



# Dashboard in a Day – Lab 1

## Accessing & Preparing Data

by Power BI Team, Microsoft



## Contents

Introduction .....	3
Power BI Desktop .....	3
Power BI Desktop – Accessing Data.....	3
Power BI Desktop – Data Preparation.....	16
References .....	31

# Introduction

Today you will learn about various key features of the Power BI service. This is an introductory course intended to teach you how to author reports using Power BI Desktop, create operational dashboards, and share content via the Power BI Service.

By the end of this lab, you will have learned:

- How to load data from Microsoft Excel and Comma-Separated Values (CSV) sources
- How to manipulate the data to prepare it for reporting
- How to prepare the tables in Power Query and load them into the model

Learning these steps will prepare you for the reporting exercises in Lab 2.

## Power BI Desktop

### Power BI Desktop – Accessing Data

In this section, you will import VanArsdel's and its competitors' USA sales data. You will then import and merge sales data from other countries.

#### Power BI Desktop - Get Data

Let's start by looking at the data files. The dataset contains sales data of VanArsdel and other competitors. We have seven years of transaction data by day, product, and zip code for each manufacturer. We are going to analyze data from seven countries.

USA sales data is in a CSV file located in the USSales subfolder within the Data folder (/Data/USSales).

Sales of all other countries is in the InternationalSales subfolder within the Data folder (/Data/InternationalSales). Each country's sales data is in a CSV file in this folder.

Product, Geography, and Manufacturer information is in a Microsoft Excel file called bi\_dimensions.xlsx in the USSales subfolder within the Data folder (/Data/USSales/).

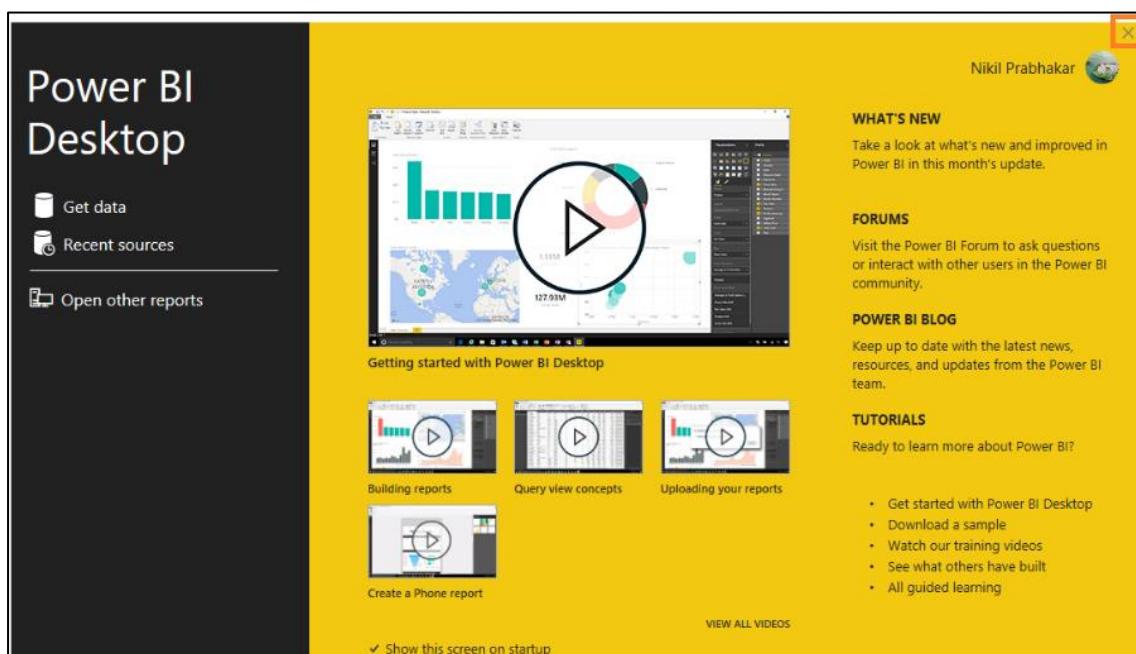
1. Open the **bi\_dimensions.xlsx** file. Notice that the first sheet has **Product** information. This sheet has a header, and product data is in a named table. Also notice that the **Category** column numerous empty cells.

The **Manufacturer sheet** has data laid out across the sheet, no column headers, several blank rows, and a note in row seven.

The **Geo sheet** has the geography information. The first few rows have data details. Actual data starts on row four.

