



Tutorial – Skripterstellung für Anfänger

Qlik Sense®

November 2024

Copyright © 1993-jjjj QlikTech International AB. Alle Rechte vorbehalten.

1 Herzlich willkommen!	4
1.1 Lerninhalte	4
1.2 Zielgruppe für diesen Kurs	4
1.3 Paketinhalt	4
1.4 Lektionen in diesem Tutorial	5
1.5 Weitere Informationsquellen und Ressourcen	5
2 Scripting in the data load editor	6
2.1 Data load editor	6
2.2 Script editor	7
Accessing syntax help for commands and functions	7
Commenting in the script	8
Indenting code	8
Inserting a prepared test script	8
3 LOAD- und SELECT-Anweisungen	9
4 Selecting and loading data	10
5 Renaming fields	19
6 Reducing data	22
7 Transforming data	28
7.1 Resident LOAD	28
7.2 Preceding LOAD	32
8 Concatenation	34
8.1 Automatic concatenation	34
8.2 Forced concatenation	37
8.3 Preventing concatenation	38
9 Zirkelbezüge	40
9.1 Auflösen von Zirkelbezügen	41
10 Synthetische Schlüssel	43
10.1 Auflösen synthetischer Schlüssel	45
11 Verwenden der Daten in einer App	47
11.1 Hinzufügen eines Diagramms	47
11.2 Hinzufügen von Dimensionen und Kennzahlen	48
Erstellen und Hinzufügen von Dimensionen	48
Erstellen und Hinzufügen von Kennzahlen	49
11.3 Vielen Dank!	53

1 Herzlich willkommen!

Willkommen bei diesem Tutorial, das die grundlegenden Schritte zur Skripterstellung in Qlik Sense erläutert.

Bevor Sie Visualisierungen in Ihrer App in Qlik Sense erstellen können, müssen Sie Ihre Daten laden. Wenn Sie sich mit der Verwendung von Ladeskripts auskennen, können Sie Ihre Daten vor dem Laden in die App vorbereiten und bearbeiten.

Daten können mit dem Datenmanager oder dem Dateneditor geladen werden. Den Dateneditor verwenden Sie, um ein Datenladeskript zu erstellen, zu bearbeiten und auszuführen.

1.1 Lerninhalte

Nachdem Sie dieses Tutorial abgeschlossen haben, sollten Sie mit Datenladevorgängen mithilfe von Skripten, der Bearbeitung von Skripten und der Umwandlung von Daten vertraut sein.

1.2 Zielgruppe für diesen Kurs

Sie sollten mit den Grundlagen von Qlik Sense vertraut sein. Dazu gehört die Erstellung von Apps und Visualisierungen.

Sie benötigen Zugriff auf den Dateneditor und müssen berechtigt sein, Daten in Qlik Sense Enterprise on Windows zu laden.

1.3 Paketinhalt

Das heruntergeladene Paket enthält die folgenden Datendateien, die Sie zum Bearbeiten des Tutorials benötigen:

- *Customers.xlsx*
- *Dates.xlsx*
- *Region.txt*
- *Sales.xlsx*

Zudem enthält das Paket eine Kopie der App *Skript-Tutorial*. Sie können die App in Ihren Hub hochladen.

Sie sollten die App wie im Tutorial beschrieben selbst erstellen, um den Lernerfolg zu maximieren. Zudem müssen Sie die Datendateien wie im Tutorial beschrieben hochladen und verbinden, damit die Datenladevorgänge funktionieren.




Wenn Sie Probleme haben, kann Sie die App aber bei der Problembehebung unterstützen. Wir haben angegeben, welche Skriptsegmente zu den einzelnen Lektionen gehören.

1.4 Lektionen in diesem Tutorial

Je nach Ihrer Erfahrung mit Qlik Sense dauert die Bearbeitung dieses Tutorials 3-4 Stunden. Die Themen sollten ihrer Reihenfolge nach bearbeitet werden. Sie können die Arbeit daran jedoch jederzeit unterbrechen und später fortsetzen. Glücklicherweise umfasst das Tutorial keine Tests.

- Einführung in das Laden von Daten
- LOAD- und SELECT-Befehle
- Auswählen und Laden von Daten
- Umbenennen von Feldern
- Reduzieren von Daten
- Umwandeln von Daten
- Zusammenfassung von Tabellen
- Zirkelbezüge
- Synthetische Schlüssel
- Verwenden der Daten in einer App

1.5 Weitere Informationsquellen und Ressourcen

-  [Qlik](#) bietet eine Vielzahl von Ressourcen, wenn Sie noch mehr erfahren möchten.
- [Qlik Online-Hilfe](#) ist verfügbar.
- Schulungen, einschließlich kostenloser Online-Kurse, stehen im  [Qlik Continuous Classroom](#) zur Verfügung.
- Diskussionsforen, Blogs und mehr finden Sie in der  [Qlik Community](#).

2 Scripting in the data load editor

Qlik Sense benutzt ein Datenladeskript, das im Dateneditor verwaltet wird, mit dem Sie eine Verbindung zur Datenbank herstellen können, um verschiedene Datenquellen abzurufen. Eine Datenquelle kann eine Datendatei sein, beispielsweise eine Excel-Datei oder eine .csv-Datei. Eine Datenquelle kann auch eine Datenbank sein, beispielsweise eine Google BigQuery- oder Salesforce-Datenbank.

Sie können Daten auch mithilfe des Datenmanagers laden. Wenn Sie jedoch ein Datenladeskript erstellen, bearbeiten und ausführen möchten, verwenden Sie den Dateneditor.

Im Skript werden die zu ladenden Felder und Tabellen festgelegt. Die Skripterstellung wird häufig verwendet, um anzugeben, welche Daten aus Ihren Datenquellen geladen werden sollen. Sie können auch die Datenstruktur ändern, indem Sie Skriptanweisungen verwenden.

Während der Ausführung des Skripts legt Qlik Sense allgemeine Felder aus verschiedenen Tabellen (Schlüsselfelder) fest, um die Daten zu verknüpfen. Die Datenstruktur der Dateien in der App können Sie im Datenmodell überprüfen. Änderungen an der Datenstruktur können durch Umbenennen der Felder vorgenommen werden, um unterschiedliche Verknüpfungen zwischen den Tabellen zu erhalten.

Nachdem die Daten in Qlik Sense geladen sind, werden sie in der App gespeichert.

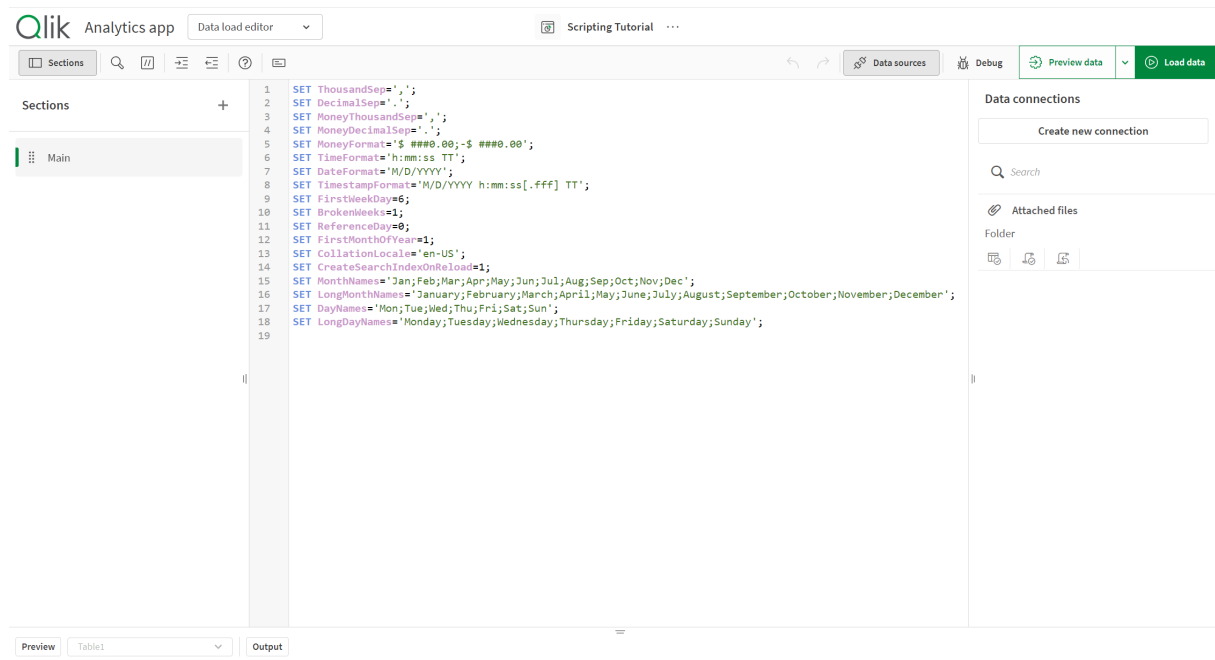
2.1 Data load editor

You can create scripts to load data in Data load editor. The editor is available from the drop-down menu in Qlik Sense.

When you open the data load editor, the script editor appears in the middle of your screen. Script sections are shown as tabs in the left menu. Qlik Sense automatically creates the **Main** section. Data connections are shown in the right menu.

The script must be written using the Qlik Sense script syntax. Qlik Sense syntax keywords are highlighted in blue.

Dateneditor



2.2 Script editor


There are a number of functions available in the editor to assist you in developing the load script.

Accessing syntax help for commands and functions

There are several ways to access syntax help for a Qlik Sense syntax keyword.

Accessing the help portal

You can access detailed help in the Qlik Sense help portal in two different ways.

- Click  in the toolbar to enter syntax help mode. In syntax help mode you can click on a syntax keyword (marked in blue and underlined) to access syntax help.
- Place the cursor inside or at the end of the keyword and press Ctrl+H.



You cannot edit the script in syntax help mode.

Using the autocomplete function


If you start to type a Qlik Sense script keyword, you get an autocomplete list of matching keywords to select from. The list is narrowed down as you continue to type, and you can select from templates with suggested syntax and parameters. A tooltip displays the syntax of the function, including parameters and additional statements, as well as a link to the help portal description of the statement or function.





*You can also use the keyboard shortcut **Ctrl+Space** to show the keyword list, and **Ctrl+Shift+Space** to show a tooltip.*

Commenting in the script

You can insert comments in the script code, or deactivate parts of the script code by using comment marks. All text on a line that follows to the right of `//` (two forward slashes) will be considered a comment and will not be executed when the script is run.

The data load editor toolbar contains a shortcut for commenting or uncommenting code. Click , or press **Ctrl + K** to comment or uncomment code.

Indenting code

You can indent the code to increase readability. Click  to indent the text (increase indentation) or, click  to outdent the text (decrease indentation).

Inserting a prepared test script

You can insert a prepared test script that will load a set of inline data fields. You can use this to quickly create a data set for test purposes. Press **Ctrl + 00** to insert the test script.

3 LOAD- und SELECT-Anweisungen

Sie können mithilfe von LOAD- und SELECT-Befehlen Daten in Qlik Sense laden. Durch jeden dieser Anweisungen wird eine interne Tabelle erstellt. LOAD wird zum Laden von Daten aus Dateien verwendet, während mit SELECT Daten aus Datenbanken geladen werden.

In diesem Tutorial verwenden Sie Daten aus Dateien. Daher werden LOAD-Anweisungen verwendet.

Sie können auch ein vorangestellte LOAD verwenden, um den Inhalt der geladenen Daten beeinflussen zu können. Beispielsweise muss das Umbenennen von Feldern in einer LOAD-Anweisung erfolgen, während die SELECT-Anweisung keine Änderungen an Feldnamen zulässt.

Beim Laden von Daten in Qlik Sense gelten die folgenden Regeln:

- Qlik Sense unterscheidet nicht zwischen Tabellen, die durch eine LOAD- oder eine SELECT-Anweisung generiert werden. Wenn mehrere Tabellen geladen werden, ist es daher unerheblich, ob die Tabellen durch LOAD- oder SELECT-Anweisungen oder eine Kombination beider Anweisungen geladen wurden.
- Die Reihenfolge der Felder in der zugrunde liegenden Datenbank ist für Qlik Sense nicht relevant.
- Feldnamen unterscheiden zwischen Groß- und Kleinschreibung und werden verwendet, um Zuordnungen zwischen Datentabellen einzurichten. Aus diesem Grund müssen Felder im Ladeskript manchmal umbenannt werden, um das gewünschte Datenmodell zu erhalten.

4 Selecting and loading data

Loading data from files, such as Microsoft Excel or any other supported file format, is easily done by using the data selection dialog in the data load editor.

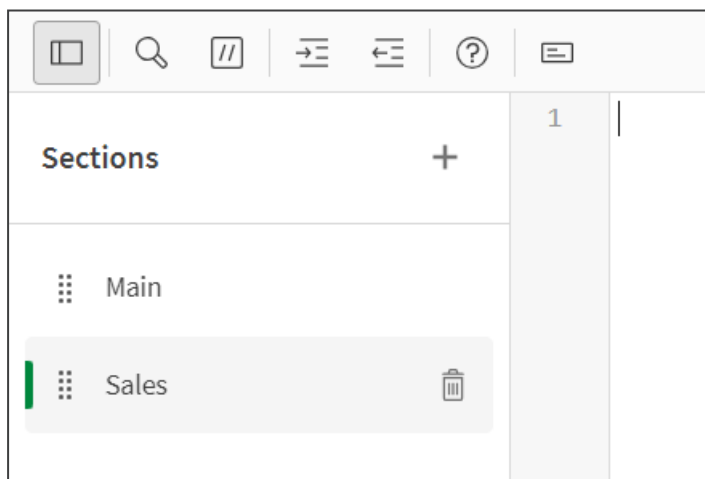
Gehen Sie folgendermaßen vor:

1. Open Qlik Sense.
2. Create a new app.
3. Name the app *Scripting Tutorial*, and then click **Create**.
4. Open the app.



