Learning module: **Associating Tables and Transforming Data**

Exercise - INSTRUCTIONS

Using the example download files on the right, along with the creation of a new application in Qlik Sense Desktop, you will practice making table associations within the script. In addition, you will manipulate the data with hand-written expressions and utilize stacked preceding load blocks to compartmentalize your data load script.

Step 1: Create a new app

- Launch Qlik Sense Desktop.
- Create a new app.
    - Name of new app: "Associating Tables and Transformation Practice".

Step 2: Prepare the data load script

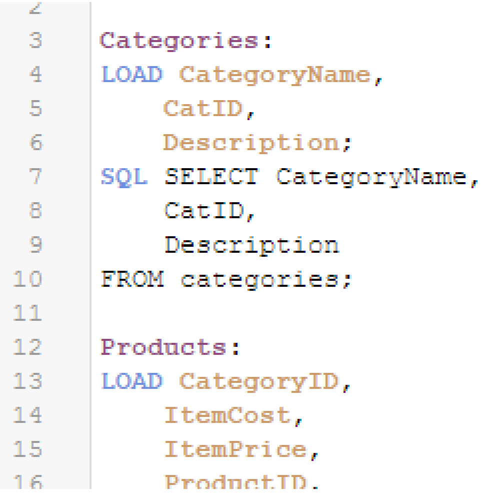
- Use the **Script editor** () option to launch the data load editor view.
- Create a new script section called "Associating".

Step 3: Create the connection

- Create a new **OLE DB** database connection.
- Use the provider: **Microsoft Jet 4.0 OLE DB Provider(32-bit)**.
- Enter the full drive letter and path where you stored the example database, along with the filename, e.g., `C:\Data\SOP.mdb`.
- Click the **Test connection** button.
- Name the connection "SOP".

Step 4: Form the data load script

- Use the **Select data** () option for your new connection to view tables in the database.
- **Insert script** for the *categories* and *products* tables.
- Name the tables appropriately (Categories: , Products:).



```
Categories:
LOAD CategoryName,
    CatID,
    Description;
SQL SELECT CategoryName,
    CatID,
    Description
FROM categories;

Products:
LOAD CategoryID,
    ItemCost,
    ItemPrice,
    ProductID,
```

Example Download

The example file provided for this exercise is a single Access **.mdb** format database.

It is located in the zip archive, available for download from the course site. Extract this file from the zip archive and place it in a convenient location on your local hard drive.

It is recommended you place the file into a new folder off the root of your drive, as you will be required to enter the entire drive and folder path during the course of this exercise.

Here is the file used in this exercise assignment:

SOP.mdb
Contains numerous tables centered around sales order processing of a fictitious company.

Step 5: Extract the data

- Execute the script by clicking the **Load data** button.
- **Are there any errors in the progress?** (Answer = No)

Step 6: Review the data model

- Check the **Data Model Viewer** (⧉) to ensure the model is correct.
- **Do the *products* and *categories* table associate with one another?** (Answer = No)
- **Can you identify the probable cause?** (Answer = *CategoryID ≠ CatID*)

Step 7: Rectify the problem

- In the data load script, alias the *CatID* field `AS` *CategoryID.*
- Click the **Load data** button.
- Check the **Data model viewer**.
- **Do the tables now associate correctly?** (Answer = Yes)

Step 8: Create a new calculated field

- Within the script, create a new field in the *Products* table.
  - Calculate the profit for each item, using the following expression construct: `ItemPrice – ItemCost`
- Alias the new field as *ItemProfit.*

Step 9: Reload the data and check the new field

- Execute the script by clicking the **Load data** button.
- Check the **Data Model Viewer** to ensure the *Products* table contains the new field.
- Check the **Preview panel** (⬓) and ensure the new field contains expected values [HINT: *ItemProfit* should always be a positive value with multiple decimal places].