A	B	C	D	E	F	
1	Source: Public Database					
2	Last Updated: Monday, February 1, 2016					
3						
4	Zip	City	State	Region	District	Country
5	22654	Star Tannery, VA, USA	VA	East	District #07	USA
6	22655	Stephens City, VA, USA	VA	East	District #07	USA
7	22656	Stephenson, VA, USA	VA	East	District #07	USA
8	22657	Strasburg, VA, USA	VA	East	District #07	USA
9	22660	Toms Brook, VA, USA	VA	East	District #07	USA
10	22663	White Post, VA, USA	VA	East	District #07	USA
11	22664	Woodstock, VA, USA	VA	East	District #07	USA
12	22701	Culpeper, VA, USA	VA	East	District #07	USA
13	22709	Aroda, VA, USA	VA	East	District #07	USA
14	22711	Banco, VA, USA	VA	East	District #07	USA
15	22712	Bedford, VA, USA	VA	East	District #07	USA

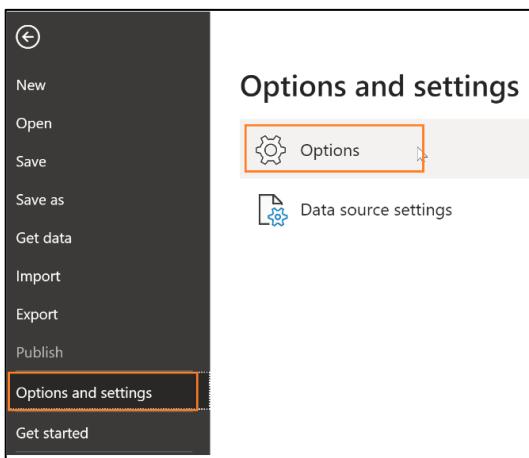
We will start by connecting to data from these different sheets, and then perform data cleaning and transformation operations.



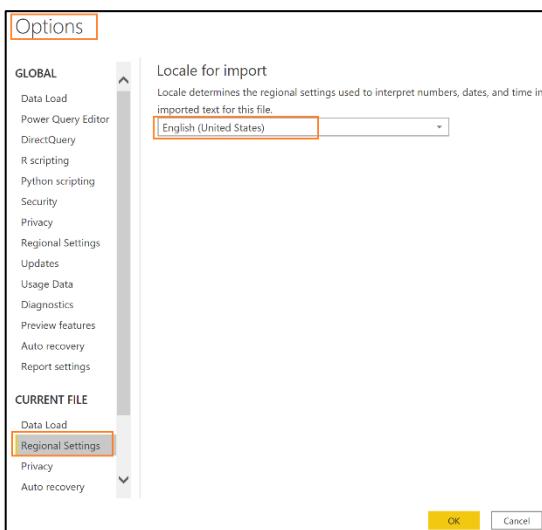
2. If you don't have the **Power BI Desktop** open, launch it now.
3. Click **Already have a Power BI Account? Sign in** option.
4. **Sign in** using your Power BI credentials.
5. You will see the startup screen opens. Click on the **X** on the top right corner of the dialog box to close it.

Let's set the **Locale** to US English to make it convenient in the rest of this lab.

6. From the ribbon, click **File**, then click **Options and settings**, then click **Options**.

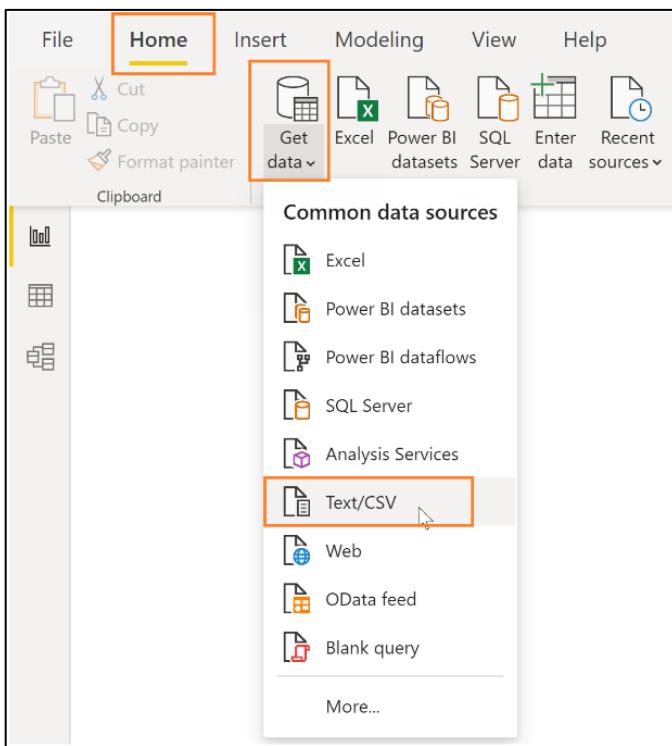


7. In the left panel of **Options** dialog box, click **Regional Settings** under **Current File**.
8. From the **Locale** drop-down, click **English (United States)**.
9. Click **OK** to close the dialog box.