*Before you load data into your app for the first time, there is an option to use **Add data** to easily load data from files. However, in this tutorial we want to see the script, so we will use the data load editor.*

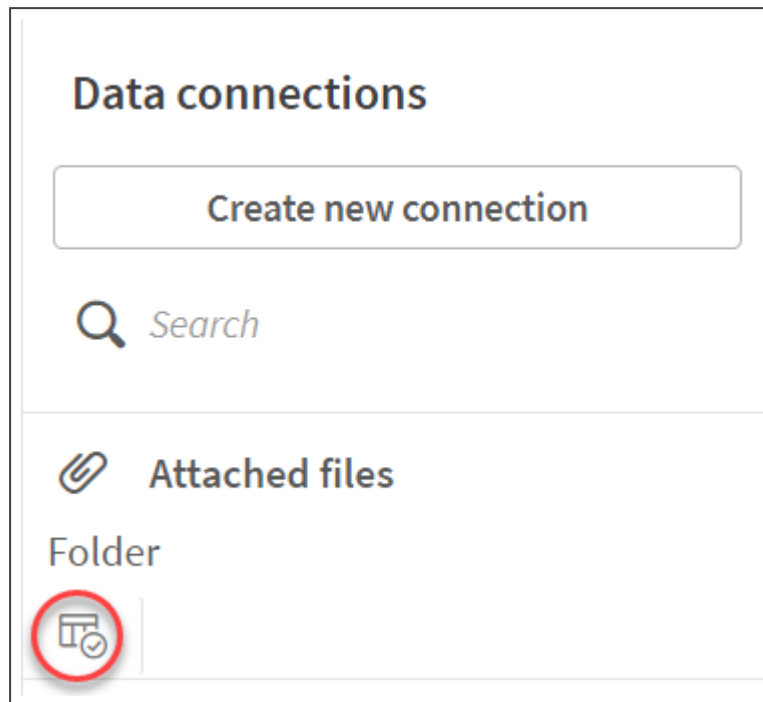
5. Open Dateneditor from the drop-down menu in the top toolbar.
6. Click **+** in the left menu to add a new script section below the section named *Main*.
Using more than one section makes it easy to keep your script organized. The script section will execute in order when you load data.
7. Give the section a name by typing *Sales*.
New Sales tab in Scripting Tutorial app



8. Klicken Sie im rechten Menü unter **AttachedFiles** auf **Daten auswählen**.

4 Selecting and loading data

Select data window



9. Upload and then select *Sales.xlsx*. A data preview window opens.

Data preview window for Sales data file

Select data from Sales.xlsx

Tables → File format: Excel (xlsx) Field names: Embedded field names

Filter tables: Sales data 20

Header size: 1 0

# of Days ...	# of Days to ...	BackO...	Cost	Customer Nu...	Date	GrossS...	Invoice ...	Invoice Nu...	Item Desc	Item Nu...	Ma...	Op
0	22.28	10000000	01/25/2011	64.56	01/25/2011	319976	Nationel Potato Chips	10847	39.7			
0	1.77	10000433	01/29/2011	0.00	01/29/2011	320435	Ebony Asparagus	10795	-1.77			
0	3.86	10000433	01/28/2011	5.65	01/28/2011	320294	Pearl Chardonnay Wine	10895	1.56			
0	8.84	10000433	01/28/2011	20.39	01/28/2011	320274	Tell Tale Firm Tello	10505	15.64			
0	4.47	10000433	01/28/2011	20.49	01/28/2011	320294	Great Muffins	10279	15.2			
0	10.96	10000433	01/28/2011	20.59	01/28/2011	320274	Fast Grape Fruit Roll	10558	8.81			
0	5.3	10000433	01/28/2011	25.25	01/28/2011	320294	Golden Waffles	10900	19.04			
0	13.16	10000433	01/28/2011	31.43	01/28/2011	320294	Ebony New Potatoes	10797	17.2			
0	15.32	10000433	01/28/2011	42.31	01/28/2011	320294	High Top Tomatoes	10167	25.3			
0	27.71	10000433	01/28/2011	45.50	01/28/2011	320294	High Top Golden Delicious Apples	10197	15.97			
0	40.81	10000433	01/28/2011	48.68	01/28/2011	320294	Fast Golden Raisins	10561	6.12			
0	26.46	10000433	01/28/2011	48.86	01/28/2011	320294	Onioner Spaghetti	10890	20.45			
0	19.55	10000433	01/28/2011	63.16	01/28/2011	320294	Bravo Beef Soup	10649	41.08			
0	23.11	10000433	01/28/2011	84.59	01/28/2011	320294	High Top Cauliflower	10166	58.1			
0	52.91	10000433	01/28/2011	104.66	01/28/2011	320263	Ebony Plums	10823	47.56			
0	55.94	10000433	01/28/2011	110.27	01/28/2011	320294	Fast Dried Apples	10554	40.92			
0	77.1	10000433	01/28/2011	156.50	01/28/2011	320265	Just Right Chicken Ramen Soup	10967	73.14			
0	85.22	10000433	01/28/2011	157.70	01/28/2011	320294	Moms Sliced Chicken	10387	66.17			
0	113.58	10000433	01/28/2011	162.74	01/28/2011	320294	High Top Golden Delicious Apples	10197	42.65			

LOAD

"# of Days Late",
"# of Days to Ship",
"BackOrder",
"Cost",
"Customer Number",
"Date",
"GrossSales",
"Invoice Date",
"Invoice Number",
"Item Desc",
"Item Number",
"Margin",

Cancel Insert script

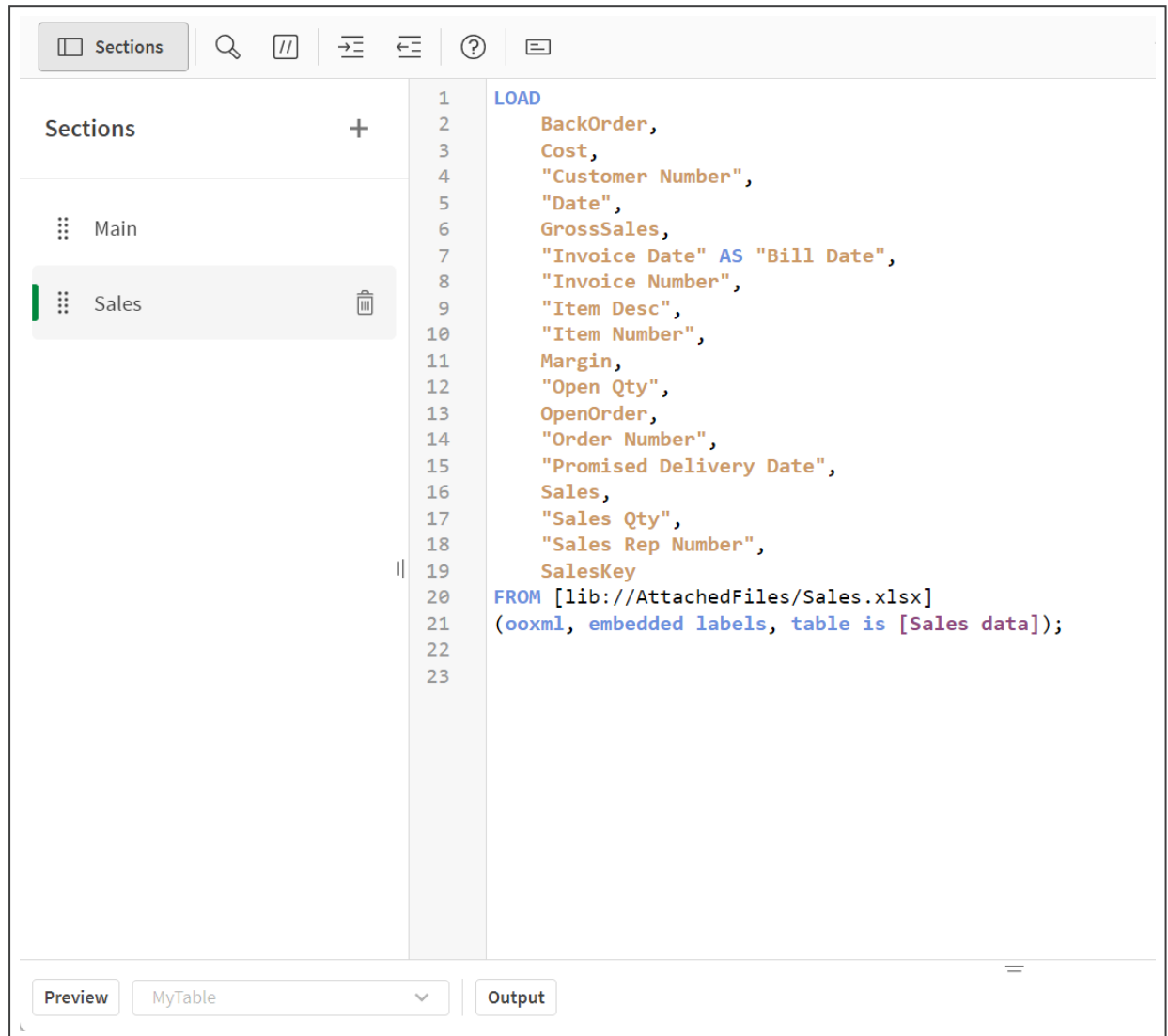
10. Deselect the fields *# of Days Late* and *# of Days to Ship*. You might need to click on the field headings to see the complete field names.
11. Search for *date* in the **Filter fields** search field.
12. Click on the heading *Invoice Date* and type *Bill Date* to rename the field.

4 Selecting and loading data

13. Klicken Sie auf **Skript einfügen**. The load script is inserted into the *Sales* section of the script editor. Note that Qlik Sense puts double quotes around field names that contain a space.

Ihr Skript sollte folgendermaßen aussehen:

Load script in Sales tab

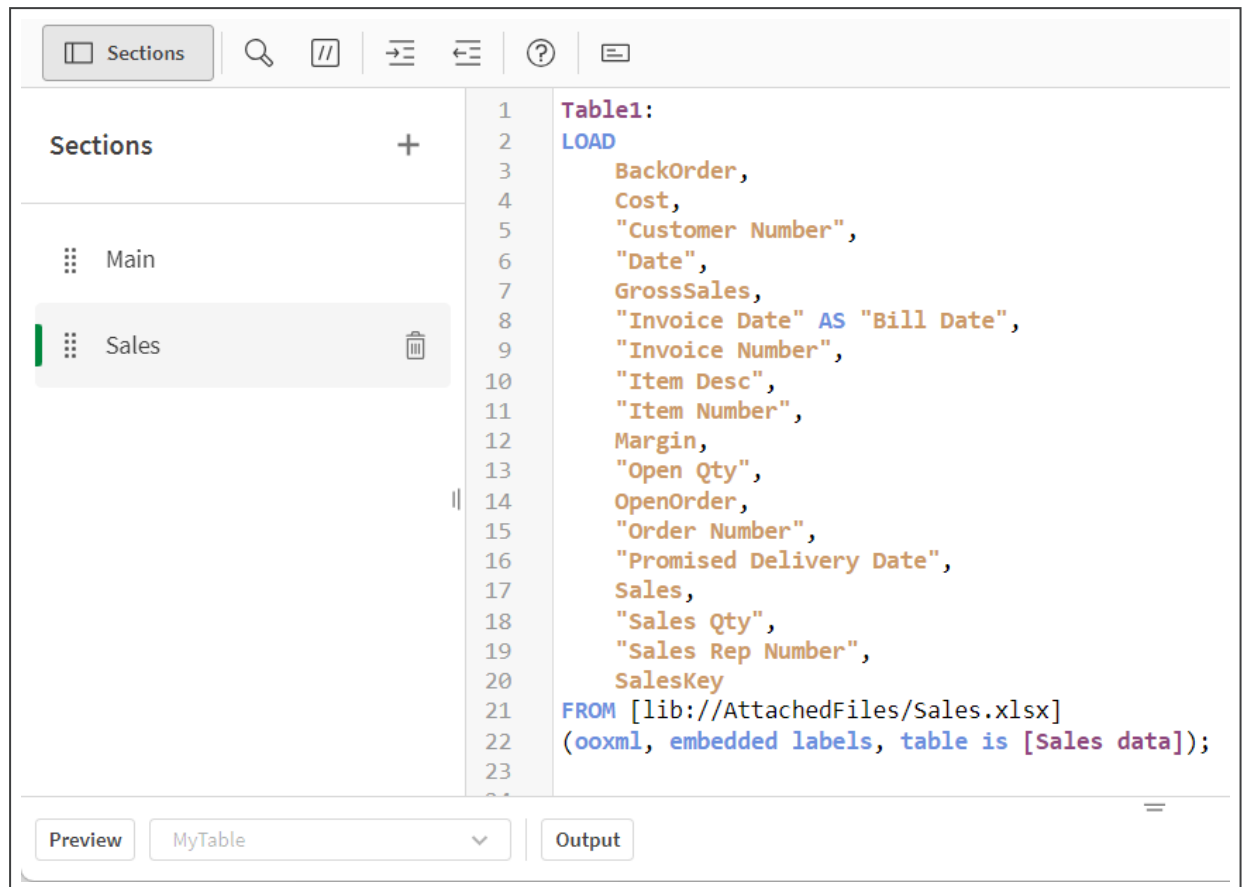


14. Add the following row above the *LOAD* statement to name the table *Table1*:
- Table1:
15. Now adjust the script to ensure that the dates are interpreted correctly. Change the *Date* field to the following:
- Date#(`Date`, 'MM/DD/YYYY') as "Date",

Ihr Skript sollte folgendermaßen aussehen:

4 Selecting and loading data

Updated load script in Sales tab








16. In the upper right corner, click **Load data**.
This will load the data into the app. A script execution progress window is displayed. When it is finished you will see a summary of possible errors and synthetic keys even if there are none.
17. Click **Close**.
18. Open the data model viewer from the drop-down menu in the top toolbar. By clicking  the data model viewer will open in a new tab.
19. Select  and  in the top menu to show the table view that is used in this tutorial. If your table is not displayed properly, you can remove the existing load script and build the script again.

Table view in data model viewer of Sales data

Table1
BackOrder
Cost
Customer Number
Date
GrossSales
Bill Date
Invoice Number
Item Desc
Item Number
Margin
Open Qty
OpenOrder
Order Number
Promised Delivery Date
Sales
Sales Qty
Sales Rep Number
SalesKey

Now, let's load another table called *Dates*. After we load the table, Qlik Sense will connect it with the *Sales* table on the *Date* field.

20. Open the **Data load editor**.
21. Klicken Sie auf **+**, um einen neuen Skriptabschnitt hinzuzufügen.
22. Name the section *Dates*. If the new section *Dates* is not already placed below *Sales*, move the pointer over , and then drag the section down below the section *Sales* to rearrange the order.
23. Click on the top row of the script and click .
Make sure `//` is added into the script.
24. Add the following text after `//`:
Loading data from Dates.xlsx

The top line of your script should now look like this:

```
// Loading data from Dates.xlsx
```

25. Klicken Sie im rechten Menü unter **AttachedFiles** auf **Daten auswählen**.

4 Selecting and loading data



Under **Field names**, make sure that **Embedded field names** is selected to include the names of the table fields when you load the data.

