Qlik

Learning module: **Introduction to the Data Load Editor**

Exercise - INSTRUCTIONS

Analysis Goal: Use several data sources to explore the basic features of the **Data load editor** to extract, transform, and load data into an app for analysis.

Create a new app

- Launch Qlik Sense Desktop.
- Create a new app.
    - Name of new app: “Data Load Script Exercise”.

Prepare to load the first table

- Use the **Script editor (** ☰ **)** option with the goal of adding data from several files to the app.
- Create a new connection, using the 📁 **All files** option, directed to the folder where you extracted the example files (as instructed to the right).
    - Name the connection: “My folder”
- Create a new section **(** ⊕ **)** for the first data source, titled: “Order Details”.
- Use your new data connection to ☑ **Select data**.
    - Select the file: *Order Details.csv*.
- Include all fields from the file and **Insert script**.

Prepare to load the second table

- Create a new section for the second data source, titled: “Product List”.
- Use your data connection to select the file: *Product List.txt.*
- Change the **Field names** drop-down to recognize: **Embedded field names**.
- Include all fields from the file and **Insert script**.

Load data

- Click the **Load data** button, review the **Data load progress** dialog, and **Close**.

Example Download

The example files provided for this exercise are of several different types.

All of the example files are provided within the zip archive, available for download from the course site. Extract the files from the zip archive and place them in a convenient **folder location** in your Windows directory.

Here are the files used in this exercise assignment:

Order Details.csv
Contains order number, quantity, sales amount, and discount values.

Product List.txt
Contains a list of products, associated with a category identifier and a supplier identifier.

Supplier Data.xlsx
Contains a list of suppliers, contact names, and countries where the supplier’s main headquarters is located.

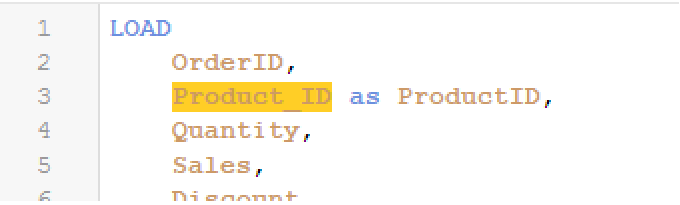
Other Tables.xlsx
Contains customer data, employee data, office locations, aggregated order data, category descriptions, and sales territory assignments.

Review the first two tables

- Use the **Data model viewer ()** in order to review the underlying data you have loaded.
- Expand the tables, and note that they are not associated (no line connects the two tables).
    - ‘Product ID’ values should associate the tables; however, in one table the field name is *Product_ID* and the other one is *ProductID.*

Edit the data load script

- Return to the **Data load editor** view.
- Use the search tool (🔍) in order to search *all sections* for the string: *Product_ID.*
- Edit the line with ‘Product_ID’ with the ‘as’ function in order to rename the field ‘ProductID’.
    - New line of script:



```
LOAD
    OrderID,
    Product_ID as ProductID,
    Quantity,
    Sales,
    Discount
```

Reload data and review changes to data model

- Click the **Load data** button, review the **Data load progress** dialog, and **Close**.
- Return to the **Data model viewer** and note that the two tables are now associated (indicated by the line which connects the two tables), due to the common field name: **ProductID**.

Prepare to load more tables

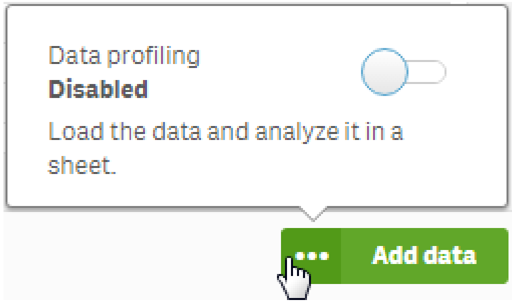
- Return to the **Data load editor**.
- Create a new section for another data source, titled: “Suppliers”.
- Use your data connection to select the file: *Supplier Data.xlsx.*
- Include all fields from the table and **Insert script**.