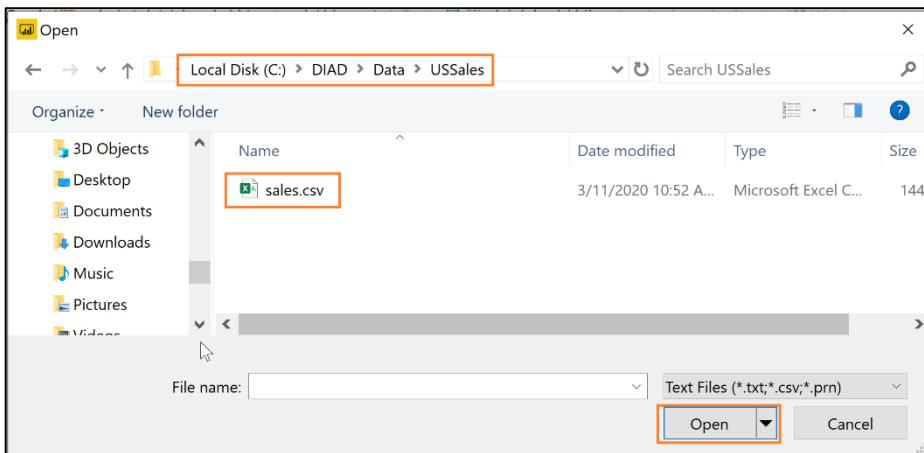
The next step is to load data to Power BI Desktop. We will load USA Sales data which is in CSV files.

10. From the ribbon, click **Home** and then click the **Get Data** drop-down arrow.
11. Click **Text/CSV**.



**Note:** Power BI Desktop has the capability to connect to 300+ data sources. We are using CSV and Excel data files in this lab for simplicity. If you would like a full list of data sources, please visit this link: <https://docs.microsoft.com/en-us/power-bi/connect-data/desktop-data-sources>

12. Browse to **DIAD**, double-click **Data**, double-click the **USSales** folder, and then click **sales.csv**.
13. Click the **Open** button.



Power BI detects the data type within each column. There are options to detect the data type based on the first 200 rows, based on the entire dataset or to not detect the data. Since our dataset is large and it will take time and resources to scan the complete dataset, we will leave the default option of selecting the dataset based on the first 200 rows.

After completing your selection, you have three options – Load, Edit or Cancel.

- **Load** adds the data from the source into Power BI Desktop for you to start creating reports.

- **Transform Data** allows you to perform data shaping operations such as merging columns, adding additional columns, changing data types of columns as well as bringing in additional data.
- **Cancel** gets you back to the main canvas.

14. Click **Transform Data** as shown in the screenshot. A new window opens.

ProductID	Date	Zip	Units	Revenue
1076	1/20/2013	72638	1	254.5725
1076	1/21/2013	47577	1	254.5725
1076	1/28/2013	34653	1	254.5725
1076	1/31/2013	84014	1	254.5725
1076	2/1/2013	75070	1	254.5725
1076	2/1/2013	87031	1	254.5725
1076	2/3/2013	72019	1	254.5725
1076	2/3/2013	72086	1	254.5725
1076	2/3/2013	77089	2	509.145
1076	2/9/2013	7649	1	254.5725
1076	2/11/2013	79705	1	254.5725
1076	2/14/2013	92624	1	254.5725
1076	2/22/2013	8527	1	254.5725
1076	2/22/2013	8816	1	254.5725
1076	2/23/2013	24740	1	254.5725

You should be in the Query Editor window as shown in the screenshot below. The Query Editor is used to perform data shaping operations. Notice that the sales file you connected to shows as a query in the left panel. You can see a preview of the data in the center panel. Power BI predicts the data type of each field (based on the first 200 rows) as indicated next to the column header. In the right panel, steps that the Query Editor performs are recorded in the Applied Steps section.

