Lab 2 – Creating a live DirectQuery connection

In the following lab, you will connect to a SQL Azure Database using a DirectQuery data connection. You will then add some Visuals to the Report page.

Exercise 1 – Connecting to a SQL Azure Database

1. Open **Power BI Desktop**.

2. If the **Welcome to Power BI Desktop** window appears, click the **Already have a Power BI account? Sign in** link at the bottom. When the **Sign In** window appears, click the **X** in the top right corner to close it.

3. On the **Canvas Background**, click the **Get data from another source** □ link, select **Azure** (the category on the left), then **Azure SQL database**. Click **Connect**.

4. In the **SQL Server database** window, configure the following settings:

- Server: **ADVLTNOW.database.windows.net**
- Database (optional): **AdventureworksLT**
- Data Connectivity mode: **DirectQuery**
- Click **OK**

5. If the authentication window appears, select **Database** in the left-hand panel.

6. Unless the Instructor provides alternative credentials, use the following:

- Username: **Student**
- Password: **Pa55w.rd**

7. Click **Connect**

8. In the **Navigator** window, expand the **AdventureworksLT** database to display the list of database objects.

9. Select the **SalesLT.SalesOrderDetail** table and review the columns within the table displayed on the right.

10. To select the following **6 tables**, tick their **checkboxes**:

- **SalesLT.Address**
- **SalesLT.Customer**
- **SalesLT.Product**
- **SalesLT.ProductCategory**
- **SalesLT.SalesOrderDetail**
- **SalesLT.SalesOrderHeader**

11. Click **Load** to connect the tables to the Power BI Desktop Model. The data is NOT loaded but the connection is made to the database and the selected tables.
12. On the **File** menu, click **Save**. Name the file **QAPBID Lab02** and save it in the **C:\Coursefiles\QAPBIDESK\lab02** folder.

Exercise 2 – Create a Report (Map / Clustered Column Chart)

1. Click the **Report** Icon. In the **Fields** pane (on the far right-hand side) expand the **SalesLT Address** table. Select the **CountryRegion** field to add it to the report page.
2. In the **Fields** pane, expand the **SalesLT SalesOrderDetail** table. Select the **OrderQty** field to add to the report.
3. In the **Visualizations** pane, select a **Clustered Column Chart** to change the map visual in the report.
4. Click the report canvas (outside the visual) to deselect the visual.
5. In the **Fields** pane, expand the **SalesLT Address** table. Select the **StateProvince** field to add the column to the report.
6. In the **Fields** pane, expand the **SalesLT SalesOrderDetail** table. Select the **OrderQty** field to add it to the report. The bubbles change size dependent upon the Order Quantity for each State or Province.
7. On the **Clustered Column Chart**, click the **United States** bar. Notice how the Map visualization automatically changes to focus in on U.S. sales. The same happens in reverse if you select a bubble on the Map visualization.
8. On the **File** menu, click **Save**.
9. Close Power BI Desktop.

Result: You created a live DirectQuery connection to an Azure SQL Database and created a report based on that DirectQuery connection.

Lab 3 – Importing Data

In the following lab, you will import data from an Excel spreadsheet and a text file. You will check the relationships in the Data Model and create a Report.

Exercise 1 – Import data from an Excel spreadsheet

1. Open **Power BI Desktop**.

2. If the **Welcome to Power BI Desktop** page appears, click the **Already have a Power BI account? Sign in** link at the bottom. When the **Sign In** dialog box appears, click the **X** in the top right corner to close it.

3. Click the **Import data from Excel** button on the Canvas background.

4. Navigate to the **C:\Coursefiles\QAPBIDESK\lab03** folder. Select the **AdventureWorks Data.xlsx** spreadsheet. Click **Open**.

5. In the **Navigator** window, select **Sheet1**. In the preview, notice that the display has an export comment in the column heading. The actual column headings are in row 2.

6. In the **Navigator** window, select the **ProductCategory** table. Notice that the content displays with the correct column headings this time. This is the difference between selecting the formatted Table instead of the Excel Sheet.

7. Click each **checkbox** to select the following **4 tables**:

· **ProductCategory**

· **Products**

· **ProductSubcategory**

· **Sales**

8. Click **Load** to load the tables into the Power BI Model.

9. The **4 tables** are displayed in the **Model** view

10. On the left panel, click the **Table icon** to change to the **Table** view. Notice the 4 tables are now also displayed in the **Fields** pane (on the right).

