisaties.

### GeoMakePoint

**GeoMakePoint()** wordt gebruikt in scripts en diagramuitdrukkingen om een punt met lengtegraad en breedtegraad te maken en aan te duiden. Door GeoMakePoint worden punten teruggegeven in volgorde van lengte- en breedtegraad.

**Syntaxis:**

```
GeoMakePoint(lat_field_name, lon_field_name)
```

**Retourgegevenstypen:** tekenreeks, opgemaakt [lengtegraad, breedtegraad]

**Argumenten:**

Argumenten

Argument	Beschrijving
lat_field_name	Een veld dat of uitdrukking die verwijst naar een veld die de breedtegraad van het punt aangeeft.
lon_field_name	Een veld dat of uitdrukking die verwijst naar een veld die de lengtegraad van het punt aangeeft.



*Maak geen gebruik van de **Group by**-clausule in de editor voor het laden van gegevens bij deze en andere niet-aggregerende geospatiale functies, omdat dit tot een fout bij het laden leidt.*

### GeoProject

**GeoProject()** wordt gebruikt in scripts en diagramuitdrukkingen voor toepassing van een projectie op een geometrie.

**Syntaxis:**

```
GeoProject(type, field_name)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

### Argumenten

Argument	Beschrijving
type	Projectietype dat wordt gebruikt bij het transformeren van de geometrie van de kaart. Hierbij zijn twee waarden mogelijk: 'eenheid' (standaard), hetgeen in een 1:1-projectie, of 'mercator', waarbij de Mercator-projectie voor het web wordt gebruikt.
field_name	Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied.



*Maak geen gebruik van de **Group by**-clausule in de editor voor het laden van gegevens bij deze en andere niet-aggregerende geospatiale functies, omdat dit tot een fout bij het laden leidt.*

Voorbeeld:

### Scriptvoorbeelden

Voorbeeld	Resultaat
In een Load-opdracht: GeoProject ( 'mercator', Area ) as GetProject	De Mercator-projectie wordt toegepast op de geometrie die is geladen als <b>Area</b> , en het resultaat wordt opgeslagen als <b>GetProject</b> .

## GeoProjectGeometry

**GeoProjectGeometry()** wordt voor aggregatie van een geometrie in een gebied en voor toepassing van een projectie.

**Syntaxis:**

```
GeoProjectGeometry (type, field_name)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

### Argumenten

Argument	Beschrijving
type	Projectietype dat wordt gebruikt bij het transformeren van de geometrie van de kaart. Hierbij zijn twee waarden mogelijk: 'eenheid' (standaard), hetgeen in een 1:1-projectie, of 'mercator', waarbij de Mercator-projectie voor het web wordt gebruikt.
field_name	Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied.

Voorbeeld:

Voorbeeld	Resultaat
In een Load-opdracht: GeoProjectGeometry ( 'mercator', AreaPolygon) as ProjectGeometry	De geometrie die is geladen als <b>AreaPolygon</b> wordt getransformeerd met de Mercator-projectie en opgeslagen als <b>ProjectGeometry</b> voor gebruik in visualisaties.

## GeoReduceGeometry

**GeoReduceGeometry()** wordt gebruikt om het aantal hoekpunten van een geometrie te verminderen, en om meerdere gebieden te aggregeren tot één gebied waarbij wel de grenslijnen van de afzonderlijke gebieden worden getoond.

**Syntaxis:**


```
GeoReduceGeometry (field_name[, value])
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

### Argumenten

Argument	Beschrijving
field_name	Een veld dat of uitdrukking die verwijst naar een veld dat de weer te geven geometrie bevat. Dit kan een punt (of reeks van punten) zijn met lengtegraad en breedtegraad, of een gebied.

Argument	Beschrijving
value	<p>De mate van reductie die moet worden toegepast op de geometrie. Het bereik loopt van 0 tot 1, waarbij 0 geen reductie is en 1 maximale reductie van de hoekpunten.</p> <div style="border: 1px solid gray; padding: 5px;"> <i>Als u een value kiest van 0,9 of hoger voor een complexe gegevensverzameling, kan het aantal hoekpunten dusdanig worden gereduceerd dat de visuele representatie onnauwkeurig is.</i></div>

**GeoReduceGeometry()** voert ook een functie uit die vergelijkbaar is met **GeoAggrGeometry()**. Een aantal gebieden wordt namelijk geaggregeerd tot één gebied. Het verschil is dat afzonderlijke grenslijnen uit de vooraf geaggregeerde gegevens op de kaart worden weergegeven als u **GeoReduceGeometry()** gebruikt.

**GeoReduceGeometry()** is een aggregatiefunctie. Wanneer u het in een script gebruikt, is een **LOAD**-instructie met de clause **Sorteren op** vereist.

Voorbeelden:

In dit voorbeeld wordt een KML-bestand met regiogegevens geladen. Daarna wordt een tabel geladen met de verminderde samengevoegde gegevens.

```
[MapSource]: LOAD [world.Name], [world.Point], [world.Area] FROM [lib://Downloads/world.kml]
(kml, Tabel is [world.shp/Features]); Map: LOAD world.Name, GeoReduceGeometry(world.Area,0.5)
as [ReducedArea] resident MapSource Group By world.Name;
```

```
Drop Table MapSource;
```

### 5.15 Interpretatiefuncties

De interpretatiefuncties evalueren de inhoud van invoertekstvelden of uitdrukkingen en leggen een specifieke gegevensopmaak op aan de resulterende numerieke waarde. Met deze functies kunt u de opmaak opgeven van het getal, in overeenstemming met het gegevenstype, met inbegrip van eigenschappen als: scheidingsteken voor decimalen, scheidingsteken voor duizendtallen en datumnotatie.

Alle interpretatiefuncties retourneren een dubbele waarde met zowel de tekenreeks als de getalwaarde, maar kunnen worden beschouwd als een conversie van tekenreeks naar getal. De functies nemen de tekstwaarde van de invoeruitdrukking en genereren een getal dat de tekenreeks vertegenwoordigt.

De opmaakfuncties, daarentegen, doen het omgekeerde: zij nemen numerieke uitdrukkingen en evalueren deze als tekenreeksen, waarbij de weergave-indeling van de resulterende tekst wordt opgegeven.

Als geen interpretatiefuncties worden gebruikt, worden de gegevens in Qlik Sense geïnterpreteerd als een combinatie van getallen, datums, tijden, tijdstempels en tekenreeksen. Daarbij wordt uitgegaan van de standaardinstellingen voor de notatie van getallen, datum en tijd zoals die in de scriptvariabelen en het besturingssysteem zijn gedefinieerd.



Alle interpretatiefuncties kunnen zowel worden gebruikt in scripts voor het laden van gegevens als in diagramuitdrukkingen.



*Alle getallen moeten worden weergegeven met een decimale punt als scheidingsteken voor decimalen.*

### Overzicht van interpretatiefuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Date#

**Date#** evalueert een uitdrukking als een datum in de opmaak die is gespecificeerd in het tweede argument, indien opgegeven. Als de opmaakcode is weggelaten, wordt de standaard datumnotatie die is ingesteld in het besturingssysteem gebruikt.

```
Date# (page 1258) (text[, format])
```

#### Interval#

**Interval#()** evalueert een tekstuitdrukking als een tijdsinterval in de opmaak die standaard is ingesteld in het besturingssysteem of in de opmaak die is gespecificeerd in het tweede argument, indien opgegeven.

```
Interval# (page 1259) (text[, format])
```

#### Money#

**Money#()** converteert een tekstenreeks naar een geldwaarde, in de indeling die is ingesteld in het load-script of in het besturingssysteem, tenzij een opmaakreeks is opgegeven. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

```
Money# (page 1260) (text[, format[, dec_sep[, thou_sep ] ] ])
```

#### Num#

**Num#()** interpreteert een tekenreeks als een numerieke waarde, wat wil zeggen dat deze functie de invoerreeks converteert naar een getal met gebruikmaking van de opmaak die is opgegeven in de tweede parameter. Als de tweede parameter wordt weggelaten, worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

```
Num# (page 1261) (text[ , format[, dec_sep[ , thou_sep]]])
```

#### Text

**Text()** dwingt af dat de uitdrukking wordt behandeld als tekst, ook als een numerieke interpretatie mogelijk is.

```
Text (expr)
```

### Time#

**Time#()** evalueert een uitdrukking als een tijdwaarde in de tijdnotatie die in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven..

```
Time# (page 1263) (text[, format])
```

### Timestamp#

**Timestamp#()** evalueert een uitdrukking als een datum- en tijdwaarde in de notatie voor tijdstempels die in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

```
Timestamp# (page 1264) (text[, format])
```

### Zie ook:

p *Opmaakfuncties (page 1221)*

### Date#

**Date#** evalueert een uitdrukking als een datum in de opmaak die is gespecificeerd in het tweede argument, indien opgegeven.

### Syntaxis:

```
Date# (text[, format])
```

**Retourgegevenstypen:** dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
text	De teksttekenreeks die moet worden geëvalueerd.
format	Tekenreeks waarmee de notatie van de te evalueren tekenreeks wordt beschreven. Als dit argument wordt weggelaten, wordt de datumopmaak gebruikt die is ingesteld in de systeemvariabelen in het load-script voor gegevens of in het besturingssysteem.

### Voorbeelden en resultaten:

In het volgende voorbeeld wordt gebruikgemaakt van de datumopmaak **M/D/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan het load-script voor gegevens.

Voeg het voorbeeldscript toe aan uw app en voer dit uit.

```
Load *,
Num(Date#(StringDate)) as Date;
LOAD * INLINE [
StringDate
8/7/97
8/6/1997
```

]

Als u een tabel maakt met **StringDate** en **Date** als dimensies, zijn de resultaten als volgt:

Resultaten

StringDate	Date
8/7/97	35649
8/6/1997	35648

### Interval#

**Interval#()** evalueert een tekstuitdrukking als een tijdsinterval in de opmaak die standaard is ingesteld in het besturingssysteem of in de opmaak die is gespecificeerd in het tweede argument, indien opgegeven.

#### Syntaxis:

```
Interval#(text[, format])
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

Argument	Beschrijving
text	De tekstoneenreeks die moet worden geëvalueerd.
format	Tekeneenreeks die de verwachte invoerindeling beschrijft bij het converteren van de tekstoneenreeks naar een numeriek interval.  Bij weglating wordt gebruikgemaakt van de korte datumnotatie, tijdsnotatie en decimaal scheidingstekens die zijn ingesteld in het besturingssysteem.

Met de functie **interval#** wordt een tijdsinterval in tekstvorm geconverteerd naar een numeriek equivalent.

Voorbeelden en resultaten:

In de onderstaande voorbeelden worden de volgende instellingen van het besturingssysteem verondersteld:

- Korte datumnotatie YY-MM-DD
- Tijdsnotatie: M/D/YY
- Scheidingstekens voor decimalen: .

Resultaten

Voorbeeld	Resultaat
<code>Interval#( A, 'D hh:mm' )</code> waarbij A='1 09:00'	1.375

## Money#

**Money#()** converteert een teksttekenreeks naar een geldwaarde, in de indeling die is ingesteld in het load-script of in het besturingssysteem, tenzij een opmaakreeks is opgegeven. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

### Syntaxis:

```
Money#(text[, format[, dec_sep [, thou_sep ] ] ])
```

**Retourgegevenstypen:** dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
text	De teksttekenreeks die moet worden geëvalueerd.
format	Tekenreeks die de verwachte invoerindeling beschrijft bij het converteren van de tekenreeks naar een numeriek interval.  Bij weglating wordt de geldopmaak die is ingesteld in het besturingssysteem gebruikt.
dec_sep	Tekenreeks die het scheidingsteken voor decimalen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor MoneyDecimalSep die is ingesteld in het load-script voor gegevens.
thou_sep	Tekenreeks die het scheidingsteken voor duizendtallen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor MoneyThousandSep die is ingesteld in het load-script voor gegevens.

De functie **money#** gedraagt zich net zoals de functie **num#**, maar haalt de standaardwaarden voor het scheidingsteken voor decimalen en duizendtallen uit de scriptvariabelen voor geldnotatie of de systeeminstellingen voor valuta.

Voorbeelden en resultaten:

In de voorbeelden hieronder worden de volgende instellingen van het besturingssysteem verondersteld:

- Standaardinstelling voor geldopmaak 1: kr # ##0,00
- Standaardinstelling voor geldopmaak 2: \$ #,##0.00

```
Money#(A , '# ##0,00 kr' )
```

waarbij A=35 648,37 kr

Resultaten

Resultaten	Instelling 1	Instelling 2
Tekenreeks	35 648.37 kr	35 648.37 kr
Getal	35648.37	3564837

Money#( A, ' \$#', '.', ', ' )  
 waarbij A= \$35.648,37

Resultaten

Resultaten	Instelling 1	Instelling 2
Tekenreeks	\$35,648.37	\$35,648.37
Getal	35648.37	35648.37

### Num#

**Num#()** interpreteert een tekenreeks als een numerieke waarde, wat wil zeggen dat deze functie de invoerreeks converteert naar een getal met gebruikmaking van de opmaak die is opgegeven in de tweede parameter. Als de tweede parameter wordt weggelaten, worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens. Aangepaste scheidingstekens voor decimalen en scheidingstekens voor duizendtallen zijn optionele parameters.

#### Syntaxis:

```
Num# (text[, format[, dec_sep [, thou_sep ] ] ] )
```

#### Retourgegevenstypen: dual

De functie **Num#()** retourneert een dubbele waarde met zowel de tekenreeks als de numerieke waarde. De functie neemt de tekstuele representatie van de invoeruitdrukking en genereert een getal. De notatie van het getal wordt niet gewijzigd: de uitvoer wordt in dezelfde notatie weergegeven als de invoer.

#### Argumenten:

Argumenten

Argument	Beschrijving
text	De teksttekenreeks die moet worden geëvalueerd.
format	Tekenreeks die specificeert welke getalopmaak wordt gebruikt in de eerste parameter. Bij weglating worden de scheidingstekens voor decimalen en duizendtallen gebruikt die zijn ingesteld in het load-script voor gegevens.
dec_sep	Tekenreeks die het scheidingsteken voor decimalen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor de variabele DecimalSep die is ingesteld in het load-script voor gegevens.

Argument	Beschrijving
thou_sep	Tekenreeks die het scheidingsteken voor duizendtallen opgeeft. Bij weglating wordt gebruikgemaakt van de waarde voor de variabele ThousandSep die is ingesteld in het load-script voor gegevens.

Voorbeelden en resultaten:

De volgende tabel toont het resultaat van *Num#( A, '#', '.', ',')* voor verschillende waarden van A.

Een	Representatie van tekenreeks	Resultaten
		Numerieke waarde (hier weergegeven met decimale punt)
35,648.31	35,648.31	35648.31
35 648.312	35 648.312	35648.312
35.648,3123	35.648,3123	-
35 648,31234	35 648,31234	-

### Text

**Text()** dwingt af dat de uitdrukking wordt behandeld als tekst, ook als een numerieke interpretatie mogelijk is.

**Syntaxis:**

**Text** (expr)

**Retourgegevenstypen:** dual

**Voorbeeld:**

Text( A )  
waarbij A=1234

Resultaten	
Tekenreeks	Getal
1234	-

**Voorbeeld:**

Text( pi( ) )

Resultaten	
Tekenreeks	Getal
3.1415926535898	-

### Time#

**Time#()** evalueert een uitdrukking als een tijdwaarde in de tijdnotatie die in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven..

#### Syntaxis:

```
time#(text[, format])
```

**Retourgegevenstypen:** dual

#### Argumenten:

##### Argumenten

Argument	Beschrijving
text	De teksttekenreeks die moet worden geëvalueerd.
format	Tekenreeks waarmee de notatie van de te evalueren tekenreeks wordt beschreven. Bij weglating wordt gebruikgemaakt van de korte datumnotatie, tijdnotatie en decimaal scheidingsteken die zijn ingesteld in het besturingssysteem.

#### Voorbeeld:

- Standaardinstelling voor tijdnotatie 1: hh:mm:ss
- Standaardinstelling voor tijdnotatie 2: hh.mm.ss

```
time#( A )  
waar A=09:00:00
```

##### Resultaten

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	09:00:00	09:00:00
Getal:	0.375	-

#### Voorbeeld:

- Standaardinstelling voor tijdnotatie 1: hh:mm:ss
- Standaardinstelling voor tijdnotatie 2: hh.mm.ss

```
time#( A, 'hh.mm' )  
waar A=09.00
```

Resultaten

Resultaten	Instelling 1	Instelling 2
Tekenreeks:	09.00	09.00
Getal:	0.375	0.375

### Timestamp#

**Timestamp#()** evalueert een uitdrukking als een datum- en tijdwaarde in de notatie voor tijdstempels die in het load-script voor gegevens of in het besturingssysteem is ingesteld, tenzij een opmaakreeks is opgegeven.

#### Syntaxis:

```
timestamp#(text[, format])
```

**Retourgegevenstypen:** dual

#### Argumenten:

Argumenten

Argument	Beschrijving
text	De tekstenreeks die moet worden geëvalueerd.
format	Tekenreeks waarmee de notatie van de te evalueren tekenreeks wordt beschreven. Bij weglating wordt gebruikgemaakt van de korte datumnotatie, tijdnotatie en decimaal scheidingstekens die zijn ingesteld in het besturingssysteem. ISO 8601 wordt ondersteund voor tijdstempels.

#### Voorbeeld:

In het volgende voorbeeld wordt gebruikgemaakt van de datumopmaak **M/D/YYYY**. De datumopmaak wordt opgegeven in de opdracht **SET DateFormat** boven aan het load-script voor gegevens.

Voeg het voorbeeldscript toe aan uw app en voer dit uit.

```
Load *,
Timestamp(Timestamp#(String)) as TS;
LOAD * INLINE [
Tekenreeks
2015-09-15T12:13:14
1952-10-16T13:14:00+0200
1109-03-01T14:15
];
```

Als u een tabel maakt met **String** en **TS** als dimensies, zijn de resultaten als volgt:



## Resultaten

Tekenreeks	TS
2015-09-15T12:13:14	9/15/2015 12:13:14 PM
1952-10-16T13:14:00+0200	10/16/1952 11:14:00 AM
1109-03-01T14:15	3/1/1109 2:15:00 PM

## 5.16 Interrecord-functies

Interrecord-functies worden gebruikt:

- In het load-script voor gegevens, wanneer een waarde uit eerder geladen records van gegevens nodig is voor de evaluatie van de huidige record.
- In een diagramuitdrukking, wanneer een andere waarde uit de gegevensverzameling van een visualisatie nodig is.



*Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als een interrecord-diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u een interrecord-diagramfunctie is een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van de interrecordfunctie. Deze beperking is niet van toepassing op de overeenkomstige scriptfunctie, indien die er is.*



*Definities van naar zichzelf verwijzende uitdrukkingen kunnen alleen op betrouwbare wijze worden gemaakt in tabellen met minder dan 100 rijen, maar dit kan variëren afhankelijk van de hardware waarop de Qlik-engine wordt uitgevoerd.*

## Rijfuncties

Deze functies kunnen alleen worden gebruikt in diagramuitdrukkingen.

Above

**Above()** evalueert een uitdrukking in een rij boven de huidige rij binnen een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De direct bovenliggende rij is de standaardwaarde. Voor andere diagrammen dan tabellen, evalueert **Above()** voor de rij boven de huidige rij in het equivalent van de standaard tabel in het diagram.

```
Above - diagramfunctie([TOTAL [<fld{,fld}>]] expr [ , offset [,count]])
```

Below

**Below()** evalueert een uitdrukking in een rij onder de huidige rij binnen een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De direct onderliggende rij is de standaardwaarde. Voor andere diagrammen dan tabellen, evalueert **Below()** voor de rij onder de huidige kolom in het equivalent van de standaard tabel in het diagram.

```
Below - diagramfunctie([TOTAL [<fld{,fld}>]] expression [ , offset [,count ]])
```

### Bottom

**Bottom()** evalueert een uitdrukking in de laatste rij (onderste rij) van een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De onderste rij is de standaardwaarde. Voor andere diagrammen dan tabellen, wordt de laatste rij van de huidige kolom in het equivalent van de strakke tabel in het diagram geëvalueerd.

```
Bottom - diagramfunctie([TOTAL[<fld{,fld}>]] expr [ , offset [,count ]])
```

### Top

**Top()** evalueert een uitdrukking in de eerste rij (bovenste rij) van een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De bovenste rij is de standaardwaarde. Voor andere diagrammen dan tabellen, wordt bij **Top()** de eerste rij van de huidige kolom in het equivalent van de standaard tabel in het diagram geëvalueerd.

```
Top - diagramfunctie([TOTAL [<fld{,fld}>]] expr [ , offset [,count ]])
```

### NoOfRows

**NoOfRows()** retourneert het aantal rijen in het huidige kolomsegment in een tabel. Voor bitmapgrafieken wordt met **NoOfRows()** het aantal rijen in het equivalent van de standaard tabel voor de grafiek geretourneerd.

```
NoOfRows - diagramfunctie([TOTAL])
```

## Kolomfuncties

Deze functies kunnen alleen worden gebruikt in diagramuitdrukkingen.

### Column

**Column()** retourneert de waarde in de kolom die overeenkomt met **ColumnNo** in een strakke tabel, waarbij dimensies buiten beschouwing worden gelaten. Zo retourneert **Column(2)** bijvoorbeeld de waarde van de tweede kolom met metingen.

```
Column - diagramfunctie(ColumnNo)
```

### Dimensionality

**Dimensionality()** retourneert het aantal dimensies voor de huidige rij. In het geval van draaitabellen retourneert de functie het totale aantal kolommen met dimensies met een niet-geaggregeerde inhoud, dat wil zeggen kolommen die geen subtotaal of samengevouwen aggregaten bevatten.

```
Dimensionality - diagramfunctie ( )
```

### Secondarydimensionality

**SecondaryDimensionality()** retourneert het aantal rijen in een draaitabel met een niet-geaggregeerde inhoud, dat wil zeggen kolommen die geen subtotaal of samengevouwen aggregaten bevatten. Deze functie is het equivalent van de functie **dimensionality()** voor horizontale dimensies in draaitabellen.

```
SecondaryDimensionality - diagramfunctie ( )
```

### Veldfuncties

FieldIndex

**FieldIndex()** retourneert de positie van de veldwaarde **value** in het veld **field\_name** (op volgorde van laden).

```
FieldIndex (field_name , value)
```

FieldValue

**FieldValue()** retourneert de waarde die is gevonden in positie **elem\_no** van het veld **field\_name** (op volgorde van laden).

```
FieldValue (field_name , elem_no)
```

FieldValueCount

**FieldValueCount()** is een functie voor **geheel getal** waarmee het aantal distinctieve waarden in een veld wordt gevonden.

```
FieldValueCount (field_name)
```

### Draaitabelfuncties

Deze functies kunnen alleen worden gebruikt in diagramuitdrukkingen.

After

retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de kolom voor de huidige kolom in een rijsegment in de draaitabel.

```
After - diagramfunctie([TOTAL] expression [ , offset [,n]])
```

Before

**Before()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de kolom voor de huidige kolom in een rijsegment in de draaitabel.

```
Before - diagramfunctie([TOTAL] expression [ , offset [,n]])
```

First

**First()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de eerste kolom van het huidige rijsegment in de draaitabel. Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.

```
First - diagramfunctie([TOTAL] expression [ , offset [,n]])
```

Last

**Last()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de laatste kolom van het huidige rijsegment in de draaitabel. Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.

```
Last - diagramfunctie([TOTAL] expression [ , offset [,n]])
```

ColumnNo

**ColumnNo()** retourneert het nummer van de huidige kolom in het huidige rijsegment in een draaitabel. De eerste kolom is nummer 1.

```
ColumnNo - diagramfunctie ([TOTAL])
```

NoOfColumns

**NoOfColumns()** retourneert het aantal kolommen in het huidige rijsegment in een draaitabel.

```
NoOfColumns - diagramfunctie ([TOTAL])
```

### Interrecord-functies in het load-script voor gegevens

**Exists**

**Exists()** bepaalt of een specifieke veldwaarde al is geladen in het veld in het script voor het laden van gegevens. De functie retourneert TRUE of FALSE, zodat deze kan worden gebruikt in de **where**-clausule van een **LOAD**-opdracht of een **IF**-opdracht.

```
Exists (field_name [, expr])
```

**LookUp**

**Lookup()** zoekt u een tabel die al is geladen en retourneert de waarde van **field\_name** die overeenkomt met het eerste exemplaar van de waarde **match\_field\_value** in het veld **match\_field\_name**. De tabel kan de huidige tabel zijn of een andere tabel die eerder is geladen.

```
LookUp (field_name, match_field_name, match_field_value [, table_name])
```

**Peek**

**Peek()** retourneert de waarde van een veld in een tabel voor een rij die al is geladen. Het rijnummer kan worden opgegeven, net als de tabel. Als er geen rijnummer is opgegeven, wordt het laatst geladen record gebruikt.

```
Peek (field_name[, row_no[, table_name ] ])
```

**Previous**

**Previous()** retourneert de waarde van de uitdrukking **expr** met gebruik van de gegevens uit de vorige invoerrecord die niet is genegeerd wegens een **where**-clausule. In de eerste record van een interne tabel retourneert de functie NULL.

```
Previous (page 1303) (expr)
```

---

**Zie ook:**

p *Bereikfuncties (page 1324)*

### Above - diagramfunctie

**Above()** evalueert een uitdrukking in een rij boven de huidige rij binnen een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De direct bovenliggende rij is de standaardwaarde. Voor andere diagrammen dan tabellen, evalueert **Above()** voor de rij boven de huidige rij in het equivalent van de standaard tabel in het diagram.

### Syntaxis:

```
Above ([TOTAL] expr [ , offset [,count]])
```

Retourgegevenstypen: dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
offset	<p>Als een groter dan 0 wordt opgegeven, wordt de evaluatie van de uitdrukking rijen verder omhoog verplaatst, geteld vanaf de huidige rij.offsetnn</p> <p>Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de huidige rij.</p> <p>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie Above op dezelfde wijze als de functie Below met het overeenkomstige positieve getal als beginwaarde.</p>
count	<p>Als een derde argument <b>count</b> groter dan 1 wordt opgegeven, retourneert de functie niet één maar een bereik van <b>count</b>-waarden. Eén waarde voor elke <b>count</b> tabelrijen, omhoog geteld vanaf de oorspronkelijke cel.</p> <p>In deze vorm kan de functie worden gebruikt als argument voor een van de speciale bereikfuncties. <i>Bereikfuncties (page 1324)</i></p>
TOTAL	Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.

Op de eerste rij van een kolomsegment wordt een NULL-waarde geretourneerd, omdat er geen rij boven de eerste rij is.



*Een kolomsegment wordt gedefinieerd als een opeenvolgende subset van cellen die dezelfde waarden hebben voor de dimensies in de huidige sorteervolgorde. Interrecord-diagramfuncties worden berekend in het kolomsegment met uitzondering van de dimensie uiterst rechts in de equivalente strakke tabel. Als er slechts één dimensie is in het diagram, of als de kwalificatie TOTAL is opgegeven, wordt de uitdrukking geëvalueerd voor de volledige tabel.*



*Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.*

### Beperkingen:

- Recursieve oproepen retourneren een NULL-waarde.
- Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.

### Voorbeelden en resultaten:

#### Example 1:

Tabelvisualisatie voor Voorbeeld 1

Customer	Sum([Sales])	Above(Sum(Sales))	Sum(Sales)+Above(Sum(Sales))	Above offset 3	Higher?
	2566	-	-	-	-
Astrida	587	-	-	-	-
Betacab	539	587	1126	-	-
Canutility	683	539	1222	-	Higher
Divadip	757	683	1440	1344	Higher

In de schermafbeelding van de tabel die in dit voorbeeld wordt weergegeven, wordt de tabelvisualisatie gemaakt op basis van de dimensie **Customer** en de metingen `Sum(Sales)` en `Above(Sum(Sales))`.

De kolom `Above(Sum(Sales))` retourneert NULL voor de rij **Customer** die **Astrida** bevat, omdat er geen rij boven de eerste rij is. Het resultaat voor de rij **Betacab** toont de waarde van `Sum(Sales)` voor **Astrida**, het resultaat voor **Canutility** toont de waarde van `Sum(Sales)` voor **Betacab**, enzovoort.

Voor de kolom met het label `Sum(Sales)+Above(Sum(Sales))` wordt in de rij voor **Betacab** het resultaat getoond van de optelsom van de waarden van `Sum(Sales)` voor de rijen **Betacab** + **Astrida** (539+587). Het resultaat voor de rij **Canutility** bevat het resultaat van de optelsom van de waarden van `Sum(Sales)` voor **Canutility** + **Betacab** (683+539).

De meting met het label `Above offset 3` die is gemaakt met behulp van de uitdrukking `Sum(Sales)+Above(Sum(Sales), 3)` bevat het argument **offset**, dat is ingesteld op 3, en neemt de waarde in de drie rijen boven de huidige rij aan. De waarde van `Sum(Sales)` voor de huidige **Customer** wordt toegevoegd aan de waarde voor de drie bovenliggende rijen van **Customer**. De waarden die worden geretourneerd voor de eerste drie rijen van **Customer** zijn null.

In de tabel worden tevens complexere metingen getoond: één die is gemaakt op basis van `Sum(Sales)+Above(Sum(Sales))` en één met het label **Higher?**, die is gemaakt op basis van `IF(Sum(Sales)>Above(Sum(Sales)), 'Higher')`.



Deze functie kan behalve in tabellen ook worden gebruikt in diagrammen, bijvoorbeeld in staafdiagrammen.



Voor andere diagramtypen converteert u het diagram naar het equivalent van de strakke tabel zodat u gemakkelijk kunt interpreteren op welke rij de functie betrekking heeft.

### Example 2:

In de schermafbeelding van tabellen die worden weergegeven in dit voorbeeld, zijn meer dimensies toegevoegd aan de visualisaties: en . **Month** en **Product**. Voor diagrammen met meer dan één dimensie zijn de resultaten van uitdrukkingen met de functies **Above**, **Below**, **Top** en **Bottom** afhankelijk van de volgorde waarin de kolom met dimensies op Qlik Sense zijn gesorteerd. Qlik Sense evalueert de functies op basis van de kolomsegmenten die het resultaat zijn van de dimensie die als laatste is gesorteerd. De sorteervolgorde van de kolom wordt beheerd in het eigenschappenvenster onder **Sorteren** en is niet per se gelijk aan de volgorde waarin de kolommen in een tabel worden weergegeven.

In de volgende schermafbeelding van tabelvisualisatie voor Voorbeeld 2 is **Month** de als laatste gesorteerde dimensie, dus de functie **Above** wordt geëvalueerd op basis van maanden. Er is een reeks van resultaten voor elke waarde van **Product** voor elke maand (**Jan t/m Aug**) - een kolomsegment. Dit wordt gevolgd door een reeks voor het volgende kolomsegment: voor elke **Month** voor het volgende **Product**. Er is een kolomsegment beschikbaar voor elke waarde van **Customer** voor elk **Product**.

Tabelvisualisatie voor Voorbeeld 2

Customer	Product	Month	Sum([Sales])	Above(Sum(Sales))
			<b>2566</b>	-
Astrida	AA	Jan	46	-
Astrida	AA	Feb	60	46
Astrida	AA	Mar	70	60
Astrida	AA	Apr	13	70
Astrida	AA	May	78	13
Astrida	AA	Jun	20	78
Astrida	AA	Jul	45	20
Astrida	AA	Aug	65	45

### Example 3:

In de schermafbeelding van de tabelvisualisatie voor Voorbeeld 3 is **Product** de als laatste gesorteerde dimensie. Dit wordt gedaan door de dimensie Product te verplaatsen naar positie 3 op het tabblad Sorteren in het eigenschappenvenster. De functie **Above** wordt geëvalueerd voor elk **Product**, en omdat er maar twee producten zijn, **AA** en **BB**, is er slechts één resultaat dat niet null is in elke reeks. In rij **BB** voor de maand **Jan** bedraagt de waarde voor **Above(Sum(Sales))** 46. Voor rij **AA** is de waarde null. De waarde in elke rij **AA** voor elke maand is altijd null, aangezien er geen waarde van **Product** boven AA is. De tweede

## 5 Script- en diagramfuncties

voor de maand **Feb**, voor de waarde van **Customer**, **Astrida**. Als alle maanden zijn geëvalueerd voor , wordt de reeks herhaald voor de tweede waarde van , enzovoort.AstridaCustomerBetacab

Tabelvisualisatie voor Voorbeeld 3

Customer	Product	Month	Sum([Sales])	Above(Sum(Sales))
			<b>2566</b>	-
Astrida	AA	Jan	46	-
Astrida	BB	Jan	46	46
Astrida	AA	Feb	60	-
Astrida	BB	Feb	60	60
Astrida	AA	Mar	70	-
Astrida	BB	Mar	70	70
Astrida	AA	Apr	13	-
Astrida	BB	Apr	13	13

Voorbeeld 4

Example 4:	Resultaat								
<p>De functie Above kan worden gebruikt als invoer voor de bereikfuncties. Bijvoorbeeld: RangeAvg (Above(Sum(Sales),1,3)).</p>	<p>In de argumenten voor de functie Above() is offset ingesteld op 1 en count ingesteld op 3. De functie retourneert de resultaten van de uitdrukking Sum (Sales) voor de drie rijen direct boven de huidige rij in het kolomsegment (waar er een rij is). Deze drie waarden worden gebruikt als invoer voor de functie RangeAvg(), die het gemiddelde van de waarden in het opgegeven bereik van cijfers als resultaat geeft.</p> <p>Een tabel met Customer als dimensie levert de volgende resultaten op voor de uitdrukking RangeAvg().</p> <table><tbody><tr><td>Astrida</td><td>-</td></tr><tr><td>Betacab</td><td>587</td></tr><tr><td>Canutility</td><td>563</td></tr><tr><td>Divadip:</td><td>603</td></tr></tbody></table>	Astrida	-	Betacab	587	Canutility	563	Divadip:	603
Astrida	-								
Betacab	587								
Canutility	563								
Divadip:	603								

Gegevens die worden gebruikt in voorbeelden:

```
Monthnames:
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
```



```
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

```
Sales2013:
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

---

### Zie ook:

[p Below - diagramfunctie \(page 1273\)](#)  
[p Bottom - diagramfunctie \(page 1277\)](#)  
[p Top - diagramfunctie \(page 1304\)](#)  
[p RangeAvg \(page 1327\)](#)

## Below - diagramfunctie

**Below()** evalueert een uitdrukking in een rij onder de huidige rij binnen een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De direct onderliggende rij is de standaardwaarde. Voor andere diagrammen dan tabellen, evalueert **Below()** voor de rij onder de huidige kolom in het equivalent van de standaard tabel in het diagram.

### Syntaxis:

```
Below([TOTAL] expr [ , offset [,count ]])
```

**Retourgegevenstypen:** dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

Argument	Beschrijving
offset	<p>Als een groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking rijen verder omlaag verplaatst, geteld vanaf de huidige rij.offsetnn</p> <p>Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de huidige rij.</p> <p>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie <b>Below</b> op dezelfde wijze als de functie <b>Above</b> met het overeenkomstige positieve getal als beginwaarde.</p>
count	<p>Als een derde parameter <b>count</b> groter dan 1 wordt opgegeven, retourneert de functie niet één maar een bereik van <b>count</b>-waarden. Eén waarde voor elke <b>count</b> tabelrijen, omlaag geteld vanaf de oorspronkelijke cel. In deze vorm kan de functie worden gebruikt als argument voor een van de speciale bereikfuncties. <i>Bereikfuncties (page 1324)</i></p>
TOTAL	<p>Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.</p>

Op de laatste rij van een kolomsegment wordt een NULL-waarde geretourneerd, omdat er geen rij onder de laatste rij is.



*Een kolomsegment wordt gedefinieerd als een opeenvolgende subset van cellen die dezelfde waarden hebben voor de dimensies in de huidige sorteervolgorde. Interrecord-diagramfuncties worden berekend in het kolomsegment met uitzondering van de dimensie uiterst rechts in de equivalente strakke tabel. Als er slechts één dimensie is in het diagram, of als de kwalificatie TOTAL is opgegeven, wordt de uitdrukking geëvalueerd voor de volledige tabel.*



*Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.*

### Beperkingen:

- Recursieve oproepen retourneren een NULL-waarde.
- Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.

### Voorbeelden en resultaten:

#### Example 1:

Tabelvisualisatie voor Voorbeeld 1

Customer	Sum([Sales])	Below(Sum(Sales))	Sum(Sales)+Below(Sum(Sales))	Below + Offset 3	Higher
	2566	-	-	-	-
Astrida	587	539	1126	1344	Higher
Betacab	539	683	1222	-	-
Canutility	683	757	1440	-	-
Divadip	757	-	-	-	-

In de schermafbeelding van Voorbeeld 1 wordt de tabelvisualisatie gemaakt op basis van de dimensie **Customer** en de metingen: `Sum(Sales)` en `Below(Sum(Sales))`.

De kolom **Below(Sum(Sales))** retourneert NULL voor de rij van **Customer** die **Divadip** bevat, omdat er geen rij onder is. Het resultaat voor de rij **Canutility** toont de waarde van `Sum(Sales)` voor **Divadip**, het resultaat voor **Betacab** toont de waarde van `Sum(Sales)` voor **Canutility**, enzovoort.

In de tabel worden tevens complexere metingen weergegeven, die u kunt bekijken in de kolommen `Sum(Sales)+Below(Sum(Sales))`, **Below +Offset 3** en **Higher?**. Deze uitdrukkingen werken zoals beschreven in de volgende alinea's.

Voor de kolom met het label **Sum(Sales)+Below(Sum(Sales))** wordt in de rij voor **Astrida** het resultaat getoond van de optelsom van de waarden van `Sum(Sales)` voor de rijen **Betacab** + **Astrida** (539+587). Het resultaat voor de rij **Betacab** bevat het resultaat van de optelsom van de waarden van `Sum(Sales)` voor **Canutility** + **Betacab** (539+683).

De meting met het label **Below +Offset 3** die is gemaakt met behulp van de uitdrukking `Sum(Sales)+Below(Sum(Sales), 3)` bevat het argument **offset**, dat is ingesteld op 3, en neemt de waarde in de drie rijen onder de huidige rij aan. De waarde van `Sum(Sales)` voor de huidige **Customer** wordt toegevoegd aan de waarde voor de drie onderliggende rijen van **Customer**. De waarden voor de laagste drie rijen van **Customer** zijn null.

De meting met het label **Higher?** wordt gemaakt met de uitdrukking: `IF(Sum(Sales)>Below(Sum(Sales)), 'Higher')`. Hierbij worden de waarden van de huidige rij in de meting `Sum(Sales)` vergeleken met de rij eronder. Als de huidige rij een grotere waarde is, wordt de tekst "Higher" weergegeven.



*Deze functie kan behalve in tabellen ook worden gebruikt in diagrammen, bijvoorbeeld in staafdiagrammen.*



*Voor andere diagramtypen converteert u het diagram naar het equivalent van de strakke tabel zodat u gemakkelijk kunt interpreteren op welke rij de functie betrekking heeft.*

## 5 Script- en diagramfuncties

Voor diagrammen met meer dan één dimensie zijn de resultaten van uitdrukkingen met de functies **Above**, **Below**, **Top** en **Bottom** afhankelijk van de volgorde waarin de kolom met dimensies op Qlik Sense zijn gesorteerd. Qlik Sense evalueert de functies op basis van de kolomsegmenten die het resultaat zijn van de dimensie die als laatste is gesorteerd. De sorteervolgorde van de kolom wordt beheerd in het eigenschappenvenster onder **Sorteren** en is niet per se gelijk aan de volgorde waarin de kolommen in een tabel worden weergegeven. Zie voor meer informatie voorbeeld 2 in de **Above**-functie.

Voorbeeld 2

Example 2:	Resultaat								
De functie <b>Below</b> kan worden gebruikt als invoer voor de bereikfuncties. Bijvoorbeeld: <code>RangeAvg(Below(Sum(Sales),1,3))</code> .	<p>In de argumenten voor de functie <b>Below()</b> is offset ingesteld op 1 en count ingesteld op 3. De functie retourneert de resultaten van de uitdrukking <b>Sum(Sales)</b> voor de drie rijen direct onder de huidige rij in het kolomsegment (waar er een rij is). Deze drie waarden worden gebruikt als invoer voor de functie <code>RangeAvg()</code>, die het gemiddelde van de waarden in het opgegeven bereik van cijfers als resultaat geeft.</p> <p>Een tabel met <b>Customer</b> als dimensie levert de volgende resultaten op voor de uitdrukking <code>RangeAvg()</code>.</p>								
	<table><tbody><tr><td>Astrida</td><td>659.67</td></tr><tr><td>Betacab</td><td>720</td></tr><tr><td>Canutility</td><td>757</td></tr><tr><td>Divadip:</td><td>-</td></tr></tbody></table>	Astrida	659.67	Betacab	720	Canutility	757	Divadip:	-
Astrida	659.67								
Betacab	720								
Canutility	757								
Divadip:	-								

Gegevens die worden gebruikt in voorbeelden:

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
May, 5
Jun, 6
Jul, 7
Aug, 8
Sep, 9
Oct, 10
Nov, 11
Dec, 12
];
```

sales2013:

```
Crosstable (MonthText, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');
```

### Zie ook:

p *Above - diagramfunctie (page 1268)*  
p *Bottom - diagramfunctie (page 1277)*  
p *Top - diagramfunctie (page 1304)*  
p *RangeAvg (page 1327)*

## Bottom - diagramfunctie

**Bottom()** evalueert een uitdrukking in de laatste rij (onderste rij) van een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De onderste rij is de standaardwaarde. Voor andere diagrammen dan tabellen, wordt de laatste rij van de huidige kolom in het equivalent van de strakke tabel in het diagram geëvalueerd.

### Syntaxis:

```
Bottom([TOTAL] expr [ , offset [,count ]])
```

**Retourgegevenstypen:** dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
offset	Als een groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking rijen boven de onderste rij geplaatst.offsetnn  Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie <b>Bottom</b> op dezelfde wijze als de functie <b>Top</b> met het overeenkomstige positieve getal als beginwaarde.
count	Als een derde parameter <b>count</b> groter dan 1 is aangegeven, retourneert de functie niet één maar een bereik van <b>count</b> waarden. Een waarde voor elk van de laatste <b>count</b> rijen van het huidige kolomsegment. In deze vorm kan de functie worden gebruikt als argument voor een van de speciale bereikfuncties. <i>Bereikfuncties (page 1324)</i>
TOTAL	Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.



Een kolomsegment wordt gedefinieerd als een opeenvolgende subset van cellen die dezelfde waarden hebben voor de dimensies in de huidige sorteervolgorde. Interrecord-diagramfuncties worden berekend in het kolomsegment met uitzondering van de dimensie uiterst rechts in de equivalente strakke tabel. Als er slechts één dimensie is in het diagram, of als de kwalificatie TOTAL is opgegeven, wordt de uitdrukking geëvalueerd voor de volledige tabel.



Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.

### Beperkingen:

- Recursieve oproepen retourneren een NULL-waarde.
- Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.

### Voorbeelden en resultaten:

#### Tabelvisualisatie voor Voorbeeld 1

Customer	Sum(Sales)	Bottom(Sum(Sales))	Sum(Sales)+Bottom(Sum(Sales))	Bottom offset 3
	2566	757	3323	3105
Astrida	587	757	1344	1126
Betacab	539	757	1296	1078
Canutility	683	757	1440	1222
Divadip	757	757	1514	1296

In de schermafbeelding van de tabel die in dit voorbeeld wordt weergegeven, wordt de tabelvisualisatie gemaakt op basis van de dimensie **Customer** en de metingen `sum(Sales)` en `Bottom(Sum(Sales))`.

De kolom **Bottom(Sum(Sales))** retourneert 757 voor alle rijen omdat dit de waarde van de onderste rij is: **Divadip**.

In de tabel worden tevens complexere metingen weergegeven: één die is gemaakt op basis van `sum(Sales)+Bottom(Sum(Sales))` en één met het label **Bottom offset 3**, die is gemaakt met behulp van de uitdrukking `sum(Sales)+Bottom(Sum(Sales), 3)` en waarvoor het argument **offset** is ingesteld op 3. Hiermee wordt de waarde **Sum(Sales)** voor de huidige rij toegevoegd aan de waarde van de rij die zich drie rijen boven de onderste rij bevindt. Met andere woorden: de huidige rij plus de waarde voor **Betacab**.

### Voorbeeld: 2

In de schermafbeelding van tabellen die worden weergegeven in dit voorbeeld, zijn meer dimensies toegevoegd aan de visualisaties: en **Month** en **Product**. Voor diagrammen met meer dan één dimensie zijn de resultaten van uitdrukkingen met de functies **Above**, **Below**, **Top** en **Bottom** afhankelijk van de volgorde waarin de kolom met dimensies op Qlik Sense zijn gesorteerd. Qlik Sense evalueert de functies op basis van de kolomsegmenten die het resultaat zijn van de dimensie die als laatste is gesorteerd. De sorteervolgorde van de kolom wordt beheerd in het eigenschappenvenster onder **Sorteren** en is niet per se gelijk aan de volgorde waarin de kolommen in een tabel worden weergegeven.

In de eerste tabel wordt de uitdrukking geëvalueerd op basis van **Month** en in de tweede tabel wordt deze geëvalueerd op basis van **Product**. De meting **End value** bevat de uitdrukking `bottom(Sum(Sales))`. De onderste rij voor **Month** is Dec en de waarde voor Dec beide waarden van **Product** die worden weergegeven de schermafbeelding zijn 22. (Sommige rijen zijn uit de schermafbeelding verwijderd om ruimte te besparen.)

*Eerste tabel voor Voorbeeld 2. De waarde van Bottom voor de meting End value gebaseerd op Month (Dec).*

Customer	Product	Month	Sum(Sales)	End value
			2566	-
Astrida	AA	Jan	46	22
Astrida	AA	Feb	60	22
Astrida	AA	Mar	70	22
Astrida	AA	Sep	78	22
Astrida	AA	Oct	12	22
Astrida	AA	Nov	78	22
Astrida	AA	Dec	22	22
Astrida	BB	Jan	46	22

*Tweede tabel voor Voorbeeld 2. De waarde van Bottom voor de meting End value gebaseerd op Product (BB voor Astrida).*

## 5 Script- en diagramfuncties

Customer	Product	Month	Sum(Sales)	End value
			<b>2566</b>	-
Astrida	AA	Jan	46	46
Astrida	BB	Jan	46	46
Astrida	AA	Feb	60	60
Astrida	BB	Feb	60	60
Astrida	AA	Mar	70	70
Astrida	BB	Mar	70	70
Astrida	AA	Apr	13	13
Astrida	BB	Apr	13	13

Zie voor meer informatie voorbeeld 2 in de **Above**-functie.

### Voorbeeld 3

Voorbeeld: 3	Resultaat
De functie <b>Bottom</b> kan worden gebruikt als invoer voor de bereikfuncties. Bijvoorbeeld: RangeAvg (Bottom(Sum(Sales),1,3)).	In de argumenten voor de functie <b>Bottom()</b> is offset ingesteld op 1 en count ingesteld op 3. De functie retourneert de resultaten van de uitdrukking <b>Sum(Sales)</b> voor de drie rijen te beginnen met de rij boven de onderste rij in het kolomsegment (omdat offset=1) en de twee rijen daarboven (waar er een rij is). Deze drie waarden worden gebruikt als invoer voor de functie RangeAvg(), die het gemiddelde van de waarden in het opgegeven bereik van cijfers als resultaat geeft.  Een tabel met <b>Customer</b> als dimensie levert de volgende resultaten op voor de uitdrukking RangeAvg().
	Astrida 659.67
	Betacab 659.67
	Canutility 659.67
	Divadip: 659.67

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
Apr, 4
```



```
May, 5  
Jun, 6  
Jul, 7  
Aug, 8  
Sep, 9  
Oct, 10  
Nov, 11  
Dec, 12  
];
```

```
Sales2013:  
Crosstable (MonthText, Sales) LOAD * inline [  
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec  
Astrida|46|60|70|13|78|20|45|65|78|12|78|22  
Betacab|65|56|22|79|12|56|45|24|32|78|55|15  
Canutility|77|68|34|91|24|68|57|36|44|90|67|27  
Divadip|57|36|44|90|67|27|57|68|47|90|80|94  
] (delimiter is '|');
```

### Zie ook:

[p Top - diagramfunctie \(page 1304\)](#)

## Column - diagramfunctie

**Column()** retourneert de waarde in de kolom die overeenkomt met **ColumnNo** in een strakke tabel, waarbij dimensies buiten beschouwing worden gelaten. Zo retourneert **Column(2)** bijvoorbeeld de waarde van de tweede kolom met metingen.


### Syntaxis:

```
Column (ColumnNo)
```

**Retourgegevenstypen:** dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
ColumnNo	Kolomnummer van een kolom in de tabel die een meting bevat.  <i>De functie Column() negeert dimensiekolommen.</i>

### Beperkingen:

- Recursieve oproepen retourneren een NULL-waarde.
- Als **ColumnNo** verwijst naar een kolom waarvoor geen meting beschikbaar is, wordt een NULL-waarde geretourneerd.

## 5 Script- en diagramfuncties

---

- Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.

### Voorbeelden en resultaten:

#### Voorbeeld: Percentage totale verkoop

Customer	Product	UnitPrice	UnitSales	Order Value	Total Sales Value	% Sales
A	AA	15	10	150	505	29.70
A	AA	16	4	64	505	12.67
A	BB	9	9	81	505	16.04
B	BB	10	5	50	505	9.90
B	CC	20	2	40	505	7.92
B	DD	25	-	0	505	0.00
C	AA	15	8	120	505	23.76
C	CC	19	-	0	505	0.00

#### Voorbeeld: Percentage verkoop voor geselecteerde klant

Customer	Product	UnitPrice	UnitSales	Order Value	Total Sales Value	% Sales
A	AA	15	10	150	295	50.85
A	AA	16	4	64	295	21.69
A	BB	9	9	81	295	27.46

### Voorbeelden en resultaten

Voorbeelden	Resultaten
Order Value wordt als meting aan de tabel toegevoegd met de uitdrukking: <code>sum (UnitPrice*Unitsales)</code> .	Het resultaat van Column(1) wordt opgehaald uit de kolom Order Value, omdat dit de eerste metingkolom is.
Total Sales Value wordt als meting toegevoegd met de uitdrukking: <code>sum(TOTAL UnitPrice*Unitsales)</code>	Het resultaat van Column(2) wordt opgehaald uit de kolom Total Sales Value, omdat dit de tweede metingkolom is.
% Sales wordt als meting toegevoegd met de uitdrukking <code>100*Column(1)/Column(2)</code>	Zie de resultaten in de kolom % Sales in het voorbeeld <i>Percentage totale verkoop (page 1282)</i> .
Voer de selectie Customer A uit.	De selectie wijzigt de Total Sales Value, en dus de %Sales. Zie het voorbeeld <i>Percentage verkoop voor geselecteerde klant (page 1282)</i> .

Gegevens die worden gebruikt in voorbeelden:

```
ProductData:
LOAD * inline [
Customer|Product|Unitsales|UnitPrice
Astrida|AA|4|16
Astrida|AA|10|15
Astrida|BB|9|9
Betacab|BB|5|10
Betacab|CC|2|20
Betacab|DD||25
Canutility|AA|8|15
Canutility|CC||19
] (delimiter is '|');
```

## Dimensionality - diagramfunctie

**Dimensionality()** retourneert het aantal dimensies voor de huidige rij. In het geval van draaitabellen retourneert de functie het totale aantal kolommen met dimensies met een niet-geaggregeerde inhoud, dat wil zeggen kolommen die geen subtotaal of samengevouwen aggregaten bevatten.

**Syntaxis:**

```
Dimensionality ( )
```

**Retourgegevenstypen:** geheel getal

**Beperkingen:**

Deze functie is alleen beschikbaar in diagrammen. Voor alle diagramtypen, met uitzondering van de draaitabel, retourneert deze functie het aantal dimensies in alle rijen met uitzondering van het totaal dat gelijk is aan 0.

Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.

### Voorbeeld: Diagramuitdrukking met behulp van Dimensionality

Voorbeeld: Diagramuitdrukking

De Dimensionality()-functie kan met een draaitabel als een diagramuitdrukking worden gebruikt wanneer u verschillende celopmaakopties wilt toepassen afhankelijk van het aantal dimensies in een rij die niet-geaggregeerde gegevens heeft. In dit voorbeeld wordt de Dimensionality()-functie gebruikt om een achtergrondkleur toe te passen of tabelcellen die overeenkomen met een bepaalde voorwaarde.

#### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

ProductSales:

```
Load * inline [
Country,Product,Sales,Budget
Sweden,AA,100000,50000
Germany,AA,125000,175000
Canada,AA,105000,98000
Norway,AA,74850,68500
Ireland,AA,49000,48000
Sweden,BB,98000,99000
Germany,BB,115000,175000
Norway,BB,71850,68500
Ireland,BB,31000,48000
] (delimiter is ',');
```

#### Diagramuitdrukking

Maak een draaitabelvisualisatie in een Qlik Sense-werkblad met **Country** en **Product** als dimensies. Voeg **Sum(Sales)**, **Sum(Budget)** en **Dimensionality()** toe als metingen.

Voer in het deelvenster **Properties** de volgende uitdrukking in als de **Background color expression** voor de **Sum(Sales)**-meting:

```
If(Dimensionality()=1 and sum(Sales)<sum(Budget),RGB(255,156,156),
If(Dimensionality()=2 and sum(Sales)<sum(Budget),RGB(178,29,29)
))
```

## Resultaat

Country <input type="text"/>		Values		
Product <input type="text"/>		Sum(Sales)	Sum([Budget])	Dimensionality()
⊖	Canada	105000	98000	1
	AA	105000	98000	2
+	Germany	240000	350000	1
⊖	Ireland	80000	96000	1
	AA	49000	48000	2
	BB	31000	48000	2
⊖	Norway	146700	137000	1
	AA	74850	68500	2
	BB	71850	68500	2
+	Sweden	198000	149000	1

## Uitleg

De uitdrukking `If(Dimensionality()=1 and Sum(Sales)<Sum(Budget),RGB(255,156,156), If(Dimensionality()=2 and Sum(Sales)<Sum(Budget),RGB(178,29,29)))` bevat conditionele statements waarmee de dimensionaliteitswaarde en de `Sum(Sales)` en `Sum(Budget)` voor elk product wordt gecontroleerd. Als aan de voorwaarden wordt voldaan, wordt een achtergrondkleur toegepast op de `Sum(Sales)`-waarde.

## Exists

**Exists()** bepaalt of een specifieke veldwaarde al is geladen in het veld in het script voor het laden van gegevens. De functie retourneert TRUE of FALSE, zodat deze kan worden gebruikt in de **where**-clausule van een **LOAD**-opdracht of een **IF**-opdracht.



*U kunt **Not Exists()** ook gebruiken om te bepalen of een veldwaarde niet is geladen, maar voorzichtigheid is geboden als u **Not Exists()** gebruikt in een where-clausule. De functie **Exists()** test zowel eerder geladen tabellen als eerder geladen waarden in de huidige tabel. Dit betekent dat alleen het eerste exemplaar wordt geladen. Wanneer een tweede exemplaar wordt gevonden, is de waarde al geladen. Bekijk de voorbeelden voor meer informatie.*


## Syntaxis:

```
Exists(field_name [, expr])
```

**Retourgegevenstypen:** Booleaanse waarde

**Argumenten:**

### Argumenten

Argument	Beschrijving
field_name	<p>De naam van het veld waarin u een waarde wilt zoeken. U kunt een expliciete veldnaam gebruiken zonder aanhalingstekens.</p> <p>Het veld moet al door het script zijn geladen. Dit betekent dat u niet naar een veld kunt verwijzen dat in een clause verderop in het script wordt geladen.</p>
expr	<p>De waarde waarvan u wilt controleren of deze bestaat. U kunt een expliciete waarde gebruiken of een uitdrukking die verwijst naar een of meer velden in de huidige load-opdracht.</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;">  <i>U kunt niet verwijzen naar velden die niet in de huidige load-opdracht zijn opgenomen.</i> </div> <p>Dit argument is optioneel. Als u dit weglaat, controleert de functie of de waarde van <b>field_name</b> al bestaat in het huidige record.</p>

Voorbeelden en resultaten:

### Voorbeeld 1

`Exists (Employee)`

Retourneert -1 (True) als de waarde van het veld **Employee** in de huidige record al bestaat in een eerder gelezen record met dat veld.

De opdrachten `Exists (Employee, Employee)` en `Exists (Employee)` zijn equivalent.

### Voorbeeld 2

`Exists(Employee, 'Bill')`

Retourneert -1 (True) als de veldwaarde **'Bill'** wordt gevonden in de huidige inhoud van het veld **Employee**.

### Voorbeeld 3

```
Employees: LOAD * inline [ Employee|ID|Salary Bill|001|20000 John|002|30000 Steve|003|35000 ]
(delimiter is '|'); Citizens: Load * inline [ Employee|Address Bill|New York Mary|London
Steve|Chicago Lucy|Madrid Lucy|Paris John|Miami ] (delimiter is '|') where Exists (Employee);
Drop Tables Employees;
```

## 5 Script- en diagramfuncties

Dit resulteert in een tabel die u kunt gebruiken in een tabelvisualisatie met de dimensies Employee en Address.

De where-clausule, `where Exists (Employee)`, betekent dat alleen de namen uit de tabel Citizens die tevens in Employees voorkomen in de nieuwe tabel worden geladen. Met de Drop-opdracht wordt de tabel Employees verwijderd om verwarring te voorkomen.

Resultaten

Employee	Address
Bill	New York
John	Miami
Steve	Chicago

### Voorbeeld 4

```
Employees: Load * inline [ Employee|ID|Salary Bill|001|20000 John|002|30000 Steve|003|35000 ]
(delimiter is '|'); Citizens: Load * inline [ Employee|Address Bill|New York Mary|London
Steve|Chicago Lucy|Madrid Lucy|Paris John|Miami ] (delimiter is '|') where not Exists
(Employee); Drop Tables Employees;
```

De where-clausule omvat `not: where not Exists (Employee)`.

Dit betekent dat alleen de namen uit de tabel Citizens die niet in Employees voorkomen in de nieuwe tabel worden geladen.

Houd er rekening mee dat hoewel er twee waarden voor Lucy in de tabel Citizens staan, slechts één in de resultaat tabel wordt opgenomen. Wanneer u de eerste rij laadt met de waarde Lucy, wordt de waarde opgenomen in het veld Employee. Dus wanneer vervolgens de tweede rij wordt gecontroleerd, bestaat de waarde al.

Resultaten

Employee	Address
Mary	London
Lucy	Madrid

### Voorbeeld 5

In dit voorbeeld wordt getoond hoe u alle waarden kunt laden.

```
Employees: Load Employee As Name; LOAD * inline [ Employee|ID|Salary Bill|001|20000
John|002|30000 Steve|003|35000 ] (delimiter is '|'); Citizens: Load * inline [
Employee|Address Bill|New York Mary|London Steve|Chicago Lucy|Madrid Lucy|Paris John|Miami ]
(delimiter is '|') where not Exists (Name, Employee); Drop Tables Employees;
```

Om alle waarden voor Lucy te kunnen ophalen, zijn er twee dingen gewijzigd:

- Een voorgaande lading van de tabel Employees was ingevoerd op de plaats waar Employee is gewijzigd in Name.  
Load Employee As Name;
- De Where-voorwaarde in Citizens is gewijzigd in:  
not Exists (Name, Employee).

Hierdoor worden velden voor Name en Employee gemaakt. Wanneer de tweede rij met Lucy wordt gecontroleerd, bestaat deze nog niet in Name.

Resultaten

Employee	Address
Mary	London
Lucy	Madrid
Lucy	Paris

### FieldIndex

**FieldIndex()** retourneert de positie van de veldwaarde **value** in het veld **field\_name** (op volgorde van laden).

#### Syntaxis:

```
FieldIndex(field_name , value)
```

**Retourgegevenstypen:** geheel getal

#### Argumenten:

Argumenten

Argument	Beschrijving
field_name	Naam van het veld waarvoor de index is vereist. Bijvoorbeeld de kolom in een tabel. Moet worden opgegeven als een tekenreekswaarde. Dit betekent dat de veldnaam tussen enkele aanhalingstekens moet staan.
value	De waarde van het veld <b>field_name</b> .

#### Beperkingen:

- Als **value** niet wordt gevonden tussen de veldwaarden van het veld **field\_name**, wordt 0 geretourneerd.
- Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie. Deze beperking is niet van toepassing op de overeenkomstige scriptfunctie.



### Voorbeelden en resultaten:

In de volgende voorbeelden wordt gebruikgemaakt van het veld: **First name** uit de tabel **Names**.

Voorbeelden en resultaten

Voorbeelden	Resultaten
Voeg de voorbeeldgegevens toe aan uw app en voer deze uit.	De tabel <b>Names</b> is geladen, zoals in de steekproefgegevens.
Diagramfunctie: Voeg in een tabel die de dimensie First name bevat het volgende in als een meting:	
FieldIndex ('First name','John')	1, aangezien 'John' als eerste wordt weergegeven in de laadvolgorde van het veld <b>First name</b> . Houd er rekening mee dat <b>John</b> als nummer 2 zou worden weergegeven in een filtervak, omdat hierbij alfabetisch wordt gesorteerd en niet op laadvolgorde.
FieldIndex ('First name', 'Peter')	4, aangezien <b>FieldIndex()</b> slechts één waarde retourneert, namelijk het eerste exemplaar in de laadvolgorde.
- scriptfunctie Gegeven dat de tabel <b>Names</b> is geladen, zoals in de voorbeeldgegevens:	
John1: Load FieldIndex('First name','John') as MyJohnPos Resident Names;	MyJohnPos=1, aangezien 'John' als eerste wordt weergegeven in de laadvolgorde van het veld <b>First name</b> . Houd er rekening mee dat <b>John</b> als nummer 2 zou worden weergegeven in een filtervak, omdat hierbij alfabetisch wordt gesorteerd en niet op laadvolgorde.
Peter1: Load FieldIndex('First name','Peter') as MyPeterPos Resident Names;	MyPeterPos=4, aangezien <b>FieldIndex()</b> slechts één waarde retourneert, namelijk het eerste exemplaar in de laadvolgorde.

### Gebruikte gegevens in voorbeeld:

Names:

```
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

John1:

```
Load FieldIndex('First name','John') as MyJohnPos
Resident Names;
```

```
Peter1:  
Load FieldIndex('First name','Peter') as MyPeterPos  
Resident Names;
```

### FieldValue

**FieldValue()** retourneert de waarde die is gevonden in positie **elem\_no** van het veld **field\_name** (op volgorde van laden).

#### Syntaxis:

```
FieldValue(field_name , elem_no)
```

**Retourgegevenstypen:** dual

#### Argumenten:

##### Argumenten

Argument	Beschrijving
field_name	Naam van het veld waarvoor de waarde is vereist. Bijvoorbeeld de kolom in een tabel. Moet worden opgegeven als een tekenreekswaarde. Dit betekent dat de veldnaam tussen enkele aanhalingstekens moet staan.
elem_no	Het positienummer (elementnummer) van het veld, na de laadvolgorde, waarvoor de waarde wordt geretourneerd. Dit kan overeenkomen met de rij in een tabel, maar het is afhankelijk van de volgorde waaraan de elementen (rijen) worden geladen.

#### Beperkingen:

- Als **elem\_no** groter is dan het aantal veldwaarden, wordt NULL geretourneerd.
- Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie. Deze beperking is niet van toepassing op de overeenkomstige scriptfunctie.

#### Voorbeeld

##### Load-script

Laad de volgende gegevens als een inline load in de editor voor laden van gegevens om het onderstaande voorbeeld te maken.

Names:

```
LOAD * inline [  
First name|Last name|Initials|Has cellphone  
John|Anderson|JA|Yes  
Sue|Brown|SB|Yes  
Mark|Carr|MC |No
```

```
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

John1:

```
Load FieldValue('First name',1) as MyPos1
Resident Names;
```

Peter1:

```
Load FieldValue('First name',5) as MyPos2
Resident Names;
```

### Een visualisatie maken

Maak een tabelvisualisatie in een Qlik Sense-werkblad. Voeg de velden **First name**, **MyPos1** en **MyPos2** toe aan de tabel.

### Resultaat

First name	MyPos1	MyPos2
Jane	John	Jane
John	John	Jane
Mark	John	Jane
Peter	John	Jane
Sue	John	Jane

### Uitleg

**FieldValue('First name','1')** resulteert in John als de waarde voor **MyPos1** voor alle voornamen omdat John als eerste verschijnt in de laadvolgorde van het veld **First name**. Houd er rekening mee dat John in een filtervak als nummer 2 van boven zou worden weergegeven, na Jane, omdat hierbij alfabetisch wordt gesorteerd en niet op laadvolgorde.

**FieldValue('First name','5')** resulteert in Jane als de waarde voor **MyPos2** voor alle voornamen omdat Jane als vijfde verschijnt in de laadvolgorde van het veld **First name**.

### FieldValueCount

**FieldValueCount()** is een functie voor **geheel getal** waarmee het aantal distinctieve waarden in een veld wordt gevonden.

Bij een gedeeltelijke lading kunnen waarden van de gegevens worden verwijderd, wat niet wordt weerspiegeld in het geretourneerde getal. Het geretourneerde getal komt overeen met alle afzonderlijke waarden die in de initiële lading of opeenvolgende gedeeltelijke ladingen zijn geladen.



Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie. Deze beperking is niet van toepassing op de overeenkomstige scriptfunctie.

### Syntaxis:

```
FieldValueCount (field_name)
```

Retourgegevenstypen: geheel getal

### Argumenten:

#### Argumenten

Argument	Beschrijving
field_name	Naam van het veld waarvoor de waarde is vereist. Bijvoorbeeld de kolom in een tabel. Moet worden opgegeven als een tekenreekswaarde. Dit betekent dat de veldnaam tussen enkele aanhalingstekens moet staan.

### Voorbeelden en resultaten:

In de volgende voorbeelden wordt gebruikgemaakt van het veld **First name** uit de tabel **Names**.

#### Voorbeelden en resultaten

Voorbeelden	Resultaten
Voeg de voorbeeldgegevens toe aan uw app en voer deze uit.	De tabel <b>Names</b> is geladen, zoals in de steekproefgegevens.
Diagramfunctie: Voeg in een tabel die de dimensie First name bevat het volgende in als een meting:	
FieldValueCount('First name')	5 aangezien <b>Peter</b> tweemaal voorkomt.
FieldValueCount('Initials')	6 aangezien <b>Initials</b> alleen distinctieve waarden heeft.
- scriptfunctie Gegeven dat de tabel <b>Names</b> is geladen, zoals in de voorbeeldgegevens:	
FieldCount1: Load FieldValueCount('First name') as MyFieldCount1 Resident Names;	MyFieldCount1=5 aangezien 'Peter' tweemaal voorkomt.
FieldCount2: Load FieldValueCount('Initials') as MyInitialsCount1 Resident Names;	MyFieldCount1=6 aangezien 'Initials' alleen distinctieve waarden heeft.

Gegevens die worden gebruikt in voorbeelden:

Names:

```
LOAD * inline [
First name|Last name|Initials|Has cellphone
John|Anderson|JA|Yes
Sue|Brown|SB|Yes
Mark|Carr|MC|No
Peter|Devonshire|PD|No
Jane|Elliot|JE|Yes
Peter|Franc|PF|Yes ] (delimiter is '|');
```

FieldCount1:

```
Load FieldValueCount('First name') as MyFieldCount1
Resident Names;
```

FieldCount2:

```
Load FieldValueCount('Initials') as MyInitialsCount1
Resident Names;
```

### LookUp

**LookUp()** zoekt u een tabel die al is geladen en retourneert de waarde van **field\_name** die overeenkomt met het eerste exemplaar van de waarde **match\_field\_value** in het veld **match\_field\_name**. De tabel kan de huidige tabel zijn of een andere tabel die eerder is geladen.

**Syntaxis:**

```
lookup(field_name, match_field_name, match_field_value [, table_name])
```

**Retourgegevenstypen:** dual

**Argumenten:**

#### Argumenten

Argument	Beschrijving
field_name	Naam van het veld waarvoor de retourwaarde is vereist. De invoerwaarde moet worden aangegeven als een tekenreeks (bijvoorbeeld literal tussen aanhalingstekens).
match_field_name	Naam van het veld waarin <b>match_field_value</b> moet worden opgezocht. De invoerwaarde moet worden aangegeven als een tekenreeks (bijvoorbeeld literal tussen aanhalingstekens).
match_field_value	Waarde die moet worden opgezocht in het veld <b>match_field_name</b> .
table_name	Naam van de tabel waarin de waarde moet worden opgezocht. De invoerwaarde moet worden aangegeven als een tekenreeks (bijvoorbeeld literal tussen aanhalingstekens). Als <b>table_name</b> wordt weggelaten, wordt de huidige tabel aangenomen.



Argumenten zonder aanhalingstekens verwijzen naar de huidige tabel. Als u naar andere tabellen wilt verwijzen, zet u een argument tussen enkele aanhalingstekens.

### Beperkingen:

De volgorde waarin de zoekopdracht wordt uitgevoerd is de laadvolgorde, tenzij de tabel het resultaat is van complexe bewerkingen, zoals joins, waarbij de volgorde niet goed is gedefinieerd. Zowel **field\_name** als **match\_field\_name** moeten velden in dezelfde tabel zijn, aangegeven met **table\_name**.

Als geen waarde wordt gevonden, wordt NULL geretourneerd.

Voorbeeld

### Load-script

Laad de volgende gegevens als een inline load in de editor voor laden van gegevens om het onderstaande voorbeeld te maken.

```
ProductList: Load * Inline [ ProductID|Product|Category|Price 1|AA|1|1 2|BB|1|3 3|CC|2|8 4|DD|3|2 ] (delimiter is '|'); OrderData: Load *, Lookup('Category', 'ProductID', ProductID, 'ProductList') as CategoryID Inline [ InvoiceID|CustomerID|ProductID|Units 1|Astrida|1|8 1|Astrida|2|6 2|Betacab|3|10 3|Divadip|3|5 4|Divadip|4|10 ] (delimiter is '|'); Drop Table ProductList;
```

### Een visualisatie maken

Maak een tabelvisualisatie in een Qlik Sense-werkblad. Voeg de velden **ProductID**, **InvoiceID**, **CustomerID**, **Units** en **CategoryID** toe aan de tabel.

### Resultaat

Resultaattabel

ProductID	InvoiceID	CustomerID	Eenheden:	CategoryID
1	1	Astrida	8	1
2	1	Astrida	6	1
3	2	Betacab	10	2
3	3	Divadip	5	2
4	4	Divadip	10	3

### Uitleg

In de steekproefgegevens wordt de functie **Lookup()** als volgt gebruikt:

```
Lookup('Category', 'ProductID', ProductID, 'ProductList')
```

De tabel **ProductList** wordt als eerste geladen.

De functie **Lookup()** wordt gebruikt voor het maken van de tabel **OrderData**. Hiermee wordt het derde argument opgegeven als **ProductID**. Dit is het veld waarvoor de waarde moet worden opgezocht in het tweede argument '**ProductID**' in de **ProductList**, zoals aangegeven door de enkele aanhalingstekens er omheen.

De functie retourneert de waarde voor '**Category**' (in de tabel **ProductList**), geladen als **CategoryID**.

Met de opdracht **drop** wordt de tabel **ProductList** verwijderd uit het gegevensmodel, omdat deze niet vereist is. Nu blijft de tabel **OrderData** over.



*De functie `Lookup()` is flexibel en kan toegang verkrijgen tot elke eerder geladen tabel. Deze functie is echter traag in vergelijking met de functie `Applymap()`.*

**Zie ook:**

p *ApplyMap* (page 1316)

### NoOfRows - diagramfunctie

**NoOfRows()** retourneert het aantal rijen in het huidige kolomsegment in een tabel. Voor bitmapgrafieken wordt met **NoOfRows()** het aantal rijen in het equivalent van de standaard tabel voor de grafiek geretourneerd.

Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.



*Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.*

**Syntaxis:**

```
NoOfRows ( [TOTAL] )
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argumenten

Argument	Beschrijving
TOTAL	Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.

### Voorbeeld: Diagramuitdrukking waarbij NoOfRows wordt gebruikt

Voorbeeld - diagramuitdrukking

#### Load-script

Laad de volgende gegevens als een inline-load in de editor voor laden van gegevens om de onderstaande voorbeelden van diagramuitdrukkingen te maken.

```
Temp:
LOAD * inline [
Region|SubRegion|RowNo()|NoOfRows()
Africa|Eastern
Africa|Western
Americas|Central
Americas|Northern
Asia|Eastern
Europe|Eastern
Europe|Northern
Europe|Western
Oceania|Australia
] (delimiter is '|');
```

#### Diagramuitdrukking

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Region** en **SubRegion** als dimensies. Voeg `RowNo()`, `NoOfRows()` en `NoOfRows(Total)` toe als metingen.

#### Resultaat

Region	SubRegion	RowNo()	NoOfRows()	NoOfRows (Total)
Africa	Eastern	1	2	9
Africa	Western	2	2	9
Americas	Central	1	2	9
Americas	Northern	2	2	9
Asia	Eastern	1	1	9
Europe	Eastern	1	3	9
Europe	Northern	2	3	9
Europe	Western	3	3	9
Oceania	Australia	1	1	9



### Uitleg

In dit voorbeeld wordt de sorteervolgorde bepaald door de eerste dimensie, Region. Dit heeft als gevolg dat elk kolomsegment bestaat uit een groep regio's met dezelfde waarde, bijvoorbeeld Afrika.

De kolom **RowNo()** toont de rijnummers voor elk kolomsegment, zo zijn er bijvoorbeeld twee rijen voor de regio Afrika. De rijnummering begint vervolgens opnieuw bij 1 voor het volgende kolomsegment, Americas.

De kolom **NoOfRows()** telt het aantal rijen in elk kolomsegment, zo heeft Europa bijvoorbeeld drie rijen in het kolomsegment.

In de kolom **NoOfRows(Total)** worden de dimensies genegeerd vanwege het argument `TOTAL` voor `NoOfRows()` en het aantal rijen in de tabel worden geteld.

Als de tabel is gesorteerd op de tweede dimensie, SubRegion, worden de kolomsegmenten gebaseerd op die dimensie, waardoor de rijnummers wijzigen voor elke SubRegion.

---

### Zie ook:

p *RowNo - diagramfunctie (page 583)*

### Peek

**Peek()** retourneert de waarde van een veld in een tabel voor een rij die al is geladen. Het rijnummer kan worden opgegeven, net als de tabel. Als er geen rijnummer is opgegeven, wordt het laatst geladen record gebruikt.

De functie `peek()` wordt het vaakst gebruikt om relevante grenzen in een eerder geladen tabel te vinden, oftewel de eerste waarde of laatste waarde van een specifiek veld. In de meeste gevallen wordt deze waarde opgeslagen in een variabele voor later gebruik, bijvoorbeeld als een voorwaarde in een `do-while` loop.

### Syntaxis:

```
Peek (  
field_name  
[, row_no[, table_name ] ])
```

**Retourgegevenstypen:** dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
field_name	Naam van het veld waarvoor de retourwaarde is vereist. De invoerwaarde moet worden aangegeven als een tekenreeks (bijvoorbeeld literal tussen aanhalingstekens).

Argument	Beschrijving
row_no	De rij in de tabel die het vereiste veld aangeeft. Kan een uitdrukking zijn, maar moet worden herleid tot een geheel getal. 0 geeft de eerste record aan, 1 de tweede, enzovoort. Negatieve nummers geven de volgorde vanaf het eind van de tabel aan. -1 geeft de laatste gelezen record aan.  Als geen <b>row_no</b> wordt aangegeven, wordt -1 verondersteld.
table_name	Een label voor een tabel zonder dubbele punt aan het eind. Als geen <b>table_name</b> wordt vermeld, wordt de huidige tabel verondersteld. Als de tabelnaam buiten de <b>LOAD</b> -opdracht wordt gebruikt of als de tabelnaam verwijst naar een andere tabel, moet de <b>table_name</b> worden opgenomen.

### Beperkingen:

De functie kan alleen waarden van reeds geladen records retourneren. Dit betekent dat in de eerste record van een tabel een aanroep die -1 als row\_no gebruikt NULL zal retourneren.

Voorbeelden en resultaten:

### Voorbeeld 1

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
EmployeeDates: Load * Inline [ EmployeeCode|StartDate|EndDate 101|02/11/2010|23/06/2012
102|01/11/2011|30/11/2013 103|02/01/2012| 104|02/01/2012|31/03/2012 105|01/04/2012|31/01/2013
106|02/11/2013| ] (delimiter is '|'); First_Last_Employee: Load EmployeeCode, Peek
('EmployeeCode',0,'EmployeeDates') As FirstCode, Peek('EmployeeCode',-1,'EmployeeDates') As
LastCode Resident EmployeeDates;
```

Resultaattabel

Employee code	StartDate	EndDate	FirstCode	LastCode
101	02/11/2010	23/06/2012	101	106
102	01/11/2011	30/11/2013	101	106
103	02/01/2012		101	106
104	02/01/2012	31/03/2012	101	106
105	01/04/2012	31/01/2013	101	106
106	02/11/2013		101	106

FirstCode = 101 omdat `Peek('EmployeeCode',0, 'EmployeeDates')` de eerste waarde retourneert van EmployeeCode in de tabel EmployeeDates.

LastCode = 106 omdat `Peek('EmployeeCode',-1, 'EmployeeDates')` de laatste waarde retourneert van EmployeeCode in de tabel EmployeeDates.

Als de waarde van het argument **row\_no** wordt vervangen, worden de waarden van de andere rijen in de tabel als volgt geretourneerd:

`Peek('EmployeeCode', 2, 'EmployeeDates')` retourneert de derde waarde, 103, in de tabel als de `FirstCode`.

Houd er echter rekening mee dat als in deze voorbeelden de tabel niet wordt opgegeven als het derde argument **table\_name**, de functie verwijst naar de huidige (in dit geval interne) tabel.

### Voorbeeld 2

Als u gegevens wilt bekijken die zich lager in een tabel bevinden, moet u dit in twee stappen doen: eerst laadt u de volledige tabel als een tijdelijke tabel en vervolgens sorteert u deze opnieuw met behulp van **Peek()**.

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
T1: LOAD * inline [ ID|value 1|3 1|4 1|6 3|7 3|8 2|1 2|11 5|2 5|78 5|13 ] (delimiter is '|');
T2: LOAD *, IF(ID=Peek('ID'), Peek('List')&', '&Value,value) AS List RESIDENT T1 ORDER BY ID
ASC; DROP TABLE T1;
```

Create a table in a sheet in your app with **ID**, **List**, and **Value** as the dimensions.

Resultaattabel

ID	Lijst	Value
1	3,4	4
1	3,4,6	6
1	3	3
2	1,11	11
2	1	1
3	7,8	8
3	7	7
5	2,78	78
5	2,78,13	13
5	2	2

De opdracht **IF()** is samengesteld op basis van de tijdelijke tabel T1.

`Peek('ID')` verwijst naar het veld ID in de vorige rij in de huidige tabel T2.

`Peek('List')` verwijst naar het veld List in de vorige rij in de tabel T2, die momenteel wordt samengesteld terwijl de uitdrukking wordt geëvalueerd.

De opdracht wordt als volgt geëvalueerd:

Als de huidige waarde van ID gelijk is aan de vorige waarde van ID, wordt de waarde van Peek('List') aaneengeschakeld met de huidige waarde van Value weggeschreven. Anders wordt alleen de huidige waarde van Value weggeschreven.

Als Peek('List') al een aaneengeschakeld resultaat bevat, wordt het nieuwe resultaat van Peek('List') hier eveneens mee aaneengeschakeld.



*Let op de **Order by**-clausule. Deze geeft aan hoe de tabel wordt geordend (op ID in oplopende volgorde). Zonder deze clausule, wordt bij de functie Peek() gebruikgemaakt van de willekeurige volgorde van de interne tabel, hetgeen tot onvoorspelbare resultaten kan leiden.*

### Voorbeeld 3

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
Amounts: Load Date#(Month,'YYYY-MM') as Month, Amount, Peek(Amount) as AmountMonthBefore  
InLine [Month,Amount 2022-01,2 2022-02,3 2022-03,7 2022-04,9 2022-05,4 2022-06,1];
```

Resultaattabel

Amount	AmountMonthBefore	Month
1	4	2022-06
2	-	2022-01
3	2	2022-02
4	9	2022-05
7	3	2022-03
9	7	2022-04

Het veld AmountMonthBefore bevat het aantal van de vorige maand.

Hier worden de parameters voor row\_no en table\_name weggelaten en worden de standaardwaarden gebruikt. In dit voorbeeld zijn de volgende functieaanroepen gelijk:

- Peek(Amount)
- Peek(Amount,-1)
- Peek(Amount,-1,'Amounts')

Als u -1 gebruikt als row\_no, wordt de waarde van de vorige rij gebruikt. Door deze waarde te vervangen, kunnen waarden van andere rijen in de tabel worden opgehaald:

Peek(Amount,2) retourneert de derde waarde in de tabel: 7.

### Voorbeeld 4

Gegevens moeten correct worden gesorteerd om de correcte resultaten te krijgen, maar helaas is dit niet altijd het geval. Bovendien kan de functie Peek() niet worden gebruikt om te verwijzen naar gegevens die nog niet zijn geladen. Door tijdelijke tabellen te gebruiken en gegevens door meerdere lagen te leiden, kunt u dergelijke problemen vermijden.

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
tmp1Amounts: Load * Inline [Month,Product,Amount 2022-01,B,3 2022-01,A,8 2022-02,B,4 2022-02,A,6 2022-03,B,1 2022-03,A,6 2022-04,A,5 2022-04,B,5 2022-05,B,6 2022-05,A,7 2022-06,A,4 2022-06,B,8]; tmp2Amounts: Load *, If(Product=Peek(Product),Peek(Amount)) as AmountMonthBefore Resident tmp1Amounts Order By Product, Month Asc; Drop Table tmp1Amounts; Amounts: Load *, If(Product=Peek(Product),Peek(Amount)) as AmountMonthAfter Resident tmp2Amounts Order By Product, Month Desc; Drop Table tmp2Amounts;
```

### Uitleg

De initiële tabel wordt gesorteerd op maand, wat betekent dat de functie peek() in de meeste gevallen het aantal voor het verkeerde product retourneert. Daarom moet deze tabel opnieuw worden gestorteerd. Dit doet u door de gegevens door een tweede laag te leiden en een nieuwe tabel tmp2Amounts te maken. Let op de Order by-clausule. De records worden eerst gesorteerd op product, vervolgens op maand in oplopende volgorde.

De functie If() is vereist omdat AmountMonthBefore alleen moet worden berekend als de vorige rij de gegevens bevat voor hetzelfde product, maar dan van de vorige maand. Door het product van de huidige rij te vergelijken met het product van de vorige rij, kan deze voorwaarde worden gevalideerd.

Wanneer de tweede tabel wordt gemaakt, wordt de eerste tabel tmp1Amounts verwijderd met de opdracht Drop Table.

Tot slot gaan de gegevens door een derde laag, maar nu worden de maanden in omgekeerde volgorde gesorteerd. Op deze manier kan AmountMonthAfter ook worden berekend.



*De clausule Order by geeft aan wat de volgorde van de tabel is; zonder deze clausule, wordt bij de functie Peek() gebruikgemaakt van de willekeurige volgorde van de interne tabel, hetgeen tot onvoorspelbare resultaten kan leiden.*

### Resultaat

Resultaattabel

Month	Product	Amount	AmountMonthBefore	AmountMonthAfter
2022-01	A	8	-	6
2022-02	B	3	-	4

## 5 Script- en diagramfuncties

Month	Product	Amount	AmountMonthBefore	AmountMonthAfter
2022-03	A	6	8	6
2022-04	B	4	3	1
2022-05	A	6	6	5
2022-06	B	1	4	5
2022-01	A	5	6	7
2022-02	B	5	1	6
2022-03	A	7	5	4
2022-04	B	6	5	8
2022-05	A	4	7	-
2022-06	B	8	6	-

### Voorbeeld 5

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
T1: Load * inline [ Quarter, value 2003q1, 10000 2003q1, 25000 2003q1, 30000 2003q2, 1250 2003q2, 55000 2003q2, 76200 2003q3, 9240 2003q3, 33150 2003q3, 89450 2003q4, 1000 2003q4, 3000 2003q4, 5000 2004q1, 1000 2004q1, 1250 2004q1, 3000 2004q2, 5000 2004q2, 9240 2004q2, 10000 2004q3, 25000 2004q3, 30000 2004q3, 33150 2004q4, 55000 2004q4, 76200 2004q4, 89450 ]; T2: Load *, rangesum(SumVal,peek('AccSumVal')) as AccSumVal; Load Quarter, sum(Value) as SumVal resident T1 group by Quarter;
```

### Resultaat

Resultaattabel

Kwartaal	SumVal	AccSumVal
2003q1	65000	65000
2003q2	132450	197450
2003q3	131840	329290
2003q4	9000	338290
2004q1	5250	343540
2004q2	24240	367780
2004q3	88150	455930
2004q4	220650	676580

### Uitleg

De LOAD-instructie **Load \*, rangesum(SumVal,peek('AccSumVal')) as AccSumVal** bevat een recursieve aanroep waarbij de vorige waarden worden toegevoegd aan de huidige waarde. Deze bewerking wordt gebruikt om een samenvoeging van waarden in het script te berekenen.

---

Zie ook:

### Previous

**Previous()** retourneert de waarde van de uitdrukking **expr** met gebruik van de gegevens uit de vorige invoerrecord die niet is genegeerd wegens een **where**-clausule. In de eerste record van een interne tabel retourneert de functie NULL.

### Syntaxis:

```
Previous (expr)
```

**Retourgegevenstypen:** dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten. De uitdrukking kan geneste functies <b>previous()</b> bevatten om toegang te krijgen tot eerdere records. Gegevens worden rechtstreeks opgehaald uit de invoerbron. Zo kan ook worden verwezen naar velden die niet zijn geladen in Qlik Sense en dus zelfs als deze niet zijn opgeslagen in de bijbehorende associatieve database.

### Beperkingen:

In de eerste record van een interne tabel retourneert de functie NULL.

### Voorbeeld:

Voer het volgende in uw load-script in

```
sales2013:
Load *, (Sales - Previous(Sales) )as Increase Inline [
Month|Sales
1|12
2|13
3|15
4|17
5|21
6|21
7|22
8|23
```

```
9|32
10|35
11|40
12|41
] (delimiter is '|');
```

Door gebruik te maken van de functie **Previous()** in de opdracht **Load**, kunnen wij de huidige waarde van Sales vergelijken met de voorafgaande waarde en deze gebruiken in een derde veld, Increase.

Resultaattabel

Maand	Sales	Toename
1	12	-
2	13	1
3	15	2
4	17	2
5	21	4
6	21	0
7	22	1
8	23	1
9	32	9
10	35	3
11	40	5
12	41	1

### Top - diagramfunctie

**Top()** evalueert een uitdrukking in de eerste rij (bovenste rij) van een kolomsegment in een tabel. De rij waarvoor deze waarde wordt berekend is afhankelijk van de waarde van **offset**, indien aanwezig. De bovenste rij is de standaardwaarde. Voor andere diagrammen dan tabellen, wordt bij **Top()** de eerste rij van de huidige kolom in het equivalent van de standaard tabel in het diagram geëvalueerd.

#### Syntaxis:

```
Top([TOTAL] expr [ , offset [,count ]])
```

**Retourgegevenstypen:** dual

#### Argumenten:

Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.



Argument	Beschrijving
offset	<p>Als een <b>offset</b> n groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n rijen omlaag verplaatst onder de bovenste rij.</p> <p>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie <b>Top</b> op dezelfde wijze als de functie <b>Bottom</b> met het overeenkomstige positieve getal als beginwaarde.</p>
count	Als een derde parameter <b>count</b> groter dan 1 is aangegeven, retourneert de functie een bereik van <b>count</b> waarden. Een waarde voor elk van de laatste <b>count</b> rijen van het huidige kolomsegment. In deze vorm kan de functie worden gebruikt als argument voor een van de speciale bereikfuncties. <i>Bereikfuncties (page 1324)</i>
TOTAL	Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.



*Een kolomsegment wordt gedefinieerd als een opeenvolgende subset van cellen die dezelfde waarden hebben voor de dimensies in de huidige sorteervolgorde. Interrecord-diagramfuncties worden berekend in het kolomsegment met uitzondering van de dimensie uiterst rechts in de equivalente strakke tabel. Als er slechts één dimensie is in het diagram, of als de kwalificatie TOTAL is opgegeven, wordt de uitdrukking geëvalueerd voor de volledige tabel.*



*Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.*

### Beperkingen:

- Recursieve oproepen retourneren een NULL-waarde.
- Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.

### Voorbeelden en resultaten:

#### Voorbeeld: 1

In de schermafbeelding van de tabel die in dit voorbeeld wordt weergegeven, wordt de tabelvisualisatie gemaakt op basis van de dimensie **Customer** en de metingen `sum(Sales)` en `Top(Sum(Sales))`.

De kolom **Top(Sum(Sales))** retourneert 587 voor alle rijen omdat dit de waarde van de bovenste rij is: **Astrida**.

## 5 Script- en diagramfuncties

In de tabel worden tevens complexere metingen weergegeven: één die is gemaakt op basis van `sum(Sales)+Top(Sum(Sales))` en één met het label **Top offset 3**, die is gemaakt met behulp van de uitdrukking `sum(Sales)+Top(Sum(Sales), 3)` en waarvoor het argument **offset** is ingesteld op 3. Hiermee wordt de waarde **Sum(Sales)** voor de huidige rij toegevoegd aan de waarde van de rij die zich drie rijen onder de bovenste rij bevindt. Met andere woorden: de huidige rij plus de waarde voor **Canutility**.

Voorbeeld 1

Top and Bottom					
Customer	Q	Sum(Sales)	Top(Sum(Sales))	Sum(Sales)+Top(Sum(Sales))	Top offset 3
Totals		2566	587	3153	3249
Astrida		587	587	1174	1270
Betacab		539	587	1126	1222
Canutility		683	587	1270	1366
Divadip		757	587	1344	1440

Voorbeeld: 2

In de schermafbeelding van tabellen die worden weergegeven in dit voorbeeld, zijn meer dimensies toegevoegd aan de visualisaties: en **Month** en **Product**. Voor diagrammen met meer dan één dimensie zijn de resultaten van uitdrukkingen met de functies **Above**, **Below**, **Top** en **Bottom** afhankelijk van de volgorde waarin de kolom met dimensies op Qlik Sense zijn gesorteerd. Qlik Sense evalueert de functies op basis van de kolomsegmenten die het resultaat zijn van de dimensie die als laatste is gesorteerd. De sorteervolgorde van de kolom wordt beheerd in het eigenschappenvenster onder **Sorteren** en is niet per se gelijk aan de volgorde waarin de kolommen in een tabel worden weergegeven.

Eerste tabel voor Voorbeeld 2. De waarde van Top voor de meting First value gebaseerd op Month (Jan).

Customer	Product	Month	Sum(Sales)	First value
			<b>2566</b>	-
Astrida	AA	Jan	46	46
Astrida	AA	Feb	60	46
Astrida	AA	Mar	70	46
Astrida	AA	Apr	13	46
Astrida	AA	May	78	46
Astrida	AA	Jun	20	46
Astrida	AA	Jul	45	46
Astrida	AA	Aug	65	46
Astrida	AA	Sep	78	46
Astrida	AA	Oct	12	46
Astrida	AA	Nov	78	46
Astrida	AA	Dec	22	46

## 5 Script- en diagramfuncties

Tweede tabel voor Voorbeeld 2. De waarde van Top voor de meting First value gebaseerd op Product (AA voor Astrida).

Customer	Product	Month	Sum(Sales)	First value
			2566	-
Astrida	AA	Jan	46	46
Astrida	BB	Jan	46	46
Astrida	AA	Feb	60	60
Astrida	BB	Feb	60	60
Astrida	AA	Mar	70	70
Astrida	BB	Mar	70	70
Astrida	AA	Apr	13	13
Astrida	BB	Apr	13	13

Zie voor meer informatie voorbeeld 2 in de **Above**-functie.

### Voorbeeld 3

Voorbeeld: 3	Resultaat
De functie <b>Top</b> kan worden gebruikt als invoer voor de bereikfuncties. Bijvoorbeeld: RangeAvg (Top(Sum(Sales), 1, 3)).	In de argumenten voor de functie <b>Top()</b> is offset ingesteld op 1 en count ingesteld op 3. De functie retourneert de resultaten van de uitdrukking <b>Sum (Sales)</b> voor de drie rijen te beginnen met de rij onder de onderste rij in het kolomsegment (omdat offset=1) en de twee rijen daaronder (waar er een rij is). Deze drie waarden worden gebruikt als invoer voor de functie RangeAvg(), die het gemiddelde van de waarden in het opgegeven bereik van cijfers als resultaat geeft.  Een tabel met <b>Customer</b> als dimensie levert de volgende resultaten op voor de uitdrukking RangeAvg().
	Astrida 603
	Betacab 603
	Canutility 603
	Divadip: 603

Monthnames:

```
LOAD *, Dual(MonthText,MonthNumber) as Month INLINE [
MonthText, MonthNumber
Jan, 1
Feb, 2
Mar, 3
```

```
Apr, 4  
May, 5  
Jun, 6  
Jul, 7  
Aug, 8  
Sep, 9  
Oct, 10  
Nov, 11  
Dec, 12  
];
```

```
sales2013:  
Crosstable (MonthText, Sales) LOAD * inline [  
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec  
Astrida|46|60|70|13|78|20|45|65|78|12|78|22  
Betacab|65|56|22|79|12|56|45|24|32|78|55|15  
Canutility|77|68|34|91|24|68|57|36|44|90|67|27  
Divadip|57|36|44|90|67|27|57|68|47|90|80|94  
] (delimiter is '|');
```

---

### Zie ook:

[p Bottom - diagramfunctie \(page 1277\)](#)  
[p Above - diagramfunctie \(page 1268\)](#)  
[p Sum - diagramfunctie \(page 346\)](#)  
[p RangeAvg \(page 1327\)](#)  
[p Bereikfuncties \(page 1324\)](#)

## SecondaryDimensionality - diagramfunctie

**SecondaryDimensionality()** retourneert het aantal rijen in een draaitabel met een niet-geaggregeerde inhoud, dat wil zeggen kolommen die geen subtotaal of samengevouwen aggregaten bevatten. Deze functie is het equivalent van de functie **dimensionality()** voor horizontale dimensies in draaitabellen.

### Syntaxis:

```
SecondaryDimensionality( )
```

**Retourgegevenstypen:** geheel getal

### Beperkingen:

- Tenzij deze wordt gebruikt in draaitabellen, retourneert de functie **SecondaryDimensionality** altijd de waarde 0.
- Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.

## After - diagramfunctie

retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de kolom voor de huidige kolom in een rijsegment in de draaitabel.

### Syntaxis:

```
after ([TOTAL] expr [, offset [, count ]])
```



*Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.*



*Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.*

### Argumenten:

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
offset	Als een <b>offset</b> n groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n rijen verder naar rechts verplaatst, geteld vanaf de huidige rij.  Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de huidige rij.  Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie <b>After</b> op dezelfde wijze als de functie <b>Before</b> met het overeenkomstige positieve getal als beginwaarde.
count	Als een derde parameter <b>count</b> groter dan 1 wordt opgegeven, retourneert de functie een bereik van waarden. Eén waarde voor elke tabelrij, met een maximum van <b>count</b> , naar rechts geteld vanaf de oorspronkelijke cel.
TOTAL	Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.

Op de laatste kolom van een rijsegment wordt een NULL-waarde geretourneerd, omdat er geen kolom na deze kolom staat.

Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onder.

### Voorbeeld:

```
after( sum( Sales ))
after( sum( Sales ), 2 )
after( total sum( Sales ))
```

rangeavg (after(sum(x),1,3)) retourneert een gemiddelde van de drie resultaten van de functie **sum(x)**, berekend aan de hand van de drie kolommen direct rechts van de huidige kolom.

### Before - diagramfunctie

**Before()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de kolom voor de huidige kolom in een rijsegment in de draaitabel.

### Syntaxis:

```
before ([TOTAL] expr [, offset [, count]])
```



*Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.*



*Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukkingkolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.*

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
offset	Als een <b>offset</b> n groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n rijen verder naar links verplaatst, geteld vanaf de huidige rij.  Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de huidige rij.  Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie <b>Before</b> op dezelfde wijze als de functie <b>After</b> met het overeenkomstige positieve getal als beginwaarde.
count	Als een derde parameter <b>count</b> groter dan 1 wordt opgegeven, retourneert de functie een bereik van waarden. Eén waarde voor elke tabelrij met een maximum van <b>count</b> , naar links geteld vanaf de oorspronkelijke cel.
TOTAL	Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.

Op de eerste kolom van een rijsegment wordt een NULL-waarde geretourneerd, daar er geen kolom voor deze kolom staat.

Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onder.

### Voorbeelden:

```
before( sum( sales ) )
before( sum( sales ), 2 )
before( total sum( sales ) )
```

rangeavg (before(sum(x),1,3)) retourneert een gemiddelde van de drie resultaten van de functie **sum(x)**, berekend aan de hand van de drie kolommen direct links van de huidige kolom.

## First - diagramfunctie

**First()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de eerste kolom van het huidige rijsegment in de draaitabel. Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.



*Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.*

### Syntaxis:

```
first([TOTAL] expr [, offset [, count]])
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
expression	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
offset	Als een <b>offset</b> n groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n rijen verder naar rechts verplaatst, geteld vanaf de huidige rij.  Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de huidige rij.  Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie <b>First</b> op dezelfde wijze als de functie <b>Last</b> met het overeenkomstige positieve getal als beginwaarde.

Argument	Beschrijving
count	Als een derde parameter <b>count</b> groter dan 1 wordt opgegeven, retourneert de functie een bereik van waarden. Eén waarde voor elke tabelrij, met een maximum van <b>count</b> , naar rechts geteld vanaf de oorspronkelijke cel.
TOTAL	Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.

Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onder.

### Voorbeelden:

```
first( sum( Sales ))
first( sum( Sales ), 2 )
first( total sum( Sales )
rangeavg ( first( sum( x ), 1, 5 ) )
```

retourneert een gemiddelde van de resultaten van de functie **sum (x)**, berekend aan de hand van de vijf meest linkse kolommen van het huidige rijsegment.

## Last - diagramfunctie

**Last()** retourneert de waarde van een uitdrukking berekend aan de hand van de dimensiewaarden van een draaitabel die staan in de laatste kolom van het huidige rijsegment in de draaitabel. Deze functie retourneert NULL in alle andere diagramtypen dan een draaitabel.



*Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukkingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.*

### Syntaxis:

```
last ([TOTAL] expr [, offset [, count]])
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.



Argument	Beschrijving
offset	<p>Als een <b>offset</b> n groter dan 1 wordt opgegeven, wordt de evaluatie van de uitdrukking n rijen verder naar links verplaatst, geteld vanaf de huidige rij.</p> <p>Als de beginwaarde 0 wordt opgegeven, wordt de uitdrukking geëvalueerd op de huidige rij.</p> <p>Als een negatief getal voor de beginwaarde wordt opgegeven, werkt de functie <b>First</b> op dezelfde wijze als de functie <b>Last</b> met het overeenkomstige positieve getal als beginwaarde.</p>
count	Als een derde parameter <b>count</b> groter dan 1 wordt opgegeven, retourneert de functie een bereik van waarden. Eén waarde voor elke tabelrij met een maximum van <b>count</b> , naar links geteld vanaf de oorspronkelijke cel.
TOTAL	Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.

Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onder.

### Voorbeeld:

```
last( sum( Sales ) )
last( sum( Sales ), 2 )
last( total sum( Sales ) )
```

rangeavg (last(sum(x),1,5)) retourneert een gemiddelde van de resultaten van de functie **sum(x)**, berekend aan de hand van de vijf meest rechtse kolommen van het huidige rijsegment.

## ColumnNo - diagramfunctie

**ColumnNo()** retourneert het nummer van de huidige kolom in het huidige rijsegment in een draaitabel. De eerste kolom is nummer 1.

### Syntaxis:

```
ColumnNo ( [total] )
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
TOTAL	Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.

Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onder.



*Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.*

### Voorbeeld:

```
if( ColumnNo( )=1, 0, sum( Sales ) / before( sum( Sales )))
```

## NoOfColumns - diagramfunctie

**NoOfColumns()** retourneert het aantal kolommen in het huidige rijsegment in een draaitabel.



*Het is niet toegestaan te sorteren op y-waarden in diagrammen of op uitdrukingskolommen in tabellen als deze diagramfunctie wordt gebruikt in een van de uitdrukkingen van het diagram. Deze sorteeropties worden daarom automatisch uitgeschakeld. Als u deze diagramfunctie in een visualisatie of tabel gebruikt, wordt het sorteren van de visualisatie hersteld naar de gesorteerde invoer van deze functie.*

### Syntaxis:

```
NoOfColumns ([total])
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
TOTAL	Als de tabel eendimensionaal is of als de kwalificatie <b>TOTAL</b> wordt gebruikt als argument, is het huidige kolomsegment altijd gelijk aan de hele kolom.

Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden. De sorteervolgorde tussen velden voor horizontale dimensies in draaitabellen wordt simpelweg gedefinieerd door de volgorde van de dimensies van boven naar onder.

### Voorbeeld:

```
if( ColumnNo( )=NoofColumns( ), 0, after( sum( Sales )))
```

## 5.17 Logische functies

In dit gedeelte worden functies beschreven die logische bewerkingen behandelen. Alle functies kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

### IsNum

Retourneert -1 (True) als de uitdrukking kan worden geïnterpreteerd als een getal, anders 0 (False).

```
IsNum( expr )
```

### IsText

Retourneert -1 (True) als de uitdrukking een tekstrepresentatie heeft, anders 0 (False).

```
IsText( expr )
```



Zowel **IsNum** als **IsText** retourneert 0 als de uitdrukking NULL is.

### Voorbeeld:

In het volgende voorbeeld wordt een inline-tabel met gemengde tekst- en numerieke waarden galden en worden twee velden toegevoegd om te controleren of de waarde een numerieke waarde respectievelijk een tekstwaarde is.

```
Load *, IsNum(Value), IsText(Value)
Inline [
Value
23
Green
Blue
12
33Red];
```

De resulterende tabel ziet er als volgt uit:

Resulting table

Value	IsNum(Value)	IsText(Value)
23	-1	0
Green	0	-1
Blue	0	-1
12	-1	0
33Red	0	-1

### 5.18 Toewijzingsfuncties

In dit hoofdstuk komen functies voor het afhandelen van toewijzingstabellen aan de orde. Een toewijzingstabel kan worden gebruikt om veldwaarden en veldnamen tijdens de uitvoering van een script te vervangen.

Toewijzingsfuncties kunnen alleen worden gebruikt in het script voor het laden van gegevens.

#### Overzicht toewijzingsfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

##### ApplyMap

De scriptfunctie **ApplyMap** wordt gebruikt voor het toewijzen van de uitvoer van een uitdrukking aan een eerdere geladen toewijzingstabel.

```
ApplyMap ('mapname', expr [ , defaultexpr ] )
```

##### MapSubstring

De scriptfunctie **MapSubstring** wordt gebruikt voor de toewijzing van onderdelen van een uitdrukking aan een eerder geladen toewijzingstabel. De toewijzing is hoofdlettergevoelig en niet-iteratief en subtekenreeksen worden van links naar rechts toegewezen.

```
MapSubstring ('mapname', expr)
```

#### ApplyMap

De scriptfunctie **ApplyMap** wordt gebruikt voor het toewijzen van de uitvoer van een uitdrukking aan een eerdere geladen toewijzingstabel.