Queries [9]

Sales

	ProductID	Date	Zip
1	1085	12/2/2019	76234
2	1085	12/2/2019	85301
3	1085	12/5/2019	32341
4	1085	12/5/2019	90745
5	1085	12/5/2019	91325
6	1085	12/6/2019	21061
7	1085	12/6/2019	21225
8	1085	12/6/2019	32765
9	1085	12/6/2019	93033
10	1085	12/6/2019	93103
11	1085	12/7/2019	58504
12	1085	12/7/2019	92346
13	1085	12/8/2019	35242
14	1085	12/8/2019	37072
15	1085	12/8/2019	38655
16	1085	12/8/2019	40701
17	1085	12/8/2019	83709
18			

6 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 9:19 AM

**Note:** You will bring in sales data from other countries as well as performing certain data shaping operations.

15. Notice that Power BI has set the **Zip** field to the data type **Whole Number**. To ensure that the leading zero is not dropped from Zip codes that start with zero, we will format them as **Text**. To do this, select the **Zip** column. Then, from the ribbon, click **Home**, click **Data Type**, and change it to **Text**.
16. The **Change Column Type** dialog box opens. Click the **Replace Current** button which overwrites Power BI's predicted data type.

IMPORTANT!  
Changing the data type is a big deal to perform later

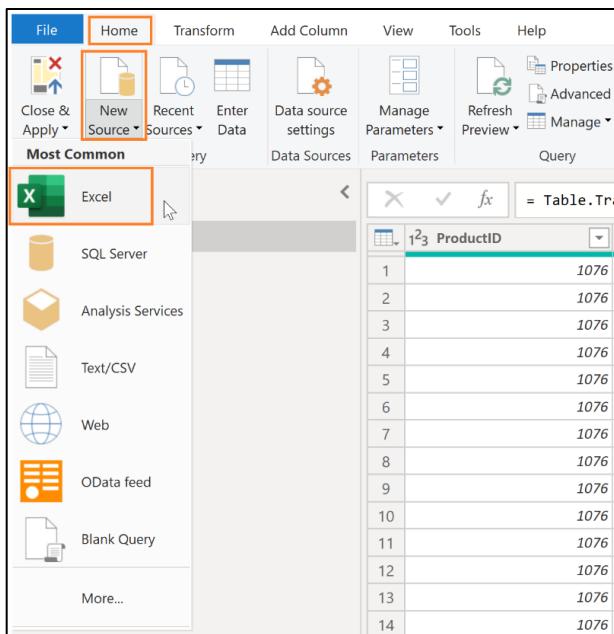
Leading zero is added

Data Type: Text

	ProductID	Date	Zip
1	1076	1/20/2013	72638
2	1076	1/21/2013	47577
3	1076	1/28/2013	34653
4	1076	1/31/2013	84014
5	1076	2/1/2013	75070
6	1076	2/1/2013	87031
7	1076	2/3/2013	72019
8	1076	2/3/2013	72086
9	1076	2/3/2013	77089
10	1076	2/9/2013	07649
11	1076	2/11/2013	79705
12	1076	2/14/2013	92624
13	1076	2/22/2013	08527
14	1076	2/22/2013	08816
15	1076	2/23/2013	24740
16	1076	2/24/2013	63023
17			

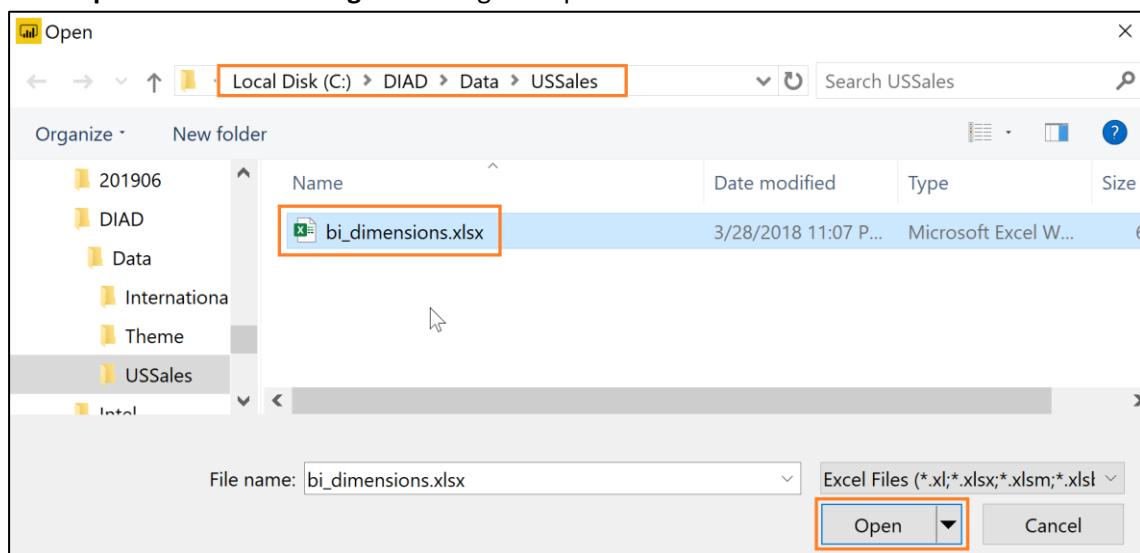
Now let's get the data that is in Excel source file.