11. On the **File** menu, click **Save**. Name the file **QAPBID Lab03**. Click **Save.**

Exercise 2 – Import a Text file

1. On the **Home** ribbon, select the **Get Data** dropdown and then **Text/CSV.**

2. In the **Open** window, select the file **CategoryManagers.txt.** Click **Open**.

3. Review the content. The column headers are **Column1** and **Column2**.

4. Click the **Transform Data** button to open the Power Query Editor.

5. On the left **Queries panel**, select the **CategoryManagers** table.

6. On the **Home** Menu Ribbon, select **Use First Row as Headers** to promote **Product Category** and **Manager Name** to be the Column heading position.

7. Click **Close and Apply** to update and save the changes to the file.

8. On the **File** menu, click **Save**.

Exercise 3 – Check the Relationships and create a Report

1. On the left panel, check that you are in the **Model** view.

2. Review the different relationships between the tables. The relationships have been automatically detected.

3. On the left panel, click the **Report** view.

4. On the **Visualizations** panel, select the **Table** Visual icon to add a new Table visual to the report canvas.

5. Select the following fields from the **Fields** pane:

- **CategoryManagers** \ Manager Name
- **CategoryManagers** \ Product Category
- **Products** \ Product Name
- **ProductSubcategory** \ Product Subcategory
- **Sales** \ Order Qty

6. Review the content of the table visual. Is the font display too small?

7. On the **Visualizations** pane, click the **Format** icon. Configure the following:

- **Values**. Set the **Text** size to **14pt**.
- **Column** headers. Set the **Text** size to 14pt.
- To improve the display, resize the **Table** visual if required.

8. On the **File** menu, click **Save**.

9. Close Power BI Desktop.

Result: You created a Data Model by importing data from an Excel spreadsheet and a text file. You used the Power Query Editor to fix the Column headers. You created a Table Report Visual.

Lab 4 – Transforming & Editing Data

In the following lab, you will import an Excel Spreadsheet. You will then edit and transform some of the column and row data by using the Power Query Editor. To finish, you will create a Matrix Report.

Exercise 1 – Import data from an Excel spreadsheet

1. Open **Power BI Desktop**.

2. If the **Welcome to Power BI Desktop** page appears, click the **Already have a Power BI account? Sign in** link at the bottom. When the **Sign In** dialog box appears, click the **X** in the top right corner to close it.

3. Click the **Import data from Excel** button on the Canvas background.

4. Navigate to the **C:\Coursefiles\QAPBIDESK\lab04** folder. Select the **ResultsByState.xlsx** spreadsheet. Click **Open**.

5. In the **Navigator** window, select the **ResultsByState** table checkbox. Review the details on the right.

6. Click the **Transform Data** button to launch the **Power Query Editor**.

7. The Table loads with Totals and NULL values in the first two columns (StateID and StateName). We need to make some changes. We'll start by removing the columns that are not required.

8. Right-click the **StateID** column and click **Remove**.

9. Right-click the **ProductID** column and click **Remove**.

10. On the **StateName** column header**,** click the **filter** button to the right. **Uncheck** the **Grand Total** and **Total** columns. Click **OK**.

11. Right-click the **StateName** column header, select **Fill** then **Down** to replace the NULL values with the StateName that precedes them.

12. On the **Home** menu tab, click the **Close and Apply** button on the ribbon to update the changes into the model.

13. On the **File** menu, click **Save**. Name the file **QAPBID Lab04**. Click **Save.**

Exercise 2 – Create a Matrix Report

1. Select the **Report** view.

2. On the **Visualizations** pane, select the **Matrix** icon to add the Matrix visual to the report. Resize and center it on the page.

3. From the **Fields** pane, expand the **ResultsByState** table. Drag the **CategoryName** field to the **Rows** bucket of the **Visualizations** pane. This adds the row headers to the matrix.
4. From the **Fields** pane, drag the **StateName** field to the **Columns** bucket of the **Visualizations** pane. This adds the column headers to the matrix.
5. From the **Fields** pane, drag the **TotalQuantity** field to the **Values** bucket of the **Visualizations** pane. This adds the values to the matrix.
6. Review the Matrix content. If necessary, click the **Format** button. **Adjust the size** of the **Column headers**, **Row headers** and **Table values**.
7. On the **File** menu, click **Save**.
8. Close Power BI Desktop.

Result: You created a Data Model by importing data from an Excel spreadsheet. You used the Power Query Editor to fix some Column headers and row entries. You created a Matrix Report.