##### Syntaxis:

```
ApplyMap('map_name', expression [ , default_mapping ] )
```

**Retourgegevenstypen:** dual

**Argumenten:**

### Argumenten

Argument	Beschrijving
map_name	<p>De naam van een toewijzingstabel die eerder is gemaakt met de opdracht <b>mapping load</b> of <b>mapping select</b>. De naam moet tussen enkele, rechte aanhalingstekens staan.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>Als u deze functie gebruikt voor een variabele die is uitgebreid met een macro en verwijst naar een toewijzingstabel die niet bestaat, geeft de functie een fout weer en wordt er geen veld gemaakt.</i> </div>
expression	De uitdrukking waarvan het resultaat moet worden toegewezen.
default_mapping	Als de uitdrukking is opgegeven, wordt deze waarde gebruikt als standaardwaarde als de toewijzingstabel geen overeenkomstige waarde bevat voor expression. Als de uitdrukking niet is opgegeven, wordt de waarde van expression ongewijzigd geretourneerd.



*Het uitvoerveld van ApplyMap mag niet dezelfde naam hebben als een van de invoervelden. Dit kan tot onverwachte resultaten leiden. Voorbeeld niet te gebruiken: `ApplyMap ('Toewijzen', A) als A.`*

**Voorbeeld:**

In dit voorbeeld laden wij een lijst met verkopers waarbij een landcode het land aangeeft waar zijn wonen. Wij gebruiken een tabel waarin een landcode wordt toegewezen aan een land ter vervanging van de landcode door de landnaam. Er zijn slechts drie landen gedefinieerd in de toewijzingstabel, andere landcode worden toegewezen aan 'Rest of the world'.

```
// Load mapping table of country codes:
map1:
mapping LOAD *
Inline [
CCode, Country
Sw, Sweden
Dk, Denmark
No, Norway
] ;

// Load list of salesmen, mapping country code to country
// If the country code is not in the mapping table, put Rest of the world
Salespersons:
LOAD *,
ApplyMap('map1', CCode,'Rest of the world') As Country
Inline [
```

```
CCode, Salesperson  
Sw, John  
Sw, Mary  
Sw, Per  
Dk, Preben  
Dk, Olle  
No, Ole  
Sf, Risttu  
] ;
```

```
// We don't need the CCode anymore
```

```
Drop Field 'CCode';
```

De resulterende tabel (Verkopers) ziet er als volgt uit:

Resulting table

Salesperson	Country
John	Sweden
Mary	Sweden
Per	Sweden
Preben	Denmark
Olle	Denmark
Ole	Norway
Risttu	Rest of the world

### MapSubstring

De scriptfunctie **MapSubstring** wordt gebruikt voor de toewijzing van onderdelen van een uitdrukking aan een eerder geladen toewijzingstabel. De toewijzing is hoofdlettergevoelig en niet-iteratief en subtekenreeksen worden van links naar rechts toegewezen.


#### Syntaxis:

```
MapSubstring('map_name', expression)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

Argument	Beschrijving
map_name	<p>De naam van een toewijzingstabel die eerder is gelezen in een opdracht <b>mapping load</b> of <b>mapping select</b>. De naam moet tussen enkele, rechte aanhalingstekens staan.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>Als u deze functie gebruikt voor een variabele die is uitgebreid met een macro en verwijst naar een toewijzingstabel die niet bestaat, geeft de functie een fout weer en wordt er geen veld gemaakt.</i> </div>
expression	De uitdrukking waarvan het resultaat door subtekenreeksen moet worden toegewezen.

**Voorbeeld:**

In dit voorbeeld laden wij een lijst met productmodellen. Elk model heeft een reeks van kenmerken die worden beschreven via een samengestelde code. Door de toewijzingstabel met MapSubstring te gebruiken, kunnen wij de kenmerkcodes uitbreiden tot een beschrijving.

```
map2:
mapping LOAD *
Inline [
AttCode, Attribute
R, Red
Y, Yellow
B, Blue
C, Cotton
P, Polyester
S, Small
M, Medium
L, Large
] ;

Productmodels:
LOAD *,
MapSubString('map2', AttCode) as Description
Inline [
Model, AttCode
Twixie, R C S
Boomer, B P L
Raven, Y P M
Seedling, R C L
SeedlingPlus, R C L with hood
Younger, B C with patch
Multistripe, R Y B C S/M/L
] ;
```

```
// We don't need the AttCode anymore  
Drop Field 'AttCode';
```

De resulterende tabel ziet er als volgt uit:

Resulting table

Model	Description
Twixie	Red Cotton Small
Boomer	Blue Polyester Large
Raven	Yellow Polyester Medium
Seedling	Red Cotton Large
SeedlingPlus	Red Cotton Large with hood
Younger	Blue Cotton with patch
MultiStripe	Red Yellow Blue Cotton Small/Medium/Large

### 5.19 Rekenkundige functies

In dit hoofdstuk komen functies voor rekenkundige constanten en Booleaanse waarden aan de orde. Deze functies hebben geen parameters, maar de haakjes zijn wel verplicht.

Alle functies kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

#### **e**

De functie retourneert het grondtal van de natuurlijke logaritmen, **e** (2,71828...).

```
e ( )
```

#### **false**

De functie retourneert een duale waarde met de tekstwaarde 'False' en numerieke waarde 0, die als logische waarde Onwaar in uitdrukkingen kan worden gebruikt.

```
false ( )
```

#### **pi**

De functie retourneert de waarde van  $\pi$  (3.14159...).

```
pi ( )
```

#### **rand**

De functie retourneert een willekeurig getal tussen 0 en 1. Hiermee kunnen steekproefgegevens worden gemaakt.

```
rand ( )
```



### Voorbeeld:

In dit voorbeeldscript wordt een tabel van 1000 records gemaakt met willekeurig geselecteerde tekens in hoofdletters, oftewel tekens in het bereik van 65 tot en met 91 (65+26).

```
Load
  Chr( Floor(rand() * 26) + 65) as UCaseChar,
  RecNo() as ID
Autogenerate 1000;
```

### true

De functie retourneert een duale waarde met de tekstwaarde 'True' en numerieke waarde -1, die als logische waarde Waar in uitdrukkingen kan worden gebruikt.

```
true ( )
```

## 5.20 NULL-functies

In dit hoofdstuk komen functies voor retournerende of detecterende NULL-waarden aan de orde.

Alle functies kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

### Overzicht NULL-functies

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### EmptyIsNull

De functie **EmptyIsNull** zet lege tekenreeksen om in NULL. Om die reden wordt NULL geretourneerd als de parameter een lege tekenreeks is, anders wordt de parameter geretourneerd.

```
EmptyIsNull (expr )
```

#### IsNull

De functie **IsNull** test of de waarde van een uitdrukking NULL is. Als dit het geval is, wordt -1 (True) geretourneerd, anders 0 (False).

```
IsNull (expr )
```

#### Null

De functie **Null** retourneert een NULL-waarde.

```
NULL ( )
```

## EmptyIsNull

De functie **EmptyIsNull** zet lege tekenreeksen om in NULL. Om die reden wordt NULL geretourneerd als de parameter een lege tekenreeks is, anders wordt de parameter geretourneerd.

### Syntaxis:

```
EmptyIsNull (exp )
```

Voorbeelden en resultaten:

Scriptvoorbeelden	
Voorbeeld	Resultaat
EmptyIsNull(AdditionalComments)	Deze uitdrukking wordt als null geretourneerd voor lege tekenreekswaarden van het veld <i>AdditionalComments</i> in plaats van lege tekenreeksen. Niet-lege tekenreeksen en getallen worden geretourneerd.
EmptyIsNull(PurgeChar(PhoneNumber, ' -()'))	Deze uitdrukking zorgt ervoor dat alle streepjes, spaties en haakjes uit het veld <i>PhoneNumber</i> worden verwijderd. Als alle tekens zijn verwijderd, retourneert de functie EmptyIsNull de lege tekenreeks als null; een leeg veld voor telefoonnummer is hetzelfde als geen telefoonnummer.

## IsNull

De functie **IsNull** test of de waarde van een uitdrukking NULL is. Als dit het geval is, wordt -1 (True) geretourneerd, anders 0 (False).

### Syntaxis:

```
IsNull (expr )
```



Een tekenreeks met een lengte van nul wordt niet beschouwd als NULL en **IsNull** retourneert hiervoor False.

### Voorbeeld: Load-script voor gegevens

In dit voorbeeld wordt een inline-tabel geladen met vier rijen, waarbij de eerste drie rijen niets, - of 'NULL' in de kolom Value bevatten. Wij converteren deze waarde naar echte NULL-waarderepresentaties met de middelste voorafgaande **LOAD** via de functie **Null**.

Met de eerste voorafgaande **LOAD** wordt een veld toegevoegd dat controleert of de waarde NULL is, met behulp van de functie **IsNull**.

```
NullsDetectedAndConverted:
```

```
LOAD *,
```

```
If(IsNull(ValueNullConv), 'T', 'F') as IsItNull;
```

```
LOAD *,  
If(len(trim(Value))= 0 or Value='NULL' or Value='-', Null(), value ) as valueNullConv;
```

```
LOAD * Inline  
[ID, Value  
0,  
1,NULL  
2,-  
3,value];
```

Dit is de resulterende tabel. In de kolom ValueNullConv worden de NULL-waarden vertegenwoordigd door -.

Resulting table

ID	Value	ValueNullConv	IsItNull
0		-	T
1	NULL	-	T
2	-	-	T
3	Value	Value	F

## NULL

De functie **Null** retourneert een NULL-waarde.

### Syntaxis:

```
Null ( )
```

### Voorbeeld: Load-script voor gegevens

In dit voorbeeld wordt een inline-tabel geladen met vier rijen, waarbij de eerste drie rijen niets, - of 'NULL' in de kolom Value bevatten. Wij willen deze waarden converteren naar echte NULL-waarderepresentaties.

De middelste voorafgaande **LOAD** voert de conversie uit met de functie **Null**.

Met de eerste voorafgaande **LOAD** wordt een veld toegevoegd dat controleert of de waarde NULL is, simpelweg voor illustratiedoeleinden in dit voorbeeld.

NullsDetectedAndConverted:

```
LOAD *,  
If(IsNull(ValueNullConv), 'T', 'F') as IsItNull;
```

```
LOAD *,  
If(len(trim(Value))= 0 or Value='NULL' or Value='-', Null(), value ) as valueNullConv;
```

```
LOAD * Inline  
[ID, Value  
0,  
1,NULL
```

2, -  
3,value];

Dit is de resulterende tabel. In de kolom ValueNullConv worden de NULL-waarden vertegenwoordigd door -.

Resulting table

ID	Value	ValueNullConv	IsItNull
0		-	T
1	NULL	-	T
2	-	-	T
3	Value	Value	F

### 5.21 Bereikfuncties

De bereikfuncties zijn functies die een reeks van waarden nemen en een enkele waarde als resultaat opleveren. Alle bereikfuncties kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

Zo kan een bereikfunctie in een visualisatie bijvoorbeeld een enkele waarde berekenen op basis van een inter-recordreeks. In het script voor het berekenen van waarden kan een bereikfunctie een enkele waarde berekenen op basis van een reeks van waarden in een interne tabel.



*Bereikfuncties vervangen de volgende algemene numerieke functies: **numsum**, **numavg**, **numcount**, **nummin** en **nummax**. Deze moeten nu als verouderd worden beschouwd.*

#### Elementaire bereikfuncties

RangeMax

**RangeMax()** retourneert de hoogste numerieke waarden die binnen de uitdrukking of het veld worden gevonden.

```
RangeMax (first_expr[, Expression])
```

RangeMaxString

**RangeMaxString()** retourneert de laatste waarde die in de gesorteerde tekst van de uitdrukking of het veld wordt gevonden.

```
RangeMaxString (first_expr[, Expression])
```

RangeMin

**RangeMin()** retourneert de laagste numerieke waarden die binnen de uitdrukking of het veld worden gevonden.

```
RangeMin (first_expr[, Expression])
```

RangeMinString

**RangeMinString()** retourneert de eerste waarde die in de gesorteerde tekst van de uitdrukking of het veld wordt gevonden.

```
RangeMinString (first_expr[, Expression])
```

RangeMode

**RangeMode()** retourneert de meest voorkomende waarde (moduswaarde) in de uitdrukking of het veld.

```
RangeMode (first_expr[, Expression])
```

RangeOnly

**RangeOnly()** is een duale functie die een waarde retourneert als de uitdrukking één enkele unieke waarde oplevert. Als dat niet het geval is, retourneert de functie de waarde **NULL**.

```
RangeOnly (first_expr[, Expression])
```

RangeSum

**RangeSum()** retourneert de som van een reeks waarden. Alle niet-numerieke waarden worden behandeld als 0.

```
RangeSum (first_expr[, Expression])
```

### Bereikfuncties voor tellers

RangeCount

**RangeCount()** retourneert het aantal waarden, zowel tekst- als numerieke waarden, in de uitdrukking of het veld.

```
RangeCount (first_expr[, Expression])
```

RangeMissingCount

**RangeMissingCount()** retourneert het aantal niet-numerieke waarden (inclusief NULL) in de uitdrukking of het veld.

```
RangeMissingCount (first_expr[, Expression])
```

RangeNullCount

**RangeNullCount()** retourneert het aantal NULL-waarden in de uitdrukking of het veld.

```
RangeNullCount (first_expr[, Expression])
```

RangeNumericCount

**RangeNumericCount()** retourneert het aantal numerieke waarden in de uitdrukking of het veld.

```
RangeNumericCount (first_expr[, Expression])
```

RangeTextCount

**RangeTextCount()** retourneert het aantal tekstwaarden in de uitdrukking of het veld.

```
RangeTextCount (first_expr[, Expression])
```

### Statistische bereikfuncties

RangeAvg

**RangeAvg()** retourneert het gemiddelde van een bereik. De invoer van de functie is een waardebereik of een uitdrukking.

```
RangeAvg (first_expr[, Expression])
```

RangeCorrel

**RangeCorrel()** retourneert de correlatiecoëfficiënt voor twee gegevensverzamelingen. De correlatiecoëfficiënt is een maat voor de relatie tussen de gegevensverzamelingen.

```
RangeCorrel (x_values , y_values[, Expression])
```

RangeFractile

**RangeFractile()** retourneert de waarde die correspondeert met het n-de **fractile** (kwantiel) van een reeks getallen.

```
RangeFractile (fractile, first_expr[, Expression])
```

RangeKurtosis

**RangeKurtosis()** retourneert de waarde die correspondeert met de piekvorm van een reeks getallen.

```
RangeKurtosis (first_expr[, Expression])
```

RangeSkew

**RangeSkew()** retourneert de waarde die correspondeert met de scheefheid van een reeks getallen.

```
RangeSkew (first_expr[, Expression])
```

RangeStdev

**RangeStdev()** retourneert de standaardafwijking van een reeks getallen.

```
RangeStdev (expr1[, Expression])
```

### Financiële bereikfuncties

RangeIRR

**RangeIRR()** retourneert het interne opbrengstpercentage voor een reeks van cashflows die worden weergegeven door de invoerwaarden.

```
RangeIRR (value[, value][, Expression])
```

RangeNPV

**RangeNPV()** retourneert de huidige nettowaarde van een investering op basis van een kortingspercentage en een reeks toekomstige periodieke betalingen (negatieve waarden) en inkomsten (positieve waarden). Het resultaat heeft de standaardgetalnotatie van **money**.

```
RangeNPV (discount_rate, value[, value][, Expression])
```

### RangeXIRR

**RangeXIRR()** retourneert het interne opbrengstpercentage voor een schema cashflows die niet noodzakelijk periodiek is. Gebruik voor de berekening van het interne opbrengstpercentage voor een reeks periodieke cashflows de functie **RangeIRR**.

```
RangeXIRR (values, dates[, Expression])
```

### RangeXNPV

**RangeXNPV()** retourneert de huidige nettowaarde voor een schema cashflows die niet noodzakelijk periodiek is. Het resultaat heeft de standaardgetalnotatie voor geld. Met de functie **RangeNPV** berekent u de huidige nettowaarde voor een reeks periodieke cashflows.

```
RangeXNPV (discount_rate, values, dates[, Expression])
```

---

### Zie ook:

p *Interrecord-functies (page 1265)*

## RangeAvg

**RangeAvg()** retourneert het gemiddelde van een bereik. De invoer van de functie is een waardebereik of een uitdrukking.

### Syntaxis:

```
RangeAvg (first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Beperkingen:

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

### Voorbeelden en resultaten:

Scriptvoorbeelden

Voorbeelden	Resultaten
RangeAvg (1,2,4)	retourneert 2,33333333
RangeAvg (1, 'xyz')	retourneert 1
RangeAvg (null( ), 'abc')	Retourneert NULL

### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
RangeTab3:
LOAD recno() as RangeID, RangeAvg(Field1,Field2,Field3) as MyRangeAvg INLINE [
Field1, Field2, Field3
10,5,6
2,3,7
8,2,8
18,11,9
5,5,9
9,4,2
];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeAvg voor elk van de records in de tabel.

Resultaattabel

RangeID	MyRangeAvg
1	7
2	4
3	6
4	12.666
5	6.333
6	5

### Voorbeeld met uitdrukking:

```
RangeAvg (Above(MyField),0,3))
```

Retourneert een glijdend gemiddelde van het resultaat van het bereik van drie waarden van **MyField** berekend op de huidige rij en de twee rijen boven de huidige rij. Doordat het derde argument wordt opgegeven als 3, retourneert de functie **Above()** drie waarden, met voldoende rijen erboven, die als invoer worden gebruikt voor de functie **RangeAvg()**.

Gegevens die worden gebruikt in voorbeelden:





Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.

#### Voorbeeldgegevens

MyField	RangeAvg (Above (MyField,0,3))	Comments
10	10	Omdat dit de bovenste rij is, bestaat het bereik uit slechts één waarde.
2	6	Er bevindt zich slechts één rij boven deze rij, dus is het bereik: 10,2.
8	6.6666666667	Het equivalent van RangeAvg(10,2,8)
18	9.3333333333	-
5	10.3333333333	-
9	10.6666666667	-

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

#### Zie ook:

p Avg - diagramfunctie (page 396)  
p Count - diagramfunctie (page 351)

## RangeCorrel

**RangeCorrel()** retourneert de correlatiecoëfficiënt voor twee gegevensverzamelingen. De correlatiecoëfficiënt is een maat voor de relatie tussen de gegevensverzamelingen.

#### Syntaxis:

```
RangeCorrel (x_value , y_value[, Expression])
```

#### Retourgegevenstypen: numeriek

Gegevensreeksen moeten worden ingevoerd als (x,y) paren. Als u bijvoorbeeld twee reeksen gegevens, array 1 en array 2, wilt evalueren, waarbij array 1 = 2,6,9 en array 2 = 3,8,4, schrijft u `RangeCorrel(2,3,6,8,9,4)`, waarmee 0.269 wordt geretourneerd.

### Argumenten:

Argumenten

Argument	Beschrijving
x-value, y-value	Elke waarde vertegenwoordigt een enkele waarde of een bereik van waarden die wordt geretourneerd door een interrecordfunctie met een derde optionele parameter. Elke waarde of bereik van waarden moet overeenkomen met een <b>x-value</b> of een bereik van <b>y-values</b> .
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Beperkingen:

De functie vereist ten minste twee coördinaatparen.

Tekstwaarden, NULL-waarden en ontbrekende waarden geven NULL retour.

### Voorbeelden en resultaten:

Voorbeelden van functies

Voorbeelden	Resultaten
RangeCorrel (2,3,6,8,9,4,8,5)	Retourneert 0,2492. Deze functie kan in het script worden geladen of toegevoegd aan een visualisatie in de uitdrukkingeditor.

### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
RangeList:
Load * Inline [
ID1|x1|y1|x2|y2|x3|y3|x4|y4|x5|y5|x6|y6
01|46|60|70|13|78|20|45|65|78|12|78|22
02|65|56|22|79|12|56|45|24|32|78|55|15
03|77|68|34|91|24|68|57|36|44|90|67|27
04|57|36|44|90|67|27|57|68|47|90|80|94
](delimiter is '|');
```

```
XY:
LOAD recno() as RangeID, * Inline [
X|Y
2|3
6|8
9|4
8|5
](delimiter is '|');
```

In een tabel met ID1 als een dimensie en de meting: RangeCorrel(x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6)), de **RangeCorrel()** functie zoekt de waarde van **Correl** over het bereik van zes x,y paar, voor elk van de ID1 values.

Resultaattabel

ID1	MyRangeCorrel
01	-0.9517
02	-0.5209
03	-0.5209
04	-0.1599

### Voorbeeld:

```
XY:  
LOAD recno() as RangeID, * Inline [  
X|Y  
2|3  
6|8  
9|4  
8|5  
](delimiter is '|');
```

In een tabel met RangelD als een dimensie en de meting: RangeCorrel(Below(X,0,4,BelowY,0,4)), de **RangeCorrel()** functie gebruikt de resultaten van de **Below()** functies, die, vanwege het feit dat het derde argument (count) op 4 is ingesteld, een bereik van vier x-y waarden produceren op basis van de geladen tabel XY.

Resultaattabel

RangelD	MyRangeCorrel2
01	0.2492
02	-0.9959
03	-1.0000
04	-

De waarde voor RangelD 01 staat gelijk aan het handmatig invoeren van RangeCorrel(2,3,6,8,9,4,8,5). Voor de andere waarden van RangelD, zijn de series die worden geproduceerd door de Below() functie: (6,8,9,4,8,5), (9,4,8,5) en (8,5); de laatste hiervan produceert een null-resultaat.

### Zie ook:

p *Correl - diagramfunctie* (page 399)

## RangeCount

**RangeCount()** retourneert het aantal waarden, zowel tekst- als numerieke waarden, in de uitdrukking of het veld.

### Syntaxis:

```
RangeCount (first_expr[, Expression])
```

**Retourgegevenstypen:** geheel getal

### Argumenten:

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden geteld.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden geteld.

### Beperkingen:

NULL-waarden worden niet meegeteld.

### Voorbeelden en resultaten:

Voorbeelden van functies

Voorbeelden	Resultaten
RangeCount (1,2,4)	retourneert 3
RangeCount (2,'xyz')	retourneert 2
RangeCount (null( ))	retourneert 0
RangeCount (2,'xyz', null())	retourneert 2

### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
RangeTab3:
LOAD recno() as RangeID, RangeCount(Field1,Field2,Field3) as MyRangeCount INLINE [
Field1, Field2, Field3
10,5,6
2,3,7
8,2,8
18,11,9
```

```
5,5,9  
9,4,2  
];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeCount voor elk van de records in de tabel.

RangeID	MyRangeCount
1	3
2	3
3	3
4	3
5	3
6	3

Voorbeeld met uitdrukking:

```
RangeCount (Above(MyField,1,3))
```

Retourneert het aantal waarden dat is opgenomen in de drie resultaten van **MyField**. Door het eerste argument van de functie **Above()** op te geven als 1 en het tweede argument als 3, worden de waarden geretourneerd van de eerste drie velden boven de huidige rij, waarbij er voldoende rijen zijn om als invoer te worden gebruikt voor de functie **RangeCount()**.

Gegevens die worden gebruikt in voorbeelden:

MyField	RangeCount(Above(MyField,1,3))
10	0
2	1
8	2
18	3
5	3
9	3

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:  
LOAD * INLINE [  
MyField  
10  
2  
8  
18  
5  
9
```

] ;

**Zie ook:**p *Count - diagramfunctie (page 351)*

## RangeFractile

**RangeFractile()** retourneert de waarde die correspondeert met het n-de **fractile** (kwantiel) van een reeks getallen.



*RangeFractile() gebruikt lineaire interpolatie tussen dichtstbijzijnde rangnummers bij het berekenen van de fractiel.*

**Syntaxis:**

```
RangeFractile(fractile, first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

Argument	Beschrijving
fractile	Een getal tussen 0 en 1 die correspondeert met de fractiel (kwantiel uitgedrukt als een fractie) die moet worden berekend.
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

**Voorbeelden en resultaten:**

Voorbeelden van functies

Voorbeelden	Resultaten
RangeFractile (0.24,1,2,4,6)	retourneert 1,72
RangeFractile(0.5,1,2,3,4,6)	retourneert 3
RangeFractile (0.5,1,2,5,6)	retourneert 3,5

### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
RangeTab:
LOAD recno() as RangeID, RangeFractile(0.5,Field1,Field2,Field3) as MyRangeFrac INLINE [
Field1, Field2, Field3
10,5,6
2,3,7
8,2,8
18,11,9
5,5,9
9,4,2
];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeFrac voor elk van de records in de tabel.

RangeID	MyRangeFrac
1	6
2	3
3	8
4	11
5	5
6	4

### Voorbeeld met uitdrukking:

```
RangeFractile (0.5, Above(Sum(MyField),0,3))
```

In dit voorbeeld bevat de inter-recordfunctie **Above()** de optionele argumenten offset en count. Dit levert een bereik van resultaten op dat kan worden gebruikt als invoer voor elk van de bereikfuncties. In dit geval retourneert `Above(Sum(MyField),0,3)` de waarden van `MyField` voor de huidige rij en de twee rijen erboven. Deze waarden vormen de invoer voor de functie **RangeFractile()**. Dus voor de onderste rij van de onderstaande tabel is dit het equivalent van `RangeFractile(0.5, 3,4,6)`, oftewel de berekening van de 0,5 fractiel voor de serie 3, 4 en 6. In de eerste twee rijen in de onderstaande tabel, wordt het aantal waarden in het bereik dienovereenkomstig verlaagd, in gevallen waarbij er geen rijen boven de huidige rij zijn. Andere inter-recordfuncties leveren vergelijkbare resultaten op.

Voorbeeldgegevens

MyField	RangeFractile(0.5, Above(Sum(MyField),0,3))
1	1
2	1.5
3	2
4	3
5	4
6	5

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
1
2
3
4
5
6
] ;
```

---

### Zie ook:

[p Above - diagramfunctie \(page 1268\)](#)

[p Fractile - diagramfunctie \(page 403\)](#)

## RangeIRR

**RangeIRR()** retourneert het interne opbrengstpercentage voor een reeks van cashflows die worden weergegeven door de invoerwaarden.

De interne rentabiliteit is het rentetarief dat wordt ontvangen voor een investering die bestaat uit betalingen (negatieve waarden) en inkomsten (positieve waarden) die zich in regelmatige perioden voordoen.

### Syntaxis:

```
RangeIRR (value[, value][, Expression])
```



**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
value	Een enkele waarde of een enkel bereik van waarden zoals geretourneerd door een interrecordfunctie met een derde optionele parameter. Voor de functie is minimaal één positieve en één negatieve waarde vereist voor de berekening.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

Voorbeeldtabel

Voorbeelden	Resultaten														
RangeIRR(-7000,12000,15000,18000,21000,26000)	retourneert 0,0866														
<p>Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.</p> <pre> RangeTab3: LOAD *, recno() as RangeID, RangeIRR(Field1,Field2,Field3) as RangeIRR; LOAD * INLINE [ Field1 Field2 Field3 -10000 5000 6000 -2000 NULL 7000 -8000 'abc' 8000 -1800 11000 9000 -5000 5000 9000 -9000 4000 2000 ] (delimiter is ' ');</pre>	<p>De resulterende tabel bevat de geretourneerde waarden van RangeIRR voor elk van de records in de tabel.</p> <table border="1"> <thead> <tr> <th>RangeID</th> <th>RangeIRR</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0.0639</td> </tr> <tr> <td>2</td> <td>0.8708</td> </tr> <tr> <td>3</td> <td>-</td> </tr> <tr> <td>4</td> <td>5.8419</td> </tr> <tr> <td>5</td> <td>0.9318</td> </tr> <tr> <td>6</td> <td>-0.2566</td> </tr> </tbody> </table>	RangeID	RangeIRR	1	0.0639	2	0.8708	3	-	4	5.8419	5	0.9318	6	-0.2566
RangeID	RangeIRR														
1	0.0639														
2	0.8708														
3	-														
4	5.8419														
5	0.9318														
6	-0.2566														

**Zie ook:**

*p Interrecord-functies (page 1265)*

## RangeKurtosis

**RangeKurtosis()** retourneert de waarde die correspondeert met de piekvorm van een reeks getallen.

**Syntaxis:**

```
RangeKurtosis(first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

**Beperkingen:**

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

**Voorbeelden en resultaten:**

Voorbeelden van functies

Voorbeelden	Resultaten
RangeKurtosis (1,2,4,7)	retourneert -0,28571428571429

**Zie ook:**

p *Kurtosis - diagramfunctie (page 410)*

## RangeMax

**RangeMax()** retourneert de hoogste numerieke waarden die binnen de uitdrukking of het veld worden gevonden.

**Syntaxis:**

```
RangeMax (first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Beperkingen:

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

### Voorbeelden en resultaten:

Funcatievoorbeelden

Voorbeelden	Resultaten
RangeMax (1,2,4)	retourneert 4
RangeMax (1,'xyz')	retourneert 1
RangeMax (null( ), 'abc')	Retourneert NULL

### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
RangeTab3:  
LOAD recno() as RangeID, RangeMax(Field1,Field2,Field3) as MyRangeMax INLINE [  
Field1, Field2, Field3  
10,5,6  
2,3,7  
8,2,8  
18,11,9  
5,5,9  
9,4,2  
];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeMax voor elk van de records in de tabel.

Resultaattabel

RangeID	MyRangeMax
1	10
2	7
3	8
4	18
5	9
6	9

### Voorbeeld met uitdrukking:

```
RangeMax (Above(MyField,0,3))
```

Retourneert de maximale waarde in het bereik van drie waarden van **MyField** berekend op de huidige rij en de twee rijen boven de huidige rij. Doordat het derde argument wordt opgegeven als 3, retourneert de functie **Above()** drie waarden, met voldoende rijen erboven, die als invoer worden gebruikt voor de functie **RangeMax()**.

Gegevens die worden gebruikt in voorbeelden:



Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.

Voorbeeldgegevens

MyField	RangeMax (Above(Sum(MyField),1,3))
10	10
2	10
8	10
18	18
5	18
9	18

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

### RangeMaxString

**RangeMaxString()** retourneert de laatste waarde die in de gesorteerde tekst van de uitdrukking of het veld wordt gevonden.

**Syntaxis:**

```
RangeMaxString(first_expr[, Expression])
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

### Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Voorbeelden en resultaten:

#### Voorbeelden van functies

Voorbeelden	Resultaten
RangeMaxString (1,2,4)	retourneert 4
RangeMaxString ('xyz', 'abc')	Retourneert 'xyz'
RangeMaxString (5, 'abc')	Retourneert 'abc'
RangeMaxString (null( ))	Retourneert NULL

### Voorbeeld met uitdrukking:

```
RangeMaxString (Above(MaxString(MyField),0,3))
```

Retourneert het laatste (in alfabetische volgorde) van de drie resultaten van de functie **MaxString (MyField)** uitgevoerd op de huidige rij en twee rijen boven de huidige rij.

Gegevens die worden gebruikt in voorbeelden:



*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

#### Voorbeeldgegevens

MyField	RangeMaxString(Above(MaxString(MyField),0,3))
10	10
abc	abc
8	abc
def	def
xyz	xyz
9	xyz

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
```

```
'abc'  
8  
'def'  
'xyz'  
9  
] ;
```

### Zie ook:

p *MaxString - diagramfunctie (page 535)*

## RangeMin

**RangeMin()** retourneert de laagste numerieke waarden die binnen de uitdrukking of het veld worden gevonden.

### Syntaxis:

```
RangeMin (first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

#### Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Beperkingen:

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

### Voorbeelden en resultaten:

#### Voorbeelden van functies

Voorbeelden	Resultaten
RangeMin (1,2,4)	retourneert 1
RangeMin (1, 'xyz')	retourneert 1
RangeMin (null( ), 'abc')	Retourneert NULL

### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

RangeTab3:

## 5 Script- en diagramfuncties

```
LOAD recno() as RangeID, RangeMin(Field1,Field2,Field3) as MyRangeMin INLINE [  
Field1, Field2, Field3  
10,5,6  
2,3,7  
8,2,8  
18,11,9  
5,5,9  
9,4,2  
];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeMin voor elk van de records in de tabel.

RangeID	MyRangeMin
1	5
2	2
3	2
4	9
5	5
6	2

Voorbeeld met uitdrukking:

```
RangeMin (Above(MyField,0,3))
```

Retourneert de minimale waarde in het bereik van drie waarden van **MyField** berekend op de huidige rij en de twee rijen boven de huidige rij. Doordat het derde argument wordt opgegeven als 3, retourneert de functie **Above()** drie waarden, met voldoende rijen erboven, die als invoer worden gebruikt voor de functie **RangeMin()**.

Gegevens die worden gebruikt in voorbeelden:

MyField	RangeMin(Above(MyField,0,3))
10	10
2	2
8	2
18	2
5	5
9	5

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:  
LOAD * INLINE [
```

```
MyField
10
2
8
18
5
9
] ;
```

### Zie ook:

p *Min - diagramfunctie (page 337)*

## RangeMinString

**RangeMinString()** retourneert de eerste waarde die in de gesorteerde tekst van de uitdrukking of het veld wordt gevonden.

### Syntaxis:

```
RangeMinString(first_expr[, Expression])
```

**Retourgegevenstypen:** tekenreeks

### Argumenten:

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

#### Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Voorbeelden en resultaten:

#### Voorbeelden van functies

Voorbeelden	Resultaten
RangeMinString (1,2,4)	retourneert 1
RangeMinString ('xyz','abc')	Retourneert 'abc'
RangeMinString (5,'abc')	retourneert 5
RangeMinString (null( ))	Retourneert NULL

### Voorbeeld met uitdrukking:

```
RangeMinString (Above(MinString(MyField),0,3))
```



Retourneert het eerste (in alfabetische volgorde) van de drie resultaten van de functie **MinString(MyField)** uitgevoerd op de huidige rij en twee rijen boven de huidige rij.

Gegevens die worden gebruikt in voorbeelden:



Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.

Voorbeeldgegevens

MyField	RangeMinString(Above(MinString(MyField),0,3))
10	10
abc	10
8	8
def	8
xyz	8
9	9

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
'def'
'xyz'
9
] ;
```

### Zie ook:

[p MinString - diagramfunctie \(page 538\)](#)

## RangeMissingCount

**RangeMissingCount()** retourneert het aantal niet-numerieke waarden (inclusief NULL) in de uitdrukking of het veld.

### Syntaxis:

```
RangeMissingCount(first_expr[, Expression])
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden geteld.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden geteld.

**Voorbeelden en resultaten:**

Voorbeelden van functies

Voorbeelden	Resultaten
RangeMissingCount (1,2,4)	retourneert 0
RangeMissingCount (5,'abc')	retourneert 1
RangeMissingCount (null( ))	retourneert 1

Voorbeeld met uitdrukking:

RangeMissingCount (Above(MinString(MyField),0,3))

Retourneert het aantal niet-numerieke waarden in de drie resultaten van de functie **MinString(MyField)**, uitgevoerd op de huidige rij en twee rijen boven de huidige rij.



*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

Voorbeeldgegevens

MyField	RangeMissingCount (Above(MinString (MyField),0,3))	Explanation
10	2	Retourneert 2, omdat er geen rijen boven deze rij staan, zodat 2 van de 3 waarden ontbreken.
abc	2	Retourneert 2, omdat er slechts 1 rij boven de huidige rij staat en de huidige rij niet-numeriek is ('abc').

MyField	RangeMissingCount (Above(MinString(MyField),0,3))	Explanation
8	1	Retourneert 1, omdat 1 van de 3 rijen een niet-numerieke waarde bevat ('abc').
def	2	Retourneert 2, omdat 2 van de 3 rijen niet-numerieke waarden bevatten ('def' en 'abc').
xyz	2	Retourneert 2, omdat 2 van de 3 rijen niet-numerieke waarden bevatten (' xyz' en 'def').
9	2	Retourneert 2, omdat 2 van de 3 rijen niet-numerieke waarden bevatten (' xyz' en 'def').

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
'def'
'xyz'
9
] ;
```

**Zie ook:**

[p MissingCount - diagramfunctie \(page 354\)](#)

## RangeMode

**RangeMode()** retourneert de meest voorkomende waarde (moduswaarde) in de uitdrukking of het veld.

**Syntaxis:**

```
RangeMode (first_expr {, Expression})
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

### Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.

Argument	Beschrijving
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Beperkingen:

Als er meerdere waarden zijn met de hoogste frequentie, wordt NULL geretourneerd.

### Voorbeelden en resultaten:

Voorbeelden van functies

Voorbeelden	Resultaten
RangeMode (1,2,9,2,4)	retourneert 2
RangeMode ('a',4,'a',4)	Retourneert NULL
RangeMode (null( ))	Retourneert NULL

### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

```
RangeTab3:
LOAD recno() as RangeID, RangeMode(Field1,Field2,Field3) as MyRangeMode INLINE [
Field1, Field2, Field3
10,5,6
2,3,7
8,2,8
18,11,9
5,5,9
9,4,2
];
```

De resulterende tabel bevat de geretourneerde waarden van **MyRangeMode** voor elk van de records in de tabel.

Resultatentabel

RangeID	MyRangMode
1	-
2	-
3	8
4	-
5	5
6	-

Voorbeeld met uitdrukking:

RangeMode (Above(MyField,0,3))

Retourneert de meest voorkomende waarde in de drie resultaten van **MyField** uitgevoerd op de huidige rij en twee rijen boven de huidige rij. Doordat het derde argument wordt opgegeven als 3, retourneert de functie **Above()** drie waarden, met voldoende rijen erboven, die als invoer worden gebruikt voor de functie **RangeMode()**.

Gebruikte gegevens in voorbeeld:

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```



Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.

Voorbeeldgegevens

MyField	RangeMode(Above(MyField,0,3))
10	Retourneert 10, omdat er geen bovenliggende rijen zijn, zodat de enkele waarde de waarde is die het meest voorkomt.
2	-
8	-
18	-
5	-
9	-

Zie ook:

[p Mode - diagramfunctie \(page 340\)](#)

### RangeNPV

**RangeNPV()** retourneert de huidige nettowaarde van een investering op basis van een kortingspercentage en een reeks toekomstige periodieke betalingen (negatieve waarden) en inkomsten (positieve waarden). Het resultaat heeft de standaardgetalnotatie van **money**.

Voor cashflows die niet noodzakelijkerwijs periodiek zijn, raadpleegt u [RangeXNPV \(page 1362\)](#).

**Syntaxis:**

```
RangeNPV (discount_rate, value[,value][, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
discount_rate	Het rentepercentage per periode.
value	Een betaling of inkomsten die aan het eind van elke periode plaatsvindt. Elke waarde kan een enkele waarde of een bereik van waarden zijn die is geretourneerd door een interrecordfunctie met een derde optionele parameter.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

Voorbeelden	Resultaten														
RangeNPV(0.1, -10000, 3000, 4200, 6800)	retourneert 1188,44														
<p>Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.</p> <pre> RangeTab3: LOAD *, recno() as RangeID, RangeNPV(Field1,Field2,Field3) as RangeNPV; LOAD * INLINE [ Field1 Field2 Field3 10 5 -6000 2 NULL 7000 8 'abc' 8000 18 11 9000 5 5 9000 9 4 2000 ] (delimiter is ' '); </pre>	<p>De resulterende tabel bevat de geretourneerde waarden van RangeNPV voor elk van de records in de tabel.</p> <table border="1"> <thead> <tr> <th>RangeID</th> <th>RangeNPV</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>\$-49.13</td> </tr> <tr> <td>2</td> <td>\$777.78</td> </tr> <tr> <td>3</td> <td>\$98.77</td> </tr> <tr> <td>4</td> <td>\$25.51</td> </tr> <tr> <td>5</td> <td>\$250.83</td> </tr> <tr> <td>6</td> <td>\$20.40</td> </tr> </tbody> </table>	RangeID	RangeNPV	1	\$-49.13	2	\$777.78	3	\$98.77	4	\$25.51	5	\$250.83	6	\$20.40
RangeID	RangeNPV														
1	\$-49.13														
2	\$777.78														
3	\$98.77														
4	\$25.51														
5	\$250.83														
6	\$20.40														

**Zie ook:**

[p Interrecord-functies \(page 1265\)](#)

### RangeNullCount

**RangeNullCount()** retourneert het aantal NULL-waarden in de uitdrukking of het veld.

**Syntaxis:**

```
RangeNullCount (first_expr [, Expression])
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

**Voorbeelden en resultaten:**

Voorbeelden van functies

Voorbeelden	Resultaten
RangeNullCount (1,2,4)	retourneert 0
RangeNullCount (5, 'abc')	retourneert 0
RangeNullCount (null( ), null( ))	retourneert 2

Voorbeeld met uitdrukking:

```
RangeNullCount (Above(Sum(MyField),0,3))
```

Retourneert het aantal NULL-waarden in de drie resultaten van de functie **Sum(MyField)** uitgevoerd op de huidige rij en twee rijen boven de huidige rij.



*Het kopiëren van **MyField** in het onderstaande voorbeeld resulteert niet in een NULL-waarde.*

Voorbeeldgegevens

MyField	RangeNullCount(Above(Sum(MyField),0,3))
10	Retourneert 2, omdat er geen rijen boven deze rij staan, zodat 2 van de 3 waarden ontbreken (=NULL).
'abc'	Retourneert 1, omdat er slechts één rij boven de huidige rij staat, zodat één van de drie waarden ontbreekt (=NULL).
8	Retourneert 0 omdat één van de drie rijen een NULL-waarde is.

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
```

```
10  
'abc'  
8  
] ;
```

### Zie ook:

[p NullCount - diagramfunctie \(page 357\)](#)

## RangeNumericCount

**RangeNumericCount()** retourneert het aantal numerieke waarden in de uitdrukking of het veld.

### Syntaxis:

```
RangeNumericCount(first_expr[, Expression])
```

**Retourgegevenstypen:** geheel getal

### Argumenten:

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

#### Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Voorbeelden en resultaten:

#### Voorbeelden van functies

Voorbeelden	Resultaten
<code>RangeNumericCount (1,2,4)</code>	retourneert 3
<code>RangeNumericCount (5,'abc')</code>	retourneert 1
<code>RangeNumericCount (null( ))</code>	retourneert 0

Voorbeeld met uitdrukking:

```
RangeNumericCount (Above(MaxString(MyField),0,3))
```

Retourneert het aantal numerieke waarden in de drie resultaten van de functie **MaxString(MyField)** uitgevoerd op de huidige rij en twee rijen boven de huidige rij.





Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.

Voorbeeldgegevens

MyField	RangeNumericCount(Above(MaxString(MyField),0,3))
10	1
abc	1
8	2
def	1
xyz	1
9	1

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
def
xyz
9
] ;
```

### Zie ook:

[p NumericCount - diagramfunctie \(page 360\)](#)

## RangeOnly

**RangeOnly()** is een duale functie die een waarde retourneert als de uitdrukking één enkele unieke waarde oplevert. Als dat niet het geval is, retourneert de functie de waarde **NULL**.

### Syntaxis:

```
RangeOnly(first_expr[, Expression])
```

**Retourgegevenstypen:** duaal

### Argumenten:

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Voorbeelden en resultaten:

Voorbeelden	Resultaten
RangeOnly (1,2,4)	Retourneert NULL
RangeOnly (5, 'abc')	Retourneert NULL
RangeOnly (null( ), 'abc')	Retourneert 'abc'
RangeOnly(10,10,10)	retourneert 10

### Zie ook:

[p Only - diagramfunctie \(page 343\)](#)

## RangeSkew

**RangeSkew()** retourneert de waarde die correspondeert met de scheefheid van een reeks getallen.

### Syntaxis:

```
RangeSkew(first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

#### Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Beperkingen:

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

### Voorbeelden en resultaten:

Voorbeelden van functies

Voorbeelden	Resultaten
rangeskew (1,2,4)	retourneert 0,93521952958283
rangeskew (above (SalesValue,0,3))	Retourneert een glijdende scheefheid van het bereik van drie waarden dat wordt geretourneerd door de functie above() berekend op de huidige rij en de twee rijen boven de huidige rij.

### Gebruikte gegevens in voorbeeld:

Voorbeeldgegevens

CustID	RangeSkew(Above(SalesValue,0,3))
1-20	-, -, 0.5676, 0.8455, 1.0127, -0.8741, 1.7243, -1.7186, 1.5518, 1.4332, 0, 1.1066, 1.3458, 1.5636, 1.5439, 0.6952, -0.3766

```
SalesTable:
LOAD recno() as CustID, * inline [
SalesValue
101
163
126
139
167
86
83
22
32
70
108
124
176
113
95
32
42
92
61
21
] ;
```

---

### Zie ook:

p *Skew - diagramfunctie (page 445)*

## RangeStdev

**RangeStdev()** retourneert de standaardafwijking van een reeks getallen.

### Syntaxis:

```
RangeStdev(first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Beperkingen:

Als er geen numerieke waarde wordt gevonden, wordt NULL geretourneerd.

### Voorbeelden en resultaten:

Voorbeelden van functies

Voorbeelden	Resultaten
RangeStdev (1,2,4)	retourneert 1,5275252316519
RangeStdev (null( ))	Retourneert NULL
RangeStdev (above (SalesValue),0,3))	Retourneert een glijdende standaard van het bereik van drie waarden dat wordt geretourneerd door de functie above() berekend op de huidige rij en de twee rijen boven de huidige rij.

Gebruikte gegevens in voorbeeld:

Voorbeeldgegevens

CustID	RangeStdev(SalesValue, 0,3))
1-20	-.43.841, 34.192, 18.771, 20.953, 41.138, 47.655, 36.116, 32.716, 25.325, 38.000, 27.737, 35.553, 33.650, 42.532, 33.858, 32.146, 25.239, 35.595

SalesTable:

```
LOAD recno() as CustID, * inline [
SalesValue
101
163
126
139
```

```
167
86
83
22
32
70
108
124
176
113
95
32
42
92
61
21
] ;
```

---

### Zie ook:

[p Stdev - diagramfunctie \(page 448\)](#)

## RangeSum

**RangeSum()** retourneert de som van een reeks waarden. Alle niet-numerieke waarden worden behandeld als 0.

### Syntaxis:

```
RangeSum(first_expr[, Expression])
```

**Retourgegevenstypen:** numeriek

### Argumenten:

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argumenten

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

### Beperkingen:

De functie **RangeSum** behandelt alle niet-numerieke waarden als 0.

### Voorbeelden en resultaten:

Voorbeelden

Voorbeelden	Resultaten
RangeSum (1,2,4)	retourneert 7
RangeSum (5, 'abc')	retourneert 5
RangeSum (null( ))	retourneert 0

### Voorbeeld:

Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.

RangeTab3:

```
LOAD recno() as RangeID, Rangesum(Field1,Field2,Field3) as MyRangeSum INLINE [
```

```
Field1, Field2, Field3
```

```
10,5,6
```

```
2,3,7
```

```
8,2,8
```

```
18,11,9
```

```
5,5,9
```

```
9,4,2
```

```
];
```

De resulterende tabel bevat de geretourneerde waarden van MyRangeSum voor elk van de records in de tabel.

Resultaattabel

RangeID	MyRangeSum
1	21
2	12
3	18
4	38
5	19
6	15

Voorbeeld met uitdrukking:

RangeSum (Above(MyField,0,3))

Retourneert de som van de drie waarden van **MyField**): van de huidige rij en de twee rijen boven de huidige rij. Doordat het derde argument wordt opgegeven als 3, retourneert de functie **Above()** drie waarden, met voldoende rijen erboven, die als invoer worden gebruikt voor de functie **RangeSum()**.

Gegevens die worden gebruikt in voorbeelden:



Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.

Voorbeeldgegevens

MyField	RangeSum(Above(MyField,0,3))
10	10
2	12
8	20
18	28
5	31
9	32

Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
2
8
18
5
9
] ;
```

Zie ook:

[p Sum - diagramfunctie \(page 346\)](#)  
[p Above - diagramfunctie \(page 1268\)](#)

## RangeTextCount

**RangeTextCount()** retourneert het aantal tekstwaarden in de uitdrukking of het veld.

**Syntaxis:**

```
RangeTextCount(first_expr[, Expression])
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

De argumenten van deze functie kunnen interrecordfuncties bevatten die op hun beurt een lijst met waarden opleveren.

Argument

Argument	Beschrijving
first_expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
Expression	Optionele uitdrukkingen of velden die het gegevensbereik bevatten dat moet worden gemeten.

**Voorbeelden en resultaten:**

Voorbeelden van functies

Voorbeelden	Resultaten
RangeTextCount (1,2,4)	retourneert 0
RangeTextCount (5, 'abc')	retourneert 1
RangeTextCount (null( ))	retourneert 0

Voorbeeld met uitdrukking:

RangeTextCount (Above(MaxString(MyField),0,3))

Retourneert het aantal tekstwaarden in de drie resultaten van de functie **MaxString(MyField)** uitgevoerd op de huidige rij en twee rijen boven de huidige rij.

Gegevens die worden gebruikt in voorbeelden:



*Schakel het sorteren van **MyField** uit om ervoor te zorgen dat het voorbeeld werkt zoals verwacht.*

Voorbeeldgegevens

MyField	MaxString(MyField)	RangeTextCount(Above(Sum(MyField),0,3))
10	10	0
abc	abc	1
8	8	1
def	def	2
xyz	xyz	2
9	9	2



Gegevens die worden gebruikt in voorbeelden:

```
RangeTab:
LOAD * INLINE [
MyField
10
'abc'
8
null()
'xyz'
9
] ;
```

**Zie ook:**

[p TextCount - diagramfunctie \(page 363\)](#)

### RangeXIRR

**RangeXIRR()** retourneert het interne opbrengstpercentage voor een schema cashflows die niet noodzakelijk periodiek is. Gebruik voor de berekening van het interne opbrengstpercentage voor een reeks periodieke cashflows de functie **RangeIRR**.

**Syntaxis:**

```
RangeXIRR(value, date{, value, date})
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

#### Argumenten

Argument	Beschrijving
value	Een cashflow of een reeks cashflows die overeenkomen met een betalingschema met datums. De reeks waarden moet ten minste een positieve en een negatieve waarde bevatten.
date	Een betaaldatum of een schema van betaaldatums die corresponderen met de cashflowbetalingen.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

Voorbeelden	Resultaten
<code>RangeXIRR(-2500, '2008-01-01', 2750, '2008-09-01')</code>	retourneert 0,1532

**Zie ook:**

p *RangeIRR* (page 1336)

**RangeXNPV**

**RangeXNPV()** retourneert de huidige nettowaarde voor een schema cashflows die niet noodzakelijk periodiek is. Het resultaat heeft de standaardgetalnotatie voor geld. Met de functie **RangeNPV** berekent u de huidige nettowaarde voor een reeks periodieke cashflows.

**Syntaxis:**

```
RangeXNPV(discount_rate, values, dates[, Expression])
```

**Retourgegevenstypen:** numeriek

**Argumenten:**

Argumenten

Argument	Beschrijving
discount_rate	Het rentepercentage per periode.
values	Een cashflow of een reeks cashflows die overeenkomen met een betalingsschema met datums. Elke waarde kan een enkele waarde of een bereik van waarden zijn die is geretourneerd door een interrecordfunctie met een derde optionele parameter. De reeks waarden moet ten minste een positieve en een negatieve waarde bevatten.
dates	Een betaaldatum of een schema van betaaldatums die corresponderen met de cashflowbetalingen.

**Beperkingen:**

Tekstwaarden, NULL-waarden en ontbrekende waarden worden genegeerd.

De kortingen voor alle betalingen zijn gebaseerd op een jaar van 365 dagen.