Load data and, again, review the data model

- Click the **Load data** button, review the **Data load progress** dialog, and **Close**.
- Click the **Output** button and compare that information with what you just reviewed in the Data load progress dialog, click the **Output** button to close that panel.
- Return to the **Data model viewer** and rearrange the layout to see the associations between these three tables.

Load data using the ‘Add data’ option

- Navigate to the **Data manager ( )** view and open in a new tab.
- Change the view to the **Tables** overview: Associations | Tables
- Use the **Add data** option to add another Excel file.
- Under **FILE LOCATIONS**, click on **My computer** to locate the Folder shortcut to the data connection you created in the Data load editor and open it.
- Select the file: *Other Tables.xlsx.*
- In the resulting dialog, check the boxes for ALL Tables listed in this data source.
- Use the triple dot button to ***Disable* Data profiling**:



Data profiling
Disabled
Load the data and analyze it in a sheet.

- Click the **Add data** button.
- Click **Close**.

Examine the Auto-generated script

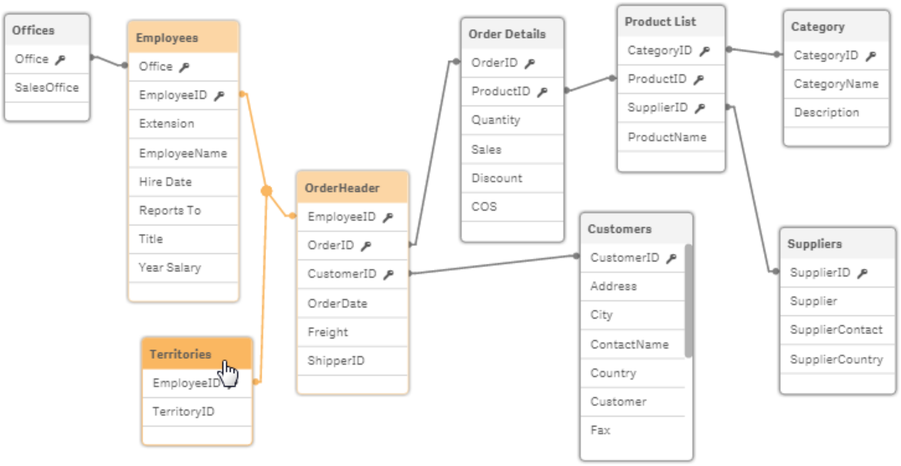
- Return to the **Data load editor** view.
- Select the **Auto-generated section**, and note that it is locked – and script cannot be edited.
- Return to the **Data manager**.
- Note that the tiles for each of the tables from the *Other Tables.xlsx* data source display **Edit (✎)** and **Delete (🗑)** icons upon mouseover – because the ‘***Add data***’ option was used. The tables loaded using the ‘***Script editor***’ option (*Order Details, Product List, & Suppliers*) do not present the edit and delete icons. Instead, they present a script ( ) icon along the bottom of the tile.

Unlock the Auto-generated script

- Return to the **Data load editor** view.
- **Unlock** the **Auto-generated section** of the script, and rename the section: “Other Tables”.
- Return to the **Data manager** view and note that Edit and Delete icons are no longer available on any tables. All table tiles now present a script ( ) icon.

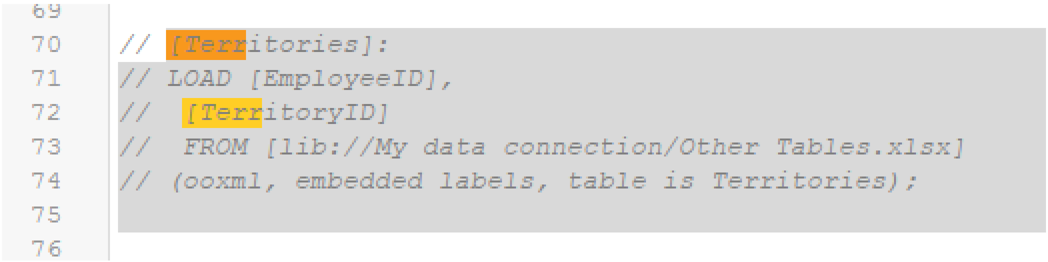
Review the data model

- Return to the **Data model viewer** and rearrange the layout to view associations between tables.
- Click on the *Territories* table and note its associations to the *Employees* table and *OrderHeader* table, indicated with orange lines.