Lab 5 – Cleansing Data

In the following lab, you will open a Power BI Report and do some data cleansing using the Power Query Editor.

Exercise 1 – Clean up some data from an Excel Spreadsheet

1. Open **Power BI Desktop**.

2. If the **Welcome to Power BI Desktop** page appears, click the **Already have a Power BI account? Sign in** link at the bottom. When the **Sign In** dialog box appears, click the **X** in the top right corner to close it.

3. Select **File**, then **Browse reports** (on the right). Navigate to the **C:\Coursefiles\QAPBIDESK\lab05** folder. Select the **QAPBID Lab 5 - Start.pbix** file. Click **Open**.

4. On the **Home** ribbon, click the **Transform Data** button to access the Power Query Editor.

5. Review the current data. The following updates need to be made. We will start making these at Step 6 below.

- The SalesDT column needs to be split into separate date and time columns.
- There is an erroneous value 27/23/2017 19:48:00 which needs to be removed.
- The Qty column should be whole number data type.
- There is a value of 'Lots' entered in the Qty column which may cause an error.
- The Region column has both UK/United Kingdom and USA/United States.
- The Code column needs to be split into ProductID & Product Name columns.
- The Colour column should be standardised as capitalised.

6. In the **Qty** column, note the current **Data type**. How can the value **'Lots'** be fixed? Can it be fixed in the source data file? On the left of the **Qty** column, click the **Data Type** button and select **Whole Number**. Right-click the **Qty** column and select **Remove Errors**.

7. Select the **SalesDT** column. Change the **Data type** to **Date/Time**. The erroneous value is replaced by **Error**. On the **Home Ribbon**, click the **Remove Rows** icon. Select **Remove Errors.**

8. Select the **Code** column. Click the **Transform** tab on the menu at the top. Click **Split Column** and select **By Delimiter**. In the window, ensure that the **~ symbol** is within the delimiter field. Click **OK**. The code is split into

two columns - **Code.1** and **Code.2**.

9. Select the **Colour** column. Click the **Format** button on the Ribbon and select **Capitalise each word**. In the **Colour** column, right-click a **Null** value and select **Replace Values.** In the **Replace Values** window, enter **No colour stated** in the field. Click **OK**.

10. Select the **SalesDT** column. Right-click and select **Duplicate Column**.

11. Reselect the **SalesDT** column. Right-click, select **Change Type** and choose **Date**.

12. Select the **SalesDT - Copy** column**.** Right-click, select **Change Type**, choose **Time**.

13. Click the **Add Column** menu ribbon at the top. To create a new column based on the Hour value, select **Time**, **Hour,** and **Hour**.

14. In the **Region** column, select the **United States** value (row 10). Right-click and replace this value with **USA**.

15. Select the **United Kingdom** value (row 55). Replace it with **UK**.

16. **Double-click** and **rename** the following columns:

- Change **Code.1** to **Product ID**
- Change **Code.2** to **Product Name**
- Change **SalesDT** to **Sales Date**
- Change **SalesDT - Copy** to **Sales Time**

17. Click the **Home** menu tab at the top, and **Close and Apply** the changes to update the Model.

18. On the **File** menu, click **Save**.

19. Close Power BI Desktop.

Result: You have used the Power Query Editor to edit and improve the quality of the data. You also added some additional columns to display the required values.

Lab 6 – Normalising Data

In the following lab, you will open a Text file and apply a unpivot operation to some data using the Power Query Editor. You will then create a Global Population Report which can be filtered by individual country.

Exercise 1 – Import data from a Text file

1. Open **Power BI Desktop**.

2. If the **Welcome to Power BI Desktop** page appears, click the **Already have a Power BI account? Sign in** link at the bottom. When the **Sign In** dialog box appears, click the **X** in the top right corner to close it.

3. On the **Home** ribbon, select the **Get Data** dropdown and then **Text/CSV.**

4. In the **Open** window, navigate to the **C:\Coursefiles\QAPBIDESK\lab06** folder. Select the **population_total.csv** file. Click **Open**.

5. In the **Preview** window, notice that the years are listed horizontally. It would be easier to use the table if the population values were rows, rather than columns.

6. Click the **Transform Data** button to open the Power Query Editor.

7. On the **Home** ribbon, click **Use First Row as Headers** to promote the first row into the column heading position.

8. **Select** the **geo** column. **Right-click** it and select **Unpivot Other Columns**. The dataset changes so that the column headings and population figures are now displayed as rows.