Voorbeeldtabel

Voorbeelden	Resultaten
<code>RangeXNPV(0.1, -2500, '2008-01-01', 2750, '2008-09-01')</code>	retourneert 80,25

Voorbeelden	Resultaten														
<p>Voeg het voorbeeldscript toe aan uw app en voer het uit. Voeg vervolgens de velden die in de resultatenkolom staan toe aan een werkblad in uw app om het resultaat te bekijken.</p> <pre> RangeTab3: LOAD *, recno() as RangeID, RangeXNPV(Field1,Field2,Field3) as RangeNPV; LOAD * INLINE [ Field1 Field2 Field3 10 5 -6000 2 NULL 7000 8 'abc' 8000 18 11 9000 5 5 9000 9 4 2000 ] (delimiter is ' '); </pre>	<p>De resulterende tabel bevat de geretourneerde waarden van RangeXNPV voor elk van de records in de tabel.</p> <table border="1"> <thead> <tr> <th>RangeID</th> <th>RangeXNPV</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>\$-49.13</td> </tr> <tr> <td>2</td> <td>\$777.78</td> </tr> <tr> <td>3</td> <td>\$98.77</td> </tr> <tr> <td>4</td> <td>\$25.51</td> </tr> <tr> <td>5</td> <td>\$250.83</td> </tr> <tr> <td>6</td> <td>\$20.40</td> </tr> </tbody> </table>	RangeID	RangeXNPV	1	\$-49.13	2	\$777.78	3	\$98.77	4	\$25.51	5	\$250.83	6	\$20.40
RangeID	RangeXNPV														
1	\$-49.13														
2	\$777.78														
3	\$98.77														
4	\$25.51														
5	\$250.83														
6	\$20.40														

## 5.22 Rangorde- en clusterfuncties

Deze functies kunnen alleen worden gebruikt in diagramuitdrukkingen.

### Rangordefuncties in diagrammen



*De onderdrukking van nulwaarden wordt automatisch uitgeschakeld als deze functies worden gebruikt. NULL-waarden worden buiten beschouwing gelaten.*

Rank

**Rank()** evalueert de rijen van het diagram in de uitdrukking en geeft voor elke rij de relatieve positie van de waarde van de dimensie die in de uitdrukking wordt geëvalueerd weer. Bij de evaluatie van de uitdrukking vergelijkt de functie het resultaat met het resultaat van de andere rijen die het huidige kolomsegment bevatten, en retourneert de rangschikking van de huidige rij in het segment.

```
Rank - diagramfunctie([TOTAL [<fld {, fld}>]] expr[, mode[, fmt]])
```

HRank

**HRank()** evalueert de uitdrukking en vergelijkt het resultaat met het resultaat van de andere kolommen die het huidige rijsegment van een draaitabel bevatten. De functie retourneert vervolgens de rangschikking van de huidige kolom binnen het segment.

```
HRank - diagramfunctie([TOTAL] expr[, mode[, fmt]])
```

### Clusterfuncties in diagrammen

KMeans2D

De eigenschapsgroep **Sitelicentie** bevat eigenschappen die gerelateerd zijn aan de licentie voor het Qlik Sense-systeem. Alle velden zijn verplicht en mogen niet leeg zijn.

### Sitelicentie-eigenschappen

Eigenschapnaam	Beschrijving
<b>Naam eigenaar</b>	De gebruikersnaam van de Qlik Sense-producteigenaar.
<b>Organisatie eigenaar</b>	De naam van de organisatie waarvan de Qlik Sense-producteigenaar lid is.
<b>Serienummer</b>	Het serienummer dat aan de Qlik Sense-software is toegewezen.
<b>Controlenummer</b>	Het controlenummer dat aan de Qlik Sense-software is toegewezen.
<b>Toegang tot LEF</b>	De License Enabler File (LEF) die aan de Qlik Sense-software is toegewezen.

**KMeans2D()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de cluster-id weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter `coordinate_1` respectievelijk `coordinate_2`. Dit zijn beide aggregaties. De parameter `num_clusters` bepaalt het aantal clusters dat wordt gemaakt. Gegevens kunnen optioneel worden genormaliseerd door de normparameter.

```
KMeans2D - diagramfunctie(num_clusters, coordinate_1, coordinate_2 [, norm])
```

#### KMeansND

**KMeansND()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de cluster-id weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter `coordinate_1` en `coordinate_2` etc. tot en met de n-kolommen Dit zijn beide aggregaties. De parameter `num_clusters` bepaalt het aantal clusters dat wordt gemaakt.

```
KMeansND - diagramfunctie(num_clusters, num_iter, coordinate_1, coordinate_2  
[, coordinate_3 [, ...]])
```

#### KMeansCentroid2D

**KMeansCentroid2D()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de gewenste coördinaat weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter `coordinate_1` respectievelijk `coordinate_2`. Dit zijn beide aggregaties. De parameter `num_clusters` bepaalt het aantal clusters dat wordt gemaakt. Gegevens kunnen optioneel worden genormaliseerd door de normparameter.

```
KMeansCentroid2D - diagramfunctie(num_clusters, coordinate_no, coordinate_1,  
coordinate_2 [, norm])
```

#### KMeansCentroidND

**KMeansCentroidND()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de gewenste coördinaat weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter `coordinate_1` respectievelijk `coordinate_2`. Dit zijn beide aggregaties. De parameter `num_clusters` bepaalt het aantal clusters dat wordt gemaakt.

```
KMeansCentroidND - diagramfunctie(num_clusters, num_iter, coordinate_no,
coordinate_1, coordinate_2 [,coordinate_3 [, ...]])
```

## Rank - diagramfunctie

**Rank()** evalueert de rijen van het diagram in de uitdrukking en geeft voor elke rij de relatieve positie van de waarde van de dimensie die in de uitdrukking wordt geëvalueerd weer. Bij de evaluatie van de uitdrukking vergelijkt de functie het resultaat met het resultaat van de andere rijen die het huidige kolomsegment bevatten, en retourneert de rangschikking van de huidige rij in het segment.

### Kolomsegmenten

	Region	Country	Population	Rank(Population)
Column segment #1	Americas	Mexico	128.932.753	2
	Americas	Canada	37.742.154	3
	Americas	United States of America	331.002.051	1
Column segment #2	Europe	Sweden	10.099.265	4
	Europe	United Kingdom	67.886.011	2
	Europe	France	65.273.511	3
	Europe	Germany	83.783.942	1

Voor andere diagrammen dan tabellen wordt het huidige kolomsegment gedefinieerd zoals deze wordt weergegeven in het equivalent van de strakke tabel.

### Syntaxis:

```
Rank ([TOTAL] expr[, mode[, fmt]])
```

### Retourgegevenstypen: dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
mode	Geeft de numerieke representatie aan van het resultaat van de functie.
fmt	Geeft de tekstrepresentatie aan van het resultaat van de functie.
TOTAL	Als het diagram eendimensionaal is of als de uitdrukking wordt voorafgegaan door de kwalificatie <b>TOTAL</b> , wordt de functie geëvalueerd voor de hele kolom. Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.

De rangorde wordt geretourneerd als een dubbele waarde. Als elke rij een unieke rangschikking heeft, is dit een geheel getal tussen 1 en het aantal rijen in het huidige kolomsegment.

Als meerdere rijen dezelfde rangschikking hebben, kan de tekstuele en numerieke representatie worden bestuurd met de parameters **mode** en **fmt**.

### mode

Het tweede argument, **mode**, kan de volgende waarden hebben:

### mode-voorbeelden

Waarde	Beschrijving
0 (standaard)	<p>Als alle rangnummers in de delende groep lager zijn dan de middenwaarde van de volledige rangschikking, krijgen alle rijen het laagste positienummer in de delende groep.</p> <p>Als alle rangnummers in de delende groep hoger zijn dan de middenwaarde van de volledige rangschikking, krijgen alle rijen het hoogste rangnummer in de delende groep.</p> <p>Als rangnummers in de delende groep zowel lager als hoger zijn dan de middenwaarde van de volledige rangschikking, krijgen alle rijen de waarde die overeenkomt met het gemiddelde van het hoogste en laagste rangnummer in het volledige kolomsegment.</p>
1	Laagste rangnummer in alle rijen.
2	Gemiddelde rangnummer in alle rijen.
3	Hoogste rangnummer in alle rijen.
4	Laagste rangnummer in eerste rij, vervolgens opgehoogd met één voor elke rij.

### fmt

Het derde argument, **fmt**, kan de volgende waarden hebben:

### fmt-voorbeelden

Waarde	Beschrijving
0 (standaard)	Lage waarde - hoge waarde in alle rijen (bijvoorbeeld 3 - 4).
1	Lage waarde in alle rijen.
2	Lage waarde in eerste rij, leeg in alle volgende rijen.

De volgorde van rijen voor **mode** 4 en **fmt** 2 wordt bepaald door de sorteervolgorde van de diagramdimensies.

### Voorbeelden en resultaten:

Maak twee visualisaties op basis van de dimensies Product en Sales en nog één op basis van Product en UnitSales. Voeg metingen toe zoals weergegeven in de volgende tabel.

### Voorbeelden van rangorde

Voorbeelden	Resultaten
Voorbeeld 1. Maak een tabel met de dimensies <code>Customer</code> en <code>Sales</code> en de meting <code>Rank(Sales)</code>	<p>Het resultaat is afhankelijk van de sorteervolgorde van de dimensies. Als de tabel wordt gesorteerd op <code>Customer</code>, bevat de tabel alle waarden van <code>Sales</code> voor <code>Astrida</code>, daarna <code>Betacab</code> enzovoort. De resultaten voor <code>Rank(Sales)</code> geven 10 aan voor de waarde 12 van <code>Sales</code>, 9 voor de waarde 13 van <code>Sales</code> enzovoort, waarbij de rangordewaarde 1 wordt geretourneerd voor de waarde 78 van <code>Sales</code>. Het volgende kolomsegment begint met <code>Betacab</code>, waarvoor de eerste waarde van <code>Sales</code> in het segment 12 bedraagt. De rangordewaarde van <code>Rank(Sales)</code> hiervoor is gegeven als 11.</p> <p>Als de tabel wordt gesorteerd op <code>Sales</code>, bestaan de kolomsegmenten uit de waarden van <code>Sales</code> en de bijbehorende <code>Customer</code>. Aangezien er twee waarden 12 zijn voor <code>Sales</code> (voor <code>Astrida</code> en <code>Betacab</code>), is de waarde van <code>Rank(Sales)</code> voor dat kolomsegment 1-2, voor elke waarde van <code>Customer</code>. Dit komt doordat er twee waarden van <code>Customer</code> zijn voor de waarde 12 van <code>Sales</code>. Als er 4 waarden waren geweest, zou het resultaat 1-4 zijn, voor alle rijen. Dit geeft aan hoe het resultaat eruitziet voor de standaardwaarde (0) van het argument <code>fmt</code>.</p>
Voorbeeld 2. Vervang de dimensie <code>Customer</code> door <code>Product</code> en voeg de meting toe <code>rank(Sales, 1, 2)</code>	Dit geeft 1 als resultaat in de eerste rij van elk kolomsegment en laat alle andere rijen leeg omdat de argumenten <b>mode</b> en <b>fmt</b> zijn ingesteld op respectievelijk 1 en 2.

Resultaten voor voorbeeld 1, waarbij de tabel is gesorteerd op `Customer`:

Resultatentabel

Customer	Sales	Rank(Sales)
Astrida	12	10
Astrida	13	9
Astrida	20	8
Astrida	22	7
Astrida	45	6
Astrida	46	5
Astrida	60	4
Astrida	65	3
Astrida	70	2

Customer	Sales	Rank(Sales)
Astrida	78	1
Betacab	12	11

Resultaten voor voorbeeld 1, waarbij de tabel is gesorteerd op Sales:

Resultatentabel

Customer	Sales	Rank(Sales)
Astrida	12	1-2
Betacab	12	1-2
Astrida	13	1
Betacab	15	1
Astrida	20	1
Astrida	22	1-2
Betacab	22	1-2
Betacab	24	1-2
Canutility	24	1-2

Gegevens die worden gebruikt in voorbeelden:

ProductData:

```
Load * inline [
```

```
Customer|Product|UnitsSales|UnitPrice
```

```
Astrida|AA|4|16
```

```
Astrida|AA|10|15
```

```
Astrida|BB|9|9
```

```
Betacab|BB|5|10
```

```
Betacab|CC|2|20
```

```
Betacab|DD|0|25
```

```
Canutility|AA|8|15
```

```
Canutility|CC|0|19
```

```
] (delimiter is '|');
```



```

Sales2013:
crosstable (Month, Sales) LOAD * inline [
Customer|Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec
Astrida|46|60|70|13|78|20|45|65|78|12|78|22
Betacab|65|56|22|79|12|56|45|24|32|78|55|15
Canutility|77|68|34|91|24|68|57|36|44|90|67|27
Divadip|57|36|44|90|67|27|57|68|47|90|80|94
] (delimiter is '|');

```

### Zie ook:

[p Sum - diagramfunctie \(page 346\)](#)

## HRank - diagramfunctie

**HRank()** evalueert de uitdrukking en vergelijkt het resultaat met het resultaat van de andere kolommen die het huidige rijsegment van een draaitabel bevatten. De functie retourneert vervolgens de rangschikking van de huidige kolom binnen het segment.

### Syntaxis:

```
HRank ( [ TOTAL ] expr [ , mode [ , fmt ] ] )
```

### Retourgegevenstypen: dual



*Deze functie werkt alleen in draaitabellen. In alle andere diagramtypen levert de functie NULL op.*

### Argumenten:

#### Argumenten

Argument	Beschrijving
expr	De uitdrukking die of het veld dat de gegevens bevat die moeten worden gemeten.
mode	Geeft de numerieke representatie aan van het resultaat van de functie.
fmt	Geeft de tekstrepresentatie aan van het resultaat van de functie.
TOTAL	Als het diagram eendimensionaal is of als de uitdrukking wordt voorafgegaan door de kwalificatie <b>TOTAL</b> , wordt de functie geëvalueerd voor de hele kolom. Als de tabel of het equivalent van de tabel meerdere verticale dimensies heeft, omvat het huidige kolomsegment alleen rijen met dezelfde waarden als de huidige rij in alle dimensiekolommen, met uitzondering van de kolom waarin de laatste dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.

## 5 Script- en diagramfuncties

---

Als de draaitabel eendimensionaal is of als de uitdrukking wordt voorafgegaan door de kwalificatie **total**, is het huidige rijsegment altijd gelijk aan de hele rij. Als de draaitabel meerdere horizontale dimensies heeft, omvat het huidige rijsegment alleen kolommen met dezelfde waarden als de huidige kolom in alle dimensierijen, met uitzondering van de rij waarin de laatste horizontale dimensie wordt weergegeven in de onderlinge sorteervolgorde van de velden.

De rangschikking wordt geretourneerd als een dubbele waarde. Als elke kolom een unieke rangschikking heeft, is dit een heel getal tussen 1 en het aantal kolommen in het huidige rijsegment.

Als meerdere kolommen dezelfde rangschikking hebben, kan de tekstuele en numerieke representatie worden bestuurd met de parameters **mode** en **format**.

Het tweede argument, **mode**, geeft de numerieke representatie aan van het resultaat van de functie:

### mode-voorbeelden

Value	Beschrijving
0 (standaard)	Als alle rangnummers in de delende groep lager zijn dan de middenwaarde van de volledige rangschikking, krijgen alle kolommen het laagste positienummer in de delende groep.  Als alle rangnummers in de delende groep hoger zijn dan de middenwaarde van de volledige rangschikking, krijgen alle kolommen het hoogste rangnummer in de delende groep.  Als rangnummers in de delende groep zowel lager als hoger zijn dan de middenwaarde van de volledige rangschikking, krijgen alle rijen de waarde die overeenkomt met het gemiddelde van het hoogste en laagste rangnummer in het volledige kolomsegment.
1	Laagste positienummer van alle kolommen in de groep.
2	Gemiddelde positienummer van alle kolommen in de groep.
3	Hoogste positienummer van alle kolommen in de groep.
4	Laagste positienummer in eerste kolom, vervolgens opgehoogd met één voor elke kolom in de groep.

Het derde argument, **format**, geeft de tekstrepresentatie aan van het resultaat van de functie:

### format-voorbeelden

Value	Beschrijving
0 (standaard)	Lage waarde&' - '&hoge waarde in alle kolommen (bijv. 3 - 4).
1	Laagste waarde van alle kolommen in de groep.
2	Laagste waarde in de eerste kolom, leeg in alle volgende kolommen in de groep.

De volgorde van kolommen voor **mode** 4 en **format** 2 wordt bepaald door de sorteervolgorde van de diagramdimensies.

### Voorbeelden:

```
HRank( sum( Sales ))  
HRank( sum( Sales ), 2 )  
HRank( sum( Sales ), 0, 1 )
```

## Optimaliseren met k-means: Een praktijkvoorbeeld

In het volgende voorbeeld wordt een voorbeeld uit de praktijk getoond waarbij functies voor K-means en middenpunten op een gegevensverzameling zijn toegepast. Met behulp van de functie K-means worden gegevenspunten gescheiden in clusters met dezelfde eigenschappen. De clusters worden compacter en verder uitgesplitst als het K-means-algoritme is toegepast op een configureerbaar aantal iteraties.

K-means wordt gebruikt in veel sectoren en gebruikscases. Enkele voorbeelden van gebruikscases waarbij clustering is toegepast zijn klantsegmentatie, fraudedetectie, het voorspellen van klantverloop, het afstemmen van stimuleringen voor klanten, het identificeren van cybercriminaliteit en optimalisatie van bezorgroutes. Het K-means clustering algoritme wordt steeds meer gebruikt door ondernemingen die patronen willen herkennen en hun diensten willen optimaliseren.

### Qlik Sense Functies voor K-means en middenpunten

Qlik Sense biedt twee functies voor K-means waarmee gegevenspunten op basis van overeenkomstigheid in clusters worden gegroepeerd. Zie *KMeans2D - diagramfunctie (page 1380)* en *KMeansND - diagramfunctie (page 1393)*. De functie **KMeans2D** accepteert twee dimensies en werkt goed voor het visualiseren van gegevens met behulp van een **spreidingsplot**. De functie **KMeansND** accepteert meer dan twee dimensies. Het is vrij eenvoudig om een 2D-resultaat op standaarddiagrammen toe te passen. In het volgende voorbeeld wordt K-means toegepast op een **spreidingsplot** met behulp van twee dimensies. K-means clustering kan worden gevisualiseerd door middel van een uitdrukking of een dimensie zoals in dit voorbeeld wordt getoond.

Middenpuntfuncties in Qlik Sense bepalen het rekenkundig gemiddelde van alle gegevenspunten in het cluster en identificeren een centraal punt, oftewel een middenpunt voor dat cluster. Voor elke diagramrij (of record) toont de middenpuntfunctie de coördinaat van het cluster waaraan dit gegevenspunt is toegewezen. Zie *KMeansCentroid2D - diagramfunctie (page 1406)* en *KMeansCentroidND - diagramfunctie (page 1408)*.

### Overzicht gebruikscase en voorbeeld

In het volgende voorbeeld wordt u door een gesimuleerd scenario uit de praktijk geleid. Een textielbedrijf in de staat New York, in de VS, moet de kosten verlagen door de leveringskosten te minimaliseren. Ze kunnen dit doen door magazijnen te verplaatsen naar locaties dichterbij hun distributeurs. Het bedrijf maakt gebruik van 118 distributeurs binnen de staat New York. In het volgende voorbeeld wordt gesimuleerd hoe een operationeel manager de distributeurs kan opdelen in vijf regio's met behulp van de functie K-means en vervolgens vijf optimale magazijnlocaties kan bepalen met behulp van de middenpuntfunctie. Het doel is om coördinaten te ontdekken die gebruikt kunnen worden om de vijf centrale magazijnlocaties te identificeren.

### De gegevensverzameling

De gegevensverzameling is gebaseerd op willekeurig gegenereerde namen en adressen in de staat New York met nauwkeurige lengte- en breedtecoördinaten. De dataset bevat de volgende tien kolommen: id, voornaam, achternaam, telefoon, adres, stad, staat, postcode, breedtegraad, lengtegraad. De gegevensverzameling is hieronder beschikbaar als een bestand dat u lokaal kunt downloaden en vervolgens kunt uploaden naar Qlik Sense of uitgelijnd voor de editor voor laden van gegevens. De app die wordt aangemaakt heet *Distributeurs K-means en middenpunt* en het eerste werkblad in de app heet *Clusteranalyse distributie*.

Selecteer de volgende link om het bestand met voorbeeldgegevens te downloaden: [DistributorData.csv](#)

*Distributor-gegevensverzameling: Uitgelijnd laden voor de editor bedoeld voor het laden van gegevens in Qlik Sense (page 1378)*

Titel: DistributorData

Totaal aantal records: 118

### De KMeans2D-functie toepassen

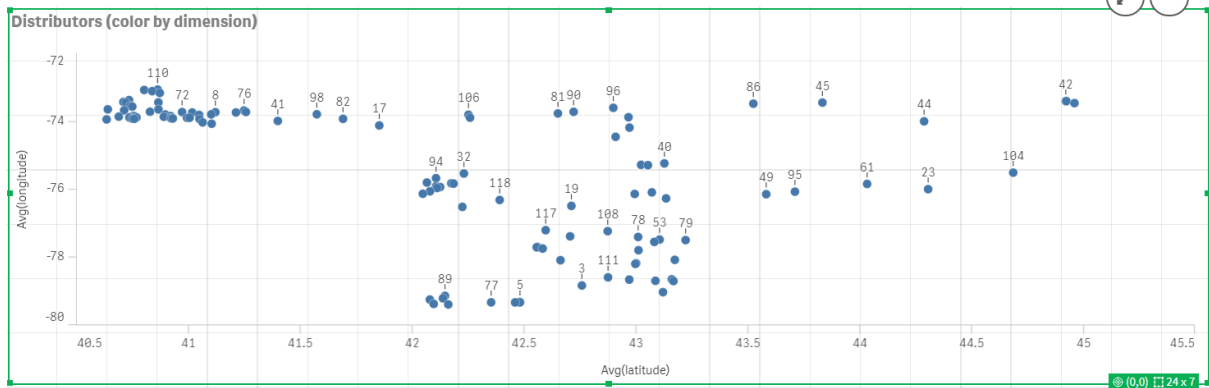
In dit voorbeeld wordt de configuratie van een **spreidingsplot** weergegeven met behulp van de gegevensverzameling *DistributorData*, de **KMeans2D**-functie is toegepast en het diagram is gekleurd op basis van dimensie.

Houd er rekening mee dat de Qlik Sense K-meansfunctie ondersteuning biedt voor automatische clustering met behulp van de methode diepteverschil (DeD). Als een gebruiker het aantal clusters instelt op 0, wordt een optimaal aantal clusters voor die gegevensverzameling bepaald. Voor dit voorbeeld is er een variabele gemaakt voor het argument **num\_clusters** (raadpleeg *KMeans2D - diagramfunctie (page 1380)* voor de syntaxis). Het gewenste aantal clusters (k=5) wordt opgegeven per variabele.

1. Er wordt een **spreidingsplot**-diagram naar het werkblad gesleept. Dit wordt *Distributeurs (per dimensie)* genoemd.
2. Er wordt een **variabele** gemaakt om het aantal clusters op te geven. De **variabele** krijgt de naam *vDistClusters*. Voor de variabele wordt voor **Definitie** =5 ingevoerd.
3. De **gegevens**configuratie voor het diagram:
  - a. Onder **Dimensies** is *id* geselecteerd voor het veld voor **Bel**. *Cluster-id* is ingevoerd voor het **label**.
  - b. Onder **Metingen**, is *Avg([latitude])* de uitdrukking voor de **X-as**.
  - c. Onder **Metingen**, is *Avg([longitude])* de uitdrukking voor de **Y-as**.
4. Configuratie voor **Uiterlijk**:
  - a. Onder **Kleuren en legenda** is **Aangepast** geselecteerd voor **Kleuren**.
  - b. **Per dimensie** is geselecteerd voor het kleuren van het diagram.
  - c. De volgende uitdrukking is ingevoerd: *=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1, 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5')*
  - d. Het selectievakje **Kleuren bij selectie behouden** is geselecteerd.

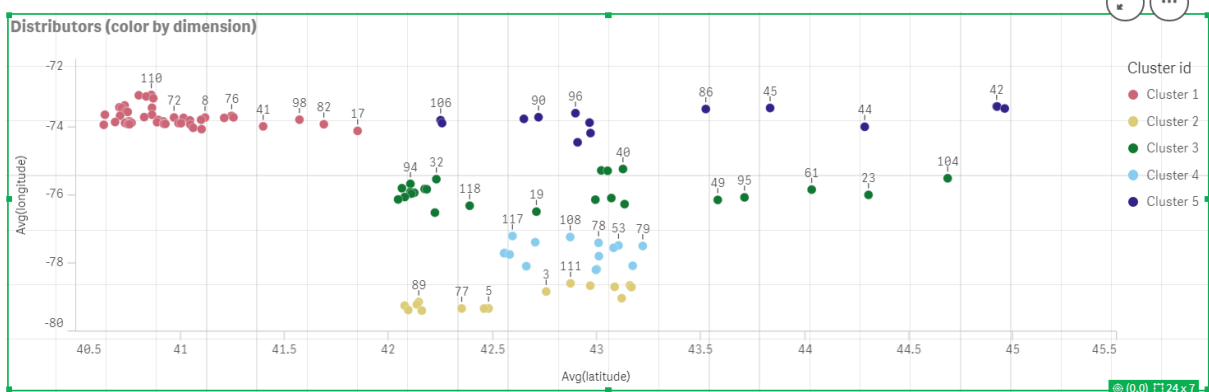
### Spreidingsplot vóór K-means kleuren per dimensie is toegepast

Distribution cluster analysis



### Spreidingsplot na K-means kleuren per dimensie is toegepast

Distribution cluster analysis



### Een tabel toevoegen: *Distributeurs*

Het kan handig zijn om een tabel bij de hand te hebben voor snelle toegang tot relevante gegevens. Het **spreidingsplot**-diagram toont weliswaar de *id's*, maar ter referentie is een tabel met de namen van de overeenkomstige distributeurs toegevoegd.

1. Er wordt een **tabel** met de naam *Distributeurs* naar het werkblad gesleept, waaraan de volgende **Kolommen** (dimensies) zijn toegevoegd: *id*, *voornaam* en *achternaam*.

Tabel: Namen distributeurs

Distributors			
id	first_name	last_name	
1	Kaiya	Snow	
2	Dean	Roy	
3	Eden	Paul	
4	Bryanna	Higgins	
5	Elisabeth	Lee	
6	Skylar	Robinson	
7	Cody	Bailey	
8	Dario	Sims	
9	Deacon	Hood	

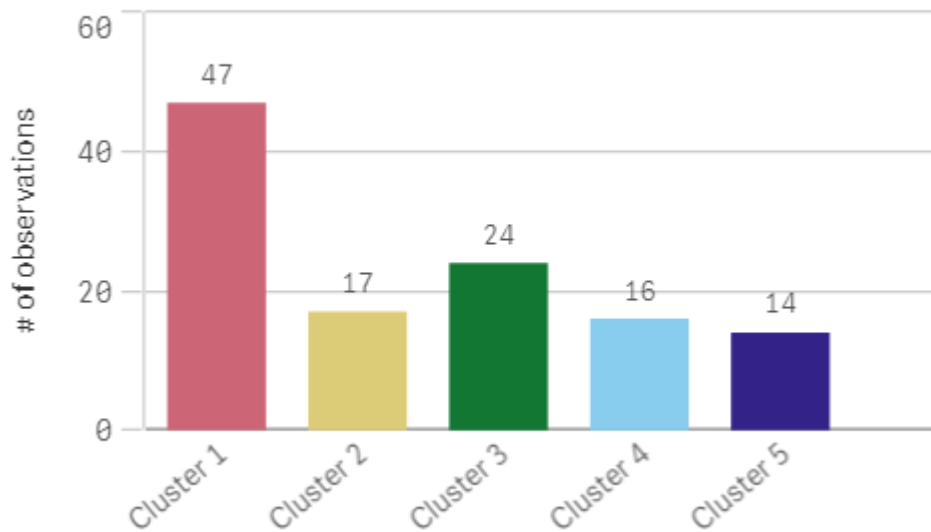
### Toevoegen van een **staafdiagram**: # observaties per cluster

Voor het scenario magazijndistributie is het nuttig te weten hoeveel distributeurs door elk magazijn zullen worden bediend. Daarom wordt er een **staafdiagram** aangemaakt, dat meet hoeveel distributeurs aan elk cluster zijn toegewezen.

1. Er wordt een **staafdiagram** naar het werkblad gesleept. Het diagram krijgt de naam: *# observaties per cluster*.
2. **Gegevens**configuratie voor het **staafdiagram**:
  - a. Er wordt een **dimensie** met het label *Clusters* toegevoegd (het label kan worden toegevoegd nadat de uitdrukking is toegepast). De volgende uitdrukking wordt ingevoerd: `=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1, 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5')`
  - b. De **Meting** met het label *# observaties* wordt toegevoegd. De volgende uitdrukking wordt ingevoerd: `=count(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id))`
3. Configuratie voor **Uiterlijk**:
  - a. Onder **Kleuren en legenda** is **Aangepast** geselecteerd voor **Kleuren**.
  - b. **Per dimensie** is geselecteerd voor het kleuren van het diagram.
  - c. De volgende uitdrukking wordt ingevoerd: `=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1, 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5')`
  - d. Het selectievakje **Kleuren bij selectie behouden** is geselecteerd.
  - e. **Legenda tonen** is uitgeschakeld.
  - f. Onder **Presentatie** staat **Waardelabels** ingesteld op **Auto**.
  - g. Onder de **x-as** is: bij **Clusters** is **Alleen labels** geselecteerd.

Staafdiagram: # observaties per cluster

### # observations per cluster



### Toepassen van de **Centroid2D**-functie

Er wordt een tweede tabel toegevoegd voor de **Centroid2D**-functie waarmee de coördinaten voor potentiële magazijnlocaties worden bepaald. Deze tabel toont de centrale locatie (middenpuntwaarden) voor de vijf geïdentificeerde distributeurgroepen.

1. Sleep een **Tabel** naar het werkblad, noem het *Cluster middenpunten* en voeg de volgende kolommen toe:
  - a. Er wordt een **dimensie** met het label *Clusters* toegevoegd. De volgende uitdrukking wordt ingevoerd: `=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1,'Warehouse 1','Warehouse 2','Warehouse 3','Warehouse 4','Warehouse 5')`
  - b. Er wordt een **Meting** met het label *breedtegraad (D1)* toegevoegd. De volgende uitdrukking wordt ingevoerd: `=only(aggr(KMeansCentroid2D(vDistClusters,0,only(latitude),only(longitude)),id))`  
 Let op: de parameter **coordinate\_no** komt overeen met de eerste dimensie(0). In dit geval wordt de dimensie *breedtegraad* uitgezet tegen de x-as. Als we met de **CentroidND**-functie zouden werken en er zouden tot zes dimensies zijn, dan zouden deze parametergegevens om het even welke van zes waarden kunnen zijn: 0, 1, 2, 3, 4 of 5.
  - c. Er wordt een **Meting** met het label *lengtegraad (D2)* toegevoegd. De volgende uitdrukking wordt ingevoerd: `=only(aggr(KMeansCentroid2D(vDistClusters,1,only(latitude),only(longitude)),id))`  
 De parameter **coordinate\_no** in deze uitdrukking komt overeen met de tweede dimensie(1). De dimensie *lengtegraad* wordt uitgezet tegen de y-as.

Tabel: Berekeningen middenpuntcluster

Cluster centroids		
Clusters	latitude (D1)	longitude (D2)
<b>Totals</b>	-	-
Warehouse 1	40.945422240426	-73.719966482979
Warehouse 2	42.590538729412	-79.067889217647
Warehouse 3	42.805089516667	-75.901621883333
Warehouse 4	42.8581692625	-77.6800485875
Warehouse 5	43.436770771429	-73.734622635714

## Toewijzen van middenpunten

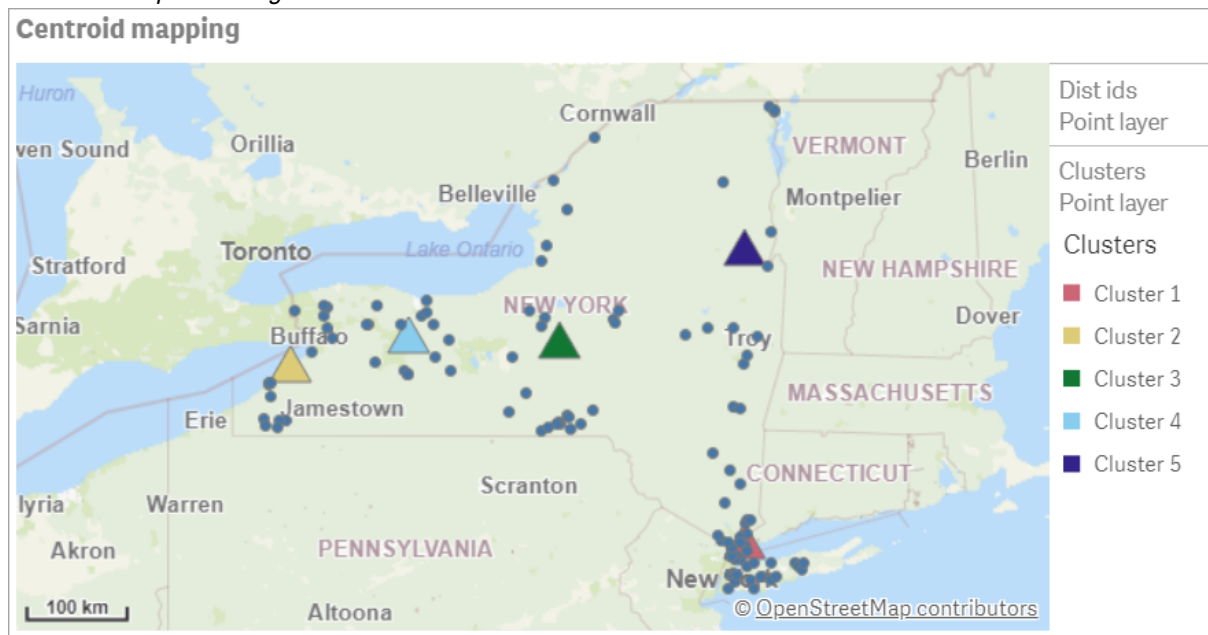
De volgende stap bestaat uit het toewijzen van de middenpunten. Het is aan de ontwikkelaar van de app of hij de visualisatie op afzonderlijke werkbladen wenst te plaatsen.

1. Er wordt een **kaart** met de naam *Toewijzen van middenpunten* naar het werkblad gesleept.
2. In de sectie **Lagen**, **Laag toevoegen** wordt geselecteerd, vervolgens wordt **Puntlaag** geselecteerd.
  - a. Het **Veld** *id* wordt geselecteerd en het **Label** voor de *Dist id*'s wordt toegevoegd.
  - b. In de sectie **Locatie** wordt het selectievakje voor de **velden Lengte- en breedtegraad** geselecteerd.
  - c. Voor **Breedtegraad** wordt het veld *breedtegraad* geselecteerd.
  - d. Voor **Lengtegraad** wordt het veld *lengtegraad* geselecteerd.
  - e. In de sectie **Grootte en vorm** wordt **Bel** geselecteerd voor **Vorm** en de **Grootte** wordt naar wens verkleind met de schuif.
  - f. In de sectie **Kleuren** wordt **Enkele kleur** geselecteerd en blauw wordt geselecteerd voor de **Kleur** en grijs voor de **Omtrek** (ook deze keuzes mag u zelf bepalen).
3. In de sectie **Lagen** wordt een tweede **Puntlaag** toegevoegd door eerst **Laag toevoegen** en dan **Puntlaag** te selecteren.
  - a. De volgende uitdrukking wordt ingevoerd: `=aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)`
  - b. Het **Label** *Clusters* wordt toegevoegd.
  - c. In de sectie **Locatie** wordt het selectievakje voor de **velden Lengte- en breedtegraad** geselecteerd.
  - d. Voor de **Breedtegraad**, die in dit geval tegen de x-as wordt uitgezet, wordt de volgende uitdrukking toegevoegd: `=aggr(KMeansCentroid2D(vDistClusters,0,only(latitude),only(longitude)),id)`
  - e. Voor de **Lengtegraad** die in dit geval langs de y-as wordt uitgezet, wordt de volgende uitdrukking toegevoegd: `=aggr(KMeansCentroid2D(vDistClusters,1,only(latitude),only(longitude)),id)`



- f. In de sectie **Grootte en vorm** wordt **Driehoek** geselecteerd voor **Vorm** en de **Grootte** wordt naar wens verkleind met de schuif.
  - g. Onder **Kleuren en legenda** is **Aangepast** geselecteerd voor **Kleuren**.
  - h. **Per dimensie** is geselecteerd voor het kleuren van het diagram. De volgende uitdrukking wordt ingevoerd: `=pick(aggr(KMeans2D(vDistClusters,only(latitude),only(longitude)),id)+1, 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5')`
  - i. De dimensie krijgt het label *Clusters*.
4. In **Kaartinstellingen** wordt **Adaptief** geselecteerd voor **Projectie**. **Metrisch** wordt geselecteerd voor **Maateenheden**.

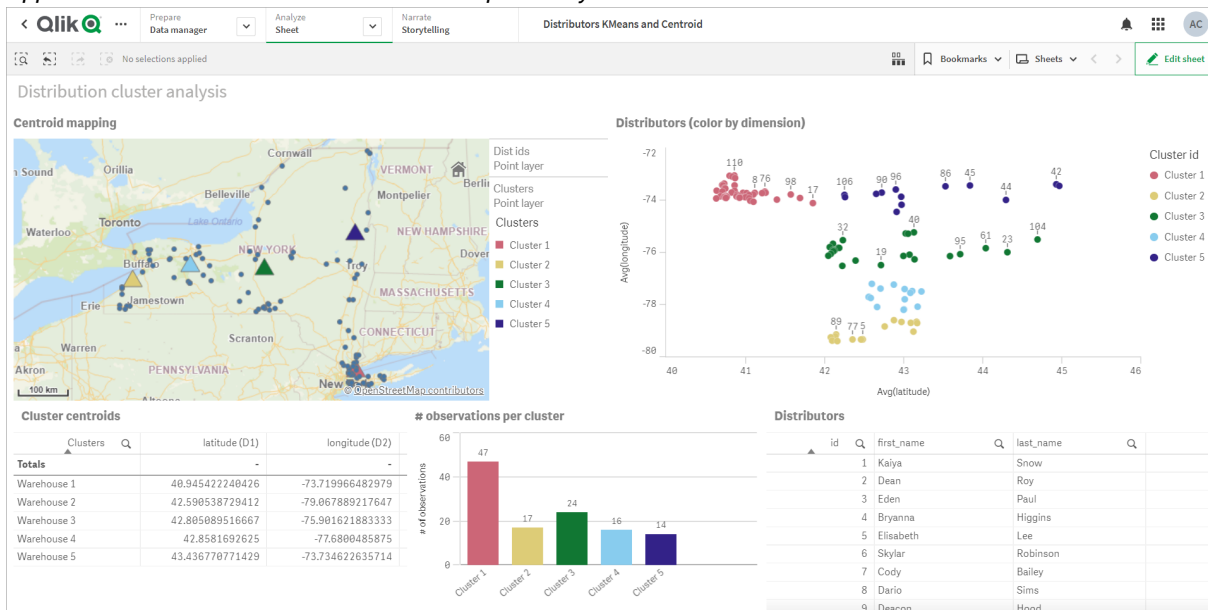
Kaart: Middenpunten toegewezen naar cluster



### Conclusie

Met behulp van de K-means-functie zijn in dit praktijkscenario distributeurs gesegmenteerd in soortgelijke groepen of clusters op basis van hun overeenkomsten; in dit geval de mate van onderlinge nabijheid. Op deze clusters is de middenpuntfunctie toegepast om vijf kaartcoördinaten te bepalen. Deze coördinaten geven een eerste centrale locatie aan voor de bouw of de vestiging van magazijnen. De middenpuntfunctie wordt toegepast op het kaartdiagram, zodat app-gebruikers kunnen zien waar de middenpunten zich ten opzichte van de omliggende clustergegevenspunten bevinden. De resulterende coördinaten staan voor potentiële locaties van magazijnen die ervoor kunnen zorgen dat de leveringskosten voor distributeurs in de staat New York kunnen worden gedrukt.

## App: Voorbeeld van een K-means- en middenpuntanalyse



## Distributor-gegevensverzameling: Uitgelijnd laden voor de editor bedoeld voor het laden van gegevens in Qlik Sense

```
DistributorData: Load * Inline [ id,first_name,last_
name,telephone,address,city,state,zip,latitude,longitude 1,Kaiya,Snow,(716) 201-1212,6231
Tonawanda Creek Rd #APT 308,Lockport,NY,14094,43.08926,-78.69313 2,Dean,Roy,(716) 201-
1588,6884 E High St,Lockport,NY,14094,43.16245,-78.65036 3,Eden,Paul,(716) 202-4596,4647
Southwestern Blvd #APT 350,Hamburg,NY,14075,42.76003,-78.83194 4,Bryanna,Higgins,(716) 203-
7041,418 Park Ave,Dunkirk,NY,14048,42.48279,-79.33088 5,Elisabeth,Lee,(716) 203-7043,36 E
Courtney St,Dunkirk,NY,14048,42.48299,-79.31928 6,Skylar,Robinson,(716) 203-7166,26 Greco
Ln,Dunkirk,NY,14048,42.4612095,-79.3317925 7,Cody,Bailey,(716) 203-7201,114 Lincoln
Ave,Dunkirk,NY,14048,42.4801269,-79.322232 8,Dario,Sims,(408) 927-1606,N Castle
Dr,Armonk,NY,10504,41.11979,-73.714864 9,Deacon,Hood,(410) 244-6221,4856 44th
St,Woodside,NY,11377,40.748372,-73.905445 10,Zackery,Levy,(410) 363-8874,61 Executive
Blvd,Farmingdale,NY,11735,40.7197457,-73.430239 11,Rey,Hawkins,(412) 344-8687,4585 Shimerville
Rd,Clarence,NY,14031,42.972075,-78.6592452 12,Phillip,Howard,(413) 269-4049,464 Main St
#101,Port Washington,NY,11050,40.8273756,-73.7009971 13,Shirley,Tyler,(434) 985-8943,114 Glann
Rd,Apalachin,NY,13732,42.0482515,-76.1229725 14,Aniyah,Jarvis,(440) 244-1808,87 N Middletown
Rd,Pearl River,NY,10965,41.0629,-74.0159 15,Alayna,Woodard,(478) 335-3704,70 W Red Oak Ln,West
Harrison,NY,10604,41.0162722,-73.7234926 16,Jermaine,Lambert,(508) 561-9836,24 Kellogg Rd,New
Hartford,NY,13413,43.0555739,-75.2793197 17,Harper,Gibbs,(239) 466-0238,Po Box
33,Cottekill,NY,12419,41.853392,-74.106082 18,Osvaldo,Graham,(252) 246-0816,6878 Sand Hill
Rd,East Syracuse,NY,13057,43.073215,-76.081448 19,Roberto,Wade,(270) 469-1211,3936 Holley
Rd,Moravia,NY,13118,42.713044,-76.481227 20,Kate,Mcguire,(270) 788-3080,6451 State 64 Rte
#3,Naples,NY,14512,42.707366,-77.380489 21,Dale,Andersen,(281) 480-5690,205 W Service
Rd,Champlain,NY,12919,44.9645392,-73.4470831 22,Lorelai,Burch,(302) 644-2133,1 Brewster
St,Glen Cove,NY,11542,40.865177,-73.633019 23,Amiyah,Flowers,(303) 223-0055,46600 Us
Interstate 81 Rte,Alexandria Bay,NY,13607,44.309626,-75.988365 24,Mckinley,Clements,(303) 918-
3230,200 Summit Lake Dr,Valhalla,NY,10595,41.101145,-73.778298 25,Marc,Gibson,(607) 203-
1233,25 Robinson St,Binghamton,NY,13901,42.107416,-75.901614 26,Kali,Norman,(607) 203-1400,1
Ely Park Blvd #APT 15,Binghamton,NY,13905,42.125866,-75.925026 27,Laci,Cain,(607) 203-1437,16
```

## 5 Script- en diagramfuncties

---

Zimmer Road, Kirkwood, NY, 13795, 42.066516, -75.792627 28, Mohammad, Perez, (607) 203-1652, 71  
Endicott Ave #APT 12, Johnson City, NY, 13790, 42.111894, -75.952187 29, Izabelle, Pham, (607) 204-  
0392, 434 State 369 Rte, Port Crane, NY, 13833, 42.185838, -75.823074 30, Kiley, Mays, (607) 204-  
0870, 244 Ballyhack Rd #14, Port Crane, NY, 13833, 42.175612, -75.814917 31, Peter, Trevino, (607) 205-  
1374, 125 Melbourne St., Vestal, NY, 13850, 42.080254, -76.051124 32, Ani, Francis, (607) 208-4067, 48  
Caswell St, Afton, NY, 13730, 42.232065, -75.525674 33, Jared, Sheppard, (716) 386-3002, 4709 430th  
Rte, Bemus Point, NY, 14712, 42.162175, -79.39176 34, Dulce, Atkinson, (914) 576-2266, 501 Pelham  
Rd, New Rochelle, NY, 10805, 40.895449, -73.782602 35, Jayla, Beasley, (716) 526-1054, 5010 474th  
Rte, Ashville, NY, 14710, 42.096859, -79.375561 36, Dane, Donovan, (718) 545-3732, 5014 31st  
Ave, Woodside, NY, 11377, 40.756967, -73.909506 37, Brendon, Clay, (585) 322-7780, 133 Cummings  
Ave, Gainesville, NY, 14066, 42.664309, -78.085651 38, Asia, Nunez, (718) 426-1472, 2407 Gilmore , East  
Elmhurst, NY, 11369, 40.766662, -73.869185 39, Dawson, Odonnell, (718) 342-2179, 5019 H  
Ave, Brooklyn, NY, 11234, 40.633245, -73.927591 40, Kyle, Collins, (315) 733-7078, 502 Rockhaven  
Rd, Utica, NY, 13502, 43.129184, -75.226726 41, Eliza, Hardin, (315) 331-8072, 502 Sladen Place, West  
Point, NY, 10996, 41.3993, -73.973003 42, Kasen, Klein, (518) 298-4581, 2407 Lake Shore  
Rd, Chazy, NY, 12921, 44.925561, -73.387373 43, Reuben, Bradford, (518) 298-4581, 33 Lake Flats  
Dr, Champlain, NY, 12919, 44.928092, -73.387884 44, Henry, Grimes, (518) 523-3990, 2407 Main St, Lake  
Placid, NY, 12946, 44.291487, -73.98474 45, Kyan, Livingston, (518) 585-7364, 241 Alexandria  
Ave, Ticonderoga, NY, 12883, 43.836553, -73.43155 46, Kaitlyn, Short, (516) 678-3189, 241 Chance  
Dr, Oceanside, NY, 11572, 40.638534, -73.63079 47, Damaris, Jacobs, (914) 664-5331, 241 Claremont  
Ave, Mount Vernon, NY, 10552, 40.919852, -73.827848 48, Alivia, Schroeder, (315) 469-4473, 241  
Lafayette Rd, Syracuse, NY, 13205, 42.996446, -76.12957 49, Bridget, Strong, (315) 298-4355, 241 Maltby  
Rd, Pulaski, NY, 13142, 43.584966, -76.136317 50, Francis, Lee, (585) 201-7021, 166 Ross  
St, Batavia, NY, 14020, 43.0031502, -78.17487 51, Makaila, Phelps, (585) 201-7422, 58 S Main  
St, Batavia, NY, 14020, 42.99941, -78.1939285 52, Jazlynn, Stephens, (585) 203-1087, 1 Sinclair  
Dr, Pittsford, NY, 14534, 43.084157, -77.545452 53, Ryann, Randolph, (585) 203-1519, 331 Eaglehead  
Rd, East Rochester, NY, 14445, 43.10785, -77.475552 54, Rosa, Baker, (585) 204-4011, 42 Ossian  
St, Dansville, NY, 14437, 42.560761, -77.70088 55, Marcel, Barry, (585) 204-4013, 42 Jefferson  
St, Dansville, NY, 14437, 42.557735, -77.702983 56, Dennis, Schmitt, (585) 204-4061, 750 Dansville  
Mount Morris Rd, Dansville, NY, 14437, 42.584458, -77.741648 57, Cassandra, Kim, (585) 204-4138, 3  
Perine Ave APT1, Dansville, NY, 14437, 42.562865, -77.69661 58, Kolton, Jacobson, (585) 206-5047, 4925  
Upper Holly Rd, Holley, NY, 14470, 43.175957, -78.074465 59, Nathanael, Donovan, (718) 393-3501, 9604  
57th Ave, Corona, NY, 11373, 40.736077, -73.864858 60, Robert, Frazier, (718) 271-3067, 300 56th  
Ave, Corona, NY, 11373, 40.735304, -73.873997 61, Jessie, Mora, (315) 405-8991, 9607 Forsyth  
Loop, Watertown, NY, 13603, 44.036466, -75.833437 62, Martha, Rollins, (347) 242-2642, 22 Main  
St, Corona, NY, 11373, 40.757727, -73.829331 63, Emely, Townsend, (718) 699-0751, 60 Sanford  
Ave, Corona, NY, 11373, 40.755466, -73.831029 64, Kylie, Cooley, (347) 561-7149, 9608 95th Ave, Ozone  
Park, NY, 11416, 40.687564, -73.845715 65, Wendy, Cameron, (585) 571-4185, 9608 Union  
St, Scottsville, NY, 14546, 43.013327, -77.7907839 66, Kayley, Peterson, (718) 654-5027, 961 E 230th  
St, Bronx, NY, 10466, 40.889275, -73.850555 67, Camden, Ochoa, (718) 760-8699, 59 Vark  
St, Yonkers, NY, 10701, 40.929322, -73.89957 68, Priscilla, Castillo, (910) 326-7233, 9359 Elm  
St, Chadwicks, NY, 13319, 43.024902, -75.26886 69, Dana, Schultz, (913) 322-4580, 99 Washington  
Ave, Hastings on Hudson, NY, 10706, 40.99265, -73.879748 70, Blaze, Medina, (914) 207-0015, 60 Elliott  
Ave, Yonkers, NY, 10705, 40.921498, -73.896682 71, Finnegan, Tucker, (914) 207-0015, 90 Hillside  
Drive, Yonkers, NY, 10705, 40.922514, -73.892911 72, Pranav, Palmer, (914) 214-8376, 5 Bruce  
Ave, Harrison, NY, 10528, 40.970916, -73.711493 73, Kolten, Wong, (914) 218-8268, 70 Barker St, Mount  
Kisco, NY, 10549, 41.211993, -73.723202 74, Jasiah, Vazquez, (914) 231-5199, 30 Broadway, Dobbs  
Ferry, NY, 10522, 41.004629, -73.879825 75, Lamar, Pierce, (914) 232-0380, 68 Ridge  
Rd, Katonah, NY, 10536, 41.256662, -73.707964 76, Carla, Coffey, (914) 232-0469, 197 Beaver Dam  
Rd, Katonah, NY, 10536, 41.247934, -73.664363 77, Brooklynn, Harmon, (716) 595-3227, 8084 Glasgow  
Rd, Cassadega, NY, 14718, 42.353861, -79.329558 78, Raquel, Hodges, (585) 398-8125, 809 County Road  
, Victor, NY, 14564, 43.011745, -77.398806 79, Jeremiah, Gardner, (585) 787-9127, 809 Houston  
Rd, Webster, NY, 14580, 43.224204, -77.491353 80, Clarence, Hammond, (720) 746-1619, 809 Pierpont  
Ave, Piermont, NY, 10968, 41.0491181, -73.918622 81, Rhys, Gill, (518) 427-7887, 81 Columbia  
St, Albany, NY, 12210, 42.652824, -73.752096 82, Edith, Parrish, (845) 452-7621, 81 Glenwood  
Ave, Poughkeepsie, NY, 12603, 41.691058, -73.910829 83, Kobe, Mcintosh, (845) 371-1101, 81 Heitman

Dr, Spring Valley, NY, 10977, 41.103227, -74.054396 84, Ayden, Waters, (516) 796-2722, 81 Kingfisher Rd, Levittown, NY, 11756, 40.738939, -73.52826 85, Francis, Rogers, (631) 427-7728, 81 Knollwood Ave, Huntington, NY, 11743, 40.864905, -73.426107 86, Jaden, Landry, (716) 496-4038, 12839 39th Rte, Chaffee, NY, 14030, 43.527396, -73.462786 87, Giancarlo, Campos, (518) 885-5717, 1284 Saratoga Rd, Ballston Spa, NY, 12020, 42.968594, -73.862847 88, Eduardo, Contreras, (716) 285-8987, 1285 Saunders Sett Rd, Niagara Falls, NY, 14305, 43.122963, -79.029274 89, Gabriela, Davidson, (716) 267-3195, 1286 Mee Rd, Falconer, NY, 14733, 42.147339, -79.137976 90, Evangeline, Case, (518) 272-9435, 1287 2nd Ave, Watervliet, NY, 12189, 42.723132, -73.703818 91, Tyrone, Ellison, (518) 843-4691, 1287 Midline Rd, Amsterdam, NY, 12010, 42.9730876, -74.1700608 92, Bryce, Bass, (518) 943-9549, 1288 Leeds Athens Rd, Athens, NY, 12015, 42.259381, -73.876897 93, Londyn, Butler, (518) 922-7095, 129 Argersinger Rd, Fultonville, NY, 12072, 42.910969, -74.441917 94, Graham, Becker, (607) 655-1318, 129 Baker Rd, Windsor, NY, 13865, 42.107271, -75.66408 95, Rolando, Fitzgerald, (315) 465-4166, 17164 County 90 Rte, Mannsville, NY, 13661, 43.713443, -76.06232 96, Grant, Hoover, (518) 692-8363, 1718 County 113 Rte, Schaghticote, NY, 12154, 42.900648, -73.585036 97, Mark, Goodwin, (631) 584-6761, 172 Cambon Ave, Saint James, NY, 11780, 40.871152, -73.146032 98, Deacon, Cantu, (845) 221-7940, 172 Carpenter Rd, Hopewell Junction, NY, 12533, 41.57388, -73.77609 99, Tristian, Walsh, (516) 997-4750, 172 E Cabot Ln, Westbury, NY, 11590, 40.7480397, -73.54819 100, Abram, Alexander, (631) 588-3817, 172 Lorenzo Cir, Ronkonkoma, NY, 11779, 40.837123, -73.09367 101, Lesly, Bush, (516) 489-3791, 172 Nassau Blvd, Garden City, NY, 11530, 40.71147, -73.660753 102, Pamela, Espinoza, (716) 201-1520, 172 Niagara St, Lockport, NY, 14094, 43.169871, -78.70093 103, Bryanna, Newton, (914) 328-4332, 172 Warren Ave, White Plains, NY, 10603, 41.047207, -73.79572 104, Marcelo, Schmitt, (315) 393-4432, 319 Mansion Ave, Ogdensburg, NY, 13669, 44.690246, -75.49992 105, Layton, Valenzuela, (631) 676-2113, 319 Singingwood Dr, Holbrook, NY, 11741, 40.801391, -73.058993 106, Roderick, Rocha, (518) 671-6037, 319 Warren St, Hudson, NY, 12534, 42.252527, -73.790629 107, Camryn, Terrell, (315) 635-1680, 3192 Olive Dr, Baldwinsville, NY, 13027, 43.136843, -76.260303 108, Summer, Callahan, (585) 394-4195, 3192 Smith Road, Canandaigua, NY, 14424, 42.875457, -77.228039 109, Pierre, Novak, (716) 665-2524, 3194 Falconer Kimball Stand Rd, Falconer, NY, 14733, 42.138439, -79.211091 110, Kennedy, Fry, (315) 543-2301, 32 College Rd, Selden, NY, 11784, 40.861624, -73.04757 111, Wyatt, Pruitt, (716) 681-4042, 277 Ransom Rd, Lancaster, NY, 14086, 42.87702, -78.591302 112, Lilly, Jensen, (631) 841-0859, 2772 Schliegel Blvd, Amityville, NY, 11701, 40.708021, -73.413015 113, Tristin, Hardin, (631) 920-0927, 278 Fulton Street, West Babylon, NY, 11704, 40.733578, -73.357321 114, Tanya, Stafford, (716) 484-0771, 278 Sampson St, Jamestown, NY, 14701, 42.0797, -79.247805 115, Paris, Cordova, (607) 589-4857, 278 Washburn Rd, Spencer, NY, 14883, 42.225046, -76.510257 116, Alfonso, Morse, (718) 359-5582, 200 Colden St, Flushing, NY, 11355, 40.750403, -73.822752 117, Maurice, Hooper, (315) 595-6694, 4435 Italy Hill Rd, Branchport, NY, 14418, 42.597957, -77.199267 118, Iris, Wolf, (607) 539-7288, 444 Harford Rd, Brooktondale, NY, 14817, 42.392164, -76.30756 ];

### KMeans2D - diagramfunctie

**KMeans2D()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de cluster-id weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter `coordinate_1` respectievelijk `coordinate_2`. Dit zijn beide aggregaties. De parameter `num_clusters` bepaalt het aantal clusters dat wordt gemaakt. Gegevens kunnen optioneel worden genormaliseerd door de `normparameter`.