26. Upload and then select *Dates.xlsx*. A data preview window opens.

Data preview window for Dates data file

Select data from Dates.xlsx

Tables: Filter tables

File format: Excel (xlsx)

Field names: Embedded field names

Header size: 1 0 +

Fields: Filter fields

Date	Month	Quarter	Week	Year
1/12/2011	Jan	Q1	3	2011
1/13/2011	Jan	Q1	3	2011
1/18/2011	Jan	Q1	3	2011
1/19/2011	Jan	Q1	4	2011
1/20/2011	Jan	Q1	4	2011
1/21/2011	Jan	Q1	4	2011
1/22/2011	Jan	Q1	4	2011
1/25/2011	Jan	Q1	4	2011
1/26/2011	Jan	Q1	5	2011
1/27/2011	Jan	Q1	5	2011
1/28/2011	Jan	Q1	5	2011
1/29/2011	Jan	Q1	5	2011
2/1/2011	Feb	Q1	5	2011
2/2/2011	Feb	Q1	6	2011
2/3/2011	Feb	Q1	6	2011
2/4/2011	Feb	Q1	6	2011
2/9/2011	Feb	Q1	6	2011
2/9/2011	Feb	Q1	7	2011
2/10/2011	Feb	Q1	7	2011

LOAD
"Date",
"Month",
Quarter,
"Week",
"Year"
FROM [lib://AttachedFiles/Dates.xlsx]
(ooxml, embedded labels, table is Dates);

Cancel Insert script

27. Klicken Sie auf **Skript einfügen**.

Ihr Skript sollte folgendermaßen aussehen:

Load script in Dates tab

Sections

Main

Sales

Date

```
//Loading data from Dates.xlsx
LOAD
    "Date",
    "Month",
    Quarter,
    "Week",
    "Year"
FROM [lib://AttachedFiles/Dates.xlsx]
(ooxml, embedded labels, table is Dates);
```

28. Add the following on the row above the *LOAD* statement to name the table *Table2*:

Table2:

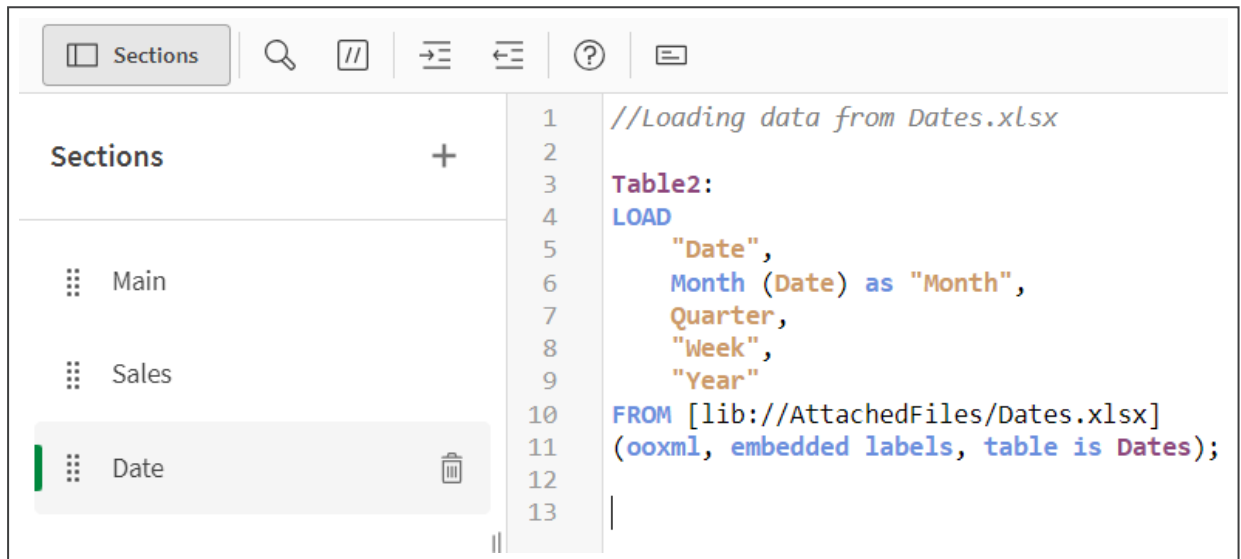
29. To ensure that the *Month* column in the file *Dates.xlsx* is interpreted correctly in Qlik Sense we need to apply the *Month* function to the *Date* field.

Change the *Month* field to the following:

Month (Date) as "Month",

Ihr Skript sollte folgendermaßen aussehen:

Updated load script in Dates tab



Now you have created a script to load the selected data from the file *Dates.xlsx*. It is time to load the data into the app.

30. In the upper right corner, click **Load data**.

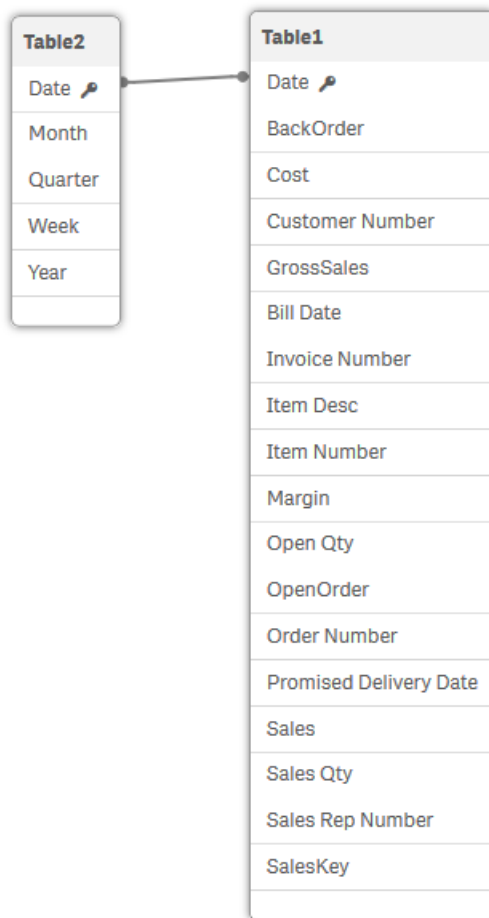
When you click **Load data**, the data is loaded into the app and the script is saved.

31. When the script execution is finished, click **Close**.

32. Öffnen Sie die **Datenmodellansicht**.

Now you can see that a connection has been made between the two fields named *Date* in the two tables.

Table view in data model viewer



33. Click **Preview** in the bottom left corner. Click on the name of the table *Table2*.

This will display information about the table. In the **Preview** field you can see that 628 rows of data have been loaded into the internal table *Table2*. If you instead click on a field in the table, you will see information about the field.

Table preview in data model viewer

Table2

Date

Month

Quarter

Week

Year

Table1

Date

BackOrder

Cost

Customer Number

GrossSales

Bill Date

Invoice Number

Item Desc

Item Number

Margin

Open Qty

OpenOrder

Order Number

▼ Preview

Table2		Preview of data				
Rows	628	Date	Month	Quarter	Week	Year
Fields	5	01/12/2011	Jan	Q1	3	2011
Keys	1	01/13/2011	Jan	Q1	3	2011
Tags	\$key \$numeric \$integer \$timestamp \$date \$ascii \$text	01/18/2011	Jan	Q1	3	2011
		01/19/2011	Jan	Q1	4	2011
		01/20/2011	Jan	Q1	4	2011
		01/21/2011	Jan	Q1	4	2011
		01/22/2011	Jan	Q1	4	2011

The data is now available to use in visualizations in an app. We will show you how later in this tutorial.

5 Renaming fields

In the previous topic, we showed you how to rename fields in the data preview window. In the heading of the data preview window, you renamed *Invoice Date* to *Bill Date*. When you inserted the load script, you could see that the field would be renamed using the keyword *AS*.

We can also perform this action directly in the script.

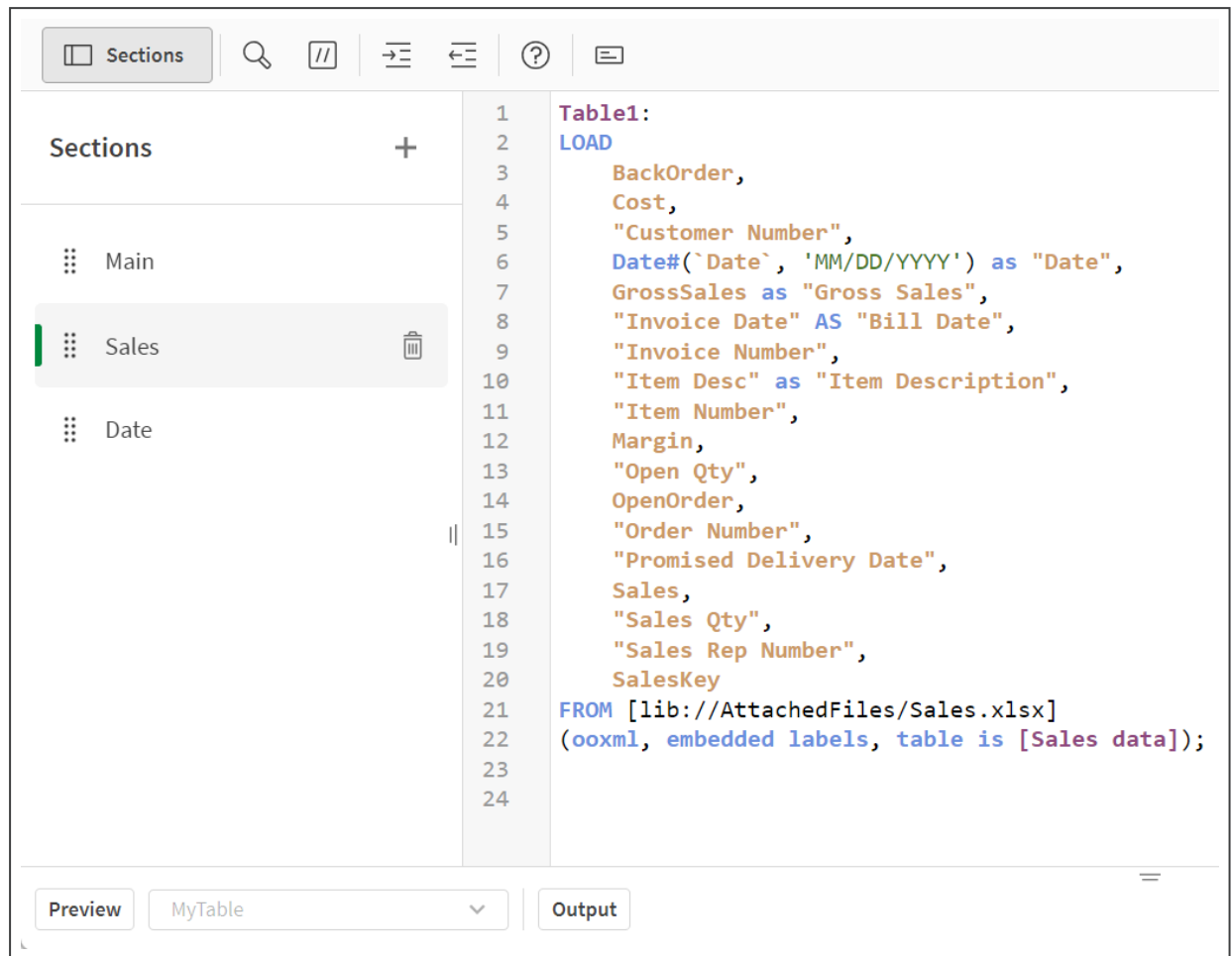
Gehen Sie folgendermaßen vor:

1. Öffnen Sie den **Dateneditor** in der *Scripting Tutorial* App.
2. Click the *Sales* tab.
3. In the load script, make the following changes. Note that you have to include parentheses around fields that contain a space.
 - i. Change *GrossSales*, to:
`GrossSales AS "Gross Sales",`
 - ii. Change *"Item Desc"*, to:
`"Item Desc" AS "Item Description",`

Ihr Skript sollte folgendermaßen aussehen:

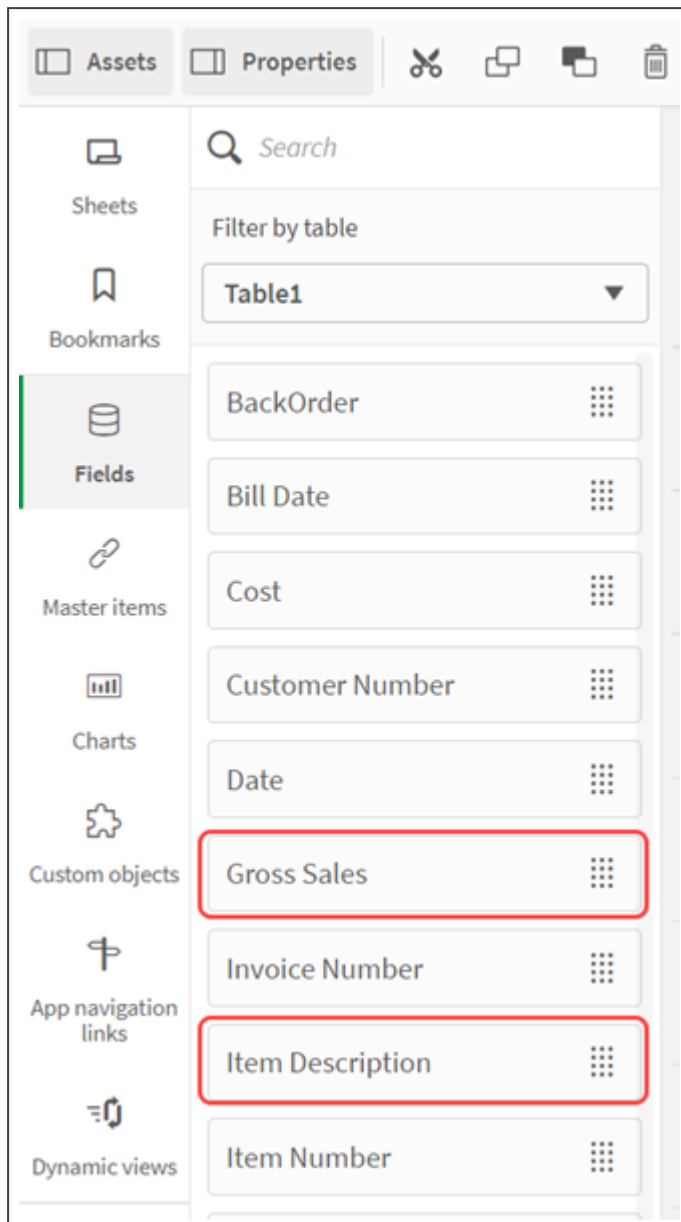
5 Renaming fields

Load script window with script to rename fields



4. Click **Load data**. The data is loaded.
5. Öffnen Sie die **Datenmodellansicht**. Confirm that the fields have been renamed.
6. You can also view all your fields in an app. Click the **Analysis/Sheet** tab in the top toolbar. The app opens in sheet view.
7. Click **Edit sheet**, and then click **Fields** in the assets panel. You can see the field names that you changed. You can use any of these fields in the visualizations that you create in your app.

Renamed fields in analysis view



6 Reducing data

Qlik Sense provides several different ways to reduce the amount of data that you load into your app. You can, for example, filter data from files or from data connectors.

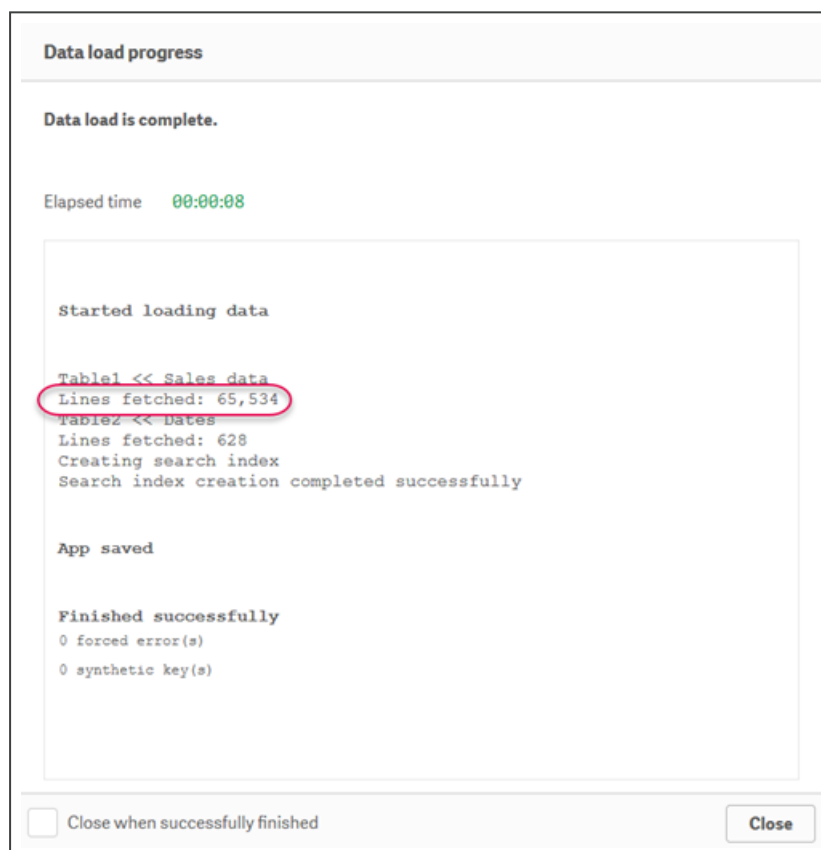
You can also reduce data directly in the load script.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie den **Dateneditor** in der *Scripting Tutorial App*.
2. Click **Load data**.