9. **Rename** the columns (by **double-clicking their column header**) as below:

- **geo** to **Country**
- **Attribute** to **Year**
- **Value** to **Population**

10. Change the **Year column** data type from **Text** to **Whole Number**.

11. On the **Home** Menu tab, click the **Close & Apply** button to update the Model.

Exercise 2 – Create a Global Population Report

1. On the left panel, select **Report** view. In the **Visualizations** pane, select a **Line Chart**. Resize it and position it in the **bottom-right quadrant** of the Canvas.

2. From the **Fields** pane, expand the **population_total** Table. Drag the **Year** field to the **X-axis** bucket on the **Visualizations** pane. Drag the **Population** field to the **Y-axis** bucket. Deselect the Line Chart visual.

3. Add a **Filled Map** visual to the **top-right quadrant** of the Canvas. Drag the **Country** field to the **Location** bucket. Deselect the Filled Map visual.
4. Add a **Slicer** visual to the **left-side** of the Canvas. From the **Fields** pane, click the **Country** field checkbox to populate it.
5. Select individual countries in the Slicer to review their historical population data and geographical location.
6. On the **File** menu, click **Save**. Name the file **QAPBID Lab06**.
7. Close Power BI Desktop.

Result: You used the Power Query Editor to unpivot some population data. You created a number of Visuals to display this population data by individual country.

Lab 7 – Merging Data

In the following lab, you will import multiple sets of data. You will cleanse and merge the sets of data into a single Query/Table using the Power Query Editor. You will then create a Scatter chart to represent the merged data.

Exercise 1 – Importing data from multiple text files

1. Open **Power BI Desktop**.

2. If the **Power BI Desktop** window displays, close it.

3. On the **Home** ribbon, click the **Get Data** dropdown and select **Text/CSV**.

4. In the **Open** window, navigate to the **C:\Coursefiles\QAPBIDESK\lab07** folder. Select the **population_total.csv** file. Click **Open**.

5. Click **Load** to load the data into the Model.

6. **Repeat steps 3 to 5 to import the additional .csv files below**:

..\lab07**income_per_person_gdppercapita_ppp_inflation_adjusted.csv**
..\lab07**life_expectancy_years.csv**
..\lab07**CountryRegion.csv**

7. On the **File** menu, click **Save**. Name the file **QAPBID Lab07**.

Exercise 2 – Cleanse the queries to be ready for merging

1. Select **Transform Data** on the Home ribbon to open the Power Query Editor. The Power Query Editor will open with the four queries/tables listed in the left panel.

2. On the **Queries Panel,** double-click and **rename** the queries as below:

- population_total to **Population**
- income_per_person_gdppercapita_ppp_inflation_adjusted to **Income**
- life_expectancy_years to **Life Expectancy**
- CountryRegion to **Country Region**

3. Select the **Population** query in the left-hand Queries panel.

4. On the **Home** tab ribbon, select **Use First Row as Headers** to promote the first row to the Column headers.

5. Select the **geo** column, **right-click** and select **unpivot other columns**.

6. **Rename** the columns:

- Geo to **Country**
- Attribute to **Year**
- Value to **Population**

7. Change the **Year** column **data type** from Text to **Whole Number**.

8. Select the **Income** query in the left-hand **Queries panel**.

9. On the **Home** tab ribbon, select **Use First Row as Headers** to promote the first row to the Column headers.

10. Select the **geo** column, **right-click** and select **unpivot other columns**.

11. **Rename** the columns:

- Geo to **Country**
- Attribute to **Year**
- Value to **Income per Capita (US$)**

12. Change the **Year** column **data type** from Text to **Whole Number**.

13. Select the **Life Expectancy** query in the left-hand **Queries panel**.

14. On the **Home** tab ribbon, select **Use First Row as Headers** to promote the first row to the Column headers.

15. Select the **geo** column, **right-click** and select **unpivot other columns**.

16. **Rename** the columns:

- Geo to **Country**
- Attribute to **Year**
- Value to **Life Expectancy (years)**

17. Change the **data type** of the **Year** column to **Whole Number**.

18. Select the **Country Region** query in the left-hand Queries panel.

19. On the **Home** tab ribbon, select **Use First Row as Headers**.

20. From the **Close and Apply** dropdown, click **Apply** to save the query updates into the Model. *This may take a few minutes.*

21. On the **File** menu, click **Save**.

Exercise 3 – Merge the data

1. Select the **Population** query in the left-hand Queries panel.

2. On the **Home** tab ribbon, in the **Combine** area, click the **dropdown arrow** next to

3. **Merge Queries.** Select **Merge Queries as New**.