**KMeans2D** retourneert één waarde per gegevenspunt. De geretourneerde waarde is een dubbele waarde en is het gehele getal dat overeenkomt met het cluster waaraan elk gegevenspunt is toegewezen.

#### Syntaxis:

```
KMeans2D(num_clusters, coordinate_1, coordinate_2 [, norm])
```

**Retourgegevenstypen:** dual

**Argumenten:**

### Argumenten

Argument	Beschrijving
num_clusters	Het geheel getal dat het aantal clusters aangeeft.
coordinate_1	De aggregatie die de eerste coördinaat berekent, meestal de x-as van de verdelingsplot die op basis van het diagram kan worden gemaakt. De aanvullende parameter, coordinate_2, berekent de tweede coördinaat.
norm	<p>De optionele normalisatiemethode die is toegepast op gegevensverzamelingen voordat K-means clustering wordt toegepast.</p> <p>Mogelijke waarden:</p> <ul style="list-style-type: none"><li>0 of 'geen' voor geen normalisatie</li><li>1 of 'zscore' for z-score normalisatie</li><li>2 of 'minmax' voor min-max normalisatie</li></ul> <p>Als er geen parameter is opgegeven of de opgegeven parameter is onjuist, wordt er geen normalisatie toegepast.</p> <p>De Z-score normaliseert gegevens op basis van de gemiddelde- en standaardafwijking. De Z-score zorgt er niet voor dat elke functie dezelfde schaal toebedeeld krijgt, maar het is een betere manier om aan te gaan met uitschieters dan de min-max-methode.</p> <p>De min-max normalisatie zorgt ervoor dat functies dezelfde schaal toebedeeld krijgen door gebruik te maken van de minimum- en maximumwaarden van elk gegevenspunt en elk gegevenspunt opnieuw te berekenen.</p>

Voorbeeld: Diagramuitdrukking

In dit voorbeeld maken we een verdelingsplot met behulp van de gegevensset *Iris* en gebruiken vervolgens KMeans om de gegevens te kleuren op uitdrukking.

Bovendien maken we een variabele voor het argument *num\_clusters* en gebruiken vervolgens een invoervak voor de variabele om het aantal clusters te wijzigen.

De gegevensset *Iris* is in verschillende indelingen openbaar toegankelijk. We hebben de gegevens ter beschikking gesteld als een uitgelijnde tabel die geladen kan worden met behulp van de editor voor laden van gegevens in Qlik Sense. Houd er rekening mee dat we in dit voorbeeld de kolom *id* hebben toegevoegd.

Nadat we de gegevens in Qlik Sense hebben geladen, gaan we als volgt te werk:

1. Sleep een **verdelingsplot** op een nieuw werkblad. Geef het *bloembladdiagram (kleuruitdrukking)* een naam.
2. Maak een variabele om het aantal clusters op te geven. Voor de variabele **Naam** voert u *KmeansPetalClusters* in. Voor de variabele **Definitie** voert u `=2` in.
3. Configureer **Uiterlijk** voor het diagram:
  - i. Onder **Dimensies** kiest u *id* voor het veld voor **Bel**. Voer de cluster-id in voor het label.
  - ii. Onder **Metingen** kiest u *Som([lengte.bloemenblad])* voor de uitdrukking voor **X-as**.
  - iii. Onder **Metingen** kiest u *Som([breedte.bloemenblad])* voor de uitdrukking voor **Y-as**.

*Gegevensinstellingen voor het bloembladdiagram (kleuruitdrukking)*

Data

**Dimensions**  
Bubble

Id > [grid icon]

Alternative dimensions

Add alternative

**Measures**  
X-axis

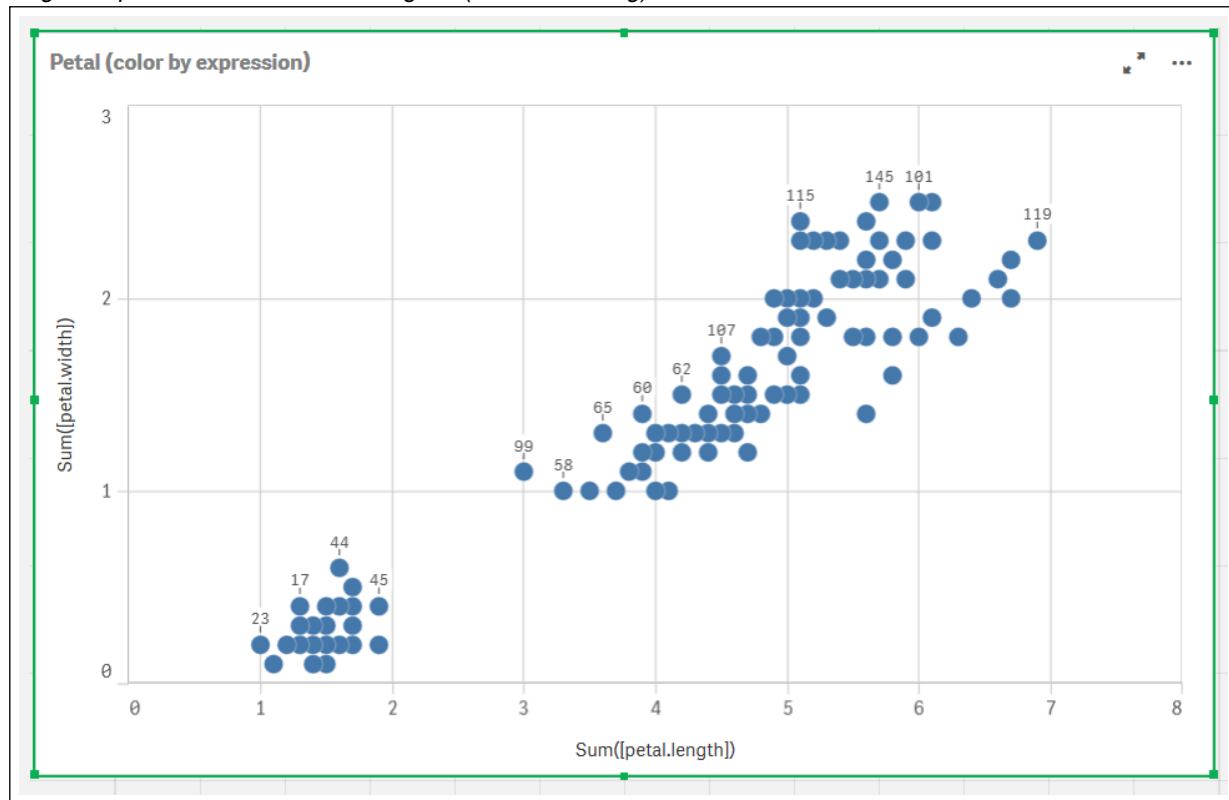
Sum	[petal.length]	>	[grid icon]
-----	----------------	---	-------------

Y-axis

Sum	[petal.width]	>	[grid icon]
-----	---------------	---	-------------

De gegevenspunten worden in het diagram geplot.

Gegevenspunten in het bloembladdiagram (kleuruitdrukking)



4. Configureer **Uiterlijk** voor het diagram:

- i. Onder **Kleuren en legenda** kiest u **Aangepast** voor **Kleuren**.
- ii. Kleur het diagram **Op uitdrukking**.
- iii. Voer het volgende in voor **Uitdrukking**: `kmeans2d($(KmeansPetalClusters), Sum([petal.length]), Sum([petal.width]))`  
 Houd er rekening mee dat `KmeansPetalClusters` de variabele is die we hebben ingesteld op 2.  
 U kunt ook het volgende invoeren: `kmeans2d(2, Sum([petal.length]), Sum([petal.width]))`
- iv. Schakel het selectievakje uit voor **De uitdrukking is een kleurcode**.

v. Voer het volgende in voor **Label**: *Cluster-id*



*Instellingen voor het uiterlijk van het bloembladdiagram (kleuruitdrukking)*

Appearance

▼ Colors and legend

Colors

Custom

By expression ▼

Expression

kmeans2d\$(KmeansPetalC *fx*)

The expression is a color code

Label

Cluster Id

Color scheme

Sequential gradient

Sequential classes

Diverging gradient

Diverging classes

Reverse colors

Range

Auto

Show legend

Auto

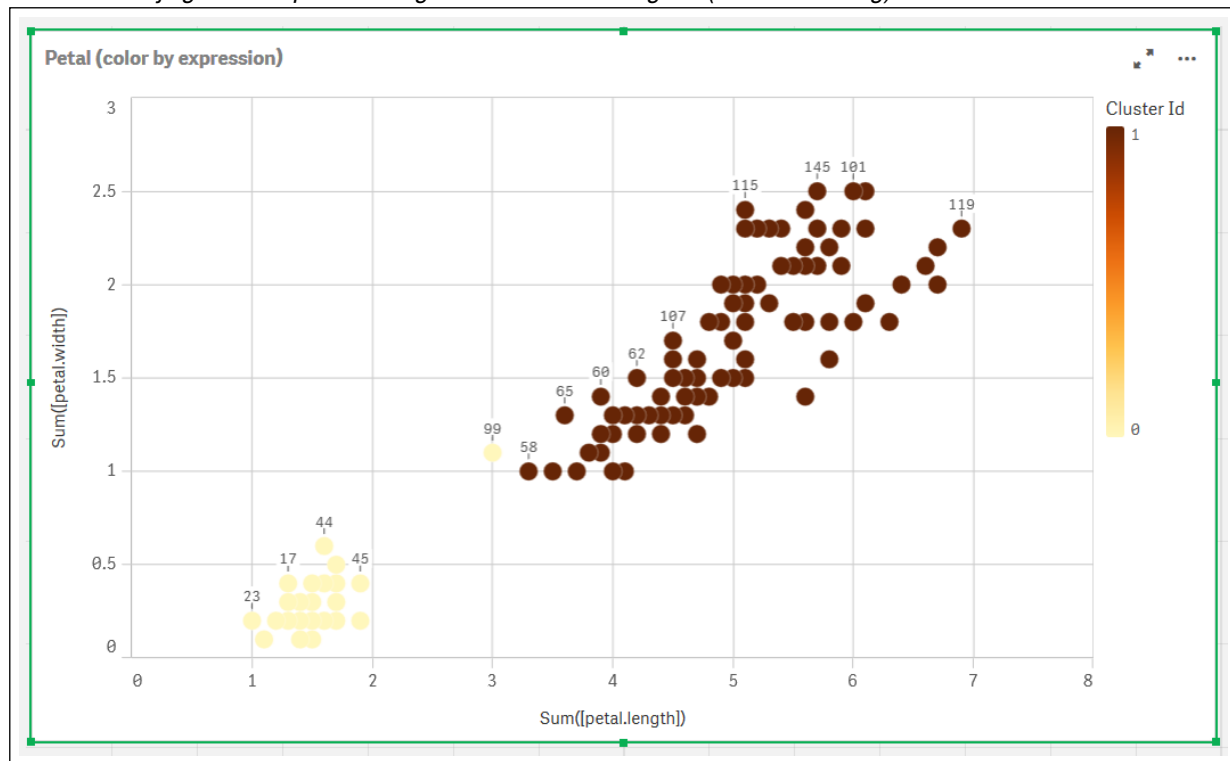
Legend position

Diagramfuncties - Qlik Sense, November 2022

Show legend title

## 5 Script- en diagramfuncties

De twee clusters van het diagram worden gekleurd door de uitdrukking KMeans.  
*Clusters die zijn gekleurd op uitdrukking in een bloembladdiagram (kleuruitdrukking)*



5. Voeg een **Invoervak voor variabelen** toe voor het aantal clusters.
  - i. Onder **Aangepaste objecten** in het **bedrijfsmiddelen**venster kiest u **Qlik Dashboard bundle**. Indien we geen toegang hadden tot de bundel dashboard, zouden we nog steeds in staat zijn om het aantal clusters te wijzigen dat de variabele gebruikt die we hebben gemaakt. Dit is ook direct mogelijk met behulp van een geheel getal in de uitdrukking.
  - ii. Sleep een **Invoervak voor variabelen** naar het werkblad.
  - iii. Onder **Uiterlijk** kiest u **Algemeen**.
  - iv. Voer het volgende in voor **Titel**: *Clusters*
  - v. Klik op **Variabele**.
  - vi. Kies de volgende variabele voor **Naam**: *KmeansPetalClusters*.
  - vii. Kies **Schuifregelaar** voor **Weergeven als**.

viii. Kies **Waarden** en configureer de instellingen zoals vereist.

*Uiterlijk voor het invoervak voor variabelen van clusters*

▼ General

Show titles  On

Title

Clusters	<i>fx</i>
----------	-----------

Subtitle

	<i>fx</i>
--	-----------

Footnote

	<i>fx</i>
--	-----------

Disable hover menu

▼ Variable

Name

KmeansPetalClusters	▼
---------------------	---

Show as

Slider	▼
--------	---

Update on drag

▼ Values

Min

2	<i>fx</i>
---	-----------

Max

10	<i>fx</i>
----	-----------

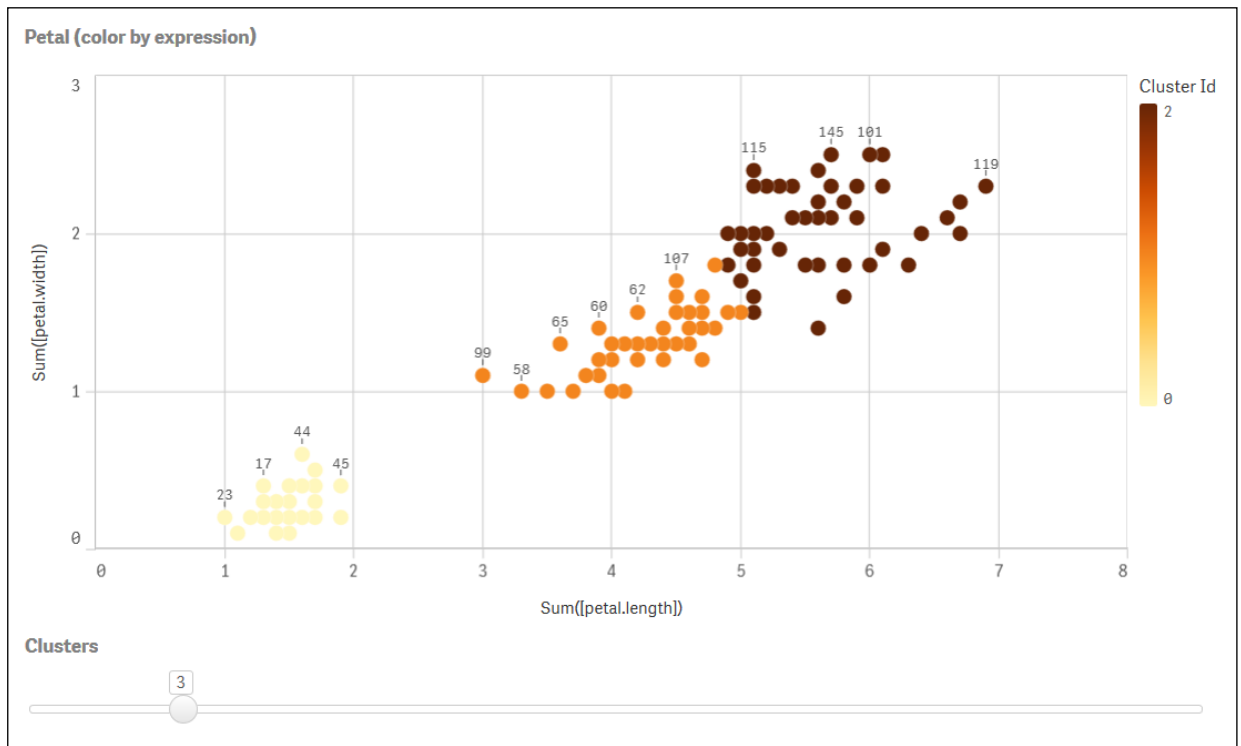
Step

1	<i>fx</i>
---	-----------

Slider label

Zodra we klaar zijn met bewerken, kunnen we het aantal clusters wijzigen met de schuifregelaar in het invoervak voor de variabele *Clusters*.

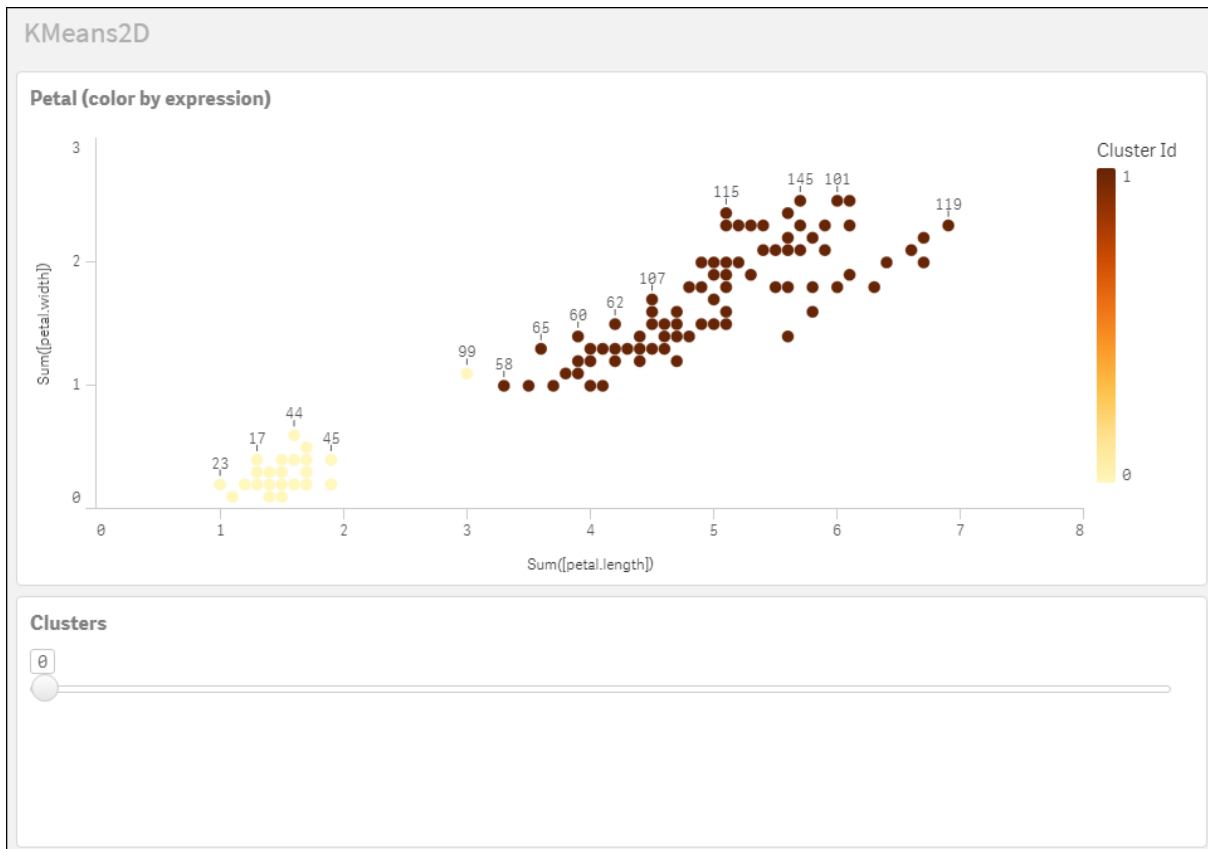
*Clusters die zijn gekleurd op uitdrukking in een bloembladdiagram (kleuruitdrukking)*



### Automatische clustering

De functies voor **KMeans** bieden ondersteuning voor automatische clustering met behulp van de methode diepteverschil (DeD). Als een gebruiker het aantal clusters instelt op 0, wordt een optimaal aantal clusters voor die gegevensverzameling bepaald. Houd er rekening mee dat een geheel getal voor het aantal clusters ( $k$ ) niet expliciet wordt geretourneerd, maar wel wordt berekend binnen het KMeans-algoritme. Als bijvoorbeeld 0 is opgegeven in de functie voor de waarde van *KmeansPetalClusters* of is ingesteld via een invoervak voor variabelen, worden clustertoewijzingen automatisch berekend voor de gegevensverzameling die is gebaseerd op een optimaal aantal clusters.

De diepteverschilmethode KMeans depth bepaalt het optimale aantal clusters als ( $k$ ) is ingesteld op 0



### Iris-gegevensverzameling: Uitgelijnde lading voor editor voor laden van gegevens in Qlik Sense

```
IrisData: Load * Inline [ sepal.length, sepal.width, petal.length, petal.width, variety, id
5.1, 3.5, 1.4, 0.2, Setosa, 1 4.9, 3, 1.4, 0.2, Setosa, 2 4.7, 3.2, 1.3, 0.2, Setosa, 3 4.6,
3.1, 1.5, 0.2, Setosa, 4 5, 3.6, 1.4, 0.2, Setosa, 5 5.4, 3.9, 1.7, 0.4, Setosa, 6 4.6, 3.4,
1.4, 0.3, Setosa, 7 5, 3.4, 1.5, 0.2, Setosa, 8 4.4, 2.9, 1.4, 0.2, Setosa, 9 4.9, 3.1, 1.5,
0.1, Setosa, 10 5.4, 3.7, 1.5, 0.2, Setosa, 11 4.8, 3.4, 1.6, 0.2, Setosa, 12 4.8, 3, 1.4,
0.1, Setosa, 13 4.3, 3, 1.1, 0.1, Setosa, 14 5.8, 4, 1.2, 0.2, Setosa, 15 5.7, 4.4, 1.5, 0.4,
Setosa, 16 5.4, 3.9, 1.3, 0.4, Setosa, 17 5.1, 3.5, 1.4, 0.3, Setosa, 18 5.7, 3.8, 1.7, 0.3,
Setosa, 19 5.1, 3.8, 1.5, 0.3, Setosa, 20 5.4, 3.4, 1.7, 0.2, Setosa, 21 5.1, 3.7, 1.5, 0.4,
Setosa, 22 4.6, 3.6, 1, 0.2, Setosa, 23 5.1, 3.3, 1.7, 0.5, Setosa, 24 4.8, 3.4, 1.9, 0.2,
Setosa, 25 5, 3, 1.6, 0.2, Setosa, 26 5, 3.4, 1.6, 0.4, Setosa, 27 5.2, 3.5, 1.5, 0.2, Setosa,
28 5.2, 3.4, 1.4, 0.2, Setosa, 29 4.7, 3.2, 1.6, 0.2, Setosa, 30 4.8, 3.1, 1.6, 0.2, Setosa,
31 5.4, 3.4, 1.5, 0.4, Setosa, 32 5.2, 4.1, 1.5, 0.1, Setosa, 33 5.5, 4.2, 1.4, 0.2, Setosa,
34 4.9, 3.1, 1.5, 0.1, Setosa, 35 5, 3.2, 1.2, 0.2, Setosa, 36 5.5, 3.5, 1.3, 0.2, Setosa, 37
4.9, 3.1, 1.5, 0.1, Setosa, 38 4.4, 3, 1.3, 0.2, Setosa, 39 5.1, 3.4, 1.5, 0.2, Setosa, 40 5,
3.5, 1.3, 0.3, Setosa, 41 4.5, 2.3, 1.3, 0.3, Setosa, 42 4.4, 3.2, 1.3, 0.2, Setosa, 43 5,
3.5, 1.6, 0.6, Setosa, 44 5.1, 3.8, 1.9, 0.4, Setosa, 45 4.8, 3, 1.4, 0.3, Setosa, 46 5.1,
3.8, 1.6, 0.2, Setosa, 47 4.6, 3.2, 1.4, 0.2, Setosa, 48 5.3, 3.7, 1.5, 0.2, Setosa, 49 5,
3.3, 1.4, 0.2, Setosa, 50 7, 3.2, 4.7, 1.4, versicolor, 51 6.4, 3.2, 4.5, 1.5, Versicolor, 52
6.9, 3.1, 4.9, 1.5, versicolor, 53 5.5, 2.3, 4, 1.3, Versicolor, 54 6.5, 2.8, 4.6, 1.5,
Versicolor, 55 5.7, 2.8, 4.5, 1.3, Versicolor, 56 6.3, 3.3, 4.7, 1.6, versicolor, 57 4.9, 2.4,
3.3, 1, Versicolor, 58 6.6, 2.9, 4.6, 1.3, Versicolor, 59 5.2, 2.7, 3.9, 1.4, versicolor, 60
5, 2, 3.5, 1, Versicolor, 61 5.9, 3, 4.2, 1.5, Versicolor, 62 6, 2.2, 4, 1, versicolor, 63
6.1, 2.9, 4.7, 1.4, versicolor, 64 5.6, 2.9, 3.6, 1.3, versicolor, 65 6.7, 3.1, 4.4, 1.4,
Versicolor, 66 5.6, 3, 4.5, 1.5, versicolor, 67 5.8, 2.7, 4.1, 1, versicolor, 68 6.2, 2.2,
```



4.5, 1.5, Versicolor, 69 5.6, 2.5, 3.9, 1.1, Versicolor, 70 5.9, 3.2, 4.8, 1.8, Versicolor, 71 6.1, 2.8, 4, 1.3, Versicolor, 72 6.3, 2.5, 4.9, 1.5, Versicolor, 73 6.1, 2.8, 4.7, 1.2, Versicolor, 74 6.4, 2.9, 4.3, 1.3, Versicolor, 75 6.6, 3, 4.4, 1.4, Versicolor, 76 6.8, 2.8, 4.8, 1.4, Versicolor, 77 6.7, 3, 5, 1.7, Versicolor, 78 6, 2.9, 4.5, 1.5, Versicolor, 79 5.7, 2.6, 3.5, 1, Versicolor, 80 5.5, 2.4, 3.8, 1.1, Versicolor, 81 5.5, 2.4, 3.7, 1, Versicolor, 82 5.8, 2.7, 3.9, 1.2, Versicolor, 83 6, 2.7, 5.1, 1.6, Versicolor, 84 5.4, 3, 4.5, 1.5, Versicolor, 85 6, 3.4, 4.5, 1.6, Versicolor, 86 6.7, 3.1, 4.7, 1.5, Versicolor, 87 6.3, 2.3, 4.4, 1.3, Versicolor, 88 5.6, 3, 4.1, 1.3, Versicolor, 89 5.5, 2.5, 4, 1.3, Versicolor, 90 5.5, 2.6, 4.4, 1.2, Versicolor, 91 6.1, 3, 4.6, 1.4, Versicolor, 92 5.8, 2.6, 4, 1.2, Versicolor, 93 5, 2.3, 3.3, 1, Versicolor, 94 5.6, 2.7, 4.2, 1.3, Versicolor, 95 5.7, 3, 4.2, 1.2, Versicolor, 96 5.7, 2.9, 4.2, 1.3, Versicolor, 97 6.2, 2.9, 4.3, 1.3, Versicolor, 98 5.1, 2.5, 3, 1.1, Versicolor, 99 5.7, 2.8, 4.1, 1.3, Versicolor, 100 6.3, 3.3, 6, 2.5, Virginica, 101 5.8, 2.7, 5.1, 1.9, Virginica, 102 7.1, 3, 5.9, 2.1, Virginica, 103 6.3, 2.9, 5.6, 1.8, Virginica, 104 6.5, 3, 5.8, 2.2, Virginica, 105 7.6, 3, 6.6, 2.1, Virginica, 106 4.9, 2.5, 4.5, 1.7, Virginica, 107 7.3, 2.9, 6.3, 1.8, Virginica, 108 6.7, 2.5, 5.8, 1.8, Virginica, 109 7.2, 3.6, 6.1, 2.5, Virginica, 110 6.5, 3.2, 5.1, 2, Virginica, 111 6.4, 2.7, 5.3, 1.9, Virginica, 112 6.8, 3, 5.5, 2.1, Virginica, 113 5.7, 2.5, 5, 2, Virginica, 114 5.8, 2.8, 5.1, 2.4, Virginica, 115 6.4, 3.2, 5.3, 2.3, Virginica, 116 6.5, 3, 5.5, 1.8, Virginica, 117 7.7, 3.8, 6.7, 2.2, Virginica, 118 7.7, 2.6, 6.9, 2.3, Virginica, 119 6, 2.2, 5, 1.5, Virginica, 120 6.9, 3.2, 5.7, 2.3, Virginica, 121 5.6, 2.8, 4.9, 2, Virginica, 122 7.7, 2.8, 6.7, 2, Virginica, 123 6.3, 2.7, 4.9, 1.8, Virginica, 124 6.7, 3.3, 5.7, 2.1, Virginica, 125 7.2, 3.2, 6, 1.8, Virginica, 126 6.2, 2.8, 4.8, 1.8, Virginica, 127 6.1, 3, 4.9, 1.8, Virginica, 128 6.4, 2.8, 5.6, 2.1, Virginica, 129 7.2, 3, 5.8, 1.6, Virginica, 130 7.4, 2.8, 6.1, 1.9, Virginica, 131 7.9, 3.8, 6.4, 2, Virginica, 132 6.4, 2.8, 5.6, 2.2, Virginica, 133 6.3, 2.8, 5.1, 1.5, Virginica, 134 6.1, 2.6, 5.6, 1.4, Virginica, 135 7.7, 3, 6.1, 2.3, Virginica, 136 6.3, 3.4, 5.6, 2.4, Virginica, 137 6.4, 3.1, 5.5, 1.8, Virginica, 138 6, 3, 4.8, 1.8, Virginica, 139 6.9, 3.1, 5.4, 2.1, Virginica, 140 6.7, 3.1, 5.6, 2.4, Virginica, 141 6.9, 3.1, 5.1, 2.3, Virginica, 142 5.8, 2.7, 5.1, 1.9, Virginica, 143 6.8, 3.2, 5.9, 2.3, Virginica, 144 6.7, 3.3, 5.7, 2.5, Virginica, 145 6.7, 3, 5.2, 2.3, Virginica, 146 6.3, 2.5, 5, 1.9, Virginica, 147 6.5, 3, 5.2, 2, Virginica, 148 6.2, 3.4, 5.4, 2.3, Virginica, 149 5.9, 3, 5.1, 1.8, Virginica, 150 ];

### KMeansND - diagramfunctie

**KMeansND()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de cluster-id weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter `coordinate_1` en `coordinate_2` etc. tot en met de `n`-kolommen Dit zijn beide aggregaties. De parameter `num_clusters` bepaalt het aantal clusters dat wordt gemaakt.

**KMeansND** retourneert één waarde per gegevenspunt. De geretourneerde waarde is een dubbele waarde en is het gehele getal dat overeenkomt met het cluster waaraan elk gegevenspunt is toegewezen.

#### Syntaxis:

```
KMeansND(num_clusters, num_iter, coordinate_1, coordinate_2 [,coordinate_3 [, ...]])
```

**Retourgegevenstypen:** dual

**Argumenten:**

Argumenten

Argument	Beschrijving
num_clusters	Het geheel getal dat het aantal clusters aangeeft.
num_iter	Het aantal iteraties voor clusters met opnieuw geïnitieerde middenpunten.
coordinate_1	De aggregatie die de eerste coördinaat berekent, meestal de x-as (van de verdelingsplot die op basis van het diagram kan worden gemaakt). De aanvullende parameters berekenen de tweede, derde en vierde coördinaat, etc.

Voorbeeld: Diagramuitdrukking

In dit voorbeeld maken we een verdelingsplot met behulp van de gegevensset *Iris* en gebruiken vervolgens KMeans om de gegevens te kleuren op uitdrukking.

Bovendien maken we een variabele voor het argument *num\_clusters* en gebruiken vervolgens een invoervak voor de variabele om het aantal clusters te wijzigen.

Bovendien maken we een variabele voor het argument *num\_iter* en gebruiken vervolgens een tweede invoervak voor de variabele om het aantal iteraties te wijzigen.

De gegevensset *Iris* is in verschillende indelingen openbaar toegankelijk. We hebben de gegevens ter beschikking gesteld als een uitgelijnde tabel die geladen kan worden met behulp van de editor voor laden van gegevens in Qlik Sense. Houd er rekening mee dat we in dit voorbeeld de kolom *id* hebben toegevoegd.

Nadat we de gegevens in Qlik Sense hebben geladen, gaan we als volgt te werk:

1. Sleep een **verdelingsplot** op een nieuw werkblad. Geef het *bloembladdiagram (kleuruitdrukking)* een naam.
2. Maak een variabele om het aantal clusters op te geven. Voor de variabele **Naam** voert u *KmeansPetalClusters* in. Voor de variabele **Definitie** voert u `=2` in.
3. Maak een variabele om het aantal iteraties op te geven. Voer voor de variabele **Naam** *KMeansNumberIterations* in. Voer voor de variabele **Definitie** `=1` in.
4. Configureer **Uiterlijk** voor het diagram:
  - i. Onder **Dimensies** kiest u *id* voor het veld voor **Bel**. Voer de cluster-id in voor het label.
  - ii. Onder **Metingen** kiest u *Som([lengte.bloemenblad])* voor de uitdrukking voor **X-as**.
  - iii. Onder **Metingen** kiest u *Som([breedte.bloemenblad])* voor de uitdrukking voor **Y-as**.

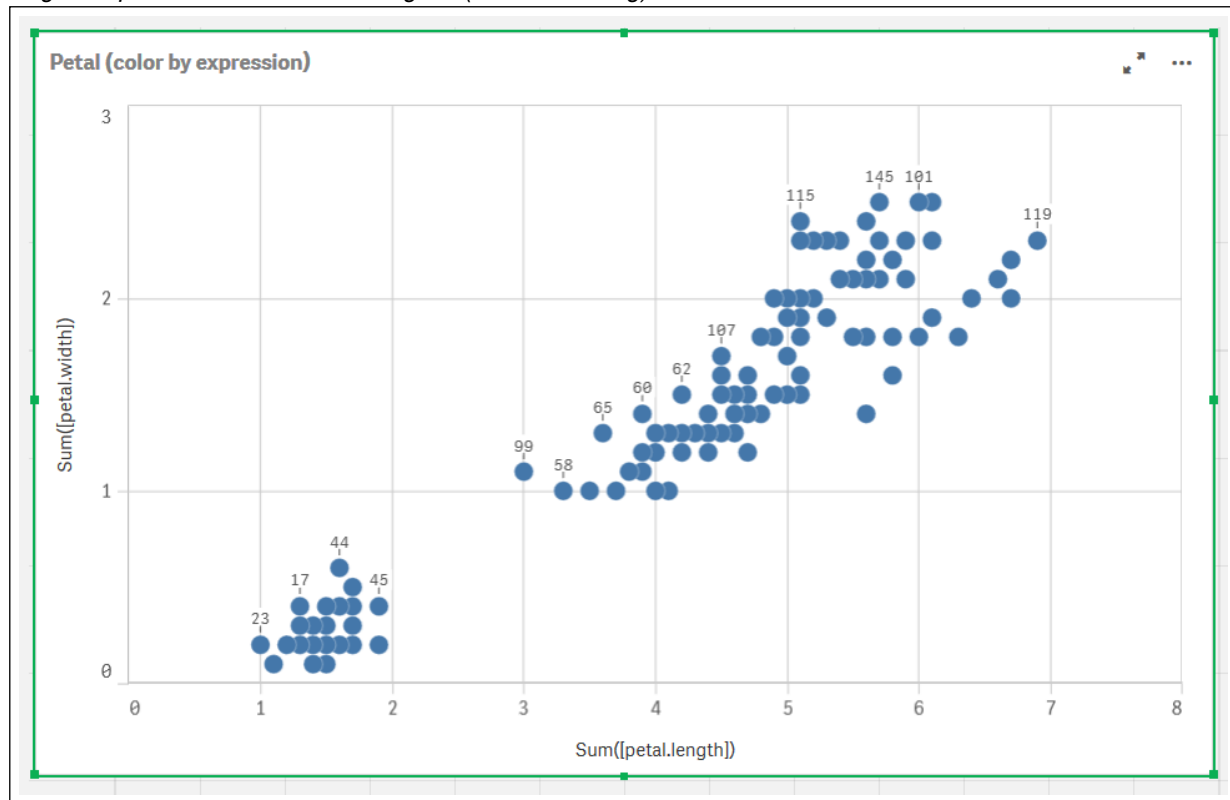
Gegevensinstellingen voor het bloembladdiagram (kleuruitdrukking)

The image shows a configuration panel for a bubble chart. It is divided into three main sections: 'Data', 'Dimensions', and 'Measures'. The 'Data' section is at the top. The 'Dimensions' section, highlighted with a red border, contains a 'Bubble' visualization type and a single dimension 'Id'. Below this is an 'Alternative dimensions' section with an 'Add alternative' button. The 'Measures' section, also highlighted with a red border, shows two measures: 'Sum [petal.length]' on the X-axis and 'Sum [petal.width]' on the Y-axis. Each measure has a dropdown arrow and a grid icon.

Section	Item
Dimensions	Bubble
Dimensions	Id
Measures	Sum [petal.length] (X-axis)
Measures	Sum [petal.width] (Y-axis)

De gegevenspunten worden in het diagram geplot.

Gegevenspunten in het bloembladdiagram (kleuruitdrukking)



5. Configureer **Uiterlijk** voor het diagram:

- i. Onder **Kleuren en legenda** kiest u **Aangepast** voor **Kleuren**.
- ii. Kleur het diagram **Op uitdrukking**.
- iii. Voer het volgende in voor **Uitdrukking**: `kmeansnd`  
 $($(KmeansPetalClusters),$(KmeansNumberIterations), Sum([petal.length]), Sum([petal.width]), Sum([sepal.length]), Sum([sepal.width]))$   
 Houd er rekening mee dat `KmeansPetalClusters` de variabele is die we hebben ingesteld op 2. We stellen de variabele `KmeansNumberIterations` in op 1.  
 U kunt ook het volgende invoeren: `kmeans2d(2, Sum([petal.length]), Sum([petal.width]))`
- iv. Schakel het selectievakje uit voor **De uitdrukking is een kleurcode**.

v. Voer het volgende in voor **Label**: *Cluster-id*

*Instellingen voor het uiterlijk van het bloembladdiagram (kleuruitdrukking)*

Appearance

▼ Colors and legend

Colors

Custom

By expression ▼

Expression

kmeansnd\$(KmeansPetal( *fx*

The expression is a color code

Label

Cluster Id

Color scheme

Sequential gradient

Sequential classes

Diverging gradient

Diverging classes

Reverse colors

Range

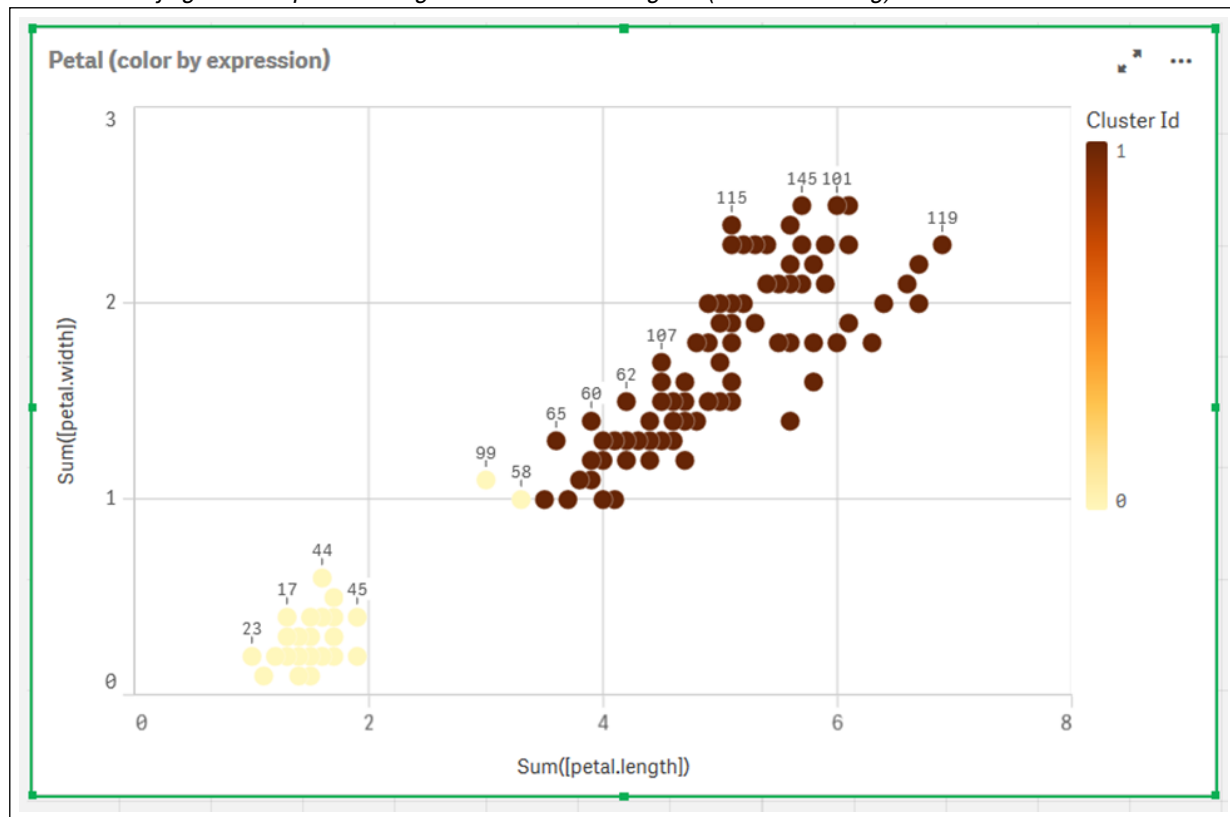
Auto

Show legend

Auto

Legend position

De twee clusters van het diagram worden gekleurd door de uitdrukking KMeans.  
*Clusters die zijn gekleurd op uitdrukking in een bloembladdiagram (kleuruitdrukking)*



6. Voeg een **Invoervak voor variabelen** toe voor het aantal clusters.
  - i. Onder **Aangepaste objecten** in het **bedrijfsmiddelen**venster kiest u **Qlik Dashboard bundle**. Indien we geen toegang hadden tot de bundel dashboard, zouden we nog steeds in staat zijn om het aantal clusters te wijzigen dat de variabele gebruikt die we hebben gemaakt. Dit is ook direct mogelijk met behulp van een geheel getal in de uitdrukking.
  - ii. Sleep een **Invoervak voor variabelen** naar het werkblad.
  - iii. Onder **Uiterlijk** kiest u **Algemeen**.
  - iv. Voer het volgende in voor **Titel**: *Clusters*
  - v. Klik op **Variabele**.
  - vi. Kies de volgende variabele voor **Naam**: *KmeansPetalClusters*.
  - vii. Kies **Schuifregelaar** voor **Weergeven als**.



viii. Kies **Waarden** en configureer de instellingen zoals vereist.

*Uiterlijk voor het invoervak voor variabelen van clusters*

▼ General

Show titles  On

Title

Clusters	<i>fx</i>
----------	-----------

Subtitle

	<i>fx</i>
--	-----------

Footnote

	<i>fx</i>
--	-----------

Disable hover menu

▼ Variable

Name

KmeansPetalClusters	▼
---------------------	---

Show as

Slider	▼
--------	---

Update on drag

▼ Values

Min

2	<i>fx</i>
---	-----------

Max

10	<i>fx</i>
----	-----------

Step

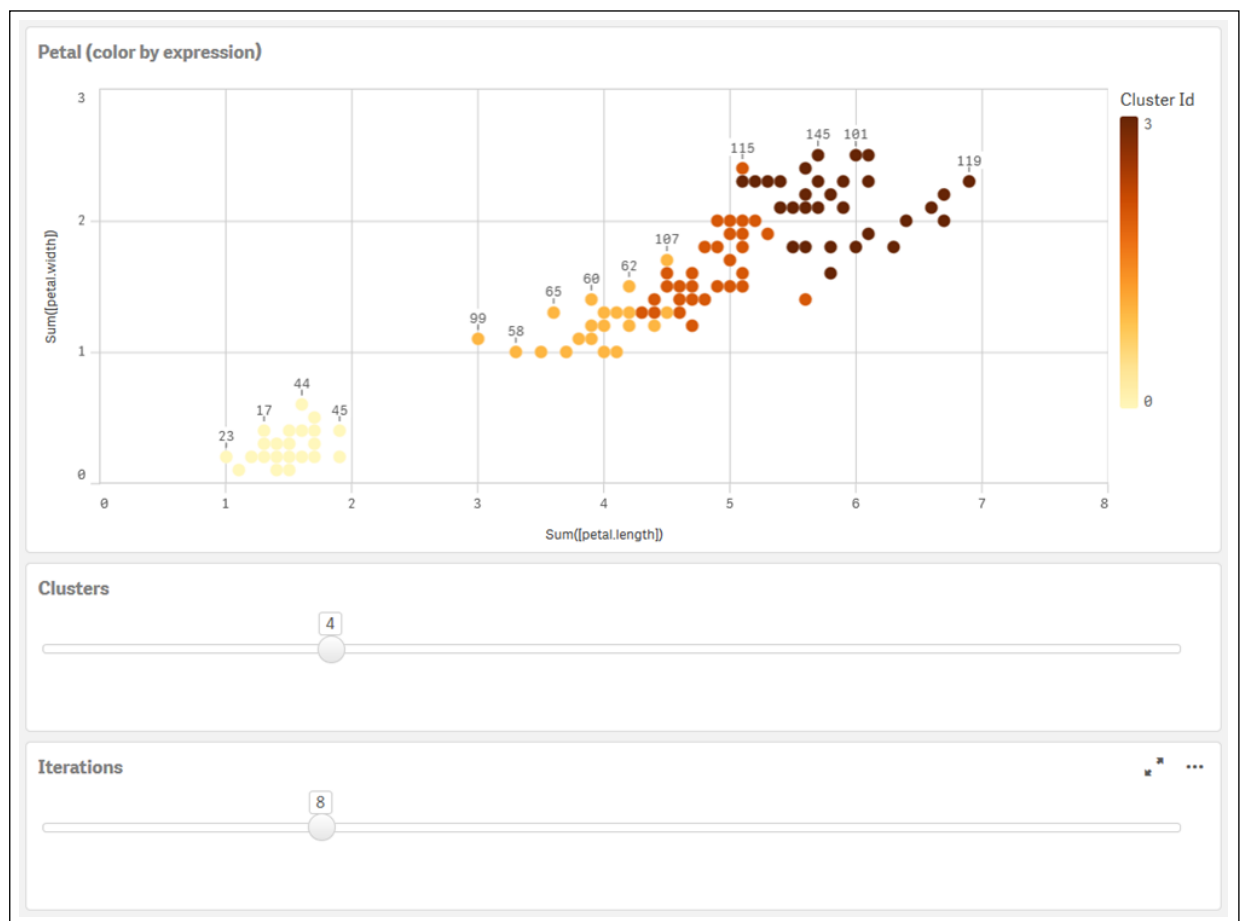
1	<i>fx</i>
---	-----------

Slider label

7. Voeg een **Invoervak voor variabelen** toe voor het aantal iteraties.
  - i. Sleep een **Invoervak voor variabelen** naar het werkblad.
  - ii. Onder **Uiterlijk** kiest u **Algemeen**.
  - iii. Voer het volgende in voor **Titel**: *Iteraties*
  - iv. Onder **Uiterlijk** kiest u **Variabele**.
  - v. Kies de volgende variabele onder **Naam**: *KmeansNumberIterations*.
  - vi. Configureer de extra instellingen zoals vereist,

We kunnen nu het aantal clusters en iteraties wijzigen met de schuifbalken in de invoervakken voor variabelen.