17. From the ribbon, click **Home**, click **New Source**, and click then **Excel**.



18. Browse to **DIAD**, double-click **Data**, double-click the **USSales** folder, and then click **bi\_dimensions.xlsx**.

19. Click the **Open** button. The **Navigator** dialog box opens.



20. The **Navigator** dialog box lists three sheets that are in the Excel workbook. It also lists the **Product** table. Click **product** in the panel on the left. In the preview panel, notice that the first row is the headers. This is not part of the data.

21. Now, deselect **product** from the left panel and click **Product\_Table**. Notice that this table has only the contents of the named table. This is the data we need.

The screenshot shows the Power BI Desktop interface. On the left, the Navigator pane displays a folder structure for 'bi\_dimensions.xlsx [4]'. Under 'Product\_Table', 'geo', 'manufacturer', and 'product' are listed, with 'product' selected and highlighted by a red box. On the right, the preview pane shows a table named 'product' with columns 'Product Details', 'Column2', 'Column3', 'Column4', and 'Column5'. The data includes rows for products like 'Abbas MA-01|All Season' and 'Abbas MA-02|All Season'.

**Navigator**

Display Options

bi\_dimensions.xlsx [4]

- Product\_Table
- geo
- manufacturer
- product**

**product**

Product Details	Column2	Column3	Column4	Column5
ProductID	Product	Category	ManufacturerID	Price
1	Abbas MA-01 All Season	Mix	1	USD 412.13
2	Abbas MA-02 All Season	null	1	USD 329.78
3	Abbas MA-03 All Season	null	1	USD 963.38
4	Abbas MA-04 All Season	null	1	USD 828.98
5	Abbas MA-05 All Season	null	1	USD 745.5

**Navigator**

Display Options

bi\_dimensions.xlsx [4]

- Product\_Table**
- geo
- manufacturer
- product

**Product\_Table**

ProductID	Product	Category	ManufacturerID	Price
1	Abbas MA-01 All Season	Mix	1	USD 412.13
2	Abbas MA-02 All Season	null	1	USD 329.78
3	Abbas MA-03 All Season	null	1	USD 963.38
4	Abbas MA-04 All Season	null	1	USD 828.98
5	Abbas MA-05 All Season	null	1	USD 745.5
7	Abbas MA-07 All Season	null	1	USD 451.45

**Note:** Table names are differentiated from Worksheet names by using different icons.

22. From the left panel, click **geo**. In the preview panel, notice that the first few rows are headers and are not part of the data. We will remove them shortly.
23. From the left panel, click **manufacturer**. In the preview panel, notice that the last couple of rows are footers and are not part of the data. We will remove them shortly.
24. Make sure that **Product\_Table**, **geo** and **manufacturer** are selected in the left panel, and then click **OK**. Notice all that three sheets are added as queries in the Query Editor.

The screenshot shows the Power BI Desktop interface. On the left, the Navigator pane displays a folder structure for 'bi\_dimensions.xlsx [4]'. Under 'Product\_Table', 'geo', and 'manufacturer' are listed, with 'geo' and 'manufacturer' selected and highlighted by a red box. On the right, the preview pane shows a table named 'manufacturer' with columns 'Column1' and 'Column2'. The data includes rows for 'ManufacturerID', 'Manufacturer' (Abbas), 'Logo' (a URL), and 'List of Suppliers and Manufacturers'.

**Navigator**

Display Options

bi\_dimensions.xlsx [4]

- Product\_Table
- geo**
- manufacturer**
- product

**manufacturer**

Column1	Column2
ManufacturerID	
Manufacturer	Abbas
Logo	https://raw.githubusercontent.com
	null
	null
List of Suppliers and Manufacturers	

**OK** **Cancel**

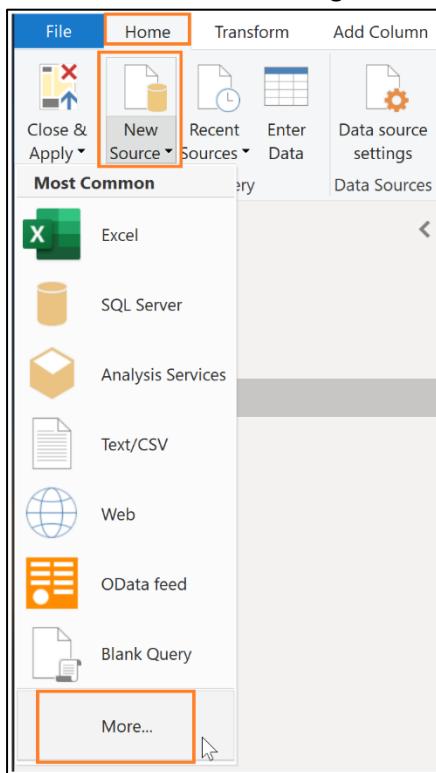
## Power BI Desktop - Adding additional data

In this scenario, the international subsidiaries have agreed to provide their sales data so that the company's sales can be analyzed together. You've created a folder where they each put their data.