4. In the **Merge** window, select the **Country** column of the **Population** table. Hold down the **CTRL key** and then select the **Year** column as well. This selects both columns for the join. Notice the numbers 1 and 2 on the column headers.

5. From the **second drop down** field, select the **Income** table.

6. Select the **Country** column. Hold down the **CTRL key** and then select the **Year** column as well. The two queries have the same columns selected with matching 1, 2 numbers. The message at the bottom of the window will show the number of row matches between the two tables. The query sources have missing data.

7. Click the **OK** button to add the new query.

8. Ensure that the **Merge1** query is selected. On the **Home** tab, in the **Combine** area, click **Merge Queries**.

9. In the Merge window, select the **Country** column of the **Merge1** table. Hold the

10. **CTRL key** and then select the **Year** column as well.

11. From the **second drop down** field, select the **Life Expectancy** table.

12. Select the **Country** column. Hold down the **CTRL key** and then select the **Year** column as well. The message at the bottom reveals some data with no matching values.

13. Click the **OK** button to merge the queries.

14. **Rename** the **Merge1** query to **Country Data**.

15. Ensure that the **Country Data** query is selected. On the **Income** Column, select the **'Expand' icon** to the right. **Uncheck** the **Country, Year** and **Use original column name as prefix** checkboxes. Click **OK**.

16. On the **Life Expectancy** Column, select the **Expand icon** to the right. **Uncheck** the

17. **Country** and **Year** checkboxes. Click **OK**.

18. The various dimensions of data have now been merged into a single table.

19. On the **Home** ribbon, select the **Close and Apply** button. The merge transformations will be applied to the .pbix file.

Exercise 4 – Create a Scatter Chart

1. Select **Report** view.

2. Click the **Scatter chart** visual to add it to the Report page. **Resize it** as necessary.

3. From the **Fields** pane, drag the following fields to the specified visual bucket:

· **Country Data** \ Income per Capita (US$) to **X-axis**
· **Country Data** \ Life Expectancy (years) to **Y-axis**
· **Country Data** \ Population to **Size**
· **Country Data** \ Country to **Legend**
· **Country Data** \ Year to **Play Axis**

Additional Tips:
You could add some 'Slicer' visuals to the page which would allow you to filter this Scatter chart data in different ways. For example, by country/countries, date range, etc.
To adjust the play rate, the 'Play Axis' visual (from the online Visual Marketplace) could be utilised as well.

4. On the **File** menu, click **Save**.
5. Close Power BI Desktop.

Result: You have used the Power Query Editor to unpivot multiple sets of data. You then merged those data sets into a single Query/Table and created a Scatter chart to represent the merged data.

Lab 8 – Importing multiple Excel files using 'M'

In the following lab, you will create a custom function using the Power Query language 'M' (Mashup). You will import multiple Excel files using that function and then create a Report based on a Map visualisation.

NB. Should be noted that when importing multiple files. All the columns in the tables must match in type and number.

Exercise 1 – Import an Excel file

1. Open **Power BI Desktop**.

2. On the **Canvas Background,** click the **Import data from Excel** button.

3. In the **Open** window, navigate to the **C:\Coursefiles\QAPBIDESK\lab08** folder. Select the **Sales - France.xlsx** file. Click **Open**.

4. In the **Navigator** window, click the checkbox next to **Sales** and review the data.

5. Click **Load** to load the data into the model.

6. On the **File** menu, click **Save**. Name the file **QAPBID Lab8**.

Exercise 2 – Use the Power Query language to create a function

1. Click **Transform Data** to open the Query Editor.

2. On the **Add Column** tab, click the **Custom Column** icon.

3. Name the column **Country**.

4. Set the Custom column formula as below: ="France"

5. Click **OK.** You have added a column with the values set as 'France'.

6. **Double-click** and **Rename** the **Sales** query as **fnReadSalesWorksheet**.

7. Click the **Home** tab, select **Advanced Editor**. In the **Display Options** dropdown, select **Display line numbers**.

8. Review the code and note that the word 'France' is currently hardcoded into the filename (line 2) and the AddColumn statement (line 5).