Based on the load script that you have written so far, Qlik Sense loads 65,534 lines from the *Sales.xlsx* data file into *Table1*. Note that *Sales data* is the name of the tab that contains your table in the original *Sales.xlsx* file.

Data load progress window

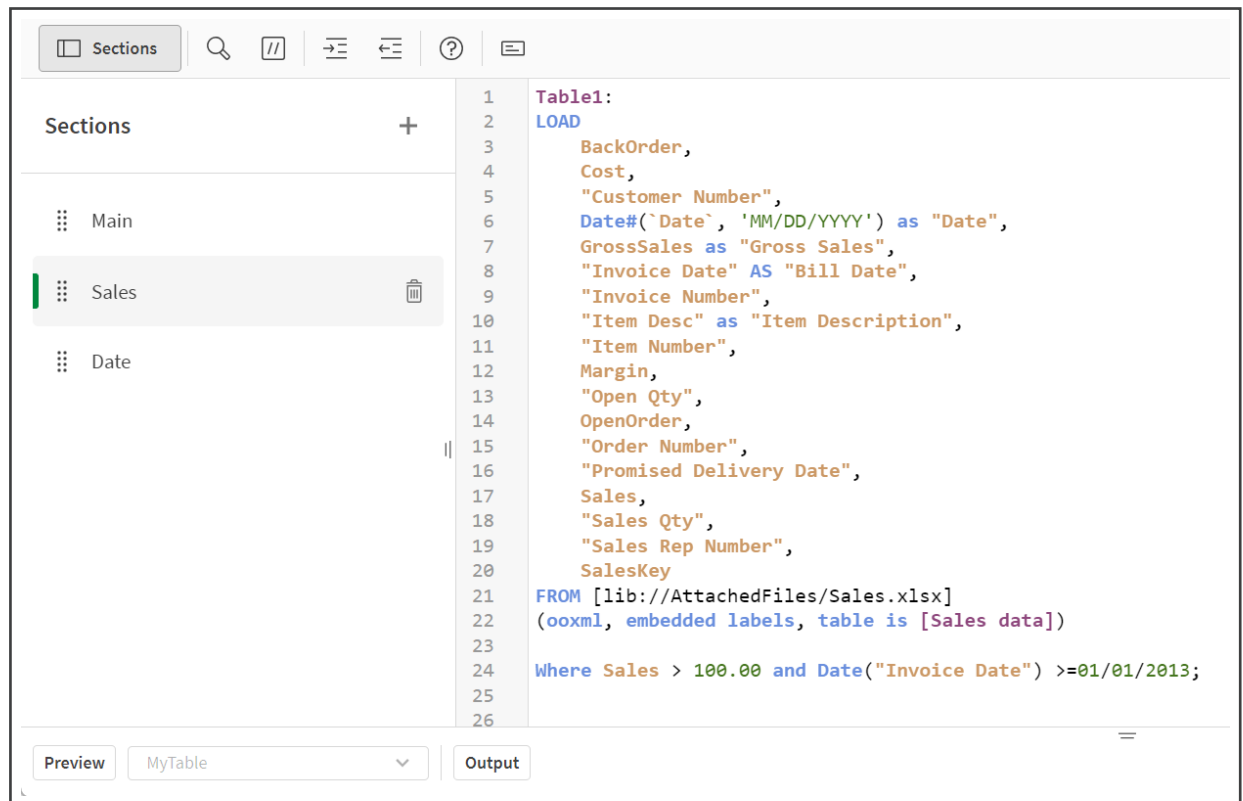


3. In the data load editor, click the *Sales* tab.
4. Delete the semicolon at the end of this line:
`(ooxml, embedded labels, table is [Sales data]);`
5. Add the following line at the end of the load script:
`where Sales > 100.00 and Date("Invoice Date") >= '01/01/2013';`

This tells Qlik Sense to only load data where sales are greater than \$100.00. It also uses the Date function to load data where the date is equal to or greater than January 1, 2013.

Ihr Skript sollte folgendermaßen aussehen:

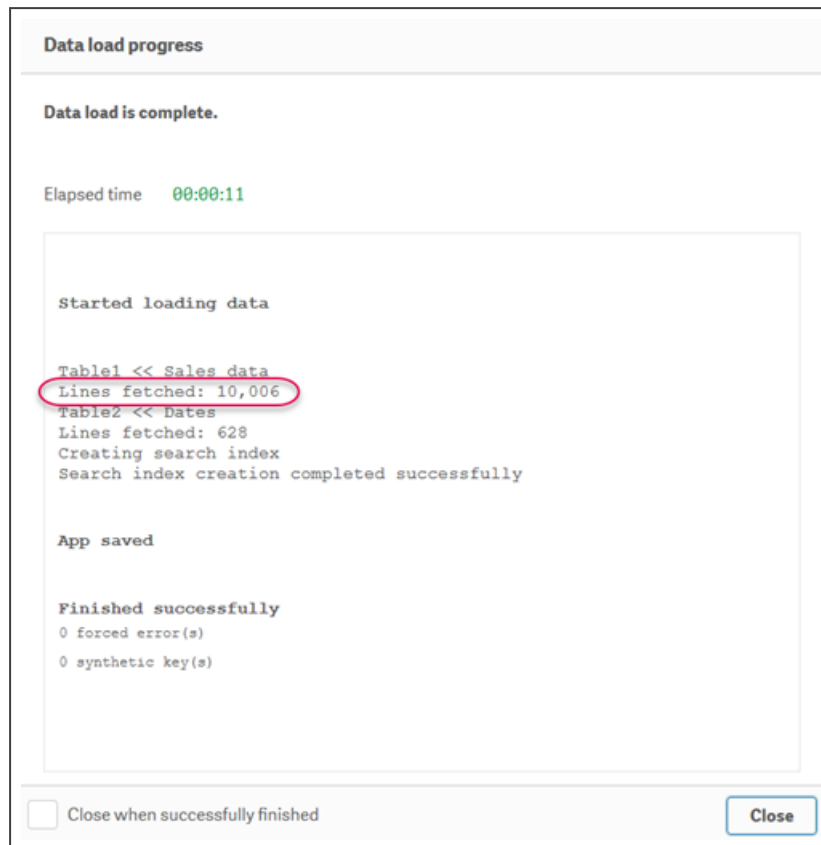
Load script window with script to reduce amount of data loaded



6. Click **Load data**.

Based on your updated load script, Qlik Sense now loads fewer lines from the *Sales.xlsx* data file.

Data load progress window with reduced data load



- If you add the data to a table in your app, you can see that only the data that conforms to the conditions that you created was loaded.

Table containing Bill Date and Sales fields

My new sheet

Click to add title

Bill Date	Sales
01/01/2015	100.47
01/01/2015	106.93
01/01/2015	107.72
01/01/2015	108.88
01/01/2015	113.94
01/01/2015	118.89
01/01/2015	122.13
01/01/2015	123.6
01/01/2015	123.79
01/01/2015	124.32

Note that we added the *Sales* field as a dimension. This is so that the *Sales* values are shown individually. If we had added *Sales* as a measure the values would have instead been aggregated per date.

Typically, you would add *Sales* as a measure. With measures you have the option to display values as currency amounts (for example, dollars) by applying number formatting to the column.

Number formatting applied to Sales measure

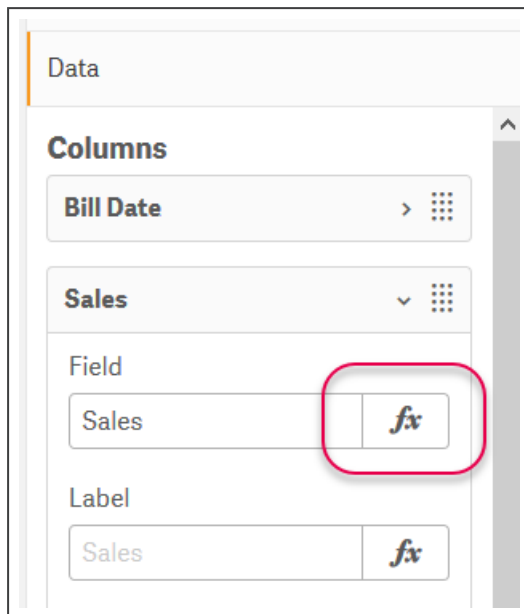
The screenshot shows the 'Data' panel in Qlik Sense. It features a table with columns 'Sum' and 'Sales'. Below the table, the 'Expression' field is set to 'Sum(Sales)' with a function icon (fx). The 'Label' field is also set to 'Sum(Sales)' with a function icon. The 'Number formatting' dropdown is set to 'Money'. The 'Format pattern' is '\$ ###0.00;-\$ ###0.00'. An example shows the result as '\$ 1000.12'.

However, as we are using *Sales* as a dimension, we need a different approach. In this case, we will use a chart expression. Even though we are not discussing chart functions in detail in this tutorial, this is a good opportunity for a quick example.

When we use *Sales* as a dimension, the values are displayed as numeric, as shown in the table above.

To fix this, you can open the chart expression editor *Sales* field by clicking **fx**, and then using the *Money* function.

Opening the expression editor



Enter the following in the expression editor.

`=Money(sales)`

Expression editor



The values in the field will now display as monetary units. Those units (in this case, dollars) are specified in the **Main** section of the load script.

Table containing Bill Date and Sales fields. Sales now shown in dollars

My new sheet	
Click to add title	
Bill Date	Money(Sales)
01/01/2015	\$100.47
01/01/2015	\$108.83
01/01/2015	\$107.72
01/01/2015	\$108.88
01/01/2015	\$115.94
01/01/2015	\$118.89
01/01/2015	\$122.13
01/01/2015	\$123.60
01/01/2015	\$123.73
01/01/2015	\$124.32

- Now that we have completed this example, comment out the Where statement. Don't forget to add a semicolon to the end of your first LOAD statement.

Ihr Skript sollte folgendermaßen aussehen:

Load script window with Where statement commented out

The screenshot shows the Qlik Sense Load Script Editor interface. On the left, the 'Sections' pane lists 'Main', 'Sales' (selected), and 'Date'. The main editor area displays a script for 'Table1' with the following code:

```

1 Table1:
2 LOAD
3     BackOrder,
4     Cost,
5     "Customer Number",
6     Date#(`Date`, 'MM/DD/YYYY') as "Date",
7     GrossSales as "Gross Sales",
8     "Invoice Date" AS "Bill Date",
9     "Invoice Number",
10    "Item Desc" as "Item Description",
11    "Item Number",
12    Margin,
13    "Open Qty",
14    OpenOrder,
15    "Order Number",
16    "Promised Delivery Date",
17    Sales,
18    "Sales Qty",
19    "Sales Rep Number",
20    SalesKey
21 FROM [lib://AttachedFiles/Sales.xlsx]
22 (ooxml, embedded labels, table is [Sales data]);
23
24 //Where Sales > 100.00 and Date("Invoice Date") >=01/01/2013;
25
26

```

At the bottom of the editor, there is a 'Preview' button, a dropdown menu showing 'MyTable', and an 'Output' button.

7 Transforming data

You can transform and manipulate data using many different techniques in Dataneditor.

One of the advantages to data manipulation is that you can choose to load only a subset of the data from a file, such as a few chosen columns from a table, to make the data handling more efficient. You can also load the data more than once to split up the raw data into several new logical tables. It is also possible to load data from more than one source and merge it into one table in Qlik Sense.

In this topic, you will perform some basic data transformation using a Resident load, and then a Preceding load.

7.1 Resident LOAD

You can use the Resident source qualifier in a LOAD statement to load data from a previously loaded table. This is also useful when you want to perform calculations on data loaded with a SELECT statement where you do not have the option to use Qlik Sense functions, such as date or numeric value handling.

In this example, you will create a new table called *Sales_Buckets* and then load the data from *Table1* using a resident load. In the *Sales_Buckets* table, you will create a variable called *quantity_threshold*, and then use a Where statement to only load data that meets that threshold.

1. Öffnen Sie den **Dataneditor** in der *Scripting Tutorial* App.
2. Click the *Sales* tab.
3. Add the following to the end of your script:

```
SET quantity_threshold = 12000;
```

```
Sales_Buckets:
LOAD
"Sales Qty" as "High_Quantity",
"Item Description" as "Item",
"Customer Number" as "Customer"
Resident Table1
where ("Sales Qty" > $(quantity_threshold));
```

Ihr Skript sollte folgendermaßen aussehen:

7 Transforming data

Load script window with script to create new table called *Sales_Buckets*

The screenshot shows the Qlik Sense Load Script window. On the left, the 'Sections' pane lists 'Main', 'Sales' (selected), and 'Date'. The main area displays a script for creating a new table named *Sales_Buckets*. The script is as follows:

```
1 Table1:
2 LOAD
3     BackOrder,
4     Cost,
5     "Customer Number",
6     Date#(`Date`, 'MM/DD/YYYY') as "Date",
7     GrossSales as "Gross Sales",
8     "Invoice Date" AS "Bill Date",
9     "Invoice Number",
10    "Item Desc" as "Item Description",
11    "Item Number",
12    Margin,
13    "Open Qty",
14    OpenOrder,
15    "Order Number",
16    "Promised Delivery Date",
17    Sales,
18    "Sales Qty",
19    "Sales Rep Number",
20    SalesKey
21 FROM [lib://AttachedFiles/Sales.xlsx]
22 (ooxml, embedded labels, table is [Sales data]);
23
24 SET quantity_threshold = 12000;
25
26 SET quantity_threshold = 12000;
27
28 Sales_Buckets:
29 LOAD
30     "Sales Qty" as "High_Quantity",
31     "Item Description" as "Item",
32     "Customer Number" as "Customer"
33 Resident Table1
34 Where ("Sales Qty" > $(quantity_threshold));
35
36
```

At the bottom, there are buttons for 'Preview', 'MyTable' (with a dropdown arrow), and 'Output'.

4. Klicken Sie auf **Daten laden**.
5. Öffnen Sie die **Datenmodellansicht**. You can see that you created a new table called *Sales_Buckets* with the data loaded according to the fields that you specified, and the threshold that you set.