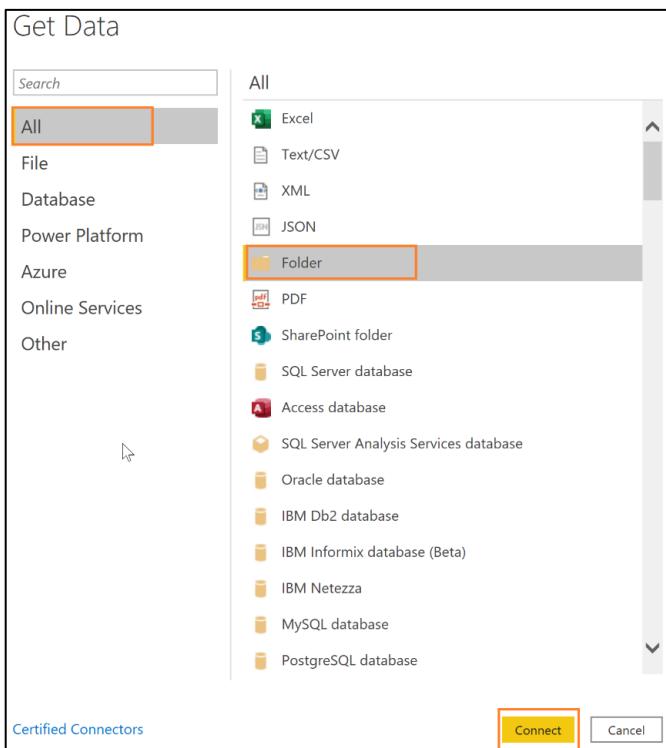
To analyze all the data together, you import the new data from each of the subsidiaries and combine it with the US Sales you loaded earlier.

You can load the files one at a time, like how you loaded the US Sales data, but Power BI provides an easier way to load all the files in a folder together.

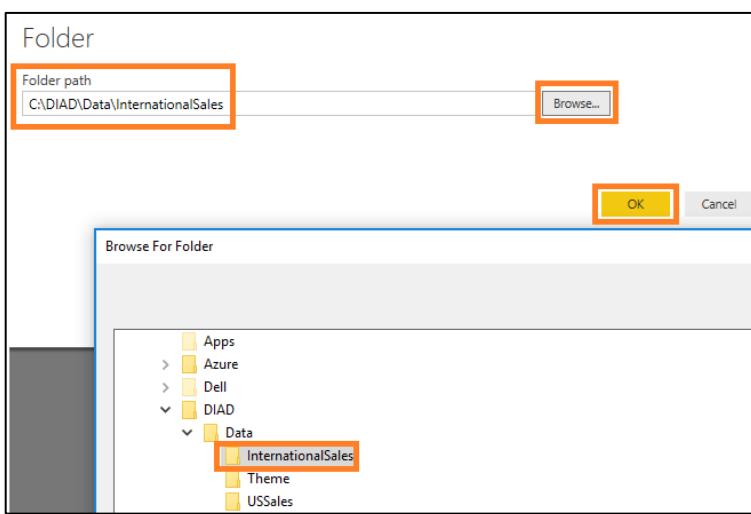
25. On the **Home** tab of the Query Editor, click on the **New Source** drop-down menu.
26. Click **More...** as shown in the figure.



27. The Get Data dialog box opens.
28. In the **Get Data** dialog box, click **Folder** as shown in the diagram.
29. Click **Connect** and the **Folder** dialog box will open.