Comment out functional lines of script

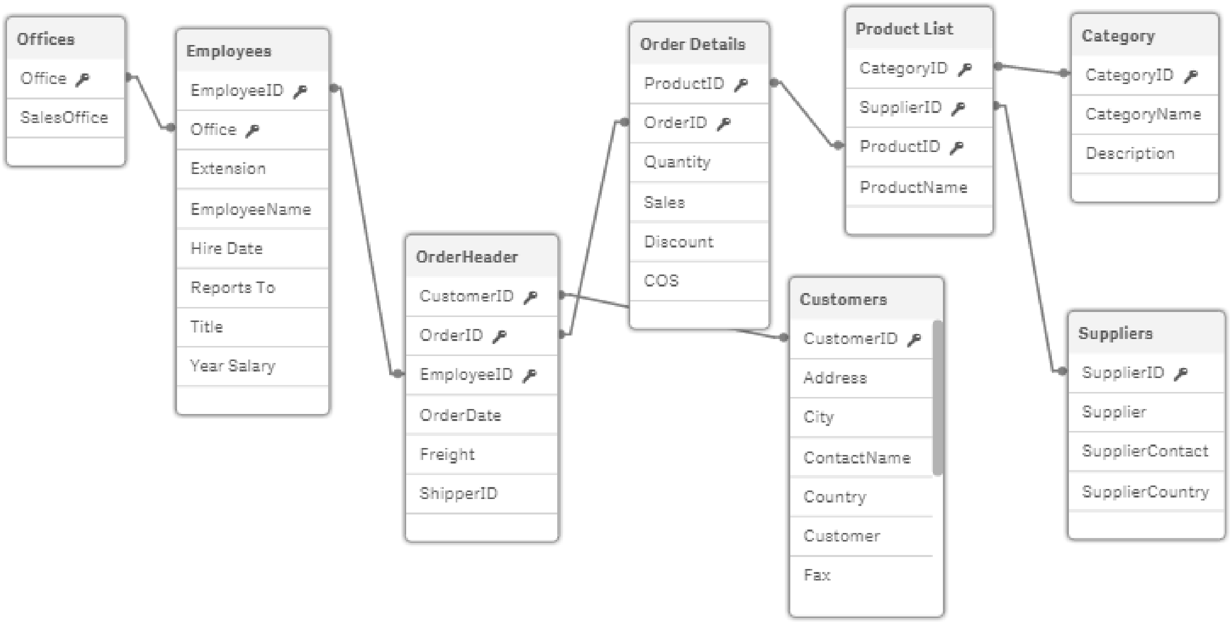
- Assume that we do not wish to include the *Territories* table in our data model, right now; however, we may wish to bring it back in at some later point in time.
- Return to the **Data load editor**.
  - Use the search tool (🔍) in order to search only the *Other Tables* section for the string: "[Terr".
- Select all lines of data load script associated with the LOAD statement which loads data from the table with the TerritoryID field.
- Use the comment tool (//) in order to change those lines of data load script to be comment lines only (in other words, no longer lines of functional data load script).



```
// [Territories]:
// LOAD [EmployeeID],
//  [TerritoryID]
//  FROM [lib://My data connection/Other Tables.xlsx]
// (ooxml, embedded labels, table is Territories);
```

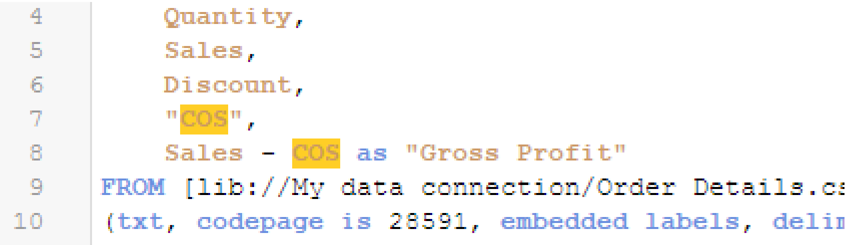
Reload data and review changes to data model