Sales_Buckets table in data model viewer

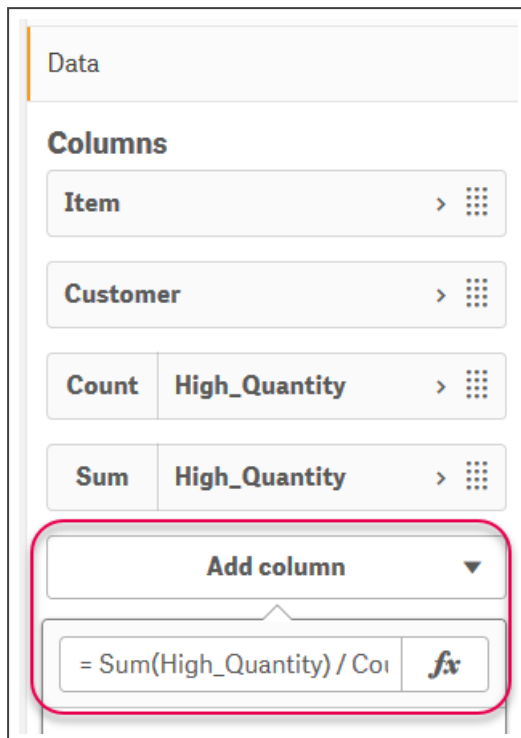
The screenshot shows the data model viewer with three tables: Sales_Buckets, Table2, and Table1. Sales_Buckets has fields High_Quantity, Item, and Customer. Table2 has fields Date, Month, Quarter, Week, and Year. Table1 has fields Date, BackOrder, Cost, Customer Number, Gross Sales, Bill Date, Invoice Number, and Item Description. A line connects the Date field in Table2 to the Date field in Table1. Below the model is a 'Preview' section with a table showing the data for Sales_Buckets.

Sales_Buckets		Preview of data		
		High_Quantity	Item	Customer
Rows	11			
Fields	3	16000	High Top Dried Mushrooms	10025737
Keys	0	12500	Blue Label Canned Peas	10017036
Tags	\$numeric \$integer \$ascii \$text	16000	High Top Dried Mushrooms	10025052
		13600	High Top Dried Mushrooms	10006919
		16000	High Top Dried Mushrooms	10006919
		13600	High Top Dried Mushrooms	10025737
		13600	High Top Dried Mushrooms	10025052

6. Add the data to a table in your app. Add *Item* and *Customer* as dimensions. Add *High-Quantity* as a measure aggregated on *Count*, and then again aggregated on *Sum*. Next, add a new column as a measure with the following formula:

$$= \text{Sum}(\text{High_Quantity}) / \text{Count}(\text{High_Quantity})$$

New measure with formula



Your table shows, for example, that *Customer* 10025737 has made 4 large orders of *High Top Dried Mushrooms*, with an average quantity of 14,800. To perform sorts on the data in the fields, close **Edit** mode by clicking **Done**.

Table showing customers that have made large orders

Item	Customer	Count(High_Quantity)	Sum(High_Quantity)	= Sum(High_Quantity) / Count(High_Quantity)
Totals		11	158100	14372.727272727
Blue Label Canned Peas	10017036	1	12500	12500
High Top Dried Mushrooms	10006919	3	43200	14400
High Top Dried Mushrooms	10025052	3	43200	14400
High Top Dried Mushrooms	10025737	4	59200	14800

- Now that we have completed this example, comment out the script for the *quantity_threshold* variable and the *Sales_Buckets* table.

The end of your script should now appear as follows:

Commented out script

```
(ooxml, embedded labels, table is [Sales data]);
// SET quantity_threshold = 12000;
//
//      Sales_Buckets:
//      LOAD
//      "Sales Qty" as "High Quantity",
//      "Item Description" as "Item",
//      "Customer Number" as "Customer"
//      Resident Table1
//      Where ("Sales Qty" > ${quantity_threshold});
```

7.2 Preceding LOAD

A preceding load allows you to perform transformations and apply filters so that you can load data in one pass. Basically, it is a LOAD statement that loads from the LOAD or SELECT statement below, without specifying a source qualifier such as From or Resident that you would normally do. You can stack any number of LOAD statements this way. The statement at the bottom will be evaluated first, then the statement above that, and so on until the top statement has been evaluated.

As mentioned earlier in this tutorial, you can load data into Qlik Sense using the LOAD and SELECT statements. Each of these statements generates an internal table. LOAD is used to load data from files or from an inline table, while SELECT is used to load data from databases. You have used data from files in this tutorial. In this example, you will use an inline table. However, it is worth noting that a preceding load can be used above a SELECT statement to manipulate your data. The basics are the same as you will see here using LOAD.

This example is not related to the data we are loading in this tutorial. It is only used to show an example of what a preceding load can look like. You will create an inline table in the data load editor called *Transactions*. Date interpretation will be performed in the preceding LOAD, where a new field called *transaction_date* will be created. This field is created from the *sale_date* field.

1. Create a new app and call it *ReformatDate*.
2. Open the data load editor, and then create a new tab called *TransactionData*.
3. Add the following script:

```
Transactions:
Load *,
Date(Date#(sale_date,'YYYYMMDD'),'DD/MM/YYYY') as transaction_date;
Load * Inline [ transaction_id, sale_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 ];
```

Ihr Skript sollte folgendermaßen aussehen:

7 Transforming data

Load script with preceding load

The screenshot shows the Qlik Sense script editor. On the left, the 'Sections' pane lists 'Main' and 'TransactionData'. The main editor area displays a script with 11 lines of code. The script starts with a 'Transactions:' section, followed by a 'Load *;' statement, and then a 'Date#' statement to create a new field 'transaction_date' from the 'sale_date' field. The script concludes with a 'Load * Inline' statement listing various transaction records with their IDs, dates, amounts, quantities, customer IDs, sizes, and color codes.

```
1 Transactions:
2 Load *,
3 Date(Date#(sale_date,'YYYYMMDD'),'DD/MM/YYYY') as transaction_date;
4 Load * Inline [ transaction_id, sale_date, transaction_amount, transaction_quantity, customer_id, size, color_code
5 3750, 20180830, 23.56, 2, 2038593, L, Red
6 3751, 20180907, 556.31, 6, 203521, m, orange
7 3752, 20180916, 5.75, 1, 5646471, S, blue
8 3753, 20180922, 125.00, 7, 3036491, l, Black
9 3754, 20180922, 484.21, 13, 049681, xs, Red
10 3756, 20180922, 59.18, 2, 2038593, M, Blue
11 3757, 20180923, 177.42, 21, 203521, XL, Black ];
```

4. Klicken Sie auf **Daten laden**.
5. Öffnen Sie die **Datenmodellansicht**. Select and expand the *Transactions* table. You can see that all the fields were loaded as specified by the * in the preceding load statement. A new field called *transaction_date* was created. The field has the reformatted date.

New field called *transaction_date* in data model viewer

The screenshot displays the Qlik Sense data model viewer. At the top, the 'Transactions' table is shown with its fields: transaction_id, sale_date, transaction_amount, transaction_quantity, customer_id, size, color_code, and transaction_date. Below this, the 'Preview' section shows a table with 7 rows of data. The first row is highlighted, showing transaction_id 3750, sale_date 20180830, transaction_amount 23.56, transaction_quantity 2, customer_id 2038593, size L, color_code Red, and transaction_date 30/08/2018. The subsequent rows show transactions 3751 through 3757, with their respective details.

transaction_id	sale_date	transaction_amount	transaction_quantity	customer_id	size	color_code	transaction_date
3750	20180830	23.56	2	2038593	L	Red	30/08/2018
3751	20180907	556.31	6	203521	m	orange	07/09/2018
3752	20180916	5.75	1	5646471	S	blue	16/09/2018
3753	20180922	125.00	7	3036491	l	Black	22/09/2018
3754	20180922	484.21	13	049681	xs	Red	22/09/2018
3756	20180922	59.18	2	2038593	M	Blue	22/09/2018
3757	20180923	177.42	21	203521	XL	Black	23/09/2018

8 Concatenation

Concatenation is an operation that takes two tables and combines them into one.

The two tables are added to each other by stacking one on top of the other, with a column for each distinct column name. The data is not changed and the resulting table contains the same number of records as the two original tables together. Several concatenate operations can be performed sequentially, so that the resulting table is concatenated from more than two tables.

8.1 Automatic concatenation

If the field names and the number of fields of two or more loaded tables are exactly the same, Qlik Sense will automatically concatenate the content of the different statements into one table.

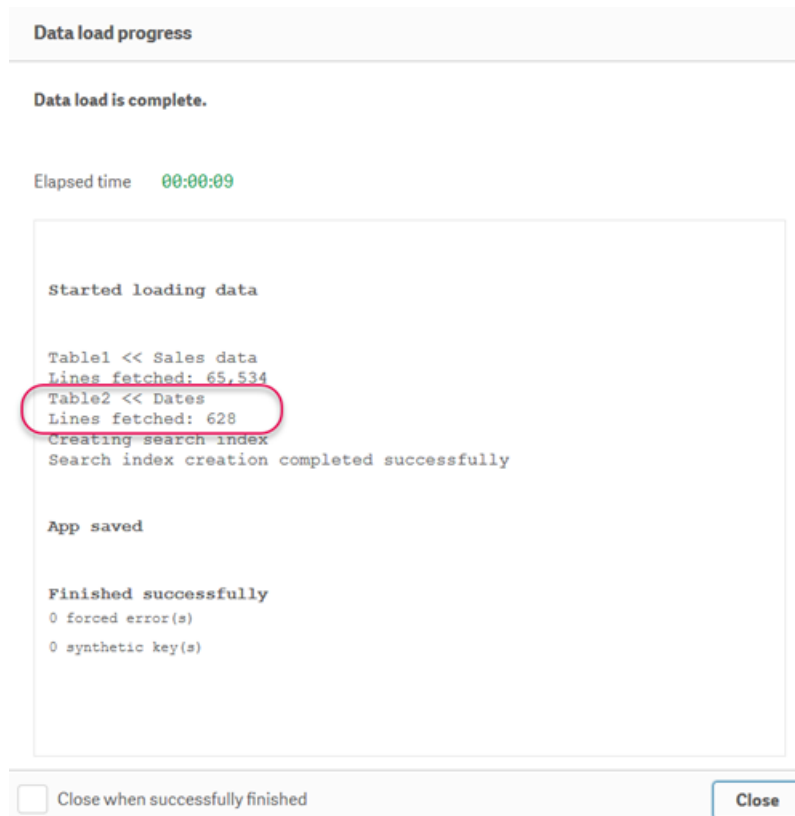
The number and names of the fields must be exactly the same for automatic concatenation to take place. The order of the two LOAD statements is arbitrary, but the table will be given the name of the table that is loaded first.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie den **Dateneditor** in der *Scripting Tutorial* App.
2. Click the **Dates** tab.
3. Klicken Sie auf **Daten laden**.

Based on the load script that you have written so far, Qlik Sense loads 628 lines from the *Dates.xlsx* data file into *Table2*.

Data load progress window



- On a new line in the script in the section *Dates*, copy and paste the LOAD statement for *Table2*. This will cause the data to be loaded twice. Name the second table *Table2a*. You could also delete the existing script, and copy and paste the following:

Table2:

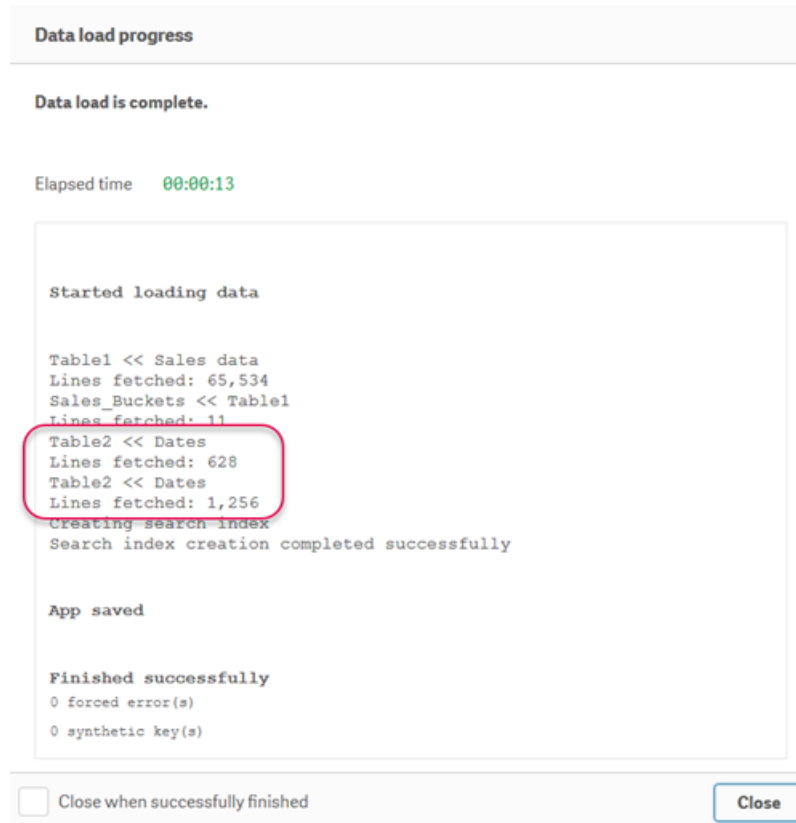
```
LOAD
    "Date",
    Month (Date) as "Month",
    Quarter,
    "Week",
    "Year"
FROM [lib://AttachedFiles/Dates.xlsx]
(ooxml, embedded labels, table is Dates);

Table2a:
LOAD
    "Date",
    Month (Date) as "Month",
    Quarter,
    "Week",
    "Year"
FROM [lib://AttachedFiles/Dates.xlsx]
(ooxml, embedded labels, table is Dates);
```

- Klicken Sie auf **Daten laden**.

Qlik Sense does not load *Table2* and then *Table2a*. Instead, it recognizes that *Table2a* has the same field names and the number of fields as *Table2*. It then adds the data of *Table2a* to *Table2*, and deletes table *Table2a*. The result is that *Table2* now has 1,256 lines.

Concatenation in data load progress window

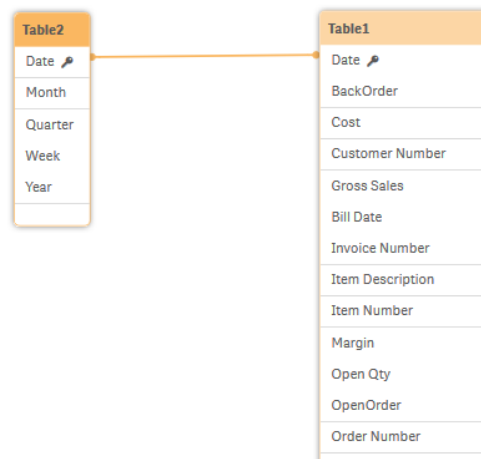


6. Öffnen Sie die **Datenmodellansicht**.

7. Click **Show preview**.

Only *Table2* was created. Select *Table2*. The table has 256 rows.

Data model viewer showing Table2



▼ Preview

Table2		Preview of data				
Rows	1256	Date	Month	Quarter	Week	Year
Fields	5	01/12/2011	Jan	Q1	3	2011
Keys	1	01/13/2011	Jan	Q1	3	2011
Tags	\$key \$numeric \$integer \$timestamp \$date \$ascii \$text	01/18/2011	Jan	Q1	3	2011
		01/19/2011	Jan	Q1	4	2011
		01/20/2011	Jan	Q1	4	2011
		01/21/2011	Jan	Q1	4	2011
		01/22/2011	Jan	Q1	4	2011

8.2 Forced concatenation

Even if two or more tables do not have exactly the same set of fields, it is still possible to force Qlik Sense to concatenate the two tables. This is done with the Concatenate prefix in the script, which concatenates a table with another named table or with the most recently created table.