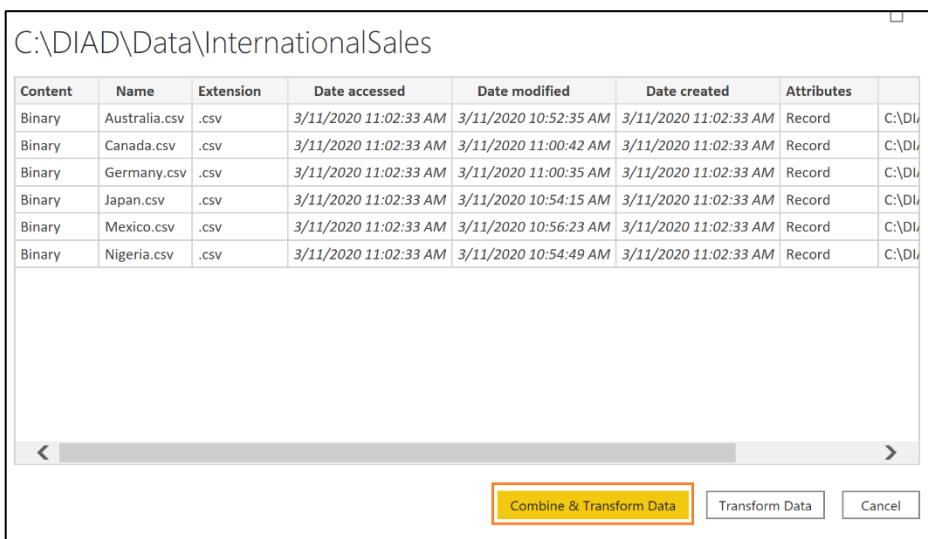
30. Click the **Browse...** button.
31. In the **Browse for Folder** dialog box, navigate to the location where you unzipped the class files.
32. Open the **DIAD** folder.
33. Open the **Data** folder.
34. Click the **InternationalSales** folder.
35. Click **OK** (to close the **Browse for Folder** dialog box).
36. Click **OK** (to close the **Folder** dialog box).



**Note:** This approach will load all the files located in the folder. This is useful when you have a group that puts files on an FTP site each month and you are not always sure of the names of the files or the number of files. All the files must be of the same file type with columns in the same order.

The dialog box will display the list of files in the folder.

**37. Click Combine & Transform Data.**

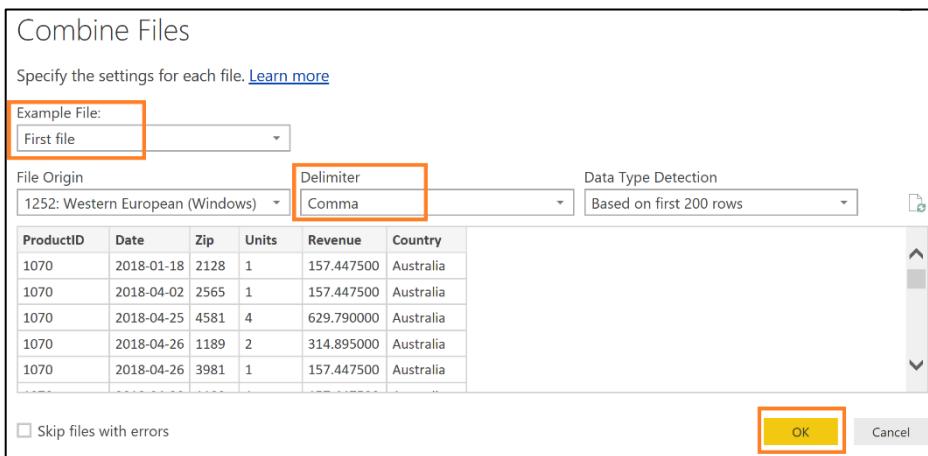


**Note:** The data in your file for **Date accessed**, **Date modified**, and **Date created** might be different than the dates displayed in the screenshot.

The **Combine Files** dialog box will open. By default, Power BI will again detect the data type based on the first 200 rows. Notice there is an option to select various file Delimiters. The file we are working with is Comma delimited, so let's leave the Delimiter option as Comma.

There is also an option to select each individual file in the folder (using **Example File** drop-down) to validate the format of the files.

**38. Click OK.**



You will now be in the **Query Editor** window with a new query named **InternationalSales**.

39. If you do not see the **Queries** pane on left, click on the > (greater than) icon to expand.
40. If you do not see the **Query Settings** pane on the right as shown in the figure, click on **View** in the ribbon and click **Query Settings** to see the pane.
41. Click on the Query **InternationalSales**.

The screenshot shows the Power BI desktop interface. The ribbon is at the top with tabs: File, Home, Transform, Add Column, View, Tools, Help. The 'View' tab is highlighted. On the far left, there's a 'Query Settings' icon. The main area has a 'Data Preview' grid showing a table with four columns: 'Source.Name', 'ProductID', 'Date', and 'Zip'. The 'Zip' column contains numerical values. Below the preview is a 'Queries [9]' pane listing various queries, including a 'Transform File from InternationalSales...' entry under the 'InternationalSales' folder.

Notice that column Zip is of the Whole Number type. Based on the first 200 rows, Power BI thinks the Zip column consists of whole numbers. But zip code could be alpha numeric in some countries or regions or contain leading zeros. If we do not change the data type, we will receive an error when we load the data shortly. So, let's change the Zip column to data type Text.