Step 10: Re-use the new field in stacked load block

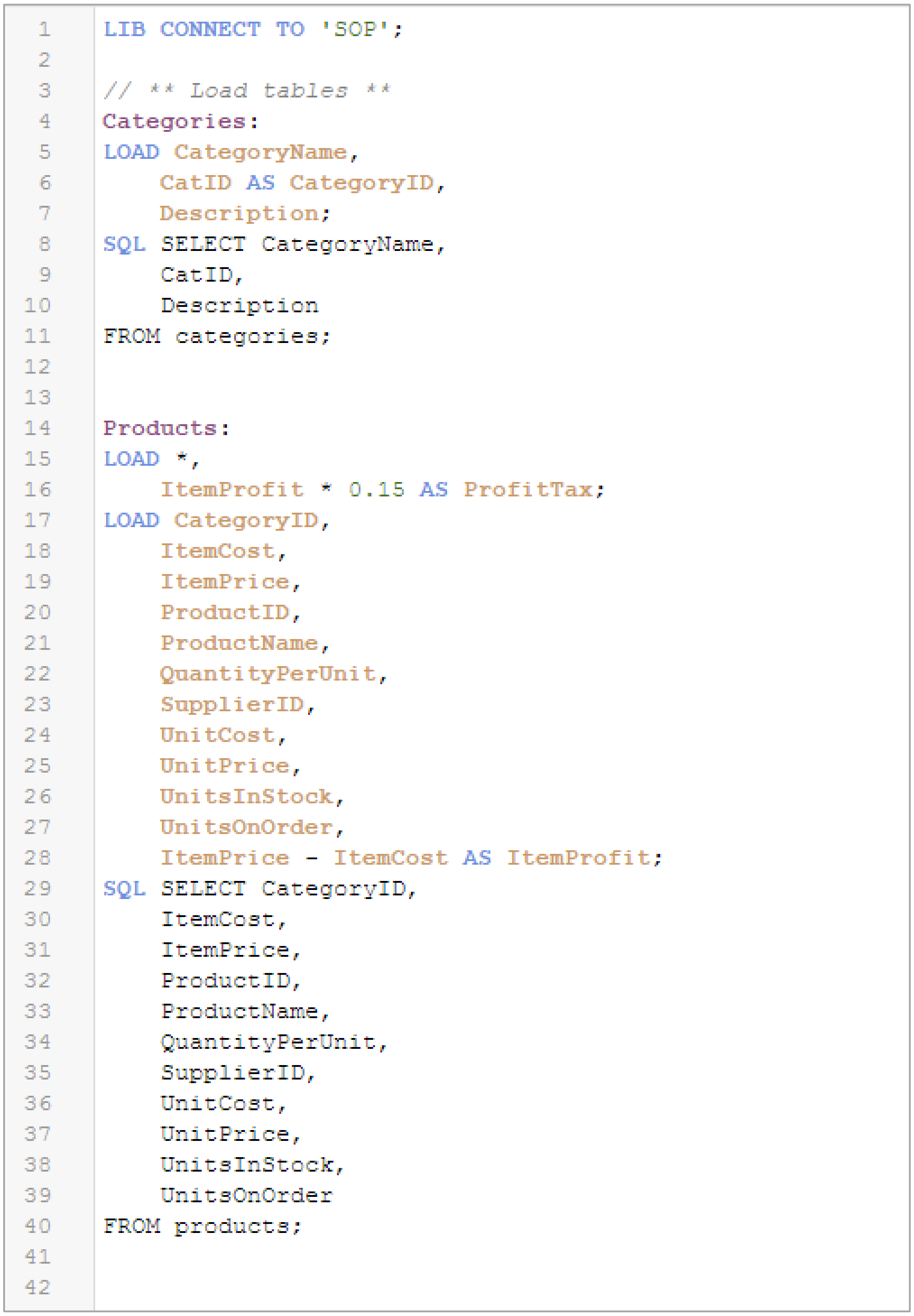
- Create another LOAD block above the existing one in the *Products* table.
- Ensure all existing fields are retained.
  [HINT: Use the asterisk (*****) and do not forget a comma.]
- Create a new field in this load block to calculate *ItemProfit * 0.15*. Alias it AS *ProfitTax*.
  [HINT: Do not forget to end the block with a semi-colon(**;**).]

Step 11: Verify the new field in the model

- Execute the script by clicking the **Load data** button.
- Check the **Data Model Viewer** to ensure the *Products* table contains the new *ProfitTax* field.
- Check the **Preview panel** and ensure the new field contains expected values.
  [HINT: The *ProfitTax* field will be at the far right side of the preview panel and may contain up to three decimal places.]

The use in this practice lab of stacking preceding load blocks is overkill for such a simplistic calculation, but it is good practice to solidify the knowledge of how you can stack load blocks in this way. Experiment yourself with different calculations and the stacking of preceding load blocks to cement the knowledge of the data being passed upwards from block to block.

The data load script you created should appear as shown below.



```
LIB CONNECT TO 'SOP';

// ** Load tables **
Categories:
LOAD CategoryName,
    CatID AS CategoryID,
    Description;
SQL SELECT CategoryName,
    CatID,
    Description
FROM categories;


Products:
LOAD *,
    ItemProfit * 0.15 AS ProfitTax;
LOAD CategoryID,
    ItemCost,
    ItemPrice,
    ProductID,
    ProductName,
    QuantityPerUnit,
    SupplierID,
    UnitCost,
    UnitPrice,
    UnitsInStock,
    UnitsOnOrder,
    ItemPrice - ItemCost AS ItemProfit;
SQL SELECT CategoryID,
    ItemCost,
    ItemPrice,
    ProductID,
    ProductName,
    QuantityPerUnit,
    SupplierID,
    UnitCost,
    UnitPrice,
    UnitsInStock,
    UnitsOnOrder
FROM products;
```