Gehen Sie folgendermaßen vor:

1. Edit the LOAD statement for *Table2a*, adding Concatenate and commenting out *Week*.
Your script should now appear as follows:

Table2a:

```
Concatenate LOAD
  "Date",
  Month (Date) as "Month",
  Quarter,
  // "Week",
  "Year"
FROM [lib://AttachedFiles/Dates.xlsx]
(ooxml, embedded labels, table is Dates);
```

By commenting out *Week*, we make sure that the tables are not identical.

2. Klicken Sie auf **Daten laden**.

3. Öffnen Sie die **Datenmodellansicht**.

Now you can see that *Table2a* has not been created.

4. Click *Table2* in the data model viewer, and then click **Preview**.

The table has the fields *Date*, *Month*, *Quarter*, *Week*, and *Year*. The field *Week* is still showing, because it was loaded from *Table2*.

5. Click *Week* in *Table2*. The preview shows that the number of non-null values for the field is 628. However, if you click any of the other fields, you see that number of non-null values is 1256. *Week* was loaded just once, from *Table2*. The number of values, or records, is the sum of the number of records in *Table2* and *Table2a*.

8.3 Preventing concatenation

If the field names and the number of fields of two or more loaded tables are exactly the same, Qlik Sense will automatically concatenate the content of the different statements into one table. This can be prevented with a `NoConcatenate` statement. The table loaded with the associated `LOAD` or `SELECT` statement will then not be concatenated with the existing table.

Gehen Sie folgendermaßen vor:

1. To be able to separate the content of the two tables completely, add `NoConcatenate` to the `LOAD` statement in *Table2a*, and rename the fields so that Qlik Sense does not create a synthetic key based on the matching fields. Uncomment *Week* in *Table2* so that the two tables have the same fields.

Your script should now appear as follows:

Table2:

```
LOAD
  "Date",
  Month (Date) as "Month",
  Quarter,
  "Week",
  "Year"
FROM [lib://AttachedFiles/Dates.xlsx]
(ooxml, embedded labels, table is Dates);
```

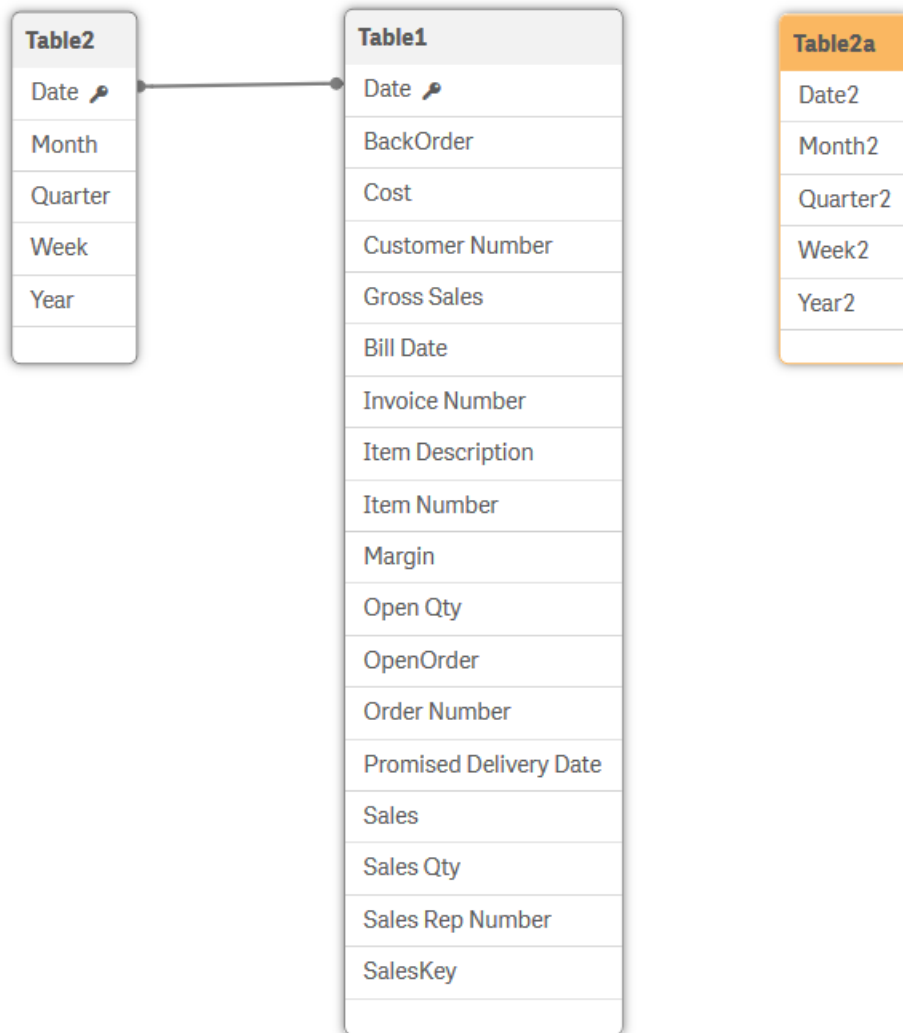
Table2a:

```
NoConcatenate LOAD
  "Date" as "Date2",
  Month (Date) as "Month2",
  Quarter as "Quarter2",
  "Week" as "Week2",
  "Year" as "Year2"
FROM [lib://AttachedFiles/Dates.xlsx]
(ooxml, embedded labels, table is Dates);
```

2. Click **Load data**.
3. Öffnen Sie die **Datenmodellansicht**.

Now you can see that the two tables are completely separated.

Data model viewer showing Table2 and Table 2a



4. Now that we have finished demonstrating concatenation, we no longer need *Table2a*. Delete all the rows in the LOAD statement for *Table2a*, and then click **Load data**.

9 Zirkelbezüge

Sind Tabellen so miteinander verknüpft, dass Werte auf mehr als einem Weg miteinander verknüpft sind, spricht man von Zirkelbezügen oder auch zirkulären Verknüpfungen ("Loops") in der Datenstruktur.

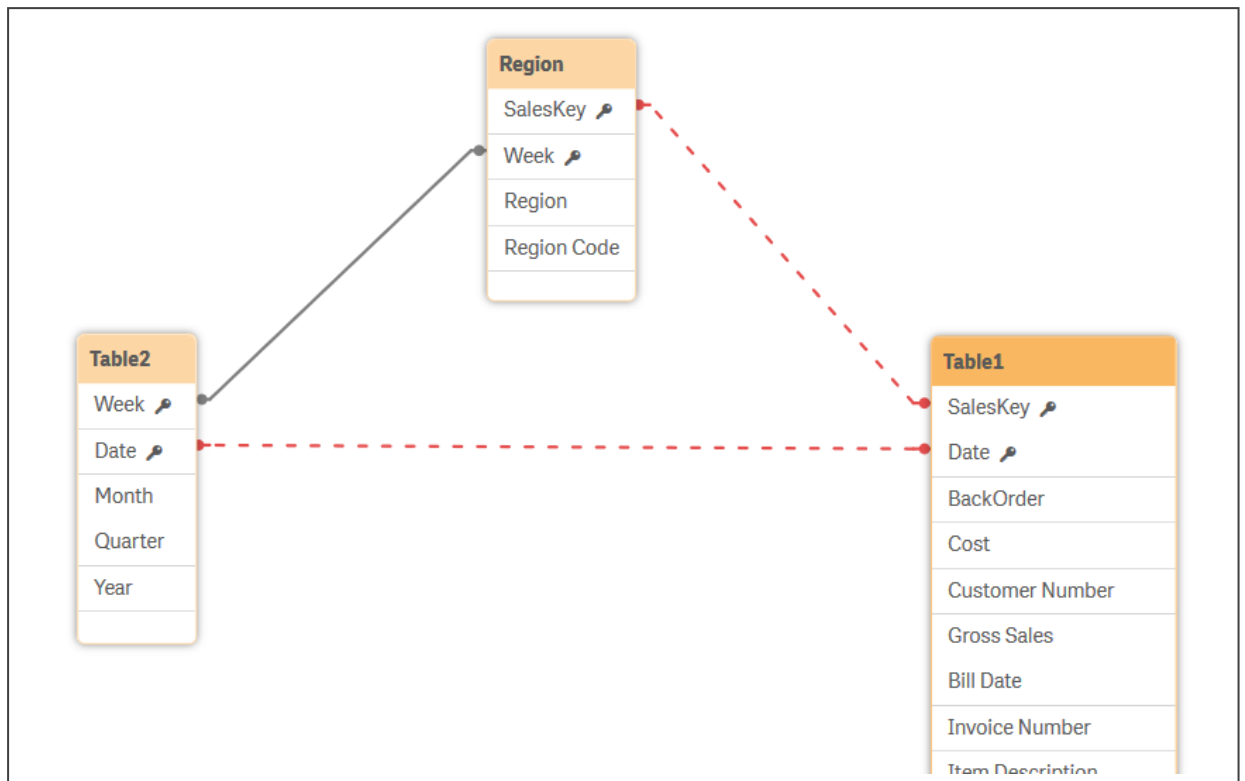
Gehen Sie folgendermaßen vor:

1. Öffnen Sie den **Dateneditor** in der *Scripting Tutorial* App.
2. Klicken Sie auf **+**, um einen neuen Skriptabschnitt hinzuzufügen.
3. Nennen Sie den Abschnitt *Region*.
4. Klicken Sie im rechten Menü unter **AttachedFiles** auf **Daten auswählen**.
5. Laden Sie *Region.txt* hoch und wählen Sie die Datei aus. Ein Datenvorschaufenster wird geöffnet.
6. Wählen Sie alle Felder aus und achten Sie darauf, dass unter **Feldnamen** der Eintrag **Eingebettete Feldnamen** ausgewählt ist, damit die Namen der Tabellenfelder beim Laden der Daten eingeschlossen sind.
7. Klicken Sie auf **Skript einfügen**.
8. Klicken Sie auf **Daten laden**.

Dieses Mal sieht es so aus, als ob beim Laden der Daten etwas schief gelaufen ist. Ein Zirkelbezug wurde erstellt. Im Fenster **Datenladefortschritt** wird eine Fehlermeldung angezeigt, die darauf hinweist, dass während des Ladevorgangs ein Zirkelbezug gefunden wurde. Der Ladevorgang wird jedoch abgeschlossen und die App wird gespeichert.

9. Öffnen Sie die **Datenmodellansicht**.
Sie können die Tabellen durch Ziehen so neu organisieren, dass die Verbindungen zwischen den Tabellen leicht sichtbar sind.

Datenmodellansicht mit Zirkelbezügen



Die rot gepunkteten Linien weisen darauf hin, dass ein Zirkelbezug erstellt wurde. Dies sollte vermieden werden, da dadurch Widersprüche bei der Interpretation der Daten entstehen können.

9.1 Auflösen von Zirkelbezügen

Um verstehen zu können, was den Zirkelbezug verursacht hat, sehen wir uns die Tabellen im **Datenmodell** genauer an.

Wenn Sie *Table1* und *Table2* im obigen Screenshot betrachten, sehen Sie, dass beiden das Feld *Date* gemein ist. Sie werden auch feststellen, dass *Table1* und *Region* das Feld *SalesKey* gemein ist. Außerdem haben *Table2* und *Region* das Feld *Week* gemeinsam. Das bedeutet, dass ein Zirkelbezug erstellt wurde. Da dieser später bei der Datenanalyse zu Problemen führen kann, werden wir ihn auflösen.

Am einfachsten lässt sich das Problem durch Umbenennen oder Entfernen eines Felds auflösen. In unserem konkreten Fall wurden Daten geladen, die für unsere App nicht erforderlich sind, weshalb wir sie entfernen können.

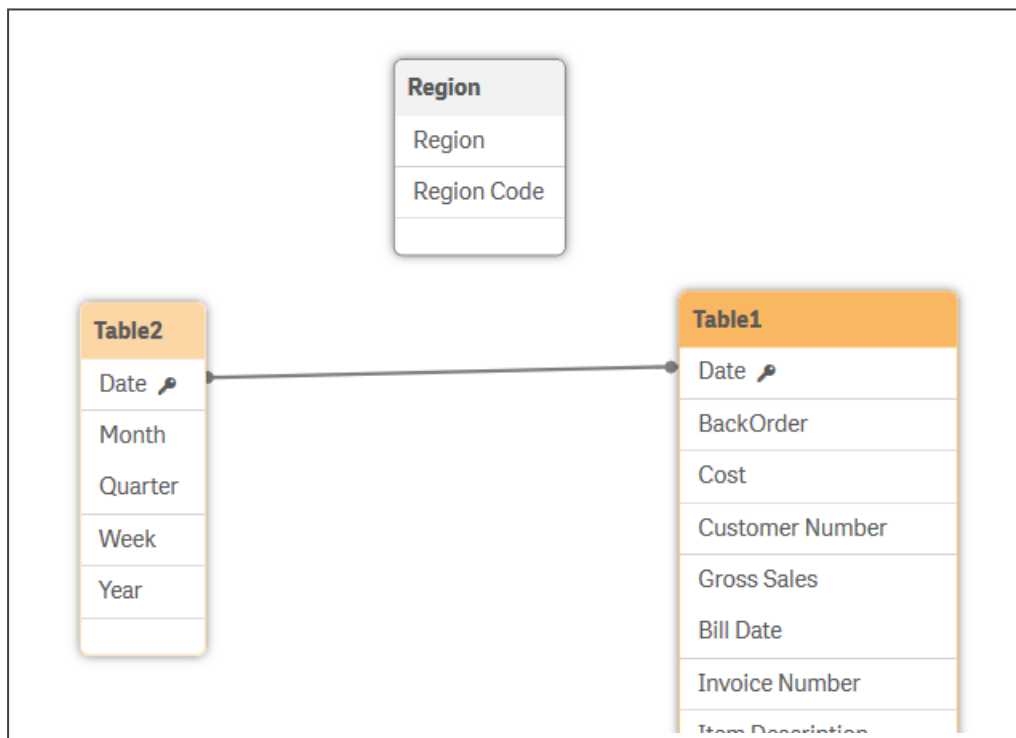
Gehen Sie folgendermaßen vor:

1. Öffnen Sie den **Dateneditor**.
2. Klicken Sie auf den Abschnitt *Region* und löschen Sie die folgenden beiden Zeilen in der *LOAD*-Anweisung:
"week",
saleskey
3. Entfernen Sie unbedingt auch das Komma nach "*Region Code*".
Ihr Skript sieht nun wie folgt aus:

```
LOAD
    Region,
    "Region Code"
FROM [lib://AttachedFiles/Region.txt]
(txt, codepage is 28591, embedded labels, delimiter is '\t', msq);
```

4. Klicken Sie auf **Daten laden**.
5. Öffnen Sie die **Datenmodellansicht**.
Die unerwünschten Bezüge zu *Region* wurden entfernt.

In der Datenmodellansicht ist zu sehen, dass die Zirkelbezüge entfernt wurden



10 Synthetische Schlüssel

Haben zwei interne Tabellen mehr als ein Feld gemeinsam, werden sie automatisch über beide Felder verknüpft, und es kann zu Fehlinterpretationen kommen. Um dieses Problem zu umgehen, generiert Qlik Sense synthetische Schlüssel. Synthetische Schlüssel enthalten alle möglichen Kombinationen der Werte zweier oder mehrerer Schlüsselfelder.

Eine höhere Anzahl zusammengesetzter Schlüssel kann den Arbeitsspeicherverbrauch erhöhen und die Leistung beeinträchtigen. Auch Datenmengen, Tabellenstruktur und andere Faktoren können eine Rolle spielen. Immer wenn es mehrere synthetische Schlüssel gibt, die voneinander abhängen, ist es eine gute Vorgehensweise, diese zu entfernen.