- Click the **Load data** button, review the *Data load progress* dialog, and **Close**.
- Return to the **Data model viewer**.
    - Note that the *Territories* table is no longer loaded



Calculate a new field with an expression in the load data script

- Return to the **Data load editor** view.
- Use the search tool (🔍) in order to search *all sections* for the string: “COS”.
- Enter a new line below the line which loads the COS field, or Cost of Sales, by typing a comma (**,**) to end the line and a carriage return to open a new line.
- Type the following expression, (*shown on line* 8) which is intended to calculate *Gross Profit* values:



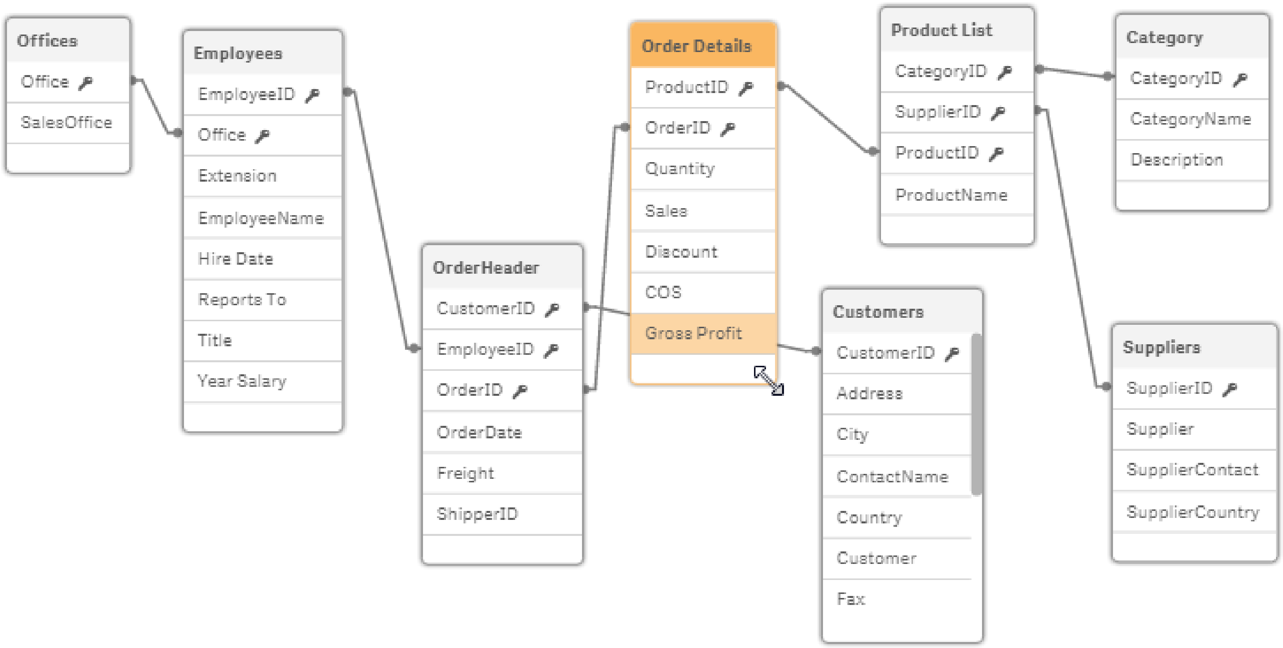
```
4       Quantity,
5       Sales,
6       Discount,
7       "COS",
8       Sales - COS as "Gross Profit"
9   FROM [lib://My data connection/Order Details.cs
10  (txt, codepage is 28591, embedded labels, delin
```

Load data

- Click the **Load data** button, review the **Data load progress** dialog, and **Close**.