*Clusters die zijn gekleurd op uitdrukking in een bloembladdiagram (kleuruitdrukking)*



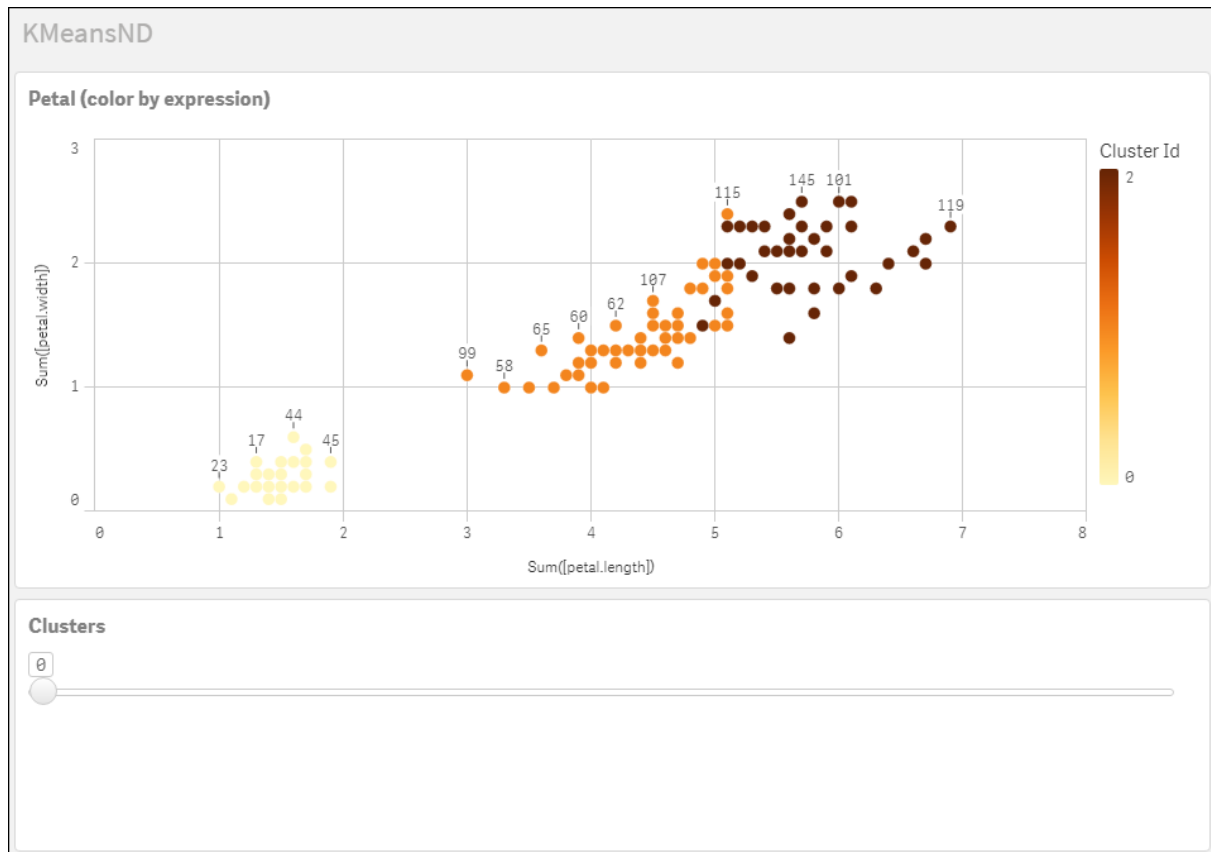
### Automatische clustering

De functies voor **KMeans** bieden ondersteuning voor automatische clustering met behulp van de methode diepteverschil (DeD). Als een gebruiker het aantal clusters instelt op 0, wordt een optimaal aantal clusters voor die gegevensverzameling bepaald. Houd er rekening mee dat een geheel getal voor het aantal clusters ( $k$ ) niet expliciet wordt geretourneerd, maar wel wordt berekend binnen het KMeans-algoritme. Als bijvoorbeeld 0 is opgegeven in de functie voor de waarde van *KmeansPetalClusters* of is ingesteld via een

## 5 Script- en diagramfuncties

invoervak voor variabelen, worden clustertoewijzingen automatisch berekend voor de gegevensverzameling die is gebaseerd op een optimaal aantal clusters. Op basis van de Iris-gegevensverzameling, als 0 is geselecteerd voor het aantal clusters, bepaalt het algoritme (automatische clustering) een optimaal aantal clusters (3) voor deze gegevensverzameling.

*De diepteverschilmethode KMeans depth bepaalt het optimale aantal clusters als (k) is ingesteld op 0*



### Iris-gegevensverzameling: Uitgelijnde lading voor editor voor laden van gegevens in Qlik Sense

```
IrisData: Load * Inline [ sepal.length, sepal.width, petal.length, petal.width, variety, id
5.1, 3.5, 1.4, 0.2, Setosa, 1 4.9, 3, 1.4, 0.2, Setosa, 2 4.7, 3.2, 1.3, 0.2, Setosa, 3 4.6,
3.1, 1.5, 0.2, Setosa, 4 5, 3.6, 1.4, 0.2, Setosa, 5 5.4, 3.9, 1.7, 0.4, Setosa, 6 4.6, 3.4,
1.4, 0.3, Setosa, 7 5, 3.4, 1.5, 0.2, Setosa, 8 4.4, 2.9, 1.4, 0.2, Setosa, 9 4.9, 3.1, 1.5,
0.1, Setosa, 10 5.4, 3.7, 1.5, 0.2, Setosa, 11 4.8, 3.4, 1.6, 0.2, Setosa, 12 4.8, 3, 1.4,
0.1, Setosa, 13 4.3, 3, 1.1, 0.1, Setosa, 14 5.8, 4, 1.2, 0.2, Setosa, 15 5.7, 4.4, 1.5, 0.4,
Setosa, 16 5.4, 3.9, 1.3, 0.4, Setosa, 17 5.1, 3.5, 1.4, 0.3, Setosa, 18 5.7, 3.8, 1.7, 0.3,
Setosa, 19 5.1, 3.8, 1.5, 0.3, Setosa, 20 5.4, 3.4, 1.7, 0.2, Setosa, 21 5.1, 3.7, 1.5, 0.4,
Setosa, 22 4.6, 3.6, 1, 0.2, Setosa, 23 5.1, 3.3, 1.7, 0.5, Setosa, 24 4.8, 3.4, 1.9, 0.2,
Setosa, 25 5, 3, 1.6, 0.2, Setosa, 26 5, 3.4, 1.6, 0.4, Setosa, 27 5.2, 3.5, 1.5, 0.2, Setosa,
28 5.2, 3.4, 1.4, 0.2, Setosa, 29 4.7, 3.2, 1.6, 0.2, Setosa, 30 4.8, 3.1, 1.6, 0.2, Setosa,
31 5.4, 3.4, 1.5, 0.4, Setosa, 32 5.2, 4.1, 1.5, 0.1, Setosa, 33 5.5, 4.2, 1.4, 0.2, Setosa,
34 4.9, 3.1, 1.5, 0.1, Setosa, 35 5, 3.2, 1.2, 0.2, Setosa, 36 5.5, 3.5, 1.3, 0.2, Setosa, 37
4.9, 3.1, 1.5, 0.1, Setosa, 38 4.4, 3, 1.3, 0.2, Setosa, 39 5.1, 3.4, 1.5, 0.2, Setosa, 40 5,
3.5, 1.3, 0.3, Setosa, 41 4.5, 2.3, 1.3, 0.3, Setosa, 42 4.4, 3.2, 1.3, 0.2, Setosa, 43 5,
3.5, 1.6, 0.6, Setosa, 44 5.1, 3.8, 1.9, 0.4, Setosa, 45 4.8, 3, 1.4, 0.3, Setosa, 46 5.1,
3.8, 1.6, 0.2, Setosa, 47 4.6, 3.2, 1.4, 0.2, Setosa, 48 5.3, 3.7, 1.5, 0.2, Setosa, 49 5,
3.3, 1.4, 0.2, Setosa, 50 7, 3.2, 4.7, 1.4, versicolor, 51 6.4, 3.2, 4.5, 1.5, Versicolor, 52
```

6.9, 3.1, 4.9, 1.5, Versicolor, 53 5.5, 2.3, 4, 1.3, Versicolor, 54 6.5, 2.8, 4.6, 1.5, Versicolor, 55 5.7, 2.8, 4.5, 1.3, Versicolor, 56 6.3, 3.3, 4.7, 1.6, Versicolor, 57 4.9, 2.4, 3.3, 1, Versicolor, 58 6.6, 2.9, 4.6, 1.3, Versicolor, 59 5.2, 2.7, 3.9, 1.4, Versicolor, 60 5, 2, 3.5, 1, Versicolor, 61 5.9, 3, 4.2, 1.5, Versicolor, 62 6, 2.2, 4, 1, Versicolor, 63 6.1, 2.9, 4.7, 1.4, Versicolor, 64 5.6, 2.9, 3.6, 1.3, Versicolor, 65 6.7, 3.1, 4.4, 1.4, Versicolor, 66 5.6, 3, 4.5, 1.5, Versicolor, 67 5.8, 2.7, 4.1, 1, Versicolor, 68 6.2, 2.2, 4.5, 1.5, Versicolor, 69 5.6, 2.5, 3.9, 1.1, Versicolor, 70 5.9, 3.2, 4.8, 1.8, Versicolor, 71 6.1, 2.8, 4, 1.3, Versicolor, 72 6.3, 2.5, 4.9, 1.5, Versicolor, 73 6.1, 2.8, 4.7, 1.2, Versicolor, 74 6.4, 2.9, 4.3, 1.3, Versicolor, 75 6.6, 3, 4.4, 1.4, Versicolor, 76 6.8, 2.8, 4.8, 1.4, Versicolor, 77 6.7, 3, 5, 1.7, Versicolor, 78 6, 2.9, 4.5, 1.5, Versicolor, 79 5.7, 2.6, 3.5, 1, Versicolor, 80 5.5, 2.4, 3.8, 1.1, Versicolor, 81 5.5, 2.4, 3.7, 1, Versicolor, 82 5.8, 2.7, 3.9, 1.2, Versicolor, 83 6, 2.7, 5.1, 1.6, Versicolor, 84 5.4, 3, 4.5, 1.5, Versicolor, 85 6, 3.4, 4.5, 1.6, Versicolor, 86 6.7, 3.1, 4.7, 1.5, Versicolor, 87 6.3, 2.3, 4.4, 1.3, Versicolor, 88 5.6, 3, 4.1, 1.3, Versicolor, 89 5.5, 2.5, 4, 1.3, Versicolor, 90 5.5, 2.6, 4.4, 1.2, Versicolor, 91 6.1, 3, 4.6, 1.4, Versicolor, 92 5.8, 2.6, 4, 1.2, Versicolor, 93 5, 2.3, 3.3, 1, Versicolor, 94 5.6, 2.7, 4.2, 1.3, Versicolor, 95 5.7, 3, 4.2, 1.2, Versicolor, 96 5.7, 2.9, 4.2, 1.3, Versicolor, 97 6.2, 2.9, 4.3, 1.3, Versicolor, 98 5.1, 2.5, 3, 1.1, Versicolor, 99 5.7, 2.8, 4.1, 1.3, Versicolor, 100 6.3, 3.3, 6, 2.5, Virginica, 101 5.8, 2.7, 5.1, 1.9, Virginica, 102 7.1, 3, 5.9, 2.1, Virginica, 103 6.3, 2.9, 5.6, 1.8, Virginica, 104 6.5, 3, 5.8, 2.2, Virginica, 105 7.6, 3, 6.6, 2.1, Virginica, 106 4.9, 2.5, 4.5, 1.7, Virginica, 107 7.3, 2.9, 6.3, 1.8, Virginica, 108 6.7, 2.5, 5.8, 1.8, Virginica, 109 7.2, 3.6, 6.1, 2.5, Virginica, 110 6.5, 3.2, 5.1, 2, Virginica, 111 6.4, 2.7, 5.3, 1.9, Virginica, 112 6.8, 3, 5.5, 2.1, Virginica, 113 5.7, 2.5, 5, 2, Virginica, 114 5.8, 2.8, 5.1, 2.4, Virginica, 115 6.4, 3.2, 5.3, 2.3, Virginica, 116 6.5, 3, 5.5, 1.8, Virginica, 117 7.7, 3.8, 6.7, 2.2, Virginica, 118 7.7, 2.6, 6.9, 2.3, Virginica, 119 6, 2.2, 5, 1.5, Virginica, 120 6.9, 3.2, 5.7, 2.3, Virginica, 121 5.6, 2.8, 4.9, 2, Virginica, 122 7.7, 2.8, 6.7, 2, Virginica, 123 6.3, 2.7, 4.9, 1.8, Virginica, 124 6.7, 3.3, 5.7, 2.1, Virginica, 125 7.2, 3.2, 6, 1.8, Virginica, 126 6.2, 2.8, 4.8, 1.8, Virginica, 127 6.1, 3, 4.9, 1.8, Virginica, 128 6.4, 2.8, 5.6, 2.1, Virginica, 129 7.2, 3, 5.8, 1.6, Virginica, 130 7.4, 2.8, 6.1, 1.9, Virginica, 131 7.9, 3.8, 6.4, 2, Virginica, 132 6.4, 2.8, 5.6, 2.2, Virginica, 133 6.3, 2.8, 5.1, 1.5, Virginica, 134 6.1, 2.6, 5.6, 1.4, Virginica, 135 7.7, 3, 6.1, 2.3, Virginica, 136 6.3, 3.4, 5.6, 2.4, Virginica, 137 6.4, 3.1, 5.5, 1.8, Virginica, 138 6, 3, 4.8, 1.8, Virginica, 139 6.9, 3.1, 5.4, 2.1, Virginica, 140 6.7, 3.1, 5.6, 2.4, Virginica, 141 6.9, 3.1, 5.1, 2.3, Virginica, 142 5.8, 2.7, 5.1, 1.9, Virginica, 143 6.8, 3.2, 5.9, 2.3, Virginica, 144 6.7, 3.3, 5.7, 2.5, Virginica, 145 6.7, 3, 5.2, 2.3, Virginica, 146 6.3, 2.5, 5, 1.9, Virginica, 147 6.5, 3, 5.2, 2, Virginica, 148 6.2, 3.4, 5.4, 2.3, Virginica, 149 5.9, 3, 5.1, 1.8, Virginica, 150 ];

### KMeansCentroid2D - diagramfunctie

**KMeansCentroid2D()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de gewenste coördinaat weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter `coordinate_1` respectievelijk `coordinate_2`. Dit zijn beide aggregaties. De parameter `num_clusters` bepaalt het aantal clusters dat wordt gemaakt. Gegevens kunnen optioneel worden genormaliseerd door de `norm` parameter.

**KMeansCentroid2D** retourneert één waarde per gegevenspunt. De geretourneerde waarde is een dubbele waarde en is een van de coördinaten van de positie die overeenkomt met het middenpunt van het cluster waaraan elk gegevenspunt is toegewezen.

#### Syntaxis:

```
KMeansCentroid2D(num_clusters, coordinate_no, coordinate_1, coordinate_2 [, norm])
```

**Retourgegevenstypen:** dual

**Argumenten:**

### Argumenten

Argument	Beschrijving
num_clusters	Het geheel getal dat het aantal clusters aangeeft.
coordinate_no	Het gewenste aantal coördinaten van de zwaartepunten (die bijvoorbeeld overeenkomen met de x-, y-, of z-as).
coordinate_1	De aggregatie die de eerste coördinaat berekent, meestal de x-as van de verdelingsplot die op basis van het diagram kan worden gemaakt. De aanvullende parameter, coordinate_2, berekent de tweede coördinaat.
norm	<p>De optionele normalisatiemethode die is toegepast op gegevensverzamelingen voordat K-means clustering wordt toegepast.</p> <p>Mogelijke waarden:</p> <ul style="list-style-type: none"><li>0 of 'geen' voor geen normalisatie</li><li>1 of 'zscore' for z-score normalisatie</li><li>2 of 'minmax' voor min-max normalisatie</li></ul> <p>Als er geen parameter is opgegeven of de opgegeven parameter is onjuist, wordt er geen normalisatie toegepast.</p> <p>De Z-score normaliseert gegevens op basis van de gemiddelde- en standaardafwijking. De Z-score zorgt er niet voor dat elke functie dezelfde schaal toebedeeld krijgt, maar het is een betere manier om aan te gaan met uitschieters dan de min-max-methode.</p> <p>De min-max normalisatie zorgt ervoor dat functies dezelfde schaal toebedeeld krijgen door gebruik te maken van de minimum- en maximumwaarden van elk gegevenspunt en elk gegevenspunt opnieuw te berekenen.</p>

### Automatische clustering

De functies voor **KMeans** bieden ondersteuning voor automatische clustering met behulp van de methode diepteverschil (DeD). Als een gebruiker het aantal clusters instelt op 0, wordt een optimaal aantal clusters voor die gegevensverzameling bepaald. Houd er rekening mee dat een geheel getal voor het aantal clusters (*k*) niet expliciet wordt geretourneerd, maar wel wordt berekend binnen het KMeans-algoritme. Als bijvoorbeeld 0 is opgegeven in de functie voor de waarde van *KmeansPetalClusters* of is ingesteld via een invoervak voor variabelen, worden clustertoewijzingen automatisch berekend voor de gegevensverzameling die is gebaseerd op een optimaal aantal clusters.

## KMeansCentroidND - diagramfunctie

**KMeansCentroidND()** evalueert de rijen van het diagram door K-means clustering toe te passen. Voor elke rij van het diagram wordt de gewenste coördinaat weergegeven van het cluster waaraan dit gegevenspunt is toegewezen. De kolommen die worden gebruikt door het clustering-algoritme worden bepaald door de parameter `coordinate_1` respectievelijk `coordinate_2`. Dit zijn beide aggregaties. De parameter `num_clusters` bepaalt het aantal clusters dat wordt gemaakt.

**KMeansCentroidND** retourneert één waarde per gegevenspunt. De geretourneerde waarde is een dubbele waarde en is een van de coördinaten van de positie die overeenkomt met het middenpunt van het cluster waaraan elk gegevenspunt is toegewezen.

### Syntaxis:

```
KMeansCentroidND(num_clusters, num_iter, coordinate_no, coordinate_1,
coordinate_2 [,coordinate_3 [, ...]])
```

**Retourgegevenstypen:** dual

### Argumenten:

#### Argumenten

Argument	Beschrijving
<code>num_clusters</code>	Het geheel getal dat het aantal clusters aangeeft.
<code>num_iter</code>	Het aantal iteraties voor clusters met opnieuw geïnitieerde middenpunten.
<code>coordinate_no</code>	Het gewenste aantal coördinaten van de zwaartepunten (die bijvoorbeeld overeenkomen met de x-, y-, of z-as).
<code>coordinate_1</code>	De aggregatie die de eerste coördinaat berekent, meestal de x-as (van de verdelingsplot die op basis van het diagram kan worden gemaakt). De aanvullende parameters berekenen de tweede, derde en vierde coördinaat, etc.

## Automatische clustering

De functies voor **KMeans** bieden ondersteuning voor automatische clustering met behulp van de methode diepteverschil (DeD). Als een gebruiker het aantal clusters instelt op 0, wordt een optimaal aantal clusters voor die gegevensverzameling bepaald. Houd er rekening mee dat een geheel getal voor het aantal clusters (*k*) niet expliciet wordt geretourneerd, maar wel wordt berekend binnen het KMeans-algoritme. Als bijvoorbeeld 0 is opgegeven in de functie voor de waarde van *KmeansPetalClusters* of is ingesteld via een invoervak voor variabelen, worden clustertoewijzingen automatisch berekend voor de gegevensverzameling die is gebaseerd op een optimaal aantal clusters.



### 5.23 Statistische verdelingsfuncties

De statistische verdelingsfuncties retourneren de waarschijnlijkheid van verschillende mogelijke uitkomsten voor een opgegeven invoerwaarde. U kunt deze functies gebruiken om de mogelijke waarden van uw gegevenspunten te berekenen.

De drie groepen van de hieronder beschreven statistische verdelingsfuncties worden via de Cephes-functiebibliotheek in Qlik Sense geïmplementeerd. Zie [≤ Cephes library](#) voor referenties en meer informatie over gebruikte algoritmen, nauwkeurigheid enzovoort. De Cephes-functiebibliotheek wordt met toestemming gebruikt.

- De waarschijnlijkheidsfuncties berekenen de waarschijnlijkheid vanaf het punt in de verdeling die door de opgegeven waarde wordt gegeven.
  - De frequentiefuncties worden gebruikt voor discrete verdelingen.
  - De dichtheidsfuncties worden gebruikt voor continue functies.
- De dist-functies berekenen de gecumuleerde waarschijnlijkheid van de verdeling vanaf het punt in de verdeling die door de opgegeven waarde wordt gegeven.
- De Inv-functies berekenen de waarde, gegeven de gecumuleerde waarschijnlijkheid van de verdeling.

Alle functies kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

### Overzicht van statistische verdelingsfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### BetaDensity

`BetaDensity()` retourneert de waarschijnlijkheid van de bèta-verdeling.

```
BetaDensity (value, alpha, beta)
```

#### BetaDist

`BetaDist()` retourneert de gecumuleerde waarschijnlijkheid van de bèta-verdeling.

```
BetaDist (value, alpha, beta)
```

#### BetaInv

`BetaInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de bèta-verdeling.

```
BetaInv (prob, alpha, beta)
```

#### BinomDist

`BinomDist()` retourneert de gecumuleerde waarschijnlijkheid van de binomiale verdeling.

```
BinomDist (value, trials, trial_probability)
```

### BinomFrequency

`BinomFrequency()` retourneert de binomiale waarschijnlijkheidsverdeling.

```
BinomFrequency (value, trials, trial_probability)
```

### BinomInv

`BinomInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de binomiale verdeling.

```
BinomInv (prob, trials, trial_probability)
```

### ChiDensity

`ChiDensity()` retourneert de eenzijdige waarschijnlijkheid van de  $\chi^2$ -verdeling. De  $\chi^2$ -dichtheidsfunctie is gekoppeld aan een  $\chi^2$ -test.

```
ChiDensity (value, degrees_freedom)
```

### ChiDist

`ChiDist()` retourneert de eenzijdige waarschijnlijkheid van de  $\chi^2$ -verdeling. De  $\chi^2$ -verdeling is gekoppeld aan een  $\chi^2$ -test.

```
ChiDist (value, degrees_freedom)
```

### ChiInv

`ChiInv()` retourneert het omgekeerde van de eenzijdige waarschijnlijkheid van de  $\chi^2$ -verdeling.

```
ChiInv (prob, degrees_freedom)
```

### FDensity

`FDensity()` retourneert de waarschijnlijkheid van de F-verdeling.

```
FDensity (value, degrees_freedom1, degrees_freedom2)
```

### FDist

`FDist()` retourneert de gecumuleerde waarschijnlijkheid van de F-verdeling.

```
FDist (value, degrees_freedom1, degrees_freedom2)
```

### FInv

`FInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de F-verdeling.

```
FInv (prob, degrees_freedom1, degrees_freedom2)
```

### GammaDensity

`GammaDensity()` retourneert de waarschijnlijkheid van de gamma-verdeling.

```
GammaDensity (value, k,  $\theta$ )
```

### GammaDist

`GammaDist()` retourneert de gecumuleerde waarschijnlijkheid van de gamma-verdeling.

```
GammaDist (value, k,  $\theta$ )
```

### GammaInv

GammaInv() retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de gamma-verdeling.

```
GammaInv (prob, k,  $\theta$ )
```

### NormDist

NormDist() retourneert de cumulatieve normale verdeling van het opgegeven gemiddelde en de opgegeven standaarddeviatie. Als mean = 0 en standard\_dev = 1, retourneert de functie de standaard normale verdeling.

```
NormDist (value, mean, standard_dev)
```

### NormInv

NormInv() retourneert het omgekeerde van de normale cumulatieve verdeling van het opgegeven gemiddelde en de opgegeven standaarddeviatie.

```
NormInv (prob, mean, standard_dev)
```

### PoissonDist

PoissonDist() retourneert de gecumuleerde waarschijnlijkheid van de poisson-verdeling.

```
PoissonDist (value, mean)
```

### PoissonFrequency

PoissonFrequency() retourneert de poisson-waarschijnlijkheidsverdeling.

```
PoissonFrequency (value, mean)
```

### PoissonInv

PoissonInv() retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de poisson-verdeling.

```
PoissonInv (prob, mean)
```

### TDensity

TDensity() retourneert de waarde van de dichtheid  $t$  van de student, waarbij een numerieke waarde een berekende waarde van  $t$  is, waarvan de waarschijnlijk moet worden berekend.

```
TDensity (value, degrees_freedom, tails)
```

### TDist

TDist() retourneert de waarschijnlijkheid van de  $t$ -verdeling van de student, waarbij een numerieke waarde een berekende waarde van  $t$  is, waarvan de waarschijnlijk moet worden berekend.

```
TDist (value, degrees_freedom, tails)
```

### TInv

TInv() retourneert de waarde  $t$  van de  $t$ -verdeling van de student als functie van de waarschijnlijkheid en de vrijheidsgraden.

```
TInv (prob, degrees_freedom)
```

### Zie ook:

p *Statistische aggregatiefuncties (page 387)*

## BetaDensity

`BetaDensity()` retourneert de waarschijnlijkheid van de bèta-verdeling.

### Syntaxis:

```
BetaDensity(value, alpha, beta)
```

**Retourgegevenstypen:** getal

#### Argumenten

Argument	Beschrijving
<code>value</code>	De waarde waarvan u de verdeling wilt bepalen. De waarde moet tussen 0 en 1 liggen.
<code>alpha</code>	Een positief getal definieert de eerste vormparameter. Het is de exponent van de toevalsvariabele.
<code>beta</code>	Een positief getal definieert de tweede vormparameter. Geeft het aantal vrijheidsgraden van de noemer aan.

## BetaDist

`BetaDist()` retourneert de gecumuleerde waarschijnlijkheid van de bèta-verdeling.

### Syntaxis:

```
BetaDist(value, alpha, beta)
```

**Retourgegevenstypen:** getal

#### Argumenten

Argument	Beschrijving
<code>value</code>	De waarde waarvan u de verdeling wilt bepalen. De waarde moet tussen 0 en 1 liggen.
<code>alpha</code>	Een positief getal definieert de eerste vormparameter. Het is de exponent van de toevalsvariabele.
<code>beta</code>	Een positief getal definieert de tweede vormparameter. Dit is de exponent die de vorm van de verdeling bepaalt.

Deze functie is op de volgende manier gerelateerd aan de functie `BetaInv`:

If `prob = BetaDist(value, alpha, beta)`, then `BetaInv(prob, alpha, beta) = value`

## BetaInv

`BetaInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de bèta-verdeling.

### Syntaxis:

```
BetaInv(prob, alpha, beta)
```

**Retourgegevenstypen:** getal

#### Argumenten

Argument	Beschrijving
prob	Een waarschijnlijkheid die is verbonden met de bèta-waarschijnlijkheidsverdeling. Dit moet een waarde zijn tussen 0 en 1.
alpha	Een positief getal definieert de eerste vormparameter. Het is de exponent van de toevalsvariabele.
beta	Een positief getal definieert de tweede vormparameter. Dit is de exponent die de vorm van de verdeling bepaalt.

Deze functie is op de volgende manier gerelateerd aan de functie `BetaDist`:

If `prob = BetaDist(value, alpha, beta)`, then `BetaInv(prob, alpha, beta) = value`

## BinomDist

`BinomDist()` retourneert de gecumuleerde waarschijnlijkheid van de binomiale verdeling.

### Syntaxis:

```
BinomDist(value, trials, trial_probability)
```

**Retourgegevenstypen:** getal

#### Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde moet een geheel getal zijn dat niet kleiner is dan 0 en niet groter is dan het aantal proeven.
trials	Een positief getal dat het aantal proeven weergeeft.
trial_probability	De kans op succes voor elke proef. Dit is altijd een getal tussen 0 en 1.

Deze functie is op de volgende manier gerelateerd aan de functie `BinomInv`:

If `prob = BinomDIST(value, trials, trial_probability)`, then `BinomInv(prob, trials, trial_probability) = value`

## BinomFrequency

`BinomFrequency()` retourneert de binomiale waarschijnlijkheidsverdeling.

### Syntaxis:

```
BinomFrequency(value, trials, trial_probability)
```

**Retourgegevenstypen:** getal

#### Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde moet een geheel getal zijn dat niet kleiner is dan 0 en niet groter is dan het aantal proeven.
trials	Een positief getal dat het aantal proeven weergeeft
trial_probability	De kans op succes voor elke proef. Dit is altijd een getal tussen 0 en 1.

## BinomInv

`BinomInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de binomiale verdeling.

### Syntaxis:

```
BinomInv(prob, trials, trial_probability)
```

**Retourgegevenstypen:** getal

#### Argumenten

Argument	Beschrijving
prob	Een waarschijnlijkheid die is verbonden met de binomiale waarschijnlijkheidsverdeling. Dit moet een waarde zijn tussen 0 en 1.
trials	Een positief getal dat het aantal proeven weergeeft.
trial_probability	De kans op succes voor elke proef. Dit is altijd een getal tussen 0 en 1.

Deze functie is op de volgende manier gerelateerd aan de functie `BinomDist`:

If `prob = BinomDist(value, trials, trial_probability)`, then `BinomInv(prob, trials, trial_probability) = value`

## ChiDensity

`ChiDensity()` retourneert de eenzijdige waarschijnlijkheid van de  $\chi^2$ -verdeling. De  $\chi^2$ -dichtheidsfunctie is gekoppeld aan een  $\chi^2$ -test.

### Syntaxis:

```
ChiDensity(value, degrees_freedom)
```

Retourgegevenstypen: getal

#### Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn.
degrees_freedom	Een positief geheel getal waarmee het aantal vrijheidsgraden van de teller wordt aangegeven.

## ChiDist

chiDist() retourneert de eenzijdige waarschijnlijkheid van de  $\chi^2$ -verdeling. De  $\chi^2$ -verdeling is gekoppeld aan een  $\chi^2$ -test.

### Syntaxis:

```
CHIDIST(value, degrees_freedom)
```

Retourgegevenstypen: getal

### Argumenten:

#### Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn.
degrees_freedom	Een positief geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven.

Deze functie is op de volgende manier gerelateerd aan de functie **ChiInv**:

If prob = CHIDIST(value,df), then CHIINV(prob, df) = value

### Beperkingen:

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
CHIDIST( 8, 15)	retourneert 0,9238

## ChiInv

chiInv() retourneert het omgekeerde van de eenzijdige waarschijnlijkheid van de  $\chi^2$ -verdeling.

**Syntaxis:**

```
CHIINV(prob, degrees_freedom)
```

**Retourgegevenstypen:** getal**Argumenten:**

## Argumenten

Argument	Beschrijving
prob	Een waarschijnlijkheid die is verbonden met de $\chi^2$ -verdeling. Dit moet een waarde zijn tussen 0 en 1.
degrees_freedom	Een geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven.

Deze functie is op de volgende manier gerelateerd aan de functie **ChiDist**:

If `prob = CHIDIST(value,df)`, then `CHIINV(prob, df) = value`

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
CHIINV(0.9237827, 15)	retourneert 8,0000

## FDensity

`FDensity()` retourneert de waarschijnlijkheid van de F-verdeling.

**Syntaxis:**

```
FDensity(value, degrees_freedom1, degrees_freedom2)
```

**Retourgegevenstypen:** getal

## Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn.
degrees_freedom1	Een positief geheel getal waarmee het aantal vrijheidsgraden van de teller wordt aangegeven.
degrees_freedom2	Een positief geheel getal waarmee het aantal vrijheidsgraden van de noemer wordt aangegeven.



### FDist

`FDist()` retourneert de gecumuleerde waarschijnlijkheid van de F-verdeling.

**Syntaxis:**

```
FDist(value, degrees_freedom1, degrees_freedom2)
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn.
degrees_freedom1	Een positief geheel getal waarmee het aantal vrijheidsgraden van de teller wordt aangegeven.
degrees_freedom2	Een positief geheel getal waarmee het aantal vrijheidsgraden van de noemer wordt aangegeven.

Deze functie is op de volgende manier gerelateerd aan de functie **FInv**:

If  $\text{prob} = \text{FDIST}(\text{value}, \text{df1}, \text{df2})$ , then  $\text{FINV}(\text{prob}, \text{df1}, \text{df2}) = \text{value}$

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
<code>FDIST(15, 8, 6)</code>	retourneert 0,0019

### FInv

`FInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de F-verdeling.

**Syntaxis:**

```
FInv(prob, degrees_freedom1, degrees_freedom2)
```

**Retourgegevenstypen:** getal

**Argumenten:**

### Argumenten

Argument	Beschrijving
prob	Een waarschijnlijkheid die is verbonden met de F-waarschijnlijkheidsverdeling. Dit moet een waarde zijn tussen 0 en 1.
degrees_freedom	Een geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven.

Deze functie is op de volgende manier gerelateerd aan de functie **FDist**:

If `prob = FDIST(value, df1, df2)`, then `FINV(prob, df1, df2) = value`

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
<code>FINV(0.0019369, 8, 6)</code>	retourneert 15,0000

## GammaDensity

`GammaDensity()` retourneert de waarschijnlijkheid van de gamma-verdeling.

**Syntaxis:**

```
GammaDensity(value, k,  $\theta$ )
```

**Retourgegevenstypen:** getal

### Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn.
k	Een positief getal dat de tweede vormparameter definieert.
$\theta$	Een positief getal dat de tweede schaalparameter definieert.

## GammaDist

`GammaDist()` retourneert de gecumuleerde waarschijnlijkheid van de gamma-verdeling.

**Syntaxis:**

```
GammaDist(value, k,  $\theta$ )
```

**Retourgegevenstypen:** getal

### Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn.
k	Een positief getal dat de tweede vormparameter definieert.
$\theta$	Een positief getal dat de tweede schaalparameter definieert.

Deze functie is op de volgende manier gerelateerd aan de functie `GammaInv`:  
If `prob = GammaDist(value, k,  $\theta$ )`, then `GammaInv(prob, k,  $\theta$ ) = value`

## GammaInv

`GammaInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de gamma-verdeling.

### Syntaxis:

```
GammaInv(prob, k,  $\theta$ )
```

**Retourgegevenstypen:** getal

### Argumenten

Argument	Beschrijving
prob	Een waarschijnlijkheid die is verbonden met de gamma-waarschijnlijkheidsverdeling. Dit moet een waarde zijn tussen 0 en 1.
k	Een positief getal dat de tweede vormparameter definieert.
$\theta$	Een positief getal dat de tweede schaalparameter definieert.

Deze functie is op de volgende manier gerelateerd aan de functie `GammaDist`:  
If `prob = GammaDist(value, k,  $\theta$ )`, then `GammaInv(prob, k,  $\theta$ ) = value`

## NormDist

`NormDist()` retourneert de cumulatieve normale verdeling van het opgegeven gemiddelde en de opgegeven standaarddeviatie. Als `mean = 0` en `standard_dev = 1`, retourneert de functie de standaard normale verdeling.

### Syntaxis:

```
NORMDIST(value, [mean], [standard_dev], [cumulative])
```

**Retourgegevenstypen:** getal

**Argumenten:**

### Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen.
mean	Optionele waarde waarmee het rekenkundige gemiddelde voor de verdeling wordt aangegeven.  Als u geen waarde voor dit argument aangeeft, is de standaardwaarde 0.
standard_dev	Optionele positieve waarde waarmee de standaardafwijking van de verdeling wordt aangegeven.  Als u geen waarde voor dit argument aangeeft, is de standaardwaarde 1.
cumulative	U kunt er optioneel voor kiezen om een standaard normale verdeling of een cumulatieve verdeling te gebruiken.  0 = standaard normale verdeling 1 = cumulatieve verdeling (standaard)

Deze functie is op de volgende manier gerelateerd aan de functie **NormInv**:  
If  $\text{prob} = \text{NORMDIST}(\text{value}, m, \text{sd})$ , then  $\text{NORMINV}(\text{prob}, m, \text{sd}) = \text{value}$

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
<code>NORMDIST(0.5, 0, 1)</code>	retourneert 0,6915

## NormInv

`NormInv()` retourneert het omgekeerde van de normale cumulatieve verdeling van het opgegeven gemiddelde en de opgegeven standaarddeviatie.

**Syntaxis:**

```
NORMINV(prob, mean, standard_dev)
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

Argument	Beschrijving
prob	Een waarschijnlijkheid die is verbonden met de normale verdeling. Dit moet een waarde zijn tussen 0 en 1.
mean	Een waarde waarmee het rekenkundige gemiddelde voor de verdeling wordt aangegeven.
standard_dev	Een positieve waarde waarmee de standaarddeviatie van de verdeling wordt aangegeven.

Deze functie is op de volgende manier gerelateerd aan de functie **NormDist**:

If `prob = NORMDIST(value, m, sd)`, then `NORMINV(prob, m, sd) = value`

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
<code>NORMINV( 0.6914625, 0, 1 )</code>	retourneert 0,5000

## PoissonDist

`PoissonDist()` retourneert de gecumuleerde waarschijnlijkheid van de poisson-verdeling.

**Syntaxis:**

```
PoissonDist (value, mean)
```

**Retourgegevenstypen:** getal

Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn.
mean	Een positief getal dat de gemiddelde uitkomst definieert.

Deze functie is op de volgende manier gerelateerd aan de functie `PoissonInv`:

If `prob = PoissonDist(value, mean)`, then `PoissonInv(prob, mean) = value`

## PoissonFrequency

`PoissonFrequency()` retourneert de poisson-waarschijnlijkheidsverdeling.

**Syntaxis:**

```
PoissonFrequency(value, mean)
```

**Retourgegevenstypen:** getal

## Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn.
mean	Een positief getal dat de gemiddelde uitkomst definieert.

### PoissonInv

`PoissonInv()` retourneert het omgekeerde van de gecumuleerde waarschijnlijkheid van de poisson-verdeling.

**Syntaxis:**

```
PoissonInv(prob, mean)
```

**Retourgegevenstypen:** getal

## Argumenten

Argument	Beschrijving
prob	Een waarschijnlijkheid die is verbonden met de poisson-waarschijnlijkheidsverdeling. Dit moet een waarde zijn tussen 0 en 1.
mean	Een positief getal dat de gemiddelde uitkomst definieert.

Deze functie is op de volgende manier gerelateerd aan de functie `PoissonDist`:  
If `prob = PoissonDist(value, mean)`, then `PoissonInv(prob, mean) = value`

### TDensity

`TDensity()` retourneert de waarde van de dichtheid  $t$  van de student, waarbij een numerieke waarde een berekende waarde van  $t$  is, waarvan de waarschijnlijk moet worden berekend.

**Syntaxis:**

```
TDensity(value, degrees_freedom)
```

**Retourgegevenstypen:** getal

## Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn.
degrees_freedom	Een positief geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven.

### TDist

`TDIST()` retourneert de waarschijnlijkheid van de  $t$ -verdeling van de student, waarbij een numerieke waarde een berekende waarde van  $t$  is, waarvan de waarschijnlijk moet worden berekend.

**Syntaxis:**

```
TDIST(value, degrees_freedom, tails)
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

Argument	Beschrijving
value	De waarde waarvan u de verdeling wilt bepalen. De waarde mag niet negatief zijn.
degrees_freedom	Een positief geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven.
tails	Moet 1 (eenzijdige verdeling) of 2 (tweezijdige verdeling) zijn.

Deze functie is op de volgende manier gerelateerd aan de functie **TInv**:

If  $prob = TDIST(value, df, 2)$ , then  $TINV(prob, df) = value$

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
<code>TDIST(1, 30, 2)</code>	retourneert 0,3253

### TInv

`TINV()` retourneert de waarde  $t$  van de  $t$ -verdeling van de student als functie van de waarschijnlijkheid en de vrijheidsgraden.

**Syntaxis:**

```
TINV(prob, degrees_freedom)
```

**Retourgegevenstypen:** getal

**Argumenten:**

Argumenten

Argument	Beschrijving
prob	Een tweezijdige waarschijnlijkheid, verbonden met de t-verdeling. Dit moet een waarde zijn tussen 0 en 1.
degrees_ freedom	Een geheel getal waarmee het aantal vrijheidsgraden wordt aangegeven.

**Beperkingen:**

Alle argumenten moeten numeriek zijn, anders wordt een NULL-waarde geretourneerd.

Deze functie is op de volgende manier gerelateerd aan de functie **TDist**:

If  $prob = TDIST(value, df, 2)$ , then  $TINV(prob, df) = value$ .

Voorbeelden en resultaten:

Voorbeeld	Resultaat
<code>TINV(0.3253086, 30)</code>	retourneert 1,0000

## 5.24 Tekenreeksfuncties

In dit hoofdstuk worden functies voor het werken met en manipuleren van tekenreeksen beschreven.

Alle functies kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen, behalve **Evaluate**, dat alleen kan worden gebruikt in het script voor het laden van gegevens.

### Overzicht van tekenreeksfuncties

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Capitalize

**Capitalize()** retourneert de tekenreeks met alle woorden in beginhoofdletter.

```
Capitalize (text)
```

#### Chr

**Chr()** retourneert het Unicode-teken dat overeenkomt met het ingevoerde gehele getal.

```
Chr (int)
```



### Evaluate

**Evaluate()** bepaalt of de tekenreeks van de invoertekst kan worden geëvalueerd als een geldige Qlik Sense-uitdrukking en retourneert, als dat inderdaad het geval is, de waarde van de uitdrukking als een tekenreeks. Als de invoertekensreeks geen geldige uitdrukking is, wordt NULL geretourneerd.

```
Evaluate (expression_text)
```

### FindOneOf

**FindOneOf()** doorzoekt een tekenreeks om de positie te vinden van enig teken uit de reeks van opgegeven tekens. De positie waarin enig teken uit de te doorzoeken set voor het eerst voorkomt wordt geretourneerd tenzij een derde argument (met een waarde groter dan 1) wordt opgegeven. Als geen waarde wordt gevonden, wordt 0 geretourneerd.

```
FindOneOf (text, char_set[, count])
```

### Hash128

**Hash128()** retourneert een 128-bits hash van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 22 tekens.

```
Hash128 (expr{, expression})
```

### Hash160

**Hash160()** retourneert een 160-bits hash van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 27 tekens.

```
Hash160 (expr{, expression})
```

### Hash256

**Hash256()** retourneert een 256-bits hash van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 43 tekens.

```
Hash256 (expr{, expression})
```

### Index

**Index()** doorzoekt een tekenreeks om de beginpositie te vinden van de n-de keer dat een opgegeven subtekenreeks voorkomt. Een optioneel derde argument levert de waarde van n, die 1 is als het argument wordt weggelaten. Bij een negatieve waarde wordt gezocht vanaf het einde van de tekenreeks. De nummering van de posities in de tekenreeks begint bij 1.

```
Index (text, substring[, count])
```

### IsJson

**IsJson()** test of een opgegeven tekenreeks geldige JSON-gegevens (JavaScript Object Notation) bevat. U kunt ook een specifiek JSON-gegevenstype valideren.

```
IsJson (json [, type])
```

### JsonGet

**JsonGet()** retourneert het pad van een JSON-tekenreeks met gegevens (JavaScript Object Notation). De JSON-gegevens moeten een geldige JSON zijn, maar kunnen extra spaties of nieuwe regels bevatten.

```
JsonGet (json, path)
```

### JsonSet

**JsonSet()** wijzigt een tekenreeks die JSON-gegevens (JavaScript Object Notation) bevat. Het kan een JSON-waarde instellen of invoegen met de nieuwe locatie die is opgegeven door het pad. De JSON-gegevens moeten een geldige JSON zijn, maar kunnen extra spaties of nieuwe regels bevatten.

```
JsonSet (json, path, value)
```

### KeepChar

**KeepChar()** retourneert een tekenreeks die bestaat uit de eerste tekenreeks, 'text', behalve de tekens die NIET voorkomen in de tweede tekenreeks, "keep\_chars".

```
KeepChar (text, keep_chars)
```

### Left

**Left()** retourneert een tekenreeks die bestaat uit de eerste (meest linker) van de invoertekenreeks, waarbij het aantal tekens wordt bepaald door het tweede argument.

```
Left (text, count)
```

### Len

**Len()** retourneert de lengte van de invoertekenreeks.

```
Len (text)
```

### LevenshteinDist

**LevenshteinDist()** retourneert de afstand van Levenshtein tussen twee tekenreeksen. Het wordt gedefinieerd als het aantal bewerkingen van één teken (invoegingen, verwijderingen of vervangingen) dat is vereist om een tekenreeks te wijzigen. De functie kan worden gebruikt om fuzzy tekenreeksen te vergelijken.

```
LevenshteinDist (text1, text2)
```

### Lower

**Lower()** converteert alle tekens in de invoertekenreeks naar kleine letters.

```
Lower (text)
```

### LTrim

**LTrim()** retourneert de invoertekenreeks, ontdaan van beginspaties.

```
LTrim (text)
```

### Mid

**Mid()** retourneert het gedeelte van de invoertekenreeks dat begint op de positie van het teken dat is gedefinieerd door het tweede argument, 'start', en retourneert het aantal tekens dat is gedefinieerd door het derde argument, 'count'. Als 'count' wordt weggelaten, wordt de rest van de invoertekenreeks getourneerd. Het eerste teken in de invoertekenreeks heeft het nummer 1.

```
Mid (text, start[, count])
```

### Ord

**Ord()** retourneert het aantal punten voor de Unicode-code point van het eerste teken van de invoertekenreeks. Unicode

```
Ord (text)
```

### PurgeChar

**PurgeChar()** retourneert een tekenreeks die bestaat uit de tekens die zijn opgenomen in de invoertekenreeks ('text'), behalve de tekens die voorkomen in het tweede argument ('remove\_chars').

```
PurgeChar (text, remove_chars)
```

### Repeat

**Repeat()** vormt een tekenreeks die bestaat uit de invoertekenreeks die het aantal keren wordt herhaald dat wordt gedefinieerd door het tweede argument.

```
Repeat (text[, repeat_count])
```

### Replace

**Replace()** retourneert een tekenreeks nadat alle treffers van een bepaalde subtekenreeks in de tekenreeks zijn vervangen door een andere subtekenreeks. De functie is niet-recursief en wordt van links naar rechts uitgevoerd.

```
Replace (text, from_str, to_str)
```

### Right

**Right()** retourneert een tekenreeks die bestaat uit de laatste tekens (meest rechts) van de invoertekenreeks, waarbij het aantal tekens wordt bepaald door het tweede argument.

```
Right (text, count)
```

### RTrim

**RTrim()** retourneert de invoertekenreeks, ontdaan van eindspaties.

```
RTrim (text)
```

### SubField

**Subfield()** wordt gebruikt voor het extraheren van onderdelen van subtekenreeksen van een bovenliggend tekenreeksveld, waarbij de oorspronkelijke recordvelden bestaat uit twee of meer onderdelen met een scheidingsteken ertussen.

```
SubField (text, delimiter[, field_no ])
```

### SubStringCount

**SubstringCount()** retourneert het aantal keren dat de opgegeven subtekenreeks voorkomt in de tekst van de invoertekenreeks. Als er geen overeenkomst wordt gevonden, wordt 0 geretourneerd.

```
SubStringCount (text, substring)
```

### TextBetween

**TextBetween()** retourneert de tekst in de invoertekenreeks die tussen de tekens staat die zijn opgegeven als scheidingstekens.

```
TextBetween (text, delimiter1, delimiter2[, n])
```

### Trim

**Trim()** retourneert de invoertekenreeks, ontdaan van alle begin- en eindspaties.

```
Trim (text)
```

### Upper

**Upper()** converteert alle tekens in de invoertekenreeks naar hoofdletters voor alle tekstekens in de uitdrukking. Cijfers en symbolen worden genegeerd.