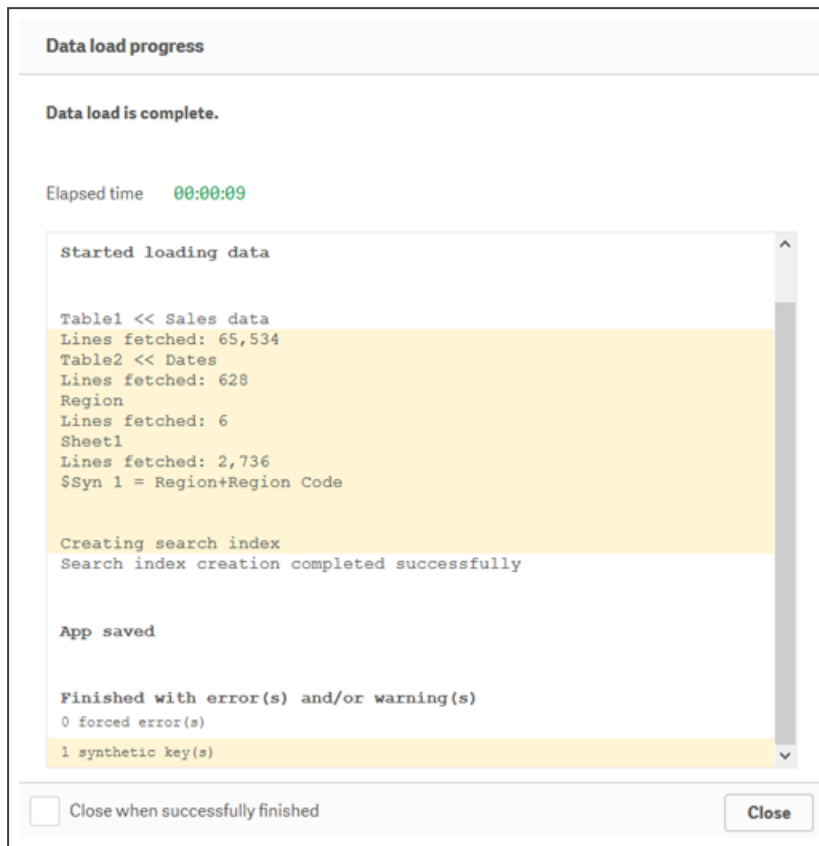
Jetzt ist es an der Zeit, den letzten Datensatz zu laden.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie den **Dateneditor** in der *Scripting Tutorial* App.
2. Klicken Sie auf **+**, um einen neuen Skriptabschnitt hinzuzufügen.
3. Nennen Sie den Abschnitt *Customers*.
4. Klicken Sie im rechten Menü unter **AttachedFiles** auf **Daten auswählen**.
5. Laden Sie *Customers.xlsx* hoch und wählen Sie die Datei aus. Das Datenvorschaufenster wird geöffnet.
6. Wählen Sie *Sheet1* aus.
7. Klicken Sie auf **Skript einfügen**.
8. Klicken Sie auf **Daten laden**.

Sie können jetzt im Fenster mit dem Datenladefortschritt sehen, dass ein synthetischer Schlüssel erstellt wurde.

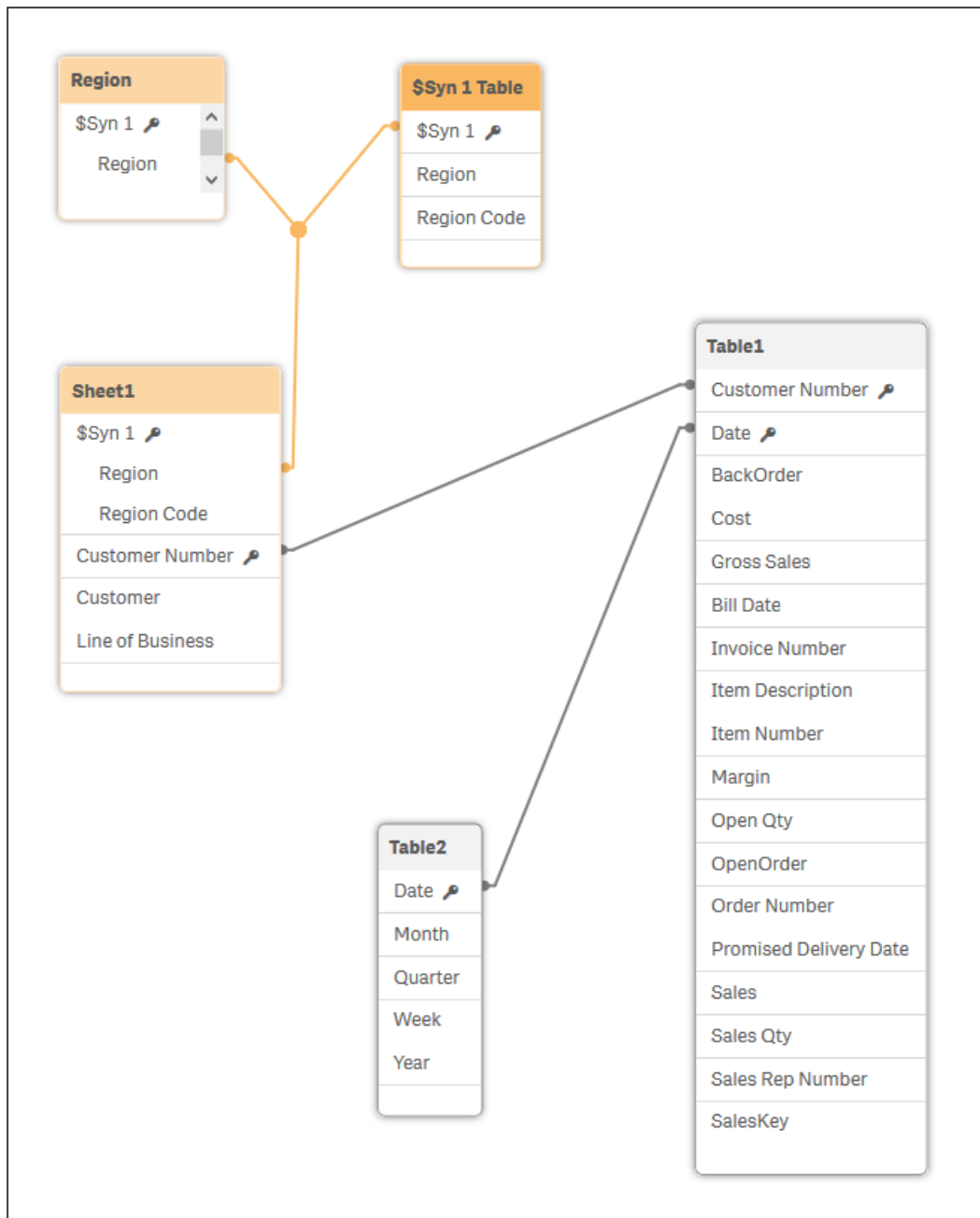
Fenster für Datenladefortschritt mit Warnung synthetischer Schlüssel



9. Öffnen Sie die **Datenmodellansicht**.

Aus der Erstellung einer neuen Tabelle *\$Syn 1 Table* ist ersichtlich, dass ein synthetischer Schlüssel erstellt wurde. Er enthält alle Felder, *Region* und *Region code*, welche die verbundenen Tabellen *Sheet1* und *Region* gemeinsam haben. In diesem Fall gestalten sich die Verbindungen dadurch etwas verwirrend und irreführend, was zu vermeiden ist.

Datenmodellansicht mit synthetischem Schlüssel



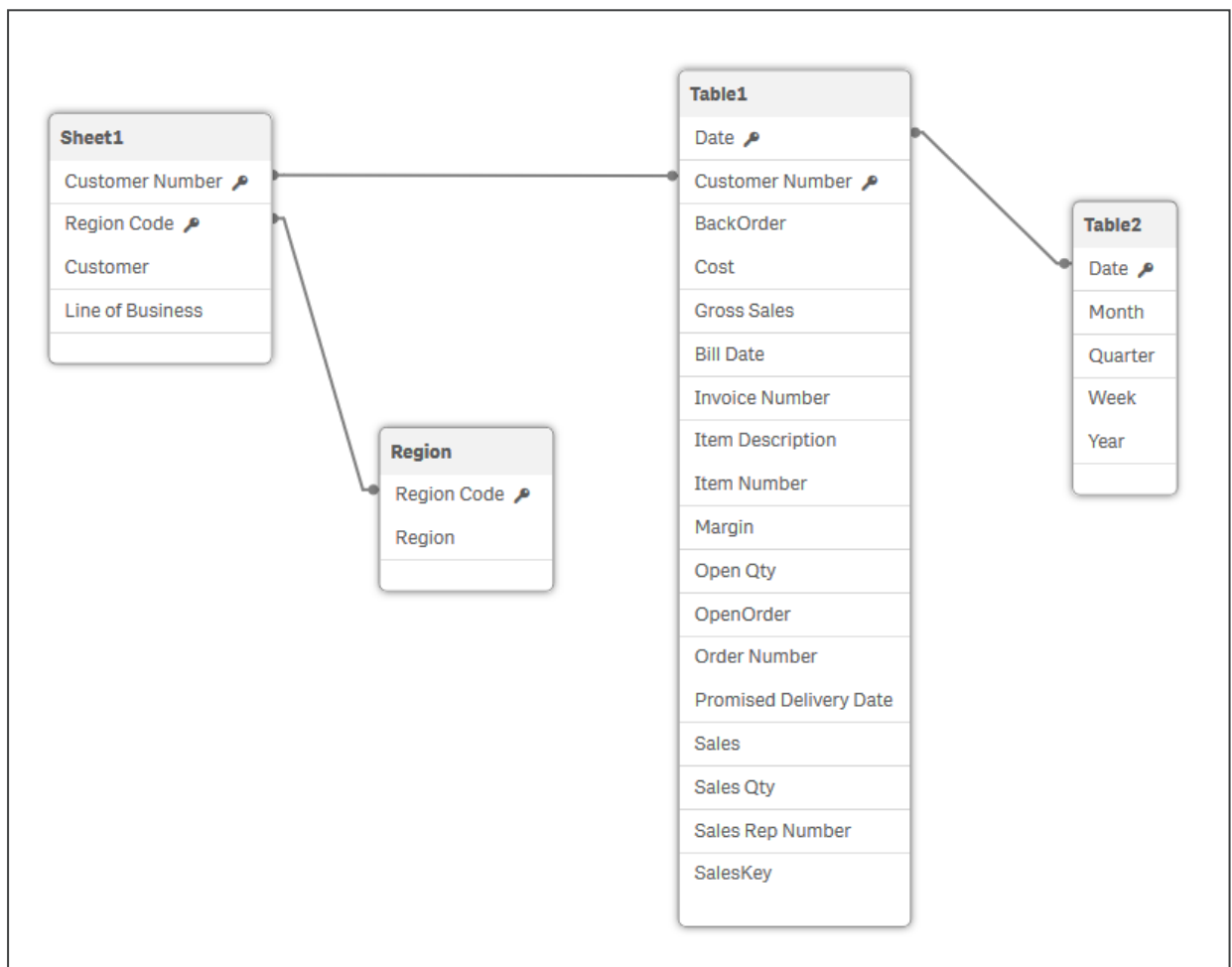
10.1 Auflösen synthetischer Schlüssel

Am einfachsten kann synthetischen Schlüsseln durch das Umbenennen von einem oder mehreren Feldern in der Tabelle entgegengewirkt werden. Dies kann beim Laden der Daten erfolgen. Jetzt werden die Schritte zur Entfernung eines synthetischen Schlüssels durchgegangen.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie den **Dateneditor**.
2. Klicken Sie auf den Abschnitt *Customers* und löschen Sie die folgende Zeile in der **LOAD**-Anweisung:
Region,
3. Klicken Sie auf **Daten laden**.
4. Öffnen Sie die **Datenmodellansicht**.
Der synthetische Schlüssel wurde entfernt.

Die Datenmodellansicht zeigt, dass der synthetische Schlüssel entfernt wurde



11 Verwenden der Daten in einer App

Zum Abschluss dieses Tutorials verwenden Sie Ihre geladenen Daten in einer Visualisierung in Ihrer App.

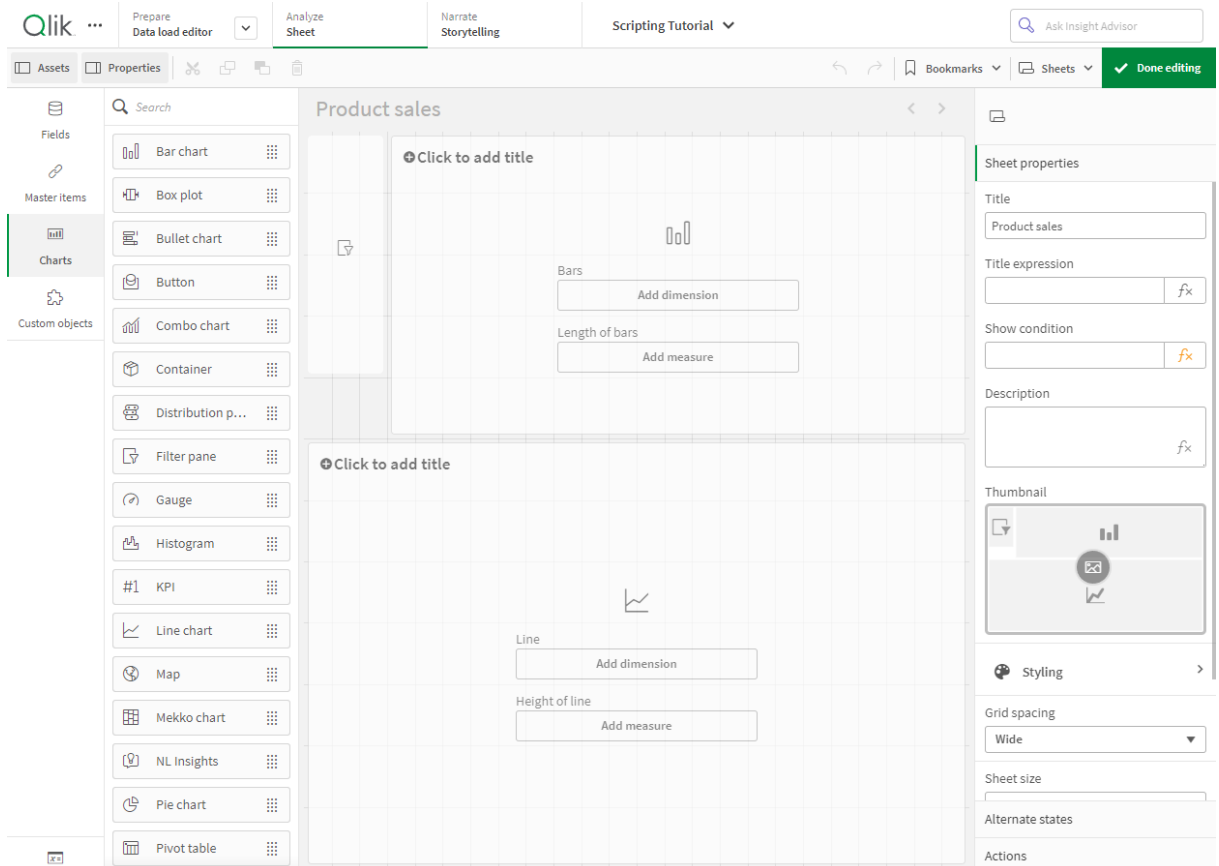
11.1 Hinzufügen eines Diagramms

Nachdem Sie Ihre Daten geladen haben, können Sie Diagramme mit den Daten erstellen. Diagramme werden oft auch als Visualisierungen bezeichnet. Ein Diagramm ist jedoch erst vollständig, nachdem Sie die erforderlichen Dimensionen und Kennzahlen hinzugefügt haben. Fügen Sie zunächst die Diagramme hinzu und fahren Sie dann mit Dimensionen und Kennzahlen fort.