9. **Enter a few lines BEFORE the ‘let’ command**. Add the following text to change the code into a function:

```
(filename as text, country as text) as table =>
```

10. Replace the hardcoded filename reference using the filename parameter as:

```
Source = Excel.Workbook(File.Contents(filename), null, true),
```

11. Replace the hardcoded “France” reference using the country parameter as:

```
#"Added Custom" = Table.AddColumn(#"Changed Type", "Country", each
country)
```

The complete code should look as follows:

```
(filename as text, country as text) as table => let
Source = Excel.Workbook(File.Contents(filename), null, true), Sales_Table =
Source{[Item="Sales",Kind="Table"]}[Data],
#"Changed Type" = Table.TransformColumnTypes(Sales_Table,{{"SalesOr
derNumber", type text},
{"Main Category", type text}, {"Sub Category", type text}, {"ProductKe y",
Int64.Type},
{"Product", type text}, {"Colour", type text}, {"OrderQuantity", Int64.T ype},
{"UnitPrice", type number},
{"OrderDate", type date}, {"DueDate", type date}, {"ShipDate", type date}}),
#"Added Custom" = Table.AddColumn(#"Changed Type", "Country", each country)
in
#"Added Custom"
```

12. Click **Done** to save the edited code. The display should now change to show a page asking for two parameters (filename and country).

13. To test the function, enter **C:\Coursefiles\QAPBIDESK\lab08\Sales - USA.xlsx** in the filename parameter and **USA** into the country parameter. Click **Invoke** to run the import operation. The query should display 818 rows.

14. Delete the **Invoked Function** query as this was only used to test the function.

15. Click **Close and Apply** to load the query updates into the model.

16. On the **File** menu, click **Save**.

Exercise 3 – Import all Excel files from a folder using the function

1. On the **Home** ribbon, click **Get Data**. Select **Folder** and click **Connect**.

2. In the **Folder** window, note that multiple file types exist within the **...\lab08** folder. Click **OK**.

3. Click the **Transform Data** button to import the folder content.

4. Using the **Extension** column, **filter the folder** content to remove all file extensions apart from the 5 **.xlsx** files.

5. Click the **Add Column** tab. Select **Custom Column**. **Name** the new column **Filename.** Enter the following formula to create a full filename for each file:

=[Folder Path] & [Name]

6. Click **OK**

7. Select the **Name** column within the query. On the **Add Column** ribbon, click the **Column From Examples dropdown** and then select **From Selection**.

8. In the **Column1** column, **double-click** the **cell** and add the text **Canada** to the first row. Press enter to move to the next line. Each of the rows should now show the correct country. Select the **OK button** above to store the calculation.

9. **Rename** the column **Country**.

10. **Remove all columns** from the lab08 query except the **Filename** and **Country** columns.

11. Click **Invoke Custom Function**.

12. In the **Invoke Custom Function** window, select the Function query **fnReadSalesWorksheet** from the dropdown. The **filename** parameter should be set to use the **Filename** column. Click the **ABC** button next to the Country parameter. Select **Column Name**. From the **dropdown**, select the **Country** column. Click **OK** to invoke the function.

13. Click the **Expand button** on the new function column. **Deselect** the **Use original column name as prefix** checkbox. Click **OK**.

14. **Remove** the first two columns from the query (**Filename** and **Country**).

15. **Rename** the query **Lab08** to **Sales**.

16. **Rename** the last column **Country.1** to **Country**.

17. **Change** the **data type** of the columns as list below:

- OrderQuantity to **Whole Number**

- UnitPrice to **Decimal**
- OrderDate to **Date**

- DueDate to **Date**
- ShipDate to **Date**

18. Select the **Home** tab, click **Close and Apply** to update the table into the model.
19. On the **File** menu, click **Save**.

Exercise 4 – Add a Report (Map)

1. In the **Visualizations** pane, click the **Map** visual to add it to the canvas. Resize the Map visual to fill the complete page area. Make sure it is **selected**.
2. From the **Fields** pane, drag the **Country** field to the **Location** bucket on the **Visualizations** pane.
3. Click and drag the **OrderQuantity** field to the **Bubble size** bucket.
4. Click and drag the **Sub Category** field to the **Legend** bucket.
5. Review the map content.
6. On the **File** menu, click **Save**.
7. Close Power BI Desktop.

Result: You created a Custom Function based on the Power Query language 'M'. You then used the Custom Function in the Query Editor to import multiple Excel files. You finished by creating a Map Visual for the Report.