Evaluate new calculated field

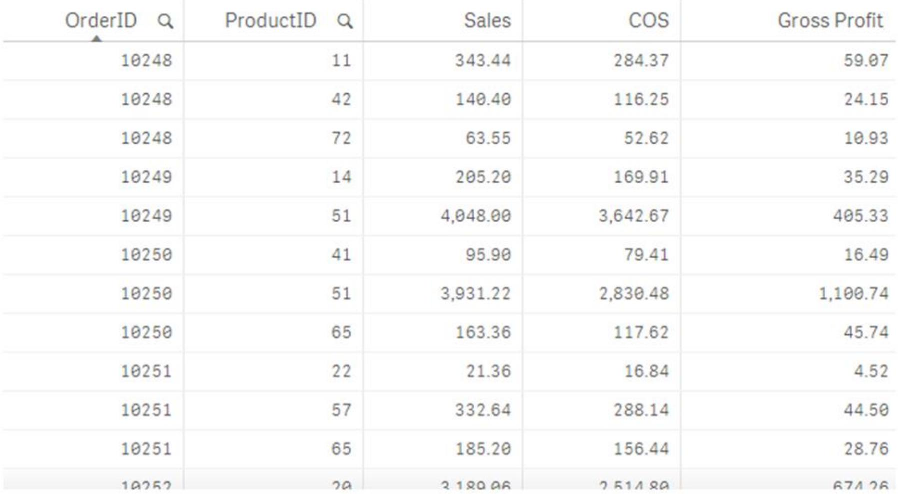
- Visit the **Data model viewer**.
  - Note the new ‘Gross Profit’ field in the *Order Details* table.



View the new calculated field in a visualization

- Navigate to the **App overview** (🏠) and launch in a new tab.
    - Open **My new sheet**, toggle the sheet into **Edit** mode, and view the **Charts** section of the assets panel.
    - Drag & drop a **Table** (▦) on the sheet.
        - Dimension: *OrderID*
        - **Add columns**
            - Dimension: *ProductID*
            - Measure: *Sales* (choose *Sum* as an aggregation)
                - Label = “Sales”
                - Number formatting = **Number** with **Simple**: **1,000.12** format
                - Totals function = **None**
            - Measure: *COS* (choose *Sum* as an aggregation)
                - Label = “COS”
                - Number formatting = **Number** with **Simple**: **1,000.12** format
                - Totals function = **None**
            - Measure: *Gross Profit* (choose *Sum* as an aggregation)
                - Label = “Gross Profit”
                - Number formatting = **Number** with **Simple**: **1,000.12** format
                - Totals function = **None**
        - Click the **Done** button in order to toggle the sheet out of Edit mode.