```
Upper (text)
```

## Capitalize

**Capitalize()** retourneert de tekenreeks met alle woorden in beginhoofdletter.

### Syntaxis:

```
Capitalize (text)
```

**Retourgegevenstypen:** tekenreeks

Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
Capitalize ( 'star trek' )	Retourneert 'Star Trek'
Capitalize ( 'AA bb cc Dd' )	Retourneert 'Aa Bb Cc Dd'

Voorbeeld: Load-script

```
Load String, Capitalize(String) Inline [String rHode iSland washingTon d.C. new york];
```

### Resultaat

Tekenreeks	Capitalize(String)
rHode iSland	Rhode Island
washingTon d.C.	Washington D.C.
new york	New York

## Chr

**Chr()** retourneert het Unicode-teken dat overeenkomt met het ingevoerde gehele getal.

### Syntaxis:

```
Chr (int)
```

**Retourgegevenstypen:** tekenreeks

Voorbeelden en resultaten:

Voorbeeld	Resultaat
Chr(65)	Retourneert de tekenreeks 'A'
Chr(163)	Retourneert de tekenreeks '£'
Chr(35)	Retourneert de tekenreeks '#'

### Evalueer

**Evalueer()** bepaalt of de tekenreeks van de invoertekst kan worden geëvalueerd als een geldige Qlik Sense-uitdrukking en retourneert, als dat inderdaad het geval is, de waarde van de uitdrukking als een tekenreeks. Als de invoertekensreeks geen geldige uitdrukking is, wordt NULL geretourneerd.

**Syntaxis:**

**Evalueer**(expression\_text)

**Retourgegevenstypen:** dual



*Deze tekenreeksfunctie kan niet in diagramuitdrukkingen worden gebruikt.*

Voorbeelden en resultaten:

Functievoorbeld	Resultaat
Evalueer ( 5 * 8 )	Retourneert '40'

### Voorbeeld van load-script

```
Load Evaluate(String) as Evaluated, String Inline [String 4 5+3 0123456789012345678 Today()];
```

**Resultaat**

Tekenreeks	Geëvalueerd
4	4
5+3	8
0123456789012345678	0123456789012345678
Today()	2022-02-02

## FindOneOf

**FindOneOf()** doorzoekt een tekenreeks om de positie te vinden van enig teken uit de reeks van opgegeven tekens. De positie waarin enig teken uit de te doorzoeken set voor het eerst voorkomt wordt geretourneerd tenzij een derde argument (met een waarde groter dan 1) wordt opgegeven. Als geen waarde wordt gevonden, wordt **0** geretourneerd.

### Syntaxis:

```
FindOneOf(text, char_set[, count])
```

**Retourgegevenstypen:** geheel getal

### Argumenten:

#### Argumenten

Argument	Beschrijving
text	De oorspronkelijke tekenreeks.
char_set	Een reeks tekens om naar te zoeken in text.
count	Definieert naar welk exemplaar van een teken moet worden gezocht. Bijvoorbeeld, bij een waarde van 2 wordt naar het tweede exemplaar gezocht.

### Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
FindoneOf( 'my example text string', 'et%s')	Retourneert '4' omdat 'e' het vierde teken is in de voorbeeldtekenreeks.
FindoneOf( 'my example text string', 'et%s', 3)	Retourneert '12' omdat wordt gezocht naar een van de tekens e, t, % of s, en t het derde exemplaar is op positie 12 van de voorbeeldtekenreeks.
FindoneOf( 'my example text string', 'æ%&')	Retourneert '0' omdat geen van de tekens æ, % of & voorkomt in de voorbeeldtekenreeks.

### Voorbeeld: Load-script

```
Load * Inline [SearchFor, occurrence et%s,1 et%s,3 æ%&,1]
```

### Resultaat

SearchFor	Exemplaar	FindOneOf('my example text string', SearchFor, Occurrence)
et%s	1	4
et%s	3	12
æ%&	1	0

## Hash128

**Hash128()** retourneert een 128-bits hash van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 22 tekens.

### Syntaxis:

```
Hash128(expr{, expression})
```

**Retourgegevenstypen:** tekenreeks

Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
Hash128 ('abc', 'xyz', '123')	Retourneert 'MA&5]6+3=:>:>G%S<U*S2+'.
Hash128 ( Region, Year, Month )	Retourneert 'G7*=6GKPJ(Z+)^KM?<\$A+'.
Note: Region, Year, and Month are table fields.	

Voorbeeld: Load-script

```
Hash_128: Load *, Hash128(Region, Year, Month) as Hash128; Load * inline [ Region, Year, Month abc, xyz, 123 EU, 2022, 01 UK, 2022, 02 US, 2022, 02 ];
```

### Resultaat

Regio	Jaar	Maand	Hash128
abc	xyz	123	MA&5]6+3=:>:>G%S<U*S2+
EU	2022	01	B40^K&[T@!;VB'XR]<5=/\$
UK	2022	02	O5T;+1?[B&"F&1//MA[MN!
US	2022	02	C6@#]4#_G-(J7EQY#KRW0

## Hash160

**Hash160()** retourneert een 160-bits hash van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 27 tekens.

### Syntaxis:

```
Hash160(expr{, expression})
```

**Retourgegevenstypen:** tekenreeks

Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
Hash160 ('abc', 'xyz', '123')	Retourneert 'MA&5]6+3=:;>G%S<U*S2!:'=X*.
Hash160 ( Region, Year, Month ) Note: Region, Year, and Month are table fields.	Retourneert 'G7*=6GKPJ (Z+)^KM?<\$'AI.)?U\$'.

Voorbeeld: Load-script

```
Hash_160: Load *, Hash160(Region, Year, Month) as Hash160; Load * inline [ Region, Year, Month abc, xyz, 123 EU, 2022, 01 UK, 2022, 02 US, 2022, 02 ];
```

**Resultaat**

Regio	Jaar	Maand	Hash160
abc	xyz	123	MA&5]6+3=:;>G%S<U*S2!:'=X*
EU	2022	01	B40^K&[T@!;VB'XR]<5=//_F853
UK	2022	02	O5T;+1?[B&"F&1//MA[MN!T"FWZ
US	2022	02	C6@#]4#_G-(J7EQY#KRW`@KF+W

## Hash256

**Hash256()** retourneert een 256-bits hash van de gecombineerde invoerwaarden van de uitdrukking. Het resultaat is een tekenreeks van 43 tekens.

**Syntaxis:**

```
Hash256 (expr{, expression})
```

**Retourgegevenstypen:** tekenreeks

Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
Hash256 ('abc', 'xyz', '123')	Retourneert 'MA&5]6+3=:;>G%S<U*S2!:'=X*A.IO*8N\%Y7Q;YEJ'.
Hash256 ( Region, Year, Month ) Note: Region, Year, and Month are table fields.	Retourneert 'G7*=6GKPJ(Z+)^KM?<\$'AI.)?U\$#X2RB [:0ZP=+Z'F:'.



Voorbeeld: Load-script

```
Hash_256: Load *, Hash256(Region, Year, Month) as Hash256; Load * inline [ Region, Year, Month abc, xyz, 123 EU, 2022, 01 UK, 2022, 02 US, 2022, 02 ];
```

### Resultaat

Regio	Jaar	Maand	Hash256
abc	xyz	123	MA&5]6+3=;>;>G%S<U*S2!:`=X*A.IO*8N\%Y7Q;YEJ
EU	2022	01	B40^K&[T@!;VB'XR]<5=//_F853?BE6'G&,YH*T'MF)
UK	2022	02	O5T;+1?[B&"F&1//MA[MN!T"FWZT=4\#V'M%6_\0C>4
US	2022	02	C6@#]4#_G-(]J7EQY#KRW`@KF+W-0]'[Z8R+#")=+0

## Index

**Index()** doorzoekt een tekenreeks om de beginpositie te vinden van de n-de keer dat een opgegeven subtekenreeks voorkomt. Een optioneel derde argument levert de waarde van n, die 1 is als het argument wordt weggelaten. Bij een negatieve waarde wordt gezocht vanaf het einde van de tekenreeks. De nummering van de posities in de tekenreeks begint bij 1.

### Syntaxis:

```
Index(text, substring[, count])
```

**Retourgegevenstypen:** geheel getal

### Argumenten:

#### Argumenten

Argument	Beschrijving
text	De oorspronkelijke tekenreeks.
substring	Een reeks van tekens om naar te zoeken in text.
count	Definieert naar welk exemplaar van <b>substring</b> moet worden gezocht. Bijvoorbeeld, bij een waarde van 2 wordt naar het tweede exemplaar gezocht.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
Index('abcdefg', 'cd')	retourneert 3
Index('abcdabcd', 'b', 2)	Retourneert 6 (het tweede exemplaar van 'b')
Index('abcdabcd', 'b', -2)	Retourneert 2 (het tweede exemplaar van 'b' te beginnen vanaf het einde)

Voorbeeld	Resultaat
Left( Date, Index( Date, '-' ) -1 ) where Date = 1997-07-14	retourneert 1997
Mid( Date, Index( Date, '-', 2 ) -2, 2 ) where Date = 1997-07-14	retourneert 07

### Voorbeeld: Script

```
T1: Load *, index(String, 'cd') as Index_CD, // returns 3 in Index_CD index
(String, 'b') as Index_B, // returns 2 in Index_B index(String, 'b', -1) as
Index_B2; // returns 2 or 6 in Index_B2 Load * inline [ String abcdefg abcdabcd ];
```

## IsJson

**IsJson()** test of een opgegeven tekenreeks geldige JSON-gegevens (JavaScript Object Notation) bevat. U kunt ook een specifiek JSON-gegevenstype valideren.

### Syntaxis:

```
value IsJson(json [, type])
```

### Retourgegevenstypen: dual

#### Argumenten

Argument	Beschrijving
json	Tekenreeks om te testen. Deze kan extra spaties of nieuwe regels bevatten.
type	Optioneel argument dat het JSON-gegevenstype aangeeft dat getest moet worden. <ul style="list-style-type: none"> <li>'waarde' (standaard)</li> <li>'object'</li> <li>'array'</li> <li>'tekenreeks'</li> <li>'getal'</li> <li>'Booleaanse waarde'</li> <li>'null'</li> </ul>

### Voorbeeld: Geldige JSON en type

Voorbeeld	Resultaat
IsJson('null')	Retourneert -1 (true)
IsJson('"abc"', 'value')	Retourneert -1 (true)
IsJson('"abc"', 'string')	Retourneert -1 (true)
IsJson(123, 'number')	Retourneert -1 (true)

Voorbeeld: Ongeldige JSON of type

Voorbeeld	Resultaat	Beschrijving
<code>IsJson('text')</code>	Retourneert 0 (false)	'text' is geen geldige JSON-waarde
<code>IsJson('"text"', 'number')</code>	Retourneert 0 (false)	""text"" is geen geldig JSON-getal
<code>IsJson('"text"', 'text')</code>	Retourneert 0 (false)	'text' is geen geldig JSON-type

### JsonGet

**JsonGet()** retourneert het pad van een JSON-tekenreeks met gegevens (JavaScript Object Notation). De JSON-gegevens moeten een geldige JSON zijn, maar kunnen extra spaties of nieuwe regels bevatten.

#### Syntaxis:

```
value JsonGet(json, path)
```

**Retourgegevenstypen:** dual

#### Argumenten

Argument	Beschrijving
json	Tekenreeks met JSON-gegevens.
path	Het pad moet zijn gespecificeerd volgens <a href="#">RFC 6901</a> . Dit maakt zoeken naar eigenschappen in json-gegevens mogelijk zonder gebruik van complexe subtekenreeks- of indexfuncties.

Voorbeeld: Geldig(e) JSON en pad

Voorbeeld	Resultaat
<code>JsonGet('{ "a": {"foo": "bar"}, "b": [123, "abc", "ABC"] }', '')</code>	Retourneert '{ "a": {"foo": "bar"}, "b": [123, "abc", "ABC"] }'
<code>JsonGet('{ "a": {"foo": "bar"}, "b": [123, "abc", "ABC"] }', '/a')</code>	Retourneert '{ "foo": "bar" }'
<code>JsonGet('{ "a": {"foo": "bar"}, "b": [123, "abc", "ABC"] }', '/a/foo')</code>	Retourneert ""bar""
<code>JsonGet('{ "a": {"foo": "bar"}, "b": [123, "abc", "ABC"] }', '/b')</code>	Retourneert '[123, "abc", "ABC"]'
<code>JsonGet('{ "a": {"foo": "bar"}, "b": [123, "abc", "ABC"] }', '/b/0')</code>	Retourneert '123'
<code>JsonGet('{ "a": {"foo": "bar"}, "b": [123, "abc", "ABC"] }', '/b/1')</code>	Retourneert ""abc""
<code>JsonGet('{ "a": {"foo": "bar"}, "b": [123, "abc", "ABC"] }', '/b/2')</code>	Retourneert ""ABC""

Voorbeeld: Ongeldig(e) JSON of type

Voorbeeld	Resultaat	Beschrijving
<code>JsonGet('{ "a": "b" }', '/b')</code>	Retourneert null	Het pad verwijst niet naar een geldig deel van de JSON-gegevens.
<code>JsonGet('{ "a" }', '/a')</code>	Retourneert null	De json-gegevens zijn geen geldige JSON (lid "a" bevat geen waarde).

## JsonSet

**JsonSet()** wijzigt een tekenreeks die JSON-gegevens (JavaScript Object Notation) bevat. Het kan een JSON-waarde instellen of invoegen met de nieuwe locatie die is opgegeven door het pad. De JSON-gegevens moeten een geldige JSON zijn, maar kunnen extra spaties of nieuwe regels bevatten.

### Syntaxis:

```
value JsonSet(json, path, value)
```

Retourgegevenstypen: dual

#### Argumenten

Argument	Beschrijving
json	Tekenreeks met JSON-gegevens.
path	Het pad moet zijn gespecificeerd volgens <a href="#">RFC 6901</a> . Dit maakt opbouwen van eigenschappen in json-gegevens mogelijk zonder gebruik van complexe subtekenreeks- of indexfuncties en aaneenschakeling.
value	De nieuwe tekenreekswaarde in JSON-indeling.

Voorbeeld: Geldig(e) JSON, pad en waarde

Voorbeeld	Resultaat
<code>JsonSet('{}', '/a', '"b"')</code>	Retourneert '{"a": "b"}'
<code>JsonSet('[]', '/0', '"x"')</code>	Retourneert '["x"]'
<code>JsonSet('"abc"', '', '123')</code>	Retourneert 123

Voorbeeld: Ongeldig(e) JSON, pad of waarde

Voorbeeld	Resultaat	Beschrijving
<code>JsonSet('"abc"', '/x', '123')</code>	Retourneert null	Het pad verwijst niet naar een geldig deel van de JSON-gegevens.

Voorbeeld	Resultaat	Beschrijving
<code>JsonSet({'a': {'b': 'c'}}', 'a/b', 'x')</code>	Retourneert null	Het pad is ongeldig.
<code>JsonSet({'a': 'b'}, '/a', 'abc')</code>	Retourneert null	De waarde is geen geldige JSON. Een tekenreeks moet tussen aanhalingstekens worden geplaatst.

### KeepChar

**KeepChar()** retourneert een tekenreeks die bestaat uit de eerste tekenreeks, 'text', behalve de tekens die NIET voorkomen in de tweede tekenreeks, "keep\_chars".

#### Syntaxis:

```
KeepChar (text, keep_chars)
```

**Retourgegevenstypen:** tekenreeks

#### Argumenten:

##### Argumenten

Argument	Beschrijving
text	De oorspronkelijke tekenreeks.
keep_chars	Een tekenreeks die de tekens in text bevat die moeten worden bewaard.

Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
<code>KeepChar ( 'a1b2c3', '123' )</code>	Retourneert '123'.
<code>KeepChar ( 'a1b2c3', '1234' )</code>	Retourneert '123'.
<code>KeepChar ( 'a1b22c3', '1234' )</code>	Retourneert '1223'.
<code>KeepChar ( 'a1b2c3', '312' )</code>	Retourneert '123'.

Voorbeeld: Load-script

```
T1: Load *, keepchar(String1, String2) as KeepChar; Load * inline [ String1, String2 'a1b2c3', '123' ];
```

#### Resultaten

Qlik Sense-tabel met de uitvoer van het gebruik van de functie *KeepChar* in het load-script.

String1	String2	KeepChar
a1b2c3	123	123

### Zie ook:

p *PurgeChar* (page 1444)

## Left

**Left()** retourneert een tekenreeks die bestaat uit de eerste (meest linker) van de invoertekenreeks, waarbij het aantal tekens wordt bepaald door het tweede argument.

### Syntaxis:

```
Left(text, count)
```

**Retourgegevenstypen:** tekenreeks

### Argumenten:

Argument	Beschrijving
text	De oorspronkelijke tekenreeks.
count	Definieert het aantal tekens dat moet worden opgenomen vanaf de linkerkant van de tekenreeks <b>text</b> .

Voorbeeld: Diagramuitdrukking

Voorbeeld	Resultaat
<code>Left('abcdef', 3)</code>	Retourneert 'abc'

Voorbeeld: Load-script

```
T1: Load *, left(Text,Start) as Left;           Load * inline [ Text, Start 'abcdef', 3 '2021-07-14', 4 '2021-07-14', 2 ];
```

### Resultaat

Qlik Sense-tabel met de uitvoer van het gebruik van de functie *Left* in het load-script.

Tekst	Start	Left
abcdef	3	abc
2021-07-14	4	2021
2021-07-14	2	20

p Zie *Index* (page 1433), waardoor complexere tekenreeksanalyse mogelijk is.

## Len

**Len()** retourneert de lengte van de invoertekenreeks.

### Syntaxis:

```
Len(text)
```

**Retourgegevenstypen:** geheel getal

Voorbeeld: Diagramuitdrukking

Voorbeeld	Resultaat
Len('Peter')	Retourneert '5'

Voorbeeld: Load-script

```
T1: Load String, First&Second as NewString; Load *, mid(String,len(First)+1) as Second; Load *, upper(left(String,1)) as First; Load * inline [ String this is a sample text string capitalize first letter only ];
```

### Resultaat

Tekenreeks	NieuweTekenreeks
this is a sample text string	This is a sample text string
capitalize first letter only	Capitalize first letter only

## LevenshteinDist

**LevenshteinDist()** retourneert de afstand van Levenshtein tussen twee tekenreeksen. Het wordt gedefinieerd als het aantal bewerkingen van één teken (invoegingen, verwijderingen of vervangingen) dat is vereist om een tekenreeks te wijzigen. De functie kan worden gebruikt om fuzzy tekenreeksen te vergelijken.

### Syntaxis:

```
LevenshteinDist(text1, text2)
```

**Retourgegevenstypen:** geheel getal

Voorbeeld: Diagramuitdrukking

Voorbeeld	Resultaat
LevenshteinDist('kitten','sitting')	retourneert 3

Voorbeeld: Load-script

### Load-script

```
T1: Load *, recno() as ID; Load 'Silver' as String_1,* inline [ String_2 Sliver SSiver SSiveer ]; T1: Load *, recno()+3 as ID; Load 'Gold' as String_1,* inline [ String_2 Bold Bool Bond ]; T1: Load *, recno()+6 as ID; Load 'Ove' as String_1,* inline [ String_2 Ove Uve Üve ]; T1: Load *, recno()+9 as ID; Load 'ABC' as String_1,* inline [ String_2 DEFG abc ビビビ ]; set nullinterpret = '<NULL>'; T1: Load *, recno()+12 as ID; Load 'X' as String_1,* inline [ String_2 '' <NULL> 1 ]; R1: Load ID, String_1, String_2, LevenshteinDist(String_1, String_2) as LevenshteinDistance resident T1; Drop table T1;
```

### Resultaat

Id	String_1	String_2	LevenshteinDistance
1	Silver	Sliver	2
2	Silver	SSiver	2
3	Silver	SSiveer	3
4	Gold	Bold	1
5	Gold	Bool	3
6	Gold	Bond	2
7	Ove	Ove	0
8	Ove	Uve	1
9	Ove	Üve	1
10	ABC	DEFG	4
11	ABC	abc	3
12	ABC	ビビビ	3
13	X		1
14	X	-	1
15	X	1	1

### Lower

**Lower()** converteert alle tekens in de invoertekenreeks naar kleine letters.

#### Syntaxis:

```
Lower (text)
```



**Retourgegevenstypen:** tekenreeks

Voorbeeld: Diagramuitdrukking

Voorbeeld	Resultaat
Lower('abcd')	Retourneert 'abcd'

Voorbeeld: Load-script

```
Load String, Lower(String) Inline [String rHode iSland washingTon d.C. new york];
```

**Resultaat**

Tekenreeks	Lower(String)
rHode iSland	rhode island
washingTon d.C.	washington d.c.
new york	new york

### LTrim

**LTrim()** retourneert de invoertekenreeks, ontdaan van beginspaties.

**Syntaxis:**

**LTrim**(text)

**Retourgegevenstypen:** tekenreeks

Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
LTrim( ' abc' )	Retourneert 'abc'
LTrim( 'abc ' )	Retourneert 'abc '

Voorbeeld: Load-script

```
Set verbatim=1; T1: Load *, len(LtrimString) as LtrimStringLength; Load *, ltrim
(String) as LtrimString; Load *, len(String) as StringLength; Load * Inline [
string ' abc ' ' def '];
```



*De opdracht "Set verbatim=1" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de ltrim-functie wordt getoond. Raadpleeg Verbatim (page 200) voor meer informatie.*

**Resultaat**

Tekenreeks	StringLength	LtrimStringLength
def	6	5
abc	10	7

**Zie ook:**

p *RTrim* (page 1447)

**Mid**

**Mid()** retourneert het gedeelte van de invoertekenreeks dat begint op de positie van het teken dat is gedefinieerd door het tweede argument, 'start', en retourneert het aantal tekens dat is gedefinieerd door het derde argument, 'count'. Als 'count' wordt weggelaten, wordt de rest van de invoertekenreeks geretourneerd. Het eerste teken in de invoertekenreeks heeft het nummer 1.

**Syntaxis:**

```
Mid(text, start[, count])
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

## Argumenten

Argument	Beschrijving
text	De oorspronkelijke tekenreeks.
start	Geheel getal dat de positie definieert van het eerste teken in text dat moet worden opgenomen.
count	Definieert de tekenreekslengte van de uitvoertekenreeks. Bij weglating, worden alle tekens opgenomen vanaf de positie die is gedefinieerd door <b>start</b> .

## Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
Mid('abcdef',3 )	Retourneert 'cdef'
Mid('abcdef',3, 2 )	Retourneert 'cd'

## Voorbeeld: Load-script

```
T1: Load *, mid(Text,Start) as Mid1, mid(Text,Start,Count) as Mid2; Load *
inline [ Text, Start, Count 'abcdef', 3, 2 'abcdef', 2, 3 '210714', 3, 2 '210714', 2, 3 ];
```

### Resultaat

Qlik Sense-tabel met de uitvoer van het gebruik van de functie *Mid* in het load-script.

Tekst	Start	Mid1	Aantal	Mid2
abcdef	2	bcdef	3	bcd
abcdef	3	cdef	2	cd
210714	2	10714	3	107
210714	3	0714	2	07

### Zie ook:

[p Index \(page 1433\)](#)

## Ord

**Ord()** retourneert het aantal punten voor de Unicode-code point van het eerste teken van de invoertekenreeks. Unicode

### Syntaxis:

**Ord** (text)

**Retourgegevenstypen:** geheel getal

Voorbeelden en resultaten:

### Voorbeeld: Diagramuitdrukking

Voorbeeld	Resultaat
<code>ord('A')</code>	Retourneert het gehele getal 65.
<code>ord('Ab')</code>	Retourneert het gehele getal 65.

### Voorbeeld: Load-script

```
//Guqin (Chinese: 古琴) - 7-stringed zithers T2: Load *, ord(Chinese) as OrdUnicode,  
ord(Western) as OrdASCII; Load * inline [ Chinese, western 古琴,  
guqin ];  
Resultaat
```

Chinees	Westers	OrdASCII	OrdUnicode
古琴	Guqin	71	21476

## PurgeChar

**PurgeChar()** retourneert een tekenreeks die bestaat uit de tekens die zijn opgenomen in de invoertekenreeks ('text'), behalve de tekens die voorkomen in het tweede argument ('remove\_chars').

### Syntaxis:

```
PurgeChar(text, remove_chars)
```

**Retourgegevenstypen:** tekenreeks

### Argumenten:

#### Argumenten

Argument	Beschrijving
text	De oorspronkelijke tekenreeks.
remove_chars	Een tekenreeks die de tekens in text bevat die moeten worden verwijderd.

**Retourgegevenstypen:** tekenreeks

Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
PurgeChar ( 'a1b2c3', '123' )	Retourneert 'abc'.
PurgeChar ( 'a1b2c3', '312' )	Retourneert 'abc'.

Voorbeeld: Load-script

```
T1: Load *, purgechar(String1, String2) as PurgeChar; Load * inline [ String1, String2 'a1b2c3', '123' ];
```

### Resultaten

Klik Sense-tabel met de uitvoer van het gebruik van de functie *PurgeChar* in het load-script.

String1	String2	PurgeChar
a1b2c3	123	abc

### Zie ook:

p *KeepChar* (page 1437)

## Repeat

**Repeat()** vormt een tekenreeks die bestaat uit de invoertekenreeks die het aantal keren wordt herhaald dat wordt gedefinieerd door het tweede argument.

### Syntaxis:

```
Repeat (text[, repeat_count])
```

Retourgegevenstypen: tekenreeks

### Argumenten:

#### Argumenten

Argument	Beschrijving
text	De oorspronkelijke tekenreeks.
repeat_count	Definieert het aantal keren dat de tekens in de tekenreeks <b>text</b> moet worden herhaald in de uitvoertekenreeks.

Voorbeeld: Diagramuitdrukking

Voorbeeld	Resultaat
Repeat( ' * ', rating ) when rating = 4	Retourneert '****'

Voorbeeld: Load-script

```
T1: Load *, repeat(String,2) as Repeat; Load * inline [ String hello world! hOw aRe you? ];
```

### Resultaat

Tekenreeks	Herhalen
hello world!	hello world!hello world!
hOw aRe you?	hOw aRe you?hOw aRe you?

## Replace

**Replace()** retourneert een tekenreeks nadat alle treffers van een bepaalde subtekenreeks in de tekenreeks zijn vervangen door een andere subtekenreeks. De functie is niet-recursief en wordt van links naar rechts uitgevoerd.

### Syntaxis:

```
Replace (text, from_str, to_str)
```

Retourgegevenstypen: tekenreeks

### Argumenten:

#### Argumenten

Argument	Beschrijving
text	De oorspronkelijke tekenreeks.

Argument	Beschrijving
from_str	Een tekenreeks die een of meer keren in de invoerreeks <b>text</b> mag voorkomen.
to_str	De tekenreeks die alle treffers van <b>from_str</b> in de tekenreeks <b>text</b> vervangt.

Voorbeelden en resultaten:

Voorbeeld	Resultaat
<code>replace('abccde', 'cc', 'xyz')</code>	Retourneert 'abxyzde'

Zie ook:

### Right

**Right()** retourneert een tekenreeks die bestaat uit de laatste tekens (meest rechts) van de invoertekenreeks, waarbij het aantal tekens wordt bepaald door het tweede argument.

**Syntaxis:**

```
Right(text, count)
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argumenten

Argument	Beschrijving
text	De oorspronkelijke tekenreeks.
count	Definieert het aantal tekens dat moet worden opgenomen vanaf de rechterkant van de tekenreeks <b>text</b> .

Voorbeeld: Diagramuitdrukking

Voorbeeld	Resultaat
<code>Right('abcdef', 3)</code>	Retourneert 'def'

Voorbeeld: Load-script

```
T1: Load *, right(Text,Start) as Right;           Load * inline [ Text, Start 'abcdef', 3
'2021-07-14', 4 '2021-07-14', 2 ];
```

### Resultaat

Qlik Sense-tabel met de uitvoer van het gebruik van de functie *Right* in het load-script.

Tekst	Start	Right
abcdef	3	def
2021-07-14	4	7-14
2021-07-14	2	14

### RTrim

**RTrim()** retourneert de invoertekenreeks, ontdaan van eindspaties.

#### Syntaxis:

**RTrim**(text)

**Retourgegevenstypen:** tekenreeks

Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
RTrim( ' abc' )	Retourneert 'abc'
RTrim( 'abc ' )	Retourneert 'abc'

Voorbeeld: Load-script

```
set verbatim=1; T1: Load *, len(RtrimString) as RtrimStringLength; Load *, rtrim
(String) as RtrimString; Load *, len(String) as StringLength; Load * Inline [
string ' abc ' ' def '];
```



*De opdracht "Set verbatim=1" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de rtrim-functie wordt getoond. Raadpleeg Verbatim (page 200) voor meer informatie.*

### Resultaat

Tekenreeks	StringLength	RtrimStringLength
def	6	4
abc	10	6

#### Zie ook:

p *LTrim* (page 1441)

## SubField

**Subfield()** wordt gebruikt voor het extraheren van onderdelen van subtekenreeksen van een bovenliggend tekenreeksveld, waarbij de oorspronkelijke recordvelden bestaat uit twee of meer onderdelen met een scheidingsteken ertussen.

De functie **Subfield()** kan bijvoorbeeld worden gebruikt om voornaam en achternaam te extraheren vanuit een lijst met records die bestaat uit volledige namen, de samenstellende delen van een padnaam of voor het extraheren van gegevens uit tabellen met door komma's gescheiden waarden.

Als u de functie **Subfield()** gebruikt in een opdracht **LOAD** terwijl de optionele parameter `field_no` wordt weggelaten, wordt één complete record gegenereerd voor elke subtekenreeks. Als meerdere velden worden geladen met **Subfield()**, worden de cartesische producten van alle combinaties gemaakt.

### Syntaxis:

```
SubField(text, delimiter[, field_no ])
```

**Retourgegevenstypen:** tekenreeks

### Argumenten:

#### Argumenten

Argument	Beschrijving
text	De oorspronkelijke tekenreeks. Dit kan een hard gecodeerde tekst, een variabele, een uitbreiding met dollarteken of een andere uitdrukking zijn.
delimiter	Een teken binnen de invoerreeks <b>text</b> die de tekenreeks opsplijst in samenstellende delen.
field_no	Het optionele derde argument is een geheel getal waarmee wordt opgegeven welke van de subtekenreeksen van de bovenliggende tekenreeks <b>text</b> moeten worden geretourneerd. Gebruik de waarde 1 om terug te keren naar de eerste subtekenreeks, 2 om terug te keren naar de tweede subtekenreeks, enzovoorts. <ul style="list-style-type: none"> <li>Als <b>field_no</b> een positieve waarde is, worden subtekenreeksen van links naar rechts geëxtraheerd.</li> <li>Als <b>field_no</b> een negatieve waarde is, worden subtekenreeksen van rechts naar links geëxtraheerd.</li> </ul>



*SubField() kan worden gebruikt in plaats van complexe combinaties van functies zoals Len(), Right(), Left(), Mid() en andere tekenreeksfuncties.*



### Voorbeelden: Script- en diagramuitdrukkingen die gebruikmaken van SubField

Voorbeelden - script- en diagramuitdrukkingen

#### Basisvoorbeelden

Voorbeeld	Resultaat
<code>SubField(S, ';' ,2)</code>	Retourneert 'cde' als <b>S</b> is 'abc;cde;efg'.
<code>subField(S, ';' ,1)</code>	Retourneert een lege tekenreeks als <b>S</b> een lege tekenreeks is.
<code>SubField(S, ';' ,1)</code>	Retourneert een lege tekenreeks als <b>S</b> ';' is.
<p>Stel dat u een variabele hebt met een padnaam <code>vMyPath</code>,</p> <pre>set vMyPath=\Users\ext_ jrb\Documents\Qlik\Sense\Apps;.</pre>	<p>In een diagram met tekst en afbeelding kunt u een meting toevoegen, zoals:  <code>subField(vMyPath, '\', -3)</code>, met het resultaat 'Qlik', omdat het de derde subtekenreeks vanaf het rechteruiteinde van de variabele <code>vMyPath</code> is.</p>

#### Scriptvoorbeeld 1

##### Load-script

Laad de volgende scriptuitdrukkingen en gegevens in de editor voor het laden van gegevens.

```
FullName:                               LOAD * inline [ Name 'Dave Owen' 'Joe Tem' ]; SepNames: LO
(Name, ' ',1) as FirstName, SubField(Name, ' ',-1) as SurName Resident FullName; Drop Table
FullName;
```

##### Een visualisatie maken

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Name**, **FirstName** en **SurName** als dimensies.

##### Resultaat

Name	FirstName	SurName
Dave Owen	Dave	Owen
Joe Tem	Joe	Tem

##### Uitleg

De functie **SubField()** haalt de eerste subtekenreeks op uit **Name** door het argument **field\_no** in te stellen op 1. Aangezien de waarde van **field\_no** positief is, wordt de volgorde links naar rechts gevolgd voor het ophalen van de subtekenreeks. Een tweede functieaanroep haalt de tweede subtekenreeks op door het argument **field\_no** in te stellen op -1, waardoor de subtekenreeks wordt opgehaald volgens de volgorde rechts naar links.

#### Scriptvoorbeeld 2

##### Load-script

Laad de volgende scriptuitdrukkingen en gegevens in de editor voor het laden van gegevens.

```
LOAD DISTINCT Instrument, SubField(Player,',') as Player, SubField(Project,',') as Project;  
Load * inline [ Instrument|Player|Project Guitar|Neil, Mike|Music, Video Guitar|Neil|Music, OST  
Synth|Neil, Jen|Music, Video, OST Synth|Jo|Music Guitar|Neil, Mike|Music, OST ] (delimiter is '|');
```

### Een visualisatie maken

Maak een tabelvisualisatie in een Qlik Sense-werkblad met **Instrument**, **Player** en **Project** als dimensies.

#### Resultaat

Instrument	Player	Project
Guitar	Mike	Music
Guitar	Mike	Video
Guitar	Mike	OST
Guitar	Neil	Music
Guitar	Neil	Video
Guitar	Neil	OST
Synth	Jen	Music
Synth	Jen	Video
Synth	Jen	OST
Synth	Jo	Music
Synth	Neil	Music
Synth	Neil	Video
Synth	Neil	OST

#### Uitleg

Dit voorbeeld laat zien dat als u meerdere exemplaren van de functie **Subfield()** gebruikt, elk met de parameter `field_no` weggelaten, binnen in dezelfde **LOAD**-opdracht, Cartesiaanse producten worden gemaakt van alle combinaties. De optie **DISTINCT** wordt gebruikt om te voorkomen dat dubbele records worden gemaakt.

## SubStringCount

**SubstringCount()** retourneert het aantal keren dat de opgegeven subtekenreeks voorkomt in de tekst van de invoertekenreeks. Als er geen overeenkomst wordt gevonden, wordt 0 geretourneerd.

#### Syntaxis:

```
SubStringCount(text, sub_string)
```

**Retourgegevenstypen:** geheel getal

**Argumenten:**

Argument	Beschrijving
text	De oorspronkelijke tekenreeks.
sub_string	Een tekenreeks die een of meer keren in de invoerreeks <b>text</b> mag voorkomen.

Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
substringCount ( 'abcdefgdcxyz', 'cd' )	Retourneert '2'
substringCount ( 'abcdefgdcxyz', 'dc' )	Retourneert '0'

Voorbeeld: Load-script

```
T1: Load *, substringcount(upper(Strings),'AB') as SubStringCount_AB; Load * inline [ Strings  
ABC:DEF:GHI:AB:CD:EF:GH aB/cd/ef/gh/Abc/abandoned ];
```

**Resultaat**

Tekenreeksen	SubStringCount_AB
aB/cd/ef/gh/Abc/abandoned	3
ABC:DEF:GHI:AB:CD:EF:GH	2

## TextBetween

**TextBetween()** retourneert de tekst in de invoertekenreeks die tussen de tekens staat die zijn opgegeven als scheidingstekens.

**Syntaxis:**

```
TextBetween(text, delimiter1, delimiter2[, n])
```

**Retourgegevenstypen:** tekenreeks

**Argumenten:**

Argument	Beschrijving
text	De oorspronkelijke tekenreeks.
delimiter1	Geeft het eerste scheidingsteken (of de eerste scheidingsreeks) op waarnaar moet worden gezocht in <b>text</b> .

Argument	Beschrijving
delimiter2	Geeft het tweede scheidingsteken (of de tweede scheidingsreeks) op waarnaar moet worden gezocht in <b>text</b> .
n	Definieert tussen welk exemplaar van het paar scheidingstekens moet worden gezocht. Een waarde van 2 retourneert bijvoorbeeld de tekens tussen het tweede exemplaar van delimiter1 en het tweede exemplaar van delimiter2.

Voorbeeld: Diagramuitdrukkingen

Voorbeeld	Resultaat
<code>TextBetween('&lt;abc&gt;', '&lt;', '&gt;')</code>	Retourneert 'abc'
<code>TextBetween('&lt;abc&gt;&lt;de&gt;', '&lt;', '&gt;', 2)</code>	Retourneert 'de'
<code>TextBetween('abc', '&lt;', '&gt;')</code> <code>TextBetween('&lt;a&lt;b', '&lt;', '&gt;')</code>	Beide voorbeelden retourneren NULL. Als één van de scheidingstekens niet wordt gevonden in de tekenreeks, wordt NULL geretourneerd.
<code>TextBetween('&lt;&gt;', '&lt;', '&gt;')</code>	Retourneert een tekenreeks met lengte nul.
<code>TextBetween('&lt;abc&gt;', '&lt;', '&gt;', 2)</code>	Retourneert NULL, want n is groter dan het aantal scheidingstekens.

Voorbeeld: Load-script

```
Load *, textbetween(Text, '<', '>') as TextBetween, textbetween(Text, '<', '>', 2) as
SecondTextBetween; Load * inline [ Text <abc><de> <def><ghi><jkl> ];
```

**Resultaat**

Tekst	TextBetween	SecondTextBetween
<abc><de>	abc	de
<def><ghi><jkl>	def	ghi

## Trim

**Trim()** retourneert de invoertekenreeks, ontdaan van alle begin- en eindspaties.

**Syntaxis:**

```
Trim(text)
```

**Retourgegevenstypen:** tekenreeks

Voorbeelden en resultaten:

**Voorbeeld: Diagramuitdrukking**

Voorbeeld	Resultaat
<code>Trim( ' abc' )</code>	Retourneert 'abc'
<code>Trim( 'abc ' )</code>	Retourneert 'abc'
<code>Trim( ' abc ' )</code>	Retourneert 'abc'

**Voorbeeld: Load-script**

```
set verbatim=1;                                T1: Load *, len(TrimString) as TrimStringLength;
(String) as TrimString; Load *, len(String) as StringLength;          Load * inline [
string ' abc ' ' def '](delimiter is '\t');
```



*De opdracht "Set verbatim=1" is opgenomen in het voorbeeld om ervoor te zorgen dat de ruimten niet automatisch worden afgekapt voordat de trim-functie wordt getoond. Raadpleeg Verbatim (page 200) voor meer informatie.*

Resultaat

Tekenreeks	StringLength	TrimStringLength
def	6	3
abc	10	3

## Upper

**Upper()** converteert alle tekens in de invoertekenreeks naar hoofdletters voor alle tekstekens in de uitdrukking. Cijfers en symbolen worden genegeerd.

**Syntaxis:**

**Upper** (text)

**Retourgegevenstypen:** tekenreeks

Voorbeeld: Diagramuitdrukking

Voorbeeld	Resultaat
<code>Upper(' abcd')</code>	Retourneert 'ABCD'

Voorbeeld: Load-script

```
Load string,Upper(String) Inline [String rHode isLand washingTon d.c. new york];
```

### Resultaat

Tekenreeks	Upper(String)
rHode iSland	RHODE ISLAND
washingTon d.C.	WASHINGTON D.C.
new york	NEW YORK

## 5.25 Systeemfuncties

Systeemfuncties zijn functies voor het benaderen van eigenschappen van het systeem, het apparaat en de Qlik Sense-app.

### Overzicht van systeemfuncties

Sommige functies worden na het overzicht nader beschreven. Bij deze functies kunt u ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Author()

Deze scriptfunctie retourneert een tekenreeks met de titel van de huidige app. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.



*De eigenschap Author kan niet worden ingesteld in de huidige versie van Qlik Sense. Als u een QlikView-document migreert, blijft de eigenschap Author behouden.*

#### ClientPlatform()

Deze functie retourneert de tekenreeks voor de gebruikersagent van de clientbrowser. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

#### Voorbeeld:

```
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/35.0.1916.114 Safari/537.36
```

#### ComputerName

Deze functie retourneert een tekenreeks met de naam van de computer volgens het besturingssysteem. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.



*Als de naam van de computer meer dan 15 tekens bevat, bevat de reeks alleen de eerste 15 tekens.*

#### ComputerName ( )

### DocumentName

Deze functie retourneert een tekenreeks met de naam van het huidige Qlik Sense-document, zonder pad maar met de extensie. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
DocumentName ( )
```

### DocumentPath

Deze functie retourneert een tekenreeks met het volledige pad naar de huidige Qlik Sense-app. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
DocumentPath ( )
```



*Deze functie wordt niet ondersteund in de standaardmodus.*

### DocumentTitle

Deze functie retourneert een tekenreeks met de titel van de huidige Qlik Sense-app. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
DocumentTitle ( )
```

### EngineVersion

Deze functie retourneert de volledige engineversie van Qlik Sense als tekenreeks.

```
EngineVersion ( )
```

### GetCollationLocale

Deze scriptfunctie retourneert de cultuurnaam van de landinstellingen voor collatie die wordt gebruikt. Als de variabele CollationLocale niet is ingesteld, wordt de daadwerkelijke landinstellingen van de gebruikersmachine geretourneerd.

```
GetCollationLocale ( )
```

### GetObjectField

**GetObjectField()** retourneert de naam van de dimensie. **Index** is een optioneel geheel getal dat de dimensie aangeeft die moet worden geretourneerd.

```
GetObjectField - diagramfunctie([index])
```

### GetRegistryString

Deze functie retourneert de waarde van een sleutel in het Windows-register. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
GetRegistryString(path, key)
```



*Deze functie wordt niet ondersteund in de standaardmodus.*

### IsPartialReload

Deze functie retourneert - 1 (True) als de huidige bewerking voor opnieuw laden gedeeltelijk is uitgevoerd, anders 0 (False).

```
IsPartialReload ( )
```

### OSUser

Deze functie retourneert een tekenreeks met de naam van de gebruiker die momenteel is verbonden. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
OSUser ( )
```



*In Qlik Sense Desktop en Qlik Sense Mobile Client Managed retourneert deze functie altijd 'PersonalMe'.*

### ProductVersion

Deze functie retourneert het volledige versie- en buildnummer van Qlik Sense als tekenreeks.

Deze functie is verouderd en vervangen door **EngineVersion()**.

```
ProductVersion ( )
```

### ReloadTime

Deze functie retourneert een tijdstempel voor het moment waarop het laden van de laatste gegevens werd voltooid. Kan zowel worden gebruikt in het script voor het laden van gegevens als in een diagramuitdrukking.

```
ReloadTime ( )
```

### StateName

**StateName()** haalt de naam op van de alternatieve state van de visualisatie waarin deze wordt gebruikt. StateName kan bijvoorbeeld worden gebruikt om visualisaties te maken met dynamische tekst en kleuren om aan te geven wanneer de state van een visualisatie is gewijzigd. U kunt de functie gebruiken in een diagramuitdrukking maar niet om de state te bepalen waarnaar de uitdrukking verwijst.

```
StateName - diagramfunctie ( )
```

## EngineVersion

Deze functie retourneert de volledige engineversie van Qlik Sense als tekenreeks.

#### Syntaxis:

```
EngineVersion ( )
```

## IsPartialReload

Deze functie retourneert - 1 (True) als de huidige bewerking voor opnieuw laden gedeeltelijk is uitgevoerd, anders 0 (False).

#### Syntaxis:

```
IsPartialReload ( )
```



### ProductVersion

Deze functie is verouderd en vervangen door `EngineVersion()`. Deze functie retourneert het volledige versie- en buildnummer van Qlik Sense als tekenreeks.

**Syntaxis:**

```
ProductVersion()
```

### StateName - diagramfunctie

**StateName()** haalt de naam op van de alternatieve state van de visualisatie waarin deze wordt gebruikt. `StateName` kan bijvoorbeeld worden gebruikt om visualisaties te maken met dynamische tekst en kleuren om aan te geven wanneer de state van een visualisatie is gewijzigd. U kunt de functie gebruiken in een diagramuitdrukking maar niet om de state te bepalen waarnaar de uitdrukking verwijst.

**Syntaxis:**

```
StateName ()
```

**Example 1:**

```
Dynamische tekst
='Region - ' & if(StateName() = '$', 'Default', StateName())
```

**Example 2:**

```
Dynamische kleuren
if(StateName() = 'Group 1', rgb(152, 171, 206),
  if(StateName() = 'Group 2', rgb(187, 200, 179),
    rgb(210, 210, 210)
  )
)
```

## 5.26 Tabelfuncties

De tabelfuncties retourneren informatie over de gegevenstabel die op dat moment wordt gelezen. Als er geen tabelnaam is opgegeven en de functie wordt gebruikt in een **LOAD**-opdracht, wordt de huidige tabelnaam verondersteld.

Alle functies kunnen worden gebruikt in het script voor het laden van gegevens, terwijl alleen **NoOfRows** kan worden gebruikt in een diagramuitdrukking.

### Overzicht van tabelfuncties

Sommige functies worden na het overzicht nader beschreven. Bij deze functies kunt u ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

### FieldName

De scriptfunctie **FieldName** retourneert de naam van het veld met het opgegeven nummer in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

```
FieldName (field_number ,table_name)
```

### FieldNumber

De scriptfunctie **FieldNumber** retourneert het nummer van een opgegeven veld in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

```
FieldNumber (field_name ,table_name)
```

### NoOfFields

De scriptfunctie **NoOfFields** retourneert het aantal velden in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

```
NoOfFields (table_name)
```

### NoOfRows

De functie **NoOfRows** retourneert het aantal rijen (records) in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

```
NoOfRows (table_name)
```

### NoOfTables

Deze scriptfunctie retourneert het aantal tabellen dat eerder is geladen.

```
NoOfTables ()
```

### TableName

Deze scriptfunctie retourneert de naam van de tabel met het opgegeven nummer.

```
TableName (table_number)
```

### TableNumber

Deze scriptfunctie retourneert het nummer van de opgegeven tabel. De eerste tabel heeft nummer 0.

Als table\_name niet bestaat, wordt NULL geretourneerd.

```
TableNumber (table_name)
```

### Voorbeeld:

In dit voorbeeld willen we een tabel maken met informatie over de tabellen en velden die zijn geladen.

Eerst gaan we een aantal steekproefgegevens laden. Hiermee worden de twee tabellen gemaakt die worden gebruikt om de tabelfuncties die in deze sectie worden beschreven te illustreren.

Characters:

```
Load Chr(RecNo()+Ord('A')-1) as Alpha, RecNo() as Num autogenerate 26;
```

ASCII:

```
Load
  if(RecNo()>=65 and RecNo()<=90,RecNo()-64) as Num,
  Chr(RecNo()) as AsciiAlpha,
  RecNo() as AsciiNum
autogenerate 255
where (RecNo()>=32 and RecNo()<=126) or RecNo()>=160 ;
```

Vervolgens itereren we door de tabellen die zijn geladen, via de functie **NoOfTables** en daarna door de velden van elke tabel, via de functie **NoOfFields**, en laden wij informatie met behulp van de tabelfuncties.

```
//Iterate through the loaded tables
For t = 0 to NoOfTables() - 1

//Iterate through the fields of table
For f = 1 to NoOfFields(TableName($(t)))
  Tables:
  Load
    TableName($(t)) as Table,
    TableNumber(TableName($(t))) as TableNo,
    NoOfRows(TableName($(t))) as TableRows,
    FieldName($(f),TableName($(t))) as Field,
    FieldNumber(FieldName($(f),TableName($(t))),TableName($(t))) as FieldNo
  Autogenerate 1;
Next f
Next t;
```

De resulterende tabel Tables ziet er als volgt uit:

Load table

Table	TableNo	TableRows	Field	FieldNo
Characters	0	26	Alpha	1
Characters	0	26	Num	2
ASCII	1	191	Num	1
ASCII	1	191	AsciiAlpha	2
ASCII	1	191	AsciiNum	3

### FieldName

De scriptfunctie **FieldName** retourneert de naam van het veld met het opgegeven nummer in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

**Syntaxis:**

```
FieldName(field_number ,table_name)
```

### Argumenten:

Argumenten

Argument	Beschrijving
field_number	Het veldnummer van het veld waarnaar u wilt verwijzen.
table_name	De tabel die het veld bevat waarnaar u wilt verwijzen.

### Voorbeeld:

```
LET a = FieldName(4,'tab1');
```

## FieldNumber

De scriptfunctie **FieldNumber** retourneert het nummer van een opgegeven veld in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

### Syntaxis:

```
FieldNumber(field_name ,table_name)
```

### Argumenten:

Argumenten

Argument	Beschrijving
field_name	De naam van het veld.
table_name	De naam van de tabel die het veld bevat.

Als het veld `field_name` niet bestaat in `table_name` of als `table_name` niet bestaat, retourneert de functie de waarde 0.

### Voorbeeld:

```
LET a = FieldNumber('Customer','tab1');
```

## NoOfFields

De scriptfunctie **NoOfFields** retourneert het aantal velden in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

### Syntaxis:

```
NoOfFields(table_name)
```

### Argumenten:

Argumenten	
Argument	Beschrijving
table_name	De naam van de tabel.

### Voorbeeld:

```
LET a = NoOfFields('tab1');
```

## NoOfRows

De functie **NoOfRows** retourneert het aantal rijen (records) in een eerder geladen tabel. Als de functie wordt gebruikt in een **LOAD**-opdracht, mag deze niet verwijzen naar de tabel die op dat moment wordt geladen.

### Syntaxis:

```
NoOfRows (table_name)
```

### Argumenten:

Argumenten	
Argument	Beschrijving
table_name	De naam van de tabel.

### Voorbeeld:

```
LET a = NoOfRows('tab1');
```

## 5.27 Trigonometrische en hyperbolische functies

In dit hoofdstuk worden functies beschreven voor trigonometrische en hyperbolische bewerkingen. In alle functies zijn de argumenten uitdrukkingen waarmee hoeken worden herleid die zijn gemeten in radialen, waarbij **x** moet worden geïnterpreteerd als een reëel getal.

Alle hoeken zijn weergegeven in radialen.

Alle functies kunnen zowel worden gebruikt in het script voor het laden van gegevens als in diagramuitdrukkingen.

### cos

Cosinus van **x**. Het resultaat is een getal tussen -1 en 1.

```
cos ( x )
```

### **acos**

Inverse cosinus van  $x$ . De functie is alleen gedefinieerd als  $-1 \leq x \leq 1$ . Het resultaat is een getal tussen 0 en  $\pi$ .

```
acos( x )
```

### **sin**

Sinus van  $x$ . Het resultaat is een getal tussen -1 en 1.

```
sin( x )
```

### **asin**

Inverse sinus van  $x$ . De functie is alleen gedefinieerd als  $-1 \leq x \leq 1$ . Het resultaat is een getal tussen  $-\pi/2$  en  $\pi/2$ .

```
asin( x )
```

### **tan**

Tangens van  $x$ . Het resultaat is een reëel getal.

```
tan( x )
```

### **atan**

Inverse tangens van  $x$ . Het resultaat is een getal tussen  $-\pi/2$  en  $\pi/2$ .

```
atan( x )
```

### **atan2**

Tweedimensionale generalisatie van de functie inverse tangens. Het resultaat is de hoek tussen de oorsprong en het punt dat wordt weergegeven door de coördinaten  $x$  en  $y$ . Het resultaat is een getal tussen  $-\pi$  en  $+\pi$ .

```
atan2( y, x )
```

### **cosh**

Hyperbolische cosinus van  $x$ . Het resultaat is een positief reëel getal.

```
cosh( x )
```

### **sinh**

Hyperbolische sinus van  $x$ . Het resultaat is een reëel getal.

```
sinh( x )
```

### **tanh**

Hyperbolische tangens van  $x$ . Het resultaat is een reëel getal.

```
tanh( x )
```

### **acosh**

Omgekeerde hyperbolische cosinus van  $x$ . Het resultaat is een positief reëel getal.

```
acosh( x )
```

### **asinh**

Omgekeerde hyperbolische cosinus van **x**. Het resultaat is een reëel getal.

```
asinh( x )
```

### **atanh**

Omgekeerde hyperbolische tangens van **x**. Het resultaat is een reëel getal.

```
atanh( x )
```

### **Voorbeelden:**

Met de volgende scriptcode wordt een steekproeftabel geladen en vervolgens wordt een tabel geladen met de berekende trigonometrische en hyperbolische bewerkingen op de waarden.

```
SampleData:
```

```
LOAD * Inline  
[Value  
-1  
0  
1];
```

```
Results:
```

```
Load *,  
cos(Value),  
acos(Value),  
sin(Value),  
asin(Value),  
tan(Value),  
atan(Value),  
atan2(Value, Value),  
cosh(Value),  
sinh(Value),  
tanh(Value)  
RESIDENT SampleData;
```

```
Drop Table SampleData;
```

### 6 Beperking van toegang tot bestandssysteem

Om beveiligingsredenen biedt Qlik Sense in de standaardmodus geen ondersteuning voor paden in het load-script voor gegevens of in functies en variabelen waarmee het bestandssysteem beschikbaar wordt gesteld.

Aangezien bestandssysteempaden echter werden ondersteund in QlikView, is het mogelijk om de standaardmodus uit te schakelen en de bestaande modus te gebruiken ten behoeve van hergebruik van QlikView load-scripts.



*Bij het uitschakelen van de standaardmodus kan een beveiligingsrisico ontstaan doordat het bestandssysteem wordt opengesteld.*

*Standaardmodus uitschakelen (page 1471)*

#### 6.1 Beveiligingsaspecten bij verbinding met op bestanden gebaseerde ODBC- en OLE DB-gegevensverbindingen

ODBC- en OLE DB-gegevensverbindingen die gebruikmaken van op bestanden gebaseerde stuurprogramma's geven het pad naar het verbonden gegevensbestand in de verbindingstekenreeks aan. Het pad kan worden weergegeven als de verbinding wordt bewerkt, in het dialoogvenster voor gegevensselectie of in bepaalde SQL-query's. Dit is het geval in zowel de standaardmodus als de bestaande modus.



*Als het weergeven van het pad in het gegevensbestand een punt van zorg is, wordt aanbevolen verbinding te maken met het gegevensbestand via een mapgegevensverbinding als dit mogelijk is.*

#### 6.2 Beperkingen in standaardmodus

Verschillende opdrachten, variabelen en functies kunnen niet worden gebruikt of hebben beperkingen in de standaardmodus. Het gebruik van niet-ondersteunde opdrachten in het load-script voor gegevens levert een fout op bij het uitvoeren van het load-script. Foutberichten zijn te vinden in het logbestand van het script. Het gebruik van niet-ondersteunde variabelen en functies resulteert niet in foutberichten of vermeldingen in het logbestand. In plaats daarvan geeft de functie NULL als resultaat.

Er is geen indicatie dat een variabele, opdracht of functie niet wordt ondersteund bij het bewerken van het load-script voor gegevens.



### Systeemvariabelen

Systeemvariabelen

Variabele	Standaardmodus	Bestaande modus	Definitie
Floppy	Niet ondersteund	Ondersteund	Retourneert de stationsletter van het eerst gevonden diskettestation. Dit is normaal gesproken a:.
CD	Niet ondersteund	Ondersteund	Retourneert de stationsletter van het eerst gevonden cd-rom-station. Als er geen cd-rom-station is gevonden, wordt c: geretourneerd.
QvPath	Niet ondersteund	Ondersteund	Retourneert het pad naar het Qlik Sense-programma.
QvRoot	Niet ondersteund	Ondersteund	Retourneert de hoofdmap van het Qlik Sense-programma.
QvWorkPath	Niet ondersteund	Ondersteund	Retourneert het pad naar de huidige Qlik Sense-app.
QvWorkRoot	Niet ondersteund	Ondersteund	Retourneert de hoofdmap van de huidige Qlik Sense-app.
WinPath	Niet ondersteund	Ondersteund	Retourneert het pad naar Windows.
WinRoot	Niet ondersteund	Ondersteund	Retourneert de hoofdmap van Windows.