Gehen Sie folgendermaßen vor:

1. Erstellen Sie ein neues Arbeitsblatt in Ihrer *Scripting Tutorial* App.
2. Ziehen Sie aus der Registerkarte **Diagramme** ein Filterfenster auf das Arbeitsblatt und passen Sie die Größe mithilfe der Griffe so an, dass es 3 Zellen breit und 4 Zellen hoch ist. Platzieren Sie es links oben im Arbeitsblatt.
3. Ziehen Sie ein Balkendiagramm nach rechts oben. Machen Sie es 5 Zellen hoch und so breit, dass es bis zur Seite des Arbeitsblatts reicht.
4. Ziehen Sie ein Liniendiagramm auf den verbleibenden Platz.
Die Symbole auf dem Arbeitsblatt zeigen, welche Art von Diagramm Sie hinzugefügt haben. Jetzt können Sie zu Ihren Diagrammen Dimensionen und Kennzahlen hinzufügen und als Visualisierungen abschließen.

Qlik Sense Arbeitsblatt mit leeren Diagrammen




11.2 Hinzufügen von Dimensionen und Kennzahlen

Als nächsten Schritt fügen wir Dimensionen und Kennzahlen hinzu. Fügen Sie zunächst Zeitdimensionen zum Filterfenster oben links hinzu. Mit einem Filterfenster können Sie Platz sparen. Sie brauchen keine eigenen Filterfenster für *Year*, *Quarter*, *Month* und *Week*, sondern nur ein einziges Filterfenster.

Erstellen und Hinzufügen von Dimensionen

Gehen Sie folgendermaßen vor:

1. Klicken Sie im Extras-Fenster links auf  **Felder**. Hier sind alle Felder in allen Tabellen aufgeführt, die Sie im Dateneditor geladen haben.
2. Scrollen Sie zum Ende der Liste und klicken Sie auf das Feld *Year*. Ziehen Sie es in die Mitte des Filterfensters oben links.
3. Fügen Sie auf dieselbe Art und Weise *Quarter*, *Month* und *Week* zum Filterfenster hinzu. Sie haben jetzt ein Filterfenster mit vier Dimensionen erstellt: *Year*, *Quarter*, *Month* und *Week*.

Erstellen und Hinzufügen von Kennzahlen

Für die meisten Visualisierungen sind sowohl Dimensionen als auch Kennzahlen erforderlich. Eine Kennzahl ist das Ergebnis einer Aggregierungsformel, die in vielen Fällen eine häufig verwendete Funktion wie **Sum**, **Max**, **Min**, **Avg** (Durchschnitt) oder **Count** ist.

Im Balkendiagramm stellen Sie die Umsätze nach Region dar.

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Felder**.
2. Klicken Sie auf das Feld *Region* und ziehen Sie es in die Mitte des Bereichs des Balkendiagramms.
3. Klicken Sie auf „**Region**“ **hinzufügen**.
4. Klicken Sie auf das Feld *Sales* und ziehen Sie es in die Mitte des Bereichs des Balkendiagramms.
5. Klicken Sie auf **Als Kennzahl hinzufügen > Sum(Sales)**.
6. Klicken Sie im Eigenschaftsfenster auf der rechten Seite auf **Darstellung** und dann auf **Präsentation**. Wählen Sie **Horizontal** aus.
Die Balken werden jetzt horizontal angezeigt.
7. Klicken Sie im Eigenschaftsfenster auf der rechten Seite auf **Sortieren**.
Die Sortierreihenfolge wird angezeigt.
8. Ziehen Sie *Sum([Sales])* über *Region*, damit die Dimensionen nach *Sum([Sales])* (Kennzahlwert) und nicht nach *Region* (Dimensionswert, alphabetisch) sortiert werden.
Das Balkendiagramm ist vollständig und zeigt die Umsatzergebnisse für die verschiedenen Regionen. Dies ist ein einfaches Balkendiagramm. Im Eigenschaftsfenster (rechts) gibt es viele Optionen, um die Gestaltung aufzuwerten. Um nur eine der Möglichkeiten zu zeigen: Sie können den Titelbereich für mehr als nur einen Titel verwenden.
9. Kopieren Sie folgenden String in das Titelfeld des Balkendiagramms:
`= 'Total sales: $' & Round(Sum(Sales)/1000000, 0.1) & 'M'`
10. Drücken Sie die Eingabetaste.

Die letzte Visualisierung auf diesem Arbeitsblatt ist ein Liniendiagramm.

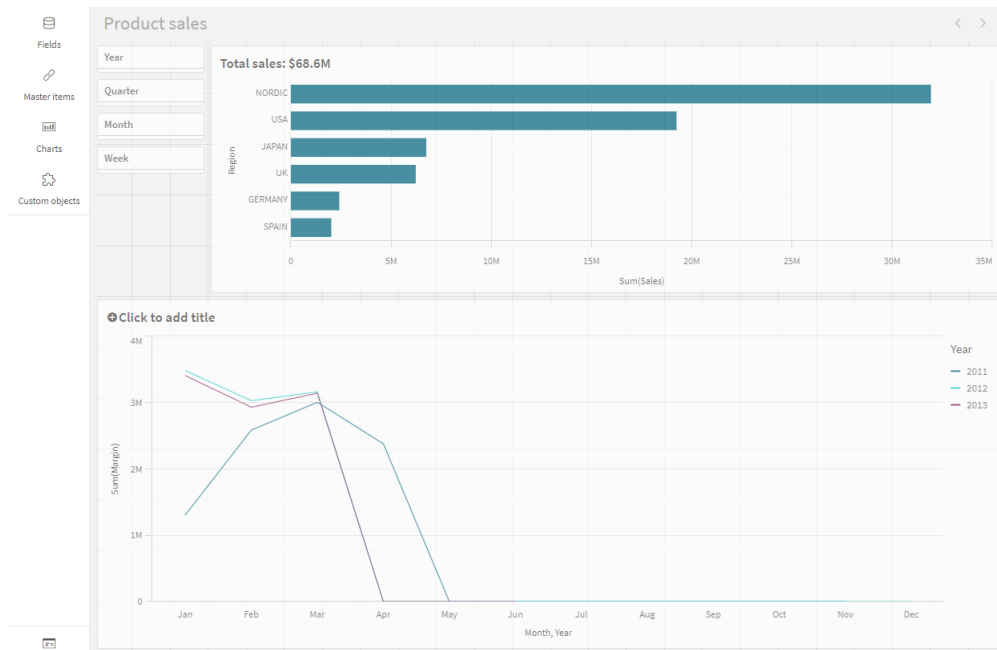
Gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Felder**.
2. Klicken Sie auf das Feld *Month* und ziehen Sie es in die Mitte des Bereichs des Liniendiagramms.
3. Klicken Sie auf „**Monat**“ **hinzufügen**.
4. Klicken Sie auf das Feld *Year* und ziehen Sie es in die Mitte des Bereichs des Liniendiagramms.
5. Klicken Sie auf „**Jahr**“ **hinzufügen**.

11 Verwenden der Daten in einer App

6. Klicken Sie auf das Feld *Margin* und ziehen Sie es in die Mitte des Bereichs des Liniendiagramms.
7. Klicken Sie auf **Als Kennzahl hinzufügen > Sum(Margin)**.
8. Geben Sie den Titel *Profit margin* oben am Liniendiagramm ein.

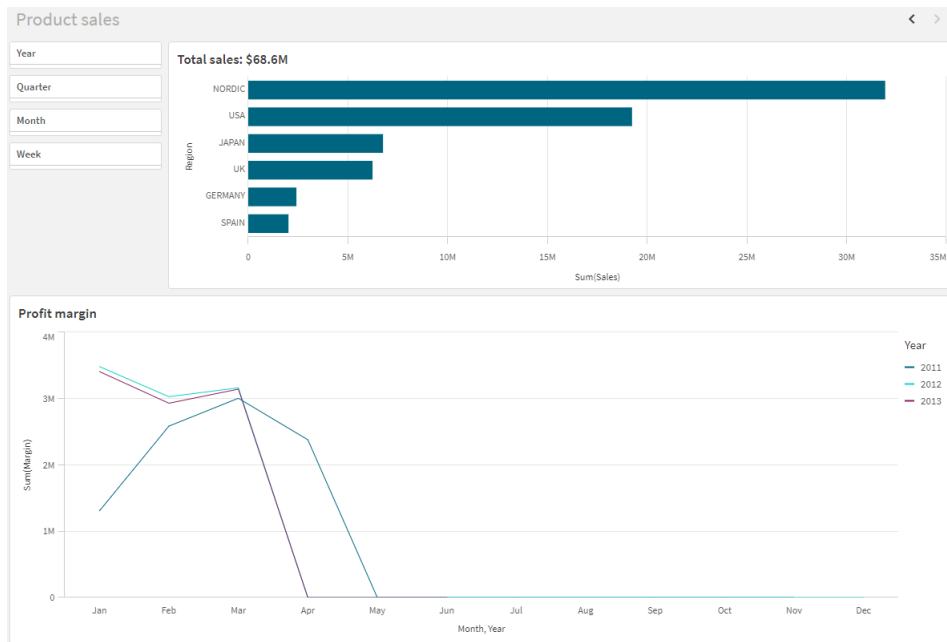
Diagramme mit Daten



9. Beenden Sie die Bearbeitung des Arbeitsblatts.
Das Arbeitsblatt ist jetzt vollständig und Sie können damit beginnen, sich durch die Daten zu klicken und mit dem Arbeitsblattinhalt zu interagieren.

11 Verwenden der Daten in einer App

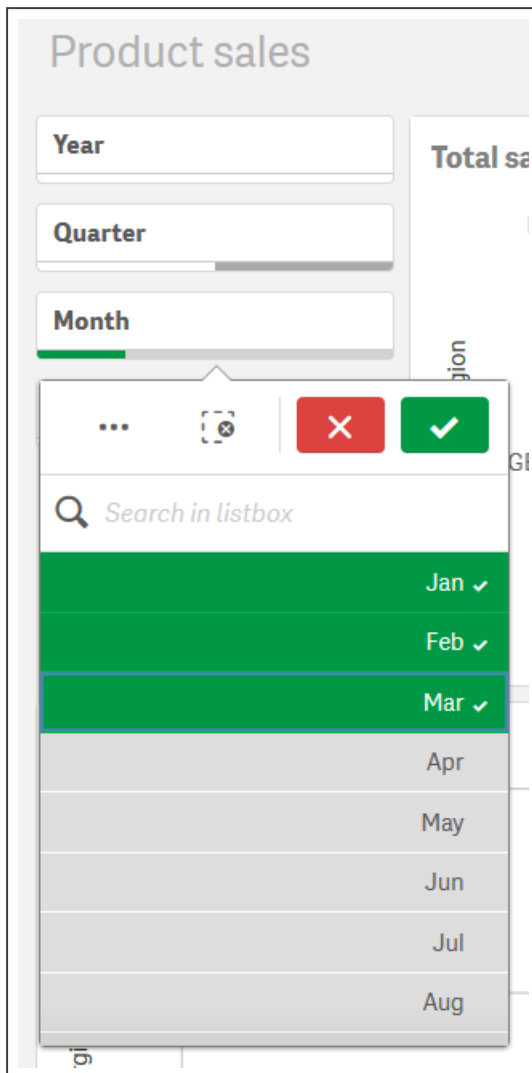
Fertiges Arbeitsblatt



Da in der ursprünglichen Datei *Sales.xlsx* nur wenige monatliche Umsatzdaten vorhanden sind, stehen im Diagramm nach Ende März der einzelnen Jahre nur wenige Daten zur Verfügung. Sie können Auswahlen im Filterfenster treffen, damit nur die ersten drei Monate der einzelnen Jahre verglichen werden.

10. Klicken Sie in der Filterbox auf das Feld *Month* und wählen Sie *Jan*, *Feb* und *Mar* aus.

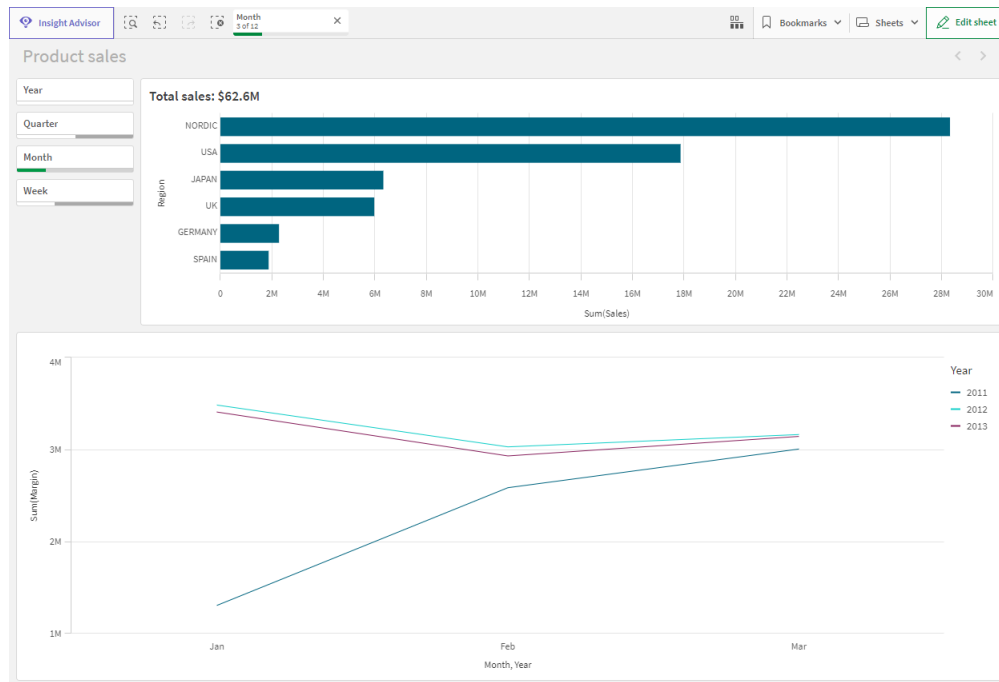
Filterbox



11. Schließen Sie das Filterfenster. Das Diagramm *Profit margin* zeigt die Daten für die ersten drei Monate der einzelnen Jahre.

11 Verwenden der Daten in einer App

Entsprechend den Auswahlen aktualisiertes Diagramm Profit margin



11.3 Vielen Dank!

Sie haben dieses Tutorial jetzt abgeschlossen und hoffentlich grundlegende Kenntnisse zur Verwendung von Skripten in Qlik Sense erworben. Besuchen Sie bitte unsere Website, um weitere Anregungen für Ihre Apps zu erhalten.