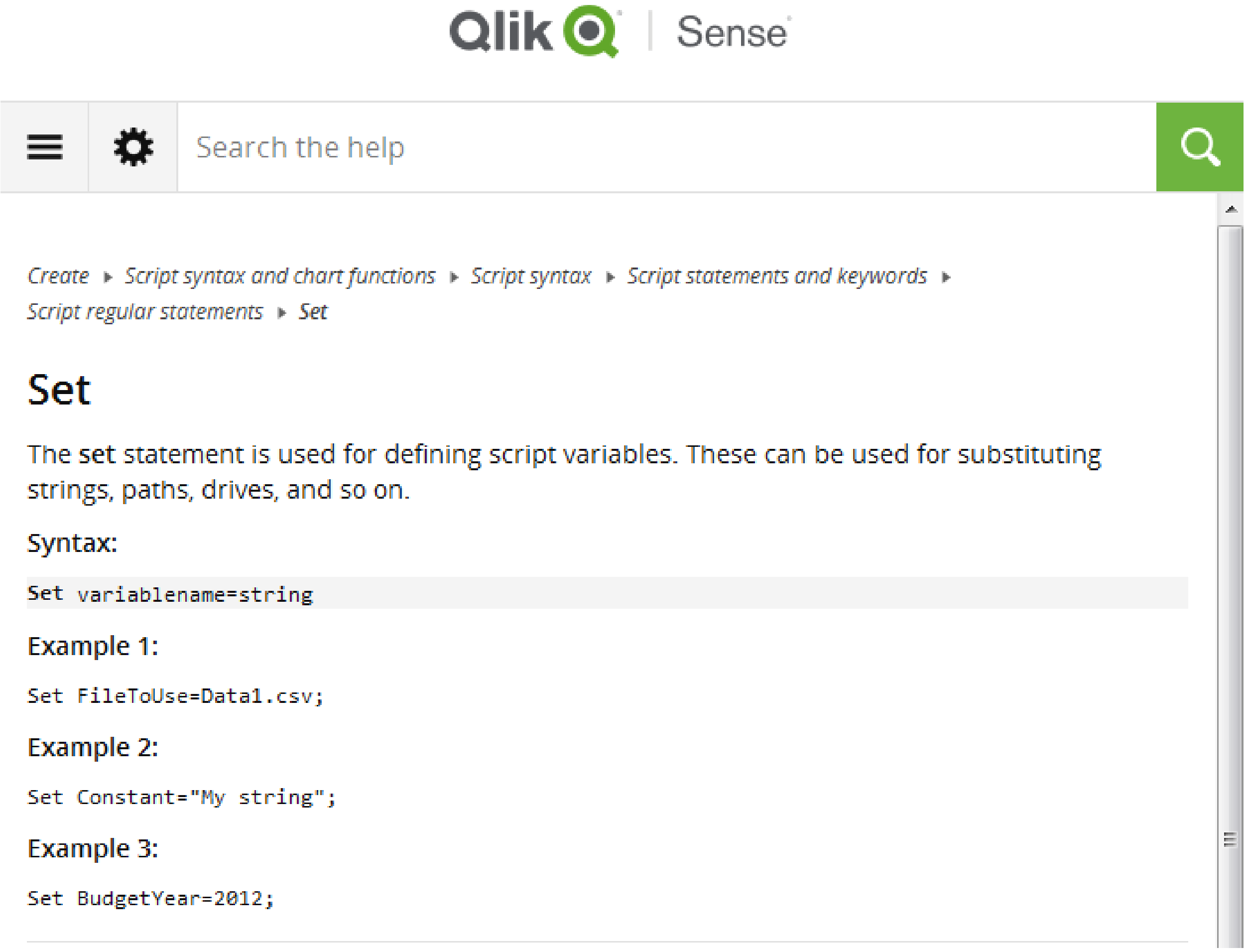
- Pull out a calculator and spot-check a few rows of data to convince yourself that the *Gross Profit* was calculated correctly (Gross Profit = Sales – COS).
- The resulting Table should look like this:



OrderID	ProductID	Sales	COS	Gross Profit
10248	11	343.44	284.37	59.07
10248	42	140.40	116.25	24.15
10248	72	63.55	52.62	10.93
10249	14	205.20	169.91	35.29
10249	51	4,048.00	3,642.67	405.33
10250	41	95.90	79.41	16.49
10250	51	3,931.22	2,830.48	1,100.74
10250	65	163.36	117.62	45.74
10251	22	21.36	16.84	4.52
10251	57	332.64	288.14	44.50
10251	65	185.20	156.44	28.76
10252	20	3,189.06	2,514.80	674.26

Recognizing Qlik Sense functions

- Return to **Data load editor**, and view the **Main** section of the data load script. Toggle the editor into **Help mode** (?) and use one of the resulting SET hyperlinks to obtain details for this function from the online help …



Create ▸ Script syntax and chart functions ▸ Script syntax ▸ Script statements and keywords ▸ Script regular statements ▸ Set

Set

The **set** statement is used for defining script variables. These can be used for substituting strings, paths, drives, and so on.

Syntax:

```
Set variablename=string
```

Example 1:

```
Set FileToUse=Data1.csv;
```

Example 2:

```
Set Constant="My string";
```

Example 3:

```
Set BudgetYear=2012;
```

Other opportunities to learn more

- Each of the apps you have created as a part of your efforts in the continuous classroom, or toward your own analysis efforts, has resulted in a data load script. Open any of these apps and take a look at the **Data load editor (☰)** view in order to relate the existing scripts to the data models which they generate.