## 6 Beperking van toegang tot bestandssysteem

Variabele	Standaardmodus	Bestaande modus	Definitie
\$(include=...)	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	De variabele <b>Include/Must_Include</b> specificeert een bestand dat tekst bevat die in het script moet worden opgenomen en moet worden geëvalueerd als scriptcode. Deze wordt niet gebruikt om gegevens toe te voegen. U kunt delen van uw scriptcode opslaan in een apart tekstbestand en dit in meerdere apps hergebruiken. Dit is een door de gebruiker gedefinieerde variabele.

### Reguliere scriptopdrachten

#### Reguliere scriptopdrachten

Opdracht	Standaardmodus	Bestaande modus	Definitie
Binary	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	De opdracht <b>binary</b> wordt gebruikt voor het laden van gegevens vanuit een andere app.
Connect	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	Met de opdracht <b>CONNECT</b> wordt de toegang van Qlik Sense tot een algemene database via de OLE DB/ODBC-interface vastgelegd. Voor ODBC moet de gegevensbron eerst worden opgegeven met behulp van ODBC-beheer.

## 6 Beperking van toegang tot bestandssysteem

Opdracht	Standaardmodus	Bestaande modus	Definitie
Directory	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	De opdracht <b>Directory</b> legt vast in welke directory moet worden gezocht naar gegevensbestanden in volgende <b>LOAD</b> -opdrachten, totdat een nieuwe <b>Directory</b> opdracht wordt gegeven.
Execute	Niet ondersteund	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	De <b>Execute</b> -opdracht wordt gebruikt voor het uitvoeren van andere programma's terwijl gegevens worden geladen in Qlik Sense. Bijvoorbeeld voor het uitvoeren van noodzakelijke conversies.
LOAD from ...	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	Met de opdracht <b>LOAD</b> worden velden geladen uit een bestand, uit gegevens die in het script zijn gedefinieerd, uit een eerder geladen tabel, van een webpagina, uit het resultaat van een daaropvolgende <b>SELECT</b> -opdracht of door gegevens automatisch te genereren.
Store into ...	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	De <b>Store</b> -instructie maakt een QVD-, CSV- of text-bestand.

### Scriptbesturingsopdrachten

Scriptbesturingsopdrachten

Opdracht	Standaardmodus	Bestaande modus	Definitie
For each... filelist mask/dirlist mask	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt  Geretourneerde output: bibliotheekverbinding	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt  Geretourneerde output: Bibliotheekverbinding of bestandssysteempad, afhankelijk van invoer	De syntaxis filelist mask produceert een lijst met door komma's gescheiden namen van alle bestanden in de huidige map die overeenkomen met het <b>filelist mask</b> . De syntaxis <b>dirlist mask</b> produceert een lijst met door komma's gescheiden namen van alle mappen in de huidige map die overeenkomen met het mapnaammasker.

### Bestandsfuncties

Bestandsfuncties

Functie	Standaardmodus	Bestaande modus	Definitie
Attribute()	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	Retourneert de waarde van de metatags van verschillende mediabestanden als tekst.
ConnectionString()	Geretourneerde output: Naam bibliotheekverbinding	De naam van de bibliotheekverbinding of daadwerkelijke verbinding, afhankelijk van invoer	Retourneert de actieve verbindingstekenreeks voor ODBC- of OLE DB-verbindingen.
FileDir()	Geretourneerde output: bibliotheekverbinding	Geretourneerde output: Bibliotheekverbinding of bestandssysteempad, afhankelijk van invoer	De functie <b>FileDir</b> retourneert een tekenreeks met het pad naar de map van het tabelbestand dat op dat moment wordt gelezen.

## 6 Beperking van toegang tot bestandssysteem

Functie	Standaardmodus	Bestaande modus	Definitie
FilePath()	Geretourneerde output: bibliotheekverbinding	Geretourneerde output: Bibliotheekverbinding of bestandssysteempad, afhankelijk van invoer	De functie <b>FilePath</b> retourneert een tekenreeks met het volledige pad naar het tabelbestand dat op dat moment wordt gelezen.
FileSize()	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	De functie retourneert een geheel getal met de grootte in bytes van het bestand of, als geen is opgegeven, van het tabelbestand dat op dat moment wordt gelezen. filenamefilename
FileTime()	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	De functie <b>FileTime</b> retourneert een tjdstempel in UTC voor de datum en tijd van de laatste wijziging van het bestand filename. Als geen filename is opgegeven, wordt het tabelbestand gebruikt dat op dat moment wordt gelezen.
GetFolderPath()	Niet ondersteund	Geretourneerde output: absoluut pad	De functie <b>GetFolderPath</b> retourneert de waarde van de functie Microsoft Windows <i>SHGetFolderPath</i> . Deze functie neemt de naam van een map in Microsoft Windows als invoer en retourneert het volledige pad van de map.

## 6 Beperking van toegang tot bestandssysteem

Functie	Standaardmodus	Bestaande modus	Definitie
QvdCreateTime()	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	Deze scriptfunctie retourneert de tijdstempel voor de XML-koptekst uit een QVD-bestand, indien aanwezig. Anders wordt NULL geretourneerd. In de tijdstempel wordt tijd weergegeven in UTC.
QvdFieldName()	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	Deze script-functie retourneert de naam van veldnummer <b>fieldno</b> in een QVD-bestand. Als het veld niet bestaat, wordt NULL geretourneerd.
QvdNoOfFields()	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	Deze scriptfunctie retourneert het aantal velden in een QVD-bestand.
QvdNoOfRecords()	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	Deze scriptfunctie retourneert het huidige aantal records in een QVD-bestand.
QvdTableName()	Ondersteunde invoer: Pad dat bibliotheekverbinding gebruikt	Ondersteunde invoer: Pad dat bibliotheekverbinding of bestandssysteem gebruikt	Deze scriptfunctie retourneert de naam van de tabel die is opgeslagen in een QVD-bestand.

## Systemfuncties

### Systemfuncties

Functie	Standaardmodus	Bestaande modus	Definitie
DocumentPath()	Niet ondersteund	Geretourneerde output: absoluut pad	Deze functie retourneert een tekenreeks met het volledige pad naar de huidige Qlik Sense-app.

## 6 Beperking van toegang tot bestandssysteem

Functie	Standaardmodus	Bestaande modus	Definitie
GetRegistryString()	Niet ondersteund	Ondersteund	Retourneert de waarde van een benoemde registersleutel met een opgegeven registerpad. Deze functie kan zowel in een diagram als een script worden gebruikt.

### 6.3 Standaardmodus uitschakelen

U kunt de standaardmodus uitschakelen, met andere woorden de bestaande modus instellen, om QlikView load-scripts die verwijzen naar absolute of relatieve bestandspaden en bibliotheekverbindingen opnieuw te gebruiken.



*Bij het uitschakelen van de standaardmodus kan een beveiligingsrisico ontstaan doordat het bestandssysteem wordt opengesteld.*

### Qlik Sense

Voor Qlik Sense, kan de standaardmodus worden uitgeschakeld in QMC met de eigenschap **Standaardmodus**.

### Qlik Sense Desktop

In Qlik Sense Desktop kunt u de standaard-/bestaande modus instellen in *Settings.ini*.

Als u Qlik Sense Desktop in de standaardinstallatielocatie hebt geïnstalleerd, bevindt *Settings.ini* zich in *C:\Users\{user}\Documents\Qlik\Sense\Settings.ini*. Als u Qlik Sense Desktop hebt geïnstalleerd in een door u geselecteerde map, bevindt *Settings.ini* zich in de map *Engine* in het installatiepad.

#### Doe het volgende:

1. Open *Settings.ini* in een teksteditor.
2. Wijzig *StandardReload=1* in *StandardReload=0*.
3. Sla het bestand op en start Qlik Sense Desktop.

Qlik Sense Desktop wordt nu in de bestaande modus uitgevoerd.

### Instellingen

De beschikbare instellingen voor *StandardReload* zijn:

- 1 (standaardmodus)
- 0 (bestaande modus)

# 6 Scripts op diagramniveau

Wanneer u diagramgegevens aanpast, gebruikt u een subset van het Qlik Sense-script dat uit verschillende opdrachten bestaat. Een opdracht kan een normale scriptopdracht of een scriptbesturingsopdracht zijn. Bepaalde opdrachten kunnen worden voorafgegaan door prefixen.

Normale opdrachten worden over het algemeen gebruikt om gegevens op een bepaalde manier te manipuleren. Deze opdrachten kunnen worden geschreven op een willekeurig aantal regels in het script en moeten altijd worden afgesloten met een puntkomma (;).

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Elke zin van een besturingsopdracht moet op één scriptregel blijven en kan worden afgesloten met een puntkomma of een regeleinde.

Prefixen kunnen worden toegevoegd aan normale opdrachten, maar nooit aan besturingsopdrachten.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

In deze sectie vindt u een alfabetische lijst met alle scriptopdrachten, besturingsopdrachten en prefixen die beschikbaar zijn in de subset van het script dat wordt gebruikt tijdens het aanpassen van diagramgegevens.

## 6.4 Scriptbesturingsopdrachten

Wanneer u diagramgegevens aanpast, gebruikt u een subset van het Qlik Sense script dat uit verschillende opdrachten bestaat. Een opdracht kan een normale scriptopdracht of een scriptbesturingsopdracht zijn.

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Elke zin van een besturingsopdracht moet op één scriptregel blijven en kan worden afgesloten met een puntkomma of een regeleinde.

Prefixen worden nooit toegepast op besturingsopdrachten.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken.

### Overzicht besturingsopdracht diagramaanpassing

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Call

De besturingsopdracht **call** roept een subroutine aan die door een vorige **sub**-opdracht is gedefinieerd.

```
Call name ( [ paramlist ] )
```



### Do..loop

De besturingsopdracht **do..loop** is een herhalingsconstructie voor gebruik in scripts waarmee één of meer opdrachten worden uitgevoerd totdat aan een logische voorwaarde wordt voldaan.

```
Do..loop [ ( while | until ) condition ] [statements]
[exit do [ ( when | unless ) condition ] [statements]
loop [ ( while | until ) condition ]
```

### End

Het sleutelwoord **End** voor scripts wordt gebruikt om **If**-, **Sub**- en **Switch**-clausules af te sluiten.

### Exit

Het sleutelwoord **Exit** voor scripts maakt onderdeel uit van de **Exit Script**-opdracht, maar kan ook worden gebruikt voor het afsluiten van **Do**-, **For**- of **Sub**-clausules.

### Exit script

Met deze besturingsopdracht wordt de uitvoering van het script stopgezet. Deze opdracht kan overal in het script worden ingevoegd.

```
Exit script [ (when | unless) condition ]
```

### For..next

De besturingsopdracht **for..next** is een herhalingsconstructie voor gebruik in scripts met een teller. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de tellervariabele tussen de opgegeven onder- en bovengrens.

```
For..next counter = expr1 to expr2 [ stepexpr3 ]
[statements]
[exit for [ ( when | unless ) condition ]
[statements]
Next [counter]
```

### For each ..next

De besturingsopdracht **for each..next** is een herhalingsconstructie voor gebruik in scripts waarmee een of meer opdrachten worden uitgevoerd voor elke waarde in een door komma's gescheiden lijst. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de lijst.

```
For each..next var in list
[statements]
[exit for [ ( when | unless ) condition ]
[statements]
next [var]
```

### If..then

De besturingsopdracht **if..then** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van één of meer logische voorwaarden, op verschillende manieren kan verlopen.



Omdat de instructie **if..then** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clausules (**if..then**, **elseif..then**, **else** en **end if**) de regelgrens overschrijden.

```
If..then..elseif..else..end if condition then
  [ statements ]
{ elseif condition then
  [ statements ] }
[ else
  [ statements ] ]
end if
```

### Next

Het sleutelwoord **Next** voor scripts wordt gebruikt om **For**-lussen af te sluiten.

### Sub

De besturingsopdracht **sub..end sub** geeft een subroutine aan die vanuit een **call**-opdracht kan worden aangeroepen.

```
Sub..end sub name [ ( paramlist ) ] statements end sub
```

### Switch

De besturingsopdracht **switch** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van de waarde van een uitdrukking, op verschillende manieren kan verlopen.

```
Switch..case..default..end switch expression {case valuelist [ statements ]}
[default statements] end switch
```

### To

Het sleutelwoord **To** voor scripts wordt gebruikt in verschillende scriptopdrachten.

## Call

De besturingsopdracht **call** roept een subroutine aan die door een vorige **sub**-opdracht is gedefinieerd.

### Syntaxis:

```
Call name ( [ paramlist ] )
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
name	De naam van de subroutine.

Argument	Beschrijving
paramlist	Een door komma's gescheiden lijst met de werkelijke parameters die worden gezonden aan de subroutine. Elk item in de lijst kan een veldnaam, een variabele of een willekeurige uitdrukking zijn.

De subroutine die wordt opgeroepen door een **call**-opdracht moet op een eerdere positie in het script door een **sub** zijn gedefinieerd.

Parameters worden gekopieerd naar de subroutine en worden, als de parameter in de **call**-instructie een variabele is en geen uitdrukking, bij het beëindigen van de subroutine weer teruggekopieerd.

### Beperkingen:

- Aangezien de **call**-opdracht een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag de opdracht niet op meerdere regels staan.
- Als u een subroutine definieert met `sub . . end sub` binnen een besturingsopdracht, bijvoorbeeld `if . . then`, kunt u de subroutine alleen oproepen binnen dezelfde besturingsopdracht.

## Do..loop

De besturingsopdracht **do..loop** is een herhalingsconstructie voor gebruik in scripts waarmee één of meer opdrachten worden uitgevoerd totdat aan een logische voorwaarde wordt voldaan.

### Syntaxis:

```
Do [ ( while | until ) condition ] [statements]
[exit do [ ( when | unless ) condition ] [statements]
loop[ ( while | until ) condition ]
```



*Omdat de instructie **do..loop** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clausules ervan (**do**, **exit do** en **loop**) de regelgrens overschrijden.*

### Argumenten:

#### Argumenten

Argument	Beschrijving
condition	Een logische uitdrukking die resulteert in True of False.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.

Argument	Beschrijving
while / until	De voorwaardelijke clausule <b>while</b> of <b>until</b> mag slechts eenmaal voorkomen in elke <b>do..loop</b> -opdracht, namelijk na <b>do</b> of na <b>loop</b> . Elke voorwaarde wordt alleen de eerste keer geïnterpreteerd wanneer deze wordt aangetroffen, maar wordt telkens in de lus geëvalueerd.
exit do	Als een clausule <b>exit do</b> binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clausule <b>loop</b> die het einde van de lus aangeeft. De clausule <b>exit do</b> kan voorwaardelijk worden gemaakt met het optionele suffix <b>when</b> of <b>unless</b> .

### End

Het sleutelwoord **End** voor scripts wordt gebruikt om **If**-, **Sub**- en **Switch**-clausules af te sluiten.

### Exit

Het sleutelwoord **Exit** voor scripts maakt onderdeel uit van de **Exit Script**-opdracht, maar kan ook worden gebruikt voor het afsluiten van **Do**-, **For**- of **Sub**-clausules.

### Exit script

Met deze besturingsopdracht wordt de uitvoering van het script stopgezet. Deze opdracht kan overal in het script worden ingevoegd.

#### Syntaxis:

```
Exit Script [ (when | unless) condition ]
```

Aangezien de **exit script**-opdracht een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag de opdracht niet op meerdere regels staan.

#### Argumenten:

Argumenten

Argument	Beschrijving
condition	Een logische uitdrukking die resulteert in True of False.
when / unless	De opdracht <b>exit script</b> kan voorwaardelijk worden gemaakt met de optionele clausule <b>when</b> of <b>unless</b> .

#### Voorbeelden:

```
//Exit script  
Exit script;
```

```
//Exit script when a condition is fulfilled
Exit Script when a=1
```

## For..next

De besturingsopdracht **for..next** is een herhalingsconstructie voor gebruik in scripts met een teller. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de tellervariabele tussen de opgegeven onder- en bovengrens.

### Syntaxis:

```
For counter = expr1 to expr2 [ step expr3 ]
[statements]
[exit for [ ( when | unless ) condition ]
[statements]
Next [counter]
```

De uitdrukkingen *expr1*, *expr2* en *expr3* worden alleen de eerste keer dat de lus wordt ingevoerd geëvalueerd. De waarde van de tellervariabele kan worden gewijzigd door opdrachten binnen de lus. Dit is echter met het oog op de gebruikelijke manier van programmeren niet aan te raden.

Als een clause **exit for** binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clause **next** die het einde van de lus aangeeft. De clause **exit for** kan voorwaardelijk worden gemaakt met het optionele suffix **when** of **unless**.



*Omdat de instructie **for..next** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clauses ervan (**for..to..step**, **exit for** en **next**) de regelgrens overschrijden.*

### Argumenten:

#### Argumenten

Argument	Beschrijving
counter	De naam van een variabele. Als <i>counter</i> wordt opgegeven na <b>next</b> moet de variabele dezelfde naam hebben als de variabele die wordt gevonden na de overeenkomende <b>for</b> .
expr1	Een uitdrukking die de eerste waarde van de variabele <i>counter</i> bepaalt waarvoor de lus wordt uitgevoerd.
expr2	Een uitdrukking die de laatste waarde van de variabele <i>counter</i> bepaalt waarvoor de lus wordt uitgevoerd.
expr3	Een uitdrukking die de waarde bepaalt van de stap waarmee de variabele <i>counter</i> toeneemt bij elke uitvoering van de lus.
condition	Een logische uitdrukking die resulteert in True of False.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.

## For each..next

De besturingsopdracht **for each..next** is een herhalingsconstructie voor gebruik in scripts waarmee een of meer opdrachten worden uitgevoerd voor elke waarde in een door komma's gescheiden lijst. De opdrachten in de lus tussen **for** en **next** worden uitgevoerd voor elke waarde van de lijst.

### Syntaxis:

Door een speciale syntaxis kunnen lijsten worden gegenereerd met namen van bestanden en mappen in de huidige map.

```
for each var in list
[statements]
[exit for [ ( when | unless ) condition ]
[statements]
next [var]
```

### Argumenten:

#### Argumenten

Argument	Beschrijving
var	De naam van een scriptvariabele die een nieuwe waarde uit de lijst ophaalt, telkens wanneer de lus wordt uitgevoerd. Als <b>var</b> wordt opgegeven na <b>next</b> moet de variabele dezelfde naam hebben als de variabele die wordt gevonden na de overeenkomende <b>for each</b> .

De waarde van de **var** kan worden gewijzigd door opdrachten binnen de lus. Dit is echter met het oog op de gebruikelijke manier van programmeren niet aan te raden.

Als een clause **exit for** binnen de lus wordt aangetroffen, gaat de uitvoering van het script over op de eerste opdracht na de clause **next** die het einde van de lus aangeeft. De clause **exit for** kan voorwaardelijk worden gemaakt met het optionele suffix **when** of **unless**.





*Omdat de instructie **for each..next** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de drie mogelijke clausules ervan (**for each**, **exit for** en **next**) de regelgrens overschrijden.*

### Syntaxis:

```
list := item { , item }
item := constant | (expression) | filelist mask | dirlist mask |
fieldvaluelist mask
```

## Argumenten

Argument	Beschrijving
constant	Elk getal of elke tekenreeks. Een tekenreeks die rechtstreeks in het script wordt opgenomen moet tussen enkele aanhalingstekens worden geplaatst. Een tekenreeks zonder enkele aanhalingstekens wordt geïnterpreteerd als een variabele en vervolgens wordt de waarde van de variabele gebruikt. Getallen hoeven niet tussen enkele aanhalingstekens te worden geplaatst.
expression	Een willekeurige uitdrukking.
mask	Een masker voor de naam van een bestand of map die geldige tekens voor bestandsnamen plus de standaardjokertekens, * en ?, kan bevatten.  U kunt absolute bestandspaden of paden lib:// gebruiken.
condition	Een logische uitdrukking die resulteert in True of False.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.
filelist mask	Deze syntaxis produceert een lijst met door komma's gescheiden namen van alle bestanden in de huidige map die overeenkomen met het bestandsnaammasker.  <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>Dit argument ondersteunt alleen bibliotheekverbindingen in de standaardmodus.</i> </div>
dirlist mask	Deze syntaxis produceert een lijst met door komma's gescheiden namen van alle mappen in de huidige map die overeenkomen met het mapnaammasker.  <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <i>Dit argument ondersteunt alleen bibliotheekverbindingen in de standaardmodus.</i> </div>
fieldvaluelist mask	Deze syntaxis itereert door de waarden van een veld die al in Qlik Sense zijn geladen.



*De Qlik Connectoren van internetopslagprovider en andere DataFiles-verbindingen bieden geen ondersteuning voor filtermaskeringen die jokertekens (\* en ?) gebruiken.*

**Example 1: Een lijst van bestanden laden**

```
// LOAD the files 1.csv, 3.csv, 7.csv and xyz.csv
for each a in 1,3,7,'xyz'
  LOAD * from file$(a).csv;
next
```

**Example 2: Een lijst van bestanden maken op schijf**

Via dit voorbeeld wordt een lijst met alle aan Qlik Sense gerelateerde bestanden in een map geladen.

```
sub DoDir (Root)
  for each Ext in 'qvw', 'qva', 'qvo', 'qvs', 'qvc', 'qvf', 'qvd'

    for each File in filelist (Root&'/*.' &Ext)

      LOAD
        '$(File)' as Name,
        FileSize( '$(File)' ) as Size,
        FileTime( '$(File)' ) as FileTime
      autogenerate 1;

    next File

  next Ext
  for each Dir in dirlist (Root&'/*' )

    call DoDir (Dir)

  next Dir

end sub

call DoDir ('lib://DataFiles')
```

### Example 3: Itereren door de waarden van een veld

Dit voorbeeld itereert door de waarden van geladen waarden van FIELD en genereert een nieuw veld, NEWFIELD. Voor elke waarde van FIELD, worden twee NEWFIELD-records gemaakt.

```
load * inline [
FIELD
one
two
three
];

FOR Each a in FieldValueList('FIELD')
LOAD '$(a)' &'-'&RecNo() as NEWFIELD AutoGenerate 2;
NEXT a
```

De resulterende tabel ziet er als volgt uit:

Example table

NEWFIELD
one-1
one-2
two-1
two-2
three-1
three-2



### If..then..elseif..else..end if

De besturingsopdracht **if..then** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van één of meer logische voorwaarden, op verschillende manieren kan verlopen.

Besturingsopdrachten worden gebruikt om het verloop van de scriptuitvoering te besturen. Gebruik in een diagramuitdrukking in plaats daarvan de **if** voorwaardelijke functie.

#### Syntaxis:

```
If condition then
  [ statements ]
{ elseif condition then
  [ statements ] }
[ else
  [ statements ] ]
end if
```

Omdat de instructie **if..then** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clausules (**if..then**, **elseif..then**, **else** en **end if**) de regelgrens overschrijden.

#### Argumenten:

##### Argumenten

Argument	Beschrijving
condition	Een logische uitdrukking die resulteert in True of False.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.

#### Example 1:

```
if a=1 then
    LOAD * from abc.csv;
    SQL SELECT e, f, g from tab1;
end if
```

#### Example 2:

```
if a=1 then; drop table xyz; end if;
```

#### Example 3:

```
if x>0 then
    LOAD * from pos.csv;
elseif x<0 then
```

```
        LOAD * from neg.csv;
else
        LOAD * from zero.txt;
end if
```

### Next

Het sleutelwoord **Next** voor scripts wordt gebruikt om **For**-lussen af te sluiten.

### Sub..end sub

De besturingsopdracht **sub..end sub** geeft een subroutine aan die vanuit een **call**-opdracht kan worden aangeroepen.

#### Syntaxis:

```
Sub name [ ( paramlist ) ] statements end sub
```

De argumenten worden in de subroutine gekopieerd en, als de corresponderende parameter in de **call**-opdracht een variabelenaam is, weer teruggekopieerd bij het verlaten van de subroutine.

Als een subroutine meer formele parameters heeft dan er aan daadwerkelijke parameters door een **call**-opdracht worden doorgegeven, worden de overtollige parameters op NULL ingesteld en kunnen ze als lokale variabelen binnen de subroutine worden gebruikt.

#### Argumenten:

Argumenten

Argument	Beschrijving
name	De naam van de subroutine.
paramlist	Een door komma's gescheiden lijst variabelen voor de formele parameters van de subroutine. Deze kunnen worden gebruikt als willekeurige variabelen binnen de subroutine.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.

#### Beperkingen:

- Omdat de instructie **sub** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de twee mogelijke clausules (**sub** en **end sub**) de regelgrens overschrijden.
- Als u een subroutine definieert met `sub . . end sub` binnen een besturingsopdracht, bijvoorbeeld `if . . then`, kunt u de subroutine alleen oproepen binnen dezelfde besturingsopdracht.

#### Example 1:

```
Sub INCR (I,J)
I = I + 1
```

```
Exit Sub when I < 10
J = J + 1
End Sub
Call INCR (X,Y)
```

### Example 2: - parameteroverdracht

```
Sub ParTrans (A,B,C)
A=A+1
B=B+1
C=C+1
End Sub
A=1
X=1
C=1
Call ParTrans (A, (X+1)*2)
```

Het resultaat van het bovenstaande is dat lokaal, binnen de subroutine, A wordt geïnitieerd op 1, B op 4 en C op NULL.

Bij het verlaten van de subroutine krijgt de globale variabele A de waarde 2 (teruggekopieerd van de subroutine). De tweede werkelijke parameter "(X+1)\*2" wordt niet teruggekopieerd omdat dit geen variabele is. Tot slot verandert de globale variabele C niet door de aanroep van de subroutine.

## Switch..case..default..end switch

De besturingsopdracht **switch** is een constructie voor scriptselectie waarmee de uitvoering van het script, afhankelijk van de waarde van een uitdrukking, op verschillende manieren kan verlopen.

### Syntaxis:

```
Switch expression {case valuelist [ statements ]} [default statements] end
switch
```



Omdat de instructie **switch** een besturingsopdracht is en dus wordt afgesloten door een puntkomma of het einde van een regel, mag geen van de vier mogelijke clausules (**switch**, **case**, **default** en **end switch**) de regelgrens overschrijden.

### Argumenten:

#### Argumenten

Argument	Beschrijving
expression	Een willekeurige uitdrukking.

Argument	Beschrijving
valuelist	Een door komma's gescheiden lijst waarden waarmee de waarde van de uitdrukking zal worden vergeleken. Uitvoering van het script wordt vervolgd met de opdrachten in de eerste groep waarvoor een waarde in valuelist is gevonden die gelijk is aan de waarde in de uitdrukking. Elke waarde in valuelist kan een willekeurige uitdrukking zijn. Als in geen van de <b>case</b> -opdrachten een overeenkomst is gevonden, worden de opdrachten achter <b>default</b> , indien gespecificeerd, uitgevoerd.
statements	Een willekeurige groep van een of meer Qlik Sense-scriptopdrachten.

### Voorbeeld:

```
Switch I
Case 1
LOAD '$(I): CASE 1' as case autogenerate 1;
Case 2
LOAD '$(I): CASE 2' as case autogenerate 1;
Default
LOAD '$(I): DEFAULT' as case autogenerate 1;
End Switch
```

## To

Het sleutelwoord **To** voor scripts wordt gebruikt in verschillende scriptopdrachten.

## 6.5 Prefixen

Prefixen kunnen worden toegevoegd aan normale opdrachten, maar nooit aan besturingsopdrachten.

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

### Overzicht prefixen diagramaanpassing

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### Add

Het voorvoegsel **Add** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat er records aan een andere tabel moeten worden toegevoegd. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Add** kan ook worden gebruikt in een **Map**-opdracht.

```
Add [only] [Concatenate [(tablename )]] (loadstatement | selectstatement)
Add [ Only ] mapstatement
```

### Replace

het voorvoegsel **Replace** kan aan elke **LOAD**- of **SELECT**-opdracht in het script worden toegevoegd om op te geven dat de geladen tabel een andere tabel moet vervangen. Dit geeft ook aan dat deze opdracht met behulp van gedeeltelijk laden moet worden uitgevoerd. Het voorvoegsel **Replace** kan ook worden gebruikt in een **Map**-opdracht.

```
Replace [only] [Concatenate[(tablename) ]] (loadstatement | selectstatement)
Replace [only] mapstatement
```

### Add

In de context van de aanpassing van diagrammen, wordt het prefix **Add** gebruikt in combinatie met **LOAD** om waarde toe te voegen aan de *HC1*-tabel. Dit vertegenwoordigt de hyperkubus die is berekend door de Qlik associative engine. U kunt één of meerdere kolommen specificeren. Ontbrekende waarden worden automatisch ingevuld door de Qlik associative engine.

#### Syntaxis:

```
Add loadstatement
```

#### Voorbeeld:

In dit voorbeeld worden twee rijen aan de kolommen *Datums* en *Verkoop* toegevoegd vanuit the inline-opdracht

```
Add Load
x as Dates,
y as Sales
Inline
[
Dates,Sales
2001/09/1,1000
2001/09/10,-300
]
```

### Replace

In de context van aanpassing van het diagram, verandert het prefix **Replace** alle waarden van de *HC1*-tabel met een berekende waarde die is gedefinieerd door het script.

#### Syntaxis:

```
Replace loadstatement
```

#### Voorbeeld:

Dit voorbeeld overschrijft alle waarden in kolom *z* met de som van *x* en *y*.

```
Replace Load
x+y as z
Resident HC1;
```

## 6.6 Reguliere scriptopdrachten

Normale opdrachten worden over het algemeen gebruikt om gegevens op een bepaalde manier te manipuleren. Deze opdrachten kunnen worden geschreven op een willekeurig aantal regels in het script en moeten altijd worden afgesloten met een puntkomma (;).

Bij het typen van sleutelwoorden in het script kunt u elke combinatie van hoofdletters en kleine letters gebruiken. Gebruikte namen van velden en variabelen in de opdrachten zijn echter hoofdlettergevoelig.

### Overzicht besturingsopdracht diagramaanpassing

Elke functie wordt verder beschreven na het overzicht. U kunt ook in de syntaxis op de functienaam klikken om meteen naar de details van die specifieke functie te gaan.

#### LOAD

In de context van aanpassing van een diagram, laadt de opdracht **LOAD** meer gegevens naar de hyperkubus uit de gegevens die in het script zijn gedefinieerd, of vanuit een eerder geladen tabel. U kunt tevens gegevens van analytische verbindingen inladen.



*De opdracht **LOAD** moet ofwel het prefix **Replace** ofwel het prefix **Add** bevatten. Zo niet, dan wordt hij afgewezen.*

```
Add | Replace Load [ distinct ] fieldlist
(
inline data [ format-spec ] |
resident table-label
) | extension pluginname.functionname([script] tabledescription)
[ where criterion | while criterion ]
[ group by groupbyfieldlist ]
[order by orderbyfieldlist ]
```

#### Let

De opdracht **let** is een aanvulling op de opdracht **set** voor het definiëren van scriptvariabelen. In tegenstelling tot de opdracht **set** evalueert de opdracht **let** de uitdrukking rechts van '=' tijdens runtime van het script voordat deze wordt toegewezen aan de variabele.

```
Let variablename=expression
```

#### Set

Met de opdracht **set** legt u scriptvariabelen vast. Deze kunnen worden gebruikt om tekenreeksen, paden, stations en dergelijke te vervangen.

```
Set variablename=string
```

#### Put

De opdracht **Put** wordt gebruikt om een numerieke waarde in de hyperkubus te stellen.

### HCValue

De opdracht **HCValue** wordt gebruikt om waarden in een rij van een bepaalde kolom op te halen.

### Load

In de context van aanpassing van een diagram, laadt de opdracht **LOAD** meer gegevens naar de hyperkubus uit de gegevens die in het script zijn gedefinieerd, of vanuit een eerder geladen tabel. U kunt tevens gegevens van analytische verbindingen inladen.



*De opdracht **LOAD** moet ofwel het prefix **Replace** ofwel het prefix **Add** bevatten. Zo niet, dan wordt hij afgewezen.*

### Syntaxis:

```
Add | Replace LOAD fieldlist
(
inline data [ format-spec ] |
resident table-label
) | extension pluginname.functionname([script] tabledescription)
[ where criterion | while criterion ]
[ group by groupbyfieldlist ]
[order by orderbyfieldlist ]
```

## Argumenten:

## Argumenten

Argument	Beschrijving
fieldlist	<p><i>fieldlist</i> ::= ( *   <i>field</i> {, *   <i>field</i> } )</p> <p>Een lijst van de velden die geladen moeten worden. Gebruik een * als veldlijst om alle velden in de tabel aan te duiden.</p> <p><i>field</i> ::= ( <i>fieldref</i>   <i>expression</i> ) [<b>as</b> <i>aliasname</i> ]</p> <p>De velddefinitie moet altijd een literal, een referentie naar een bestaand veld of een uitdrukking bevatten.</p> <p><i>fieldref</i> ::= ( <i>fieldname</i>  @<i>fieldnumber</i>  @<i>startpos:endpos</i> [ <b>I</b>   <b>U</b>   <b>R</b>   <b>B</b>   <b>T</b> ] )</p> <p><i>fieldname</i> is een tekst die identiek is aan een veldnaam in de tabel. Houd er rekening mee dat de veldnaam tussen rechte dubbele aanhalingstekens of vierkante haken moet staan als deze bijvoorbeeld spaties bevat. Soms zijn veldnamen niet expliciet beschikbaar. In dat geval wordt een andere notatie gebruikt:</p> <p>@<i>fieldnumber</i> geeft het veldnummer in een tabelbestand met scheidingstekens aan. Het moet een positief geheel getal zijn, voorafgegaan door "@". De nummering is altijd vanaf 1 optellend tot het aantal velden.</p> <p>@<i>startpos:endpos</i> geeft de begin- en eindposities van een veld aan in een bestand met een vaste recordlengte. De posities moeten beide positieve gehele getallen zijn. De twee getallen moeten worden voorafgegaan door "@" en gescheiden door een dubbele punt. De nummering is altijd vanaf 1 optellend tot het aantal posities. In het laatste veld wordt <i>n</i> gebruikt als eindpositie.</p> <ul style="list-style-type: none"> <li>Als @<i>startpos:endpos</i> direct wordt gevolgd door de tekens <b>I</b> of <b>U</b>, worden de gelezen bytes geïnterpreteerd als een binair ondertekend (<b>I</b>) of niet-ondertekend (<b>U</b>) geheel getal (Intel-bytevolgorde). Het aantal gelezen posities moet 1, 2 of 4 zijn.</li> <li>Als @<i>startpos:endpos</i> direct wordt gevolgd door het teken <b>R</b>, worden de bytes geïnterpreteerd als een binair reëel getal (IEEE 32-bits of 64-bits drijvende komma). Het aantal gelezen posities moet 4 of 8 zijn.</li> <li>Als @<i>startpos:endpos</i> direct wordt gevolgd door het teken <b>B</b>, worden de gelezen bytes geïnterpreteerd als een BCD (Binary Coded Decimal) volgens de COMP-3-standaard. U kunt een willekeurig aantal bytes opgeven.</li> </ul> <p><i>expression</i> kan een numerieke functie of een tekenreeksfunctie zijn, gebaseerd op een of meer andere velden in dezelfde tabel. Zie de syntaxis van uitdrukkingen voor meer informatie.</p> <p><b>as</b> wordt gebruikt om een nieuwe naam aan een veld toe te wijzen.</p>



Argument	Beschrijving
inline	<p><b>inline</b> wordt gebruikt als gegevens moeten worden getypt in het script en niet uit een bestand moeten worden geladen.</p> <p><i>data ::= [ text ]</i></p> <p>Gegevens die u invoert met een <b>inline</b>-clausule moeten tussen dubbele aanhalingstekens of tussen vierkante haakjes worden geplaatst. De tekst ertussen wordt net zo geïnterpreteerd als de inhoud van een bestand. Waar u bijvoorbeeld een nieuwe regel in een tekstbestand zou invoegen, moet u dat ook doen in de tekst van een <b>inline</b>-clausule. U drukt dus op Enter bij het typen van het script. Het aantal kolommen wordt bepaald door de eerste regel.</p> <p><i>format-spec ::= ( fspec-item {, fspec-item } )</i></p> <p>De opmaakspecificatie bestaat uit een lijst met verschillende opmaakspecificaties, tussen haakjes.</p>
resident	<p><b>resident</b> wordt gebruikt als gegevens moeten worden geladen uit een eerder geladen tabel.</p> <p><i>table label</i> is een label dat vooraf gaat aan de opdracht <b>LOAD</b> of waarmee de oorspronkelijke tabel is gemaakt. Het label moet worden opgegeven met een puntkomma aan het eind.</p>

Argument	Beschrijving
extension	<p>U kunt gegevens via analytische verbindingen inladen. Als u een functie wilt aanroepen die in de SSE-plugin is gedefinieerd, moet u de clause <b>extension</b> gebruiken of een script evalueren.</p> <p>U kunt één tabel naar de SSE-plugin verzenden, waarop één gegevenstabel wordt teruggegeven. Als door de plugin niet de namen worden aangegeven van de velden die worden teruggegeven, krijgen de velden de namen Field1, Field2 enzovoort.</p> <pre data-bbox="475 600 1390 640">Extension pluginname.functionname( tabledescription );</pre> <ul data-bbox="528 651 1310 869" style="list-style-type: none"> <li>• Gegevens inladen met een functie in een SSE-plugin <i>tabledescription ::= (table { ,tablefield} )</i> Als u geen tabelvelden aangeeft, worden de velden gebruikt in de volgorde waarin deze zijn ingeladen.</li> <li>• Gegevens inladen door een script in een SSE-plugin te evalueren <i>tabledescription ::= ( script, table { ,tablefield} )</i></li> </ul> <p><b>Afhandeling van gegevenssoorten in de tabelvelddefinitie</b></p> <p>Gegevenssoorten in analytische verbindingen worden automatisch gedetecteerd. Als de gegevens geen getalswaarden bevatten en ten minste één teksttekenreeks die niet leeg is, wordt het veld als tekstveld beschouwd. In alle andere gevallen wordt het veld als getalsveld beschouwd.</p> <p>U kunt de gegevenssoort afdwingen door een veldnaam op te nemen met <b>String()</b> of <b>Mixed()</b> voor gemengd.</p> <ul data-bbox="528 1256 1374 1406" style="list-style-type: none"> <li>• Met <b>String()</b> maakt u het veld een tekstveld. Als het veld numeriek is, wordt het tekstgedeelte van de dubbele waarde geëxtraheerd. Er wordt niet geconverteerd.</li> <li>• Met <b>Mixed()</b> maakt u het veld een veld met dubbele waarden.</li> </ul> <p><b>String()</b> en <b>Mixed()</b> kunnen niet buiten tabelvelddefinities van <b>extension</b> worden gebruikt, en u kunt geen andere Qlik Sense -functies in een tabelvelddefinitie gebruiken.</p>
where	<p><b>where</b> is een clause die wordt gebruikt om aan te geven of een record wel of niet in de selectie moet worden opgenomen. De selectie wordt opgenomen als <i>criterion</i> de waarde True heeft. <i>criterion</i> is een logische uitdrukking.</p>
while	<p><b>while</b> is een clause die aangeeft of een record herhaaldelijk moet worden gelezen. Hetzelfde record wordt gelezen zolang <i>criterion</i> de waarde True heeft. Een <b>while</b>-clause is vaak pas nuttig wanneer u deze samen met de functie <b>IteNo( )</b> gebruikt. <i>criterion</i> is een logische uitdrukking.</p>

Argument	Beschrijving
group by	<p><b>group by</b> is een clause die wordt gebruikt om te definiëren over welke velden de gegevens moeten worden geaggregeerd (gegroepeerd). De aggregatievelden moeten op de een of andere manier in de geladen uitdrukkingen worden opgenomen. U mag geen andere velden dan de aggregatievelden gebruiken buiten aggregatiefuncties in de geladen uitdrukkingen.</p> <p><i>groupbyfieldlist ::= (fieldname { ,fieldname } )</i></p>
order by	<p><b>order by</b> is een clause die wordt gebruikt om de records van een residentie tabel te sorteren voordat ze worden verwerkt door de <b>load</b>-opdracht. De residentie tabel kan worden gesorteerd op één of meer velden in oplopende of aflopende volgorde. Het sorteren gebeurt primair op numerieke waarde en secundair op nationale sorteervolgorde. U kunt deze clause alleen gebruiken wanneer de gegevensbron een residentie tabel is. De volgordevelden geven aan op welk veld de residentie tabel wordt gesorteerd. U kunt het veld opgeven met z'n naam of met z'n nummer in de residentie tabel (het eerste veld is nummer 1).</p> <p><i>orderbyfieldlist ::= fieldname [ sortorder ] { , fieldname [ sortorder ] }</i></p> <p><i>sortorder</i> is <i>asc</i> voor oplopend of <i>desc</i> voor aflopend. Als u geen <i>sortorder</i> opgeeft, wordt <i>asc</i> verondersteld.</p> <p><i>fieldname, path, filename</i> en <i>aliasname</i> zijn tekenreeksen die aangeven waar de respectievelijke namen voor staan. Elk veld in de brontabel kan worden gebruikt als <i>fieldname</i>. Velden die zijn gemaakt via de <i>as</i>-clause (<i>aliasname</i>) vallen echter buiten het bereik en kunnen niet worden gebruikt binnen dezelfde <b>load</b>-opdracht.</p>

## Let

De opdracht **let** is een aanvulling op de opdracht **set** voor het definiëren van scriptvariabelen. In tegenstelling tot de opdracht **set** evalueert de opdracht **let** de uitdrukking rechts van '=' tijdens runtime van het script voordat deze wordt toegewezen aan de variabele.

### Syntaxis:

```
Let variablename=expression
```

Voorbeelden en resultaten:

Voorbeeld	Resultaat
<pre>Set x=3+4; Let y=3+4; z=\$(y)+1;</pre>	<p>\$(x) wordt geëvalueerd als ' 3+4 '</p> <p>\$(y) wordt geëvalueerd als ' 7 '</p> <p>\$(z) wordt geëvalueerd als ' 8 '</p> <p>Let op het verschil tussen de <b>Set</b>- en <b>Let</b>-instructies. De <b>Set</b>-instructie wijst de reeks '3+4' toe aan de variabele, terwijl de <b>Let</b>-instructie de reeks evalueert en 7 toewijst aan de variabele.</p>
<pre>Let T=now( );</pre>	<p>\$(T) krijgt de waarde van de huidige tijd.</p>

## Set

Met de opdracht **set** legt u scriptvariabelen vast. Deze kunnen worden gebruikt om tekenreeksen, paden, stations en dergelijke te vervangen.

### Syntaxis:

```
Set variablename=string
```

#### Example 1:

```
Set FileToUse=Data1.csv;
```

#### Example 2:

```
Set Constant="My string";
```

#### Example 3:

```
Set BudgetYear=2012;
```

## Put

De opdracht **put** wordt gebruikt om een numerieke waarde in de hyperkubus te stellen.

De kolommen zijn toegankelijk via labels. U kunt kolommen en rijen ook openen op volgorde van declaratie. Zie de onderstaande voorbeelden voor meer details.

### Syntaxis:

```
put column(position)=value
```

#### Example 1:

De kolommen zijn toegankelijk via labels.

In dit voorbeeld wordt een waarde van 1 gesteld in de positie van de kolom met het label *Verkoop*.

```
Put sales(1) = 1;
```

### Example 2:

U hebt toegang tot metingkolommen op volgorde van declaratie met gebruik van de `#hc1.measure`-indeling voor metingen.

In dit voorbeeld wordt de waarde 1000 in de tiende positie geplaatst van de laatst gesorteerde hyperkubus.

```
Put #hc1.measure.2(10) = 1000;
```

### Example 3:

U kunt de dimensierijen openen op volgorde van declaratie met de `#hc1.dimension`-indeling voor dimensies.

In dit voorbeeld wordt de waarde van de constante Pi in de vijfde rij van de derde gedeclareerde dimensie.

```
Put #hc1.dimension.3(5) = Pi();
```



*Als dat soort dimensies of uitdrukkingen niet aanwezig zijn, in waarden of labels, dan wordt er een foutmelding gegeneerd dat de kolom niet is gevonden. Als de indexwaarde voor de kolom buiten bereik ligt, wordt er NULL geretourneerd.*

## HCValue

De functie **HCValue** wordt gebruikt om waarden in een rij van een bepaalde kolom op te halen.

### Syntaxis:

```
HCValue(column, position)
```

### Example 1:

In dit voorbeeld wordt de waarde uit de eerste positie van de kolom met het label Verkoop geretourneerd.

```
HCValue(Sales,1)
```

### Example 2:

In dit voorbeeld wordt de waarde uit de tiende positie van de gesorteerde hyperkubus geretourneerd.

```
HCValue(#hc1.measure.2,10)
```

### Example 3:

In dit voorbeeld wordt de waarde uit de vijfde rij in de derde dimensie geretourneerd.

```
HCValue(#hc1.dimension.3,5)
```



*Als dat soort dimensies of uitdrukkingen niet aanwezig zijn, in waarde of labels, dan wordt er een foutmelding gegeneerd dat de kolom niet is gevonden. Als de indexwaarde voor de kolom buiten bereik ligt, wordt er NULL geretourneerd.*

# 7 QlikView-functies en -opdrachten die niet worden ondersteund in Qlik Sense

De meeste functies en opdrachten die kunnen worden gebruikt in load-scripts en diagramuitdrukkingen in QlikView worden tevens ondersteund in Qlik Sense. Maar er zijn enkele uitzonderingen, zoals hier beschreven.

## 7.1 Scriptopdrachten die niet worden ondersteund in Qlik Sense

QlikView Scriptopdrachten die niet worden ondersteund in Qlik Sense

Opdracht	Commentaar
Command	Gebruik in plaats hiervan <b>SQL</b> .
InputField	

## 7.2 Functies die niet worden ondersteund in Qlik Sense

Deze lijst bevat script- en diagramfuncties in QlikView die niet worden ondersteund in Qlik Sense.

- **GetCurrentField**
- **GetExtendedProperty**
- **Input**
- **InputAvg**
- **InputSum**
- **MsgBox**
- **NoOfReports**
- **ReportComment**
- **ReportId**
- **ReportName**
- **ReportNumber**

## 7.3 Prefixes die niet worden ondersteund in Qlik Sense

Deze lijst bevat pefixes in QlikView die niet worden ondersteund in Qlik Sense.

- **Bundle**
- **Image\_Size**
- **Info**

### 8 Functies en opdrachten die niet worden aanbevolen in Qlik Sense

De meeste functies en opdrachten die kunnen worden gebruikt in load-scripts en diagramuitdrukkingen in QlikView worden tevens ondersteund in Qlik Sense, maar voor sommige hiervan wordt het gebruik in Qlik Sense niet aanbevolen. Er zijn ook functies en opdrachten beschikbaar in eerdere versies van Qlik Sense die zijn verouderd.

Om compatibiliteitsredenen werken zij nog steeds zoals bedoeld, maar het is raadzaam om de code bij te werken op basis van de aanbevelingen in deze sectie, aangezien zij mogelijk worden verwijderd in komende versies.

#### 8.1 Scriptopdrachten die niet worden aanbevolen in Qlik Sense

Deze tabel bevat scriptopdrachten die niet worden aanbevolen voor gebruik in Qlik Sense.

Scriptopdrachten die niet worden aanbevolen

Opdracht	Aanbeveling
Command	Gebruik in plaats hiervan <b>SQL</b> .
CustomConnect	Gebruik in plaats hiervan <b>Custom Connect</b> .

#### 8.2 Parameters voor scriptopdrachten die niet worden aanbevolen in Qlik Sense

Deze tabel bevat parameters voor scriptopdrachten die niet worden aanbevolen voor gebruik in Qlik Sense.

Parameters voor scriptopdrachten die niet worden aanbevolen

Opdracht	Parameters
Buffer	Gebruik <b>Incremental</b> in plaats van: <ul style="list-style-type: none"><li>• <b>Inc</b> (niet aanbevolen)</li><li>• <b>Incr</b> (niet aanbevolen)</li></ul>



## 8 Functies en opdrachten die niet worden aanbevolen in Qlik Sense

---

Opdracht	Parameters
LOAD	<p>De volgende trefwoorden voor parameters worden gegenereerd door QlikView-wizards voor bestandstransformatie. De functionaliteit blijft behouden als gegevens opnieuw worden geladen, maar Qlik Sense biedt geen gerichte ondersteuning/wizards voor het genereren van de opdracht met deze parameters:</p> <ul style="list-style-type: none"><li>• <b>Bottom</b></li><li>• <b>Cellvalue</b></li><li>• <b>Col</b></li><li>• <b>Colmatch</b></li><li>• <b>Colsplit</b></li><li>• <b>Colxtr</b></li><li>• <b>Compound</b></li><li>• <b>Contain</b></li><li>• <b>Equal</b></li><li>• <b>Every</b></li><li>• <b>Expand</b></li><li>• <b>Filters</b></li><li>• <b>Intarray</b></li><li>• <b>Interpret</b></li><li>• <b>Length</b></li><li>• <b>Longer</b></li><li>• <b>Numerical</b></li><li>• <b>Pos</b></li><li>• <b>Remove</b></li><li>• <b>Rotate</b></li><li>• <b>Row</b></li><li>• <b>Rowcnd</b></li><li>• <b>Shorter</b></li><li>• <b>Start</b></li><li>• <b>Strcnd</b></li><li>• <b>Top</b></li><li>• <b>Transpose</b></li><li>• <b>Unwrap</b></li><li>• <b>XML: XMLSAX and Pattern is Path</b></li></ul>

### 8.3 Functies die niet worden aanbevolen in Qlik Sense

Deze tabel bevat script- en diagramfuncties die niet worden aanbevolen voor gebruik in Qlik Sense.

## 8 Functies en opdrachten die niet worden aanbevolen in Qlik Sense

### Functies die niet worden aanbevolen

Functie	Aanbeveling
<b>NumAvg</b>	Gebruik in plaats hiervan Range-functies.
<b>NumCount</b>	<i>Bereikfuncties (page 1324)</i>
<b>NumMax</b>	
<b>NumMin</b>	
<b>NumSum</b>	
<b>Color()</b>	Gebruik in plaats hiervan kleurfuncties. <b>QliktechBlue()</b> kan worden vervangen door <b>RGB(8, 18, 90)</b> en <b>QliktechGray</b> kan worden vervangen door <b>RGB(158, 148, 137)</b> om dezelfde kleuren te krijgen.
<b>QliktechBlue</b>	
<b>QliktechGray</b>	<i>Kleurfuncties (page 549)</i>
<b>QlikViewVersion</b>	Gebruik in plaats hiervan <b>EngineVersion</b> . <i>EngineVersion (page 1456)</i>
<b>ProductVersion</b>	Gebruik in plaats hiervan <b>EngineVersion</b> . <i>EngineVersion (page 1456)</i>
<b>QVUser</b>	
<b>Year2Date</b>	Gebruik in plaats hiervan <b>YearToDate</b> .
<b>Vrank</b>	Gebruik in plaats hiervan <b>Rank</b> .
<b>WildMatch5</b>	Gebruik in plaats hiervan <b>WildMatch</b> .

### Kwalificatie ALL

In QlikView kan de kwalificatie **ALL** voorkomen vóór een uitdrukking. Dit is gelijk aan het gebruik van **{1}** **TOTAL**. In een dergelijk geval wordt de berekening gemaakt op basis van alle waarden van het veld in het document, en worden grafiekdimensies en huidige selecties genegeerd. Dezelfde waarde wordt altijd geretourneerd, ongeacht de logische status in het document. Als de kwalificatie **ALL** wordt gebruikt, kan de uitdrukking set niet worden gebruikt omdat de kwalificatie **ALL** zelf een set definieert. Om de functionaliteit uit eerdere versies te behouden werkt de kwalificatie **ALL** nog in deze versie van Qlik Sense. Het is echter mogelijk dat deze kwalificatie in toekomstige versies wordt verwijderd.