42. Highlight the Zip column and change the **Data Type** to **Text**.

43. The **Change Column Type** dialog box will open. Click the **Replace Current** button.

The screenshot shows the 'Change Column Type' dialog box in Power BI. The 'Data Type' dropdown is set to 'Text'. Other options listed include Decimal Number, Fixed decimal number, Whole Number, Percentage, Date/Time, Date, Time, Date/Time/Timezone, Duration, and True/False. The 'Text' option is highlighted with a red box. In the background, the Power BI interface shows the same table and queries as the previous screenshot.

In the Queries panel, notice that a Transform File from the InternationalSales folder is created. This contains the function used to load each of the files into the folder.

The screenshot shows the Power BI desktop interface. The ribbon is at the top with tabs: File, Home, Transform, Add Column, View, Tools, Help. The Home tab is selected. Below the ribbon is the 'Queries [9]' pane. Under 'Transform File from InternationalSales...', there are three helper queries: 'Sample File', 'Parameter1 (Sample File)', and 'Transform File'. Under 'Other Queries [5]', there are five queries: 'sales', 'Product\_Table', 'geo', 'manufacturer', and 'InternationalSales'. The 'InternationalSales' query is highlighted with an orange border. The main area shows a table with the following data:

	Source.Name	ProductID	Date	Country
1	Australia.csv	1070	1/18/2019	Australia
2	Australia.csv	1070	4/2/2019	Australia
3	Australia.csv	1070	4/25/2019	Australia
4	Australia.csv	1070	4/26/2019	Australia
5	Australia.csv	1070	4/26/2019	Australia
6	Australia.csv	1070	4/30/2019	Australia
7	Australia.csv	1070	5/14/2019	Australia
8	Australia.csv	1070	5/27/2019	Australia
9	Australia.csv	1070	5/30/2019	Australia
10	Australia.csv	1070	6/7/2019	Australia
11	Australia.csv	1070	6/20/2019	Australia
12	Australia.csv	1070	6/27/2019	Australia
13	Australia.csv	1070	6/27/2019	Australia
14	Australia.csv	1070	6/27/2019	Australia
15	Australia.csv	1070	6/27/2019	Australia
16	Australia.csv	1070	6/28/2019	Australia

If you compare the **InternationalSales** and the **sales** table, you will see the **InternationalSales** table contains two new columns, **Source.Name** and **Country**.

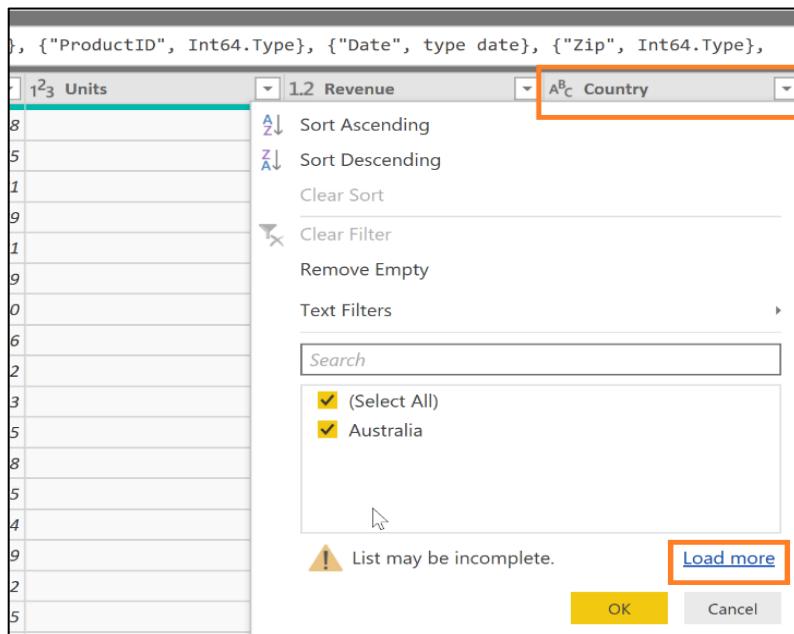
44. We do not need the **Source.Name** column. Click the **Source.Name** column and from the ribbon, click **Home**, click **Remove Columns**, and then click **Remove Columns** again.

The screenshot shows the Power BI desktop interface. The ribbon is at the top with tabs: File, Home, Transform, Add Column, View, Tools, Help. The Home tab is selected. Below the ribbon is the 'Queries [9]' pane. The 'InternationalSales' query is selected and highlighted with an orange border. The main area shows a table with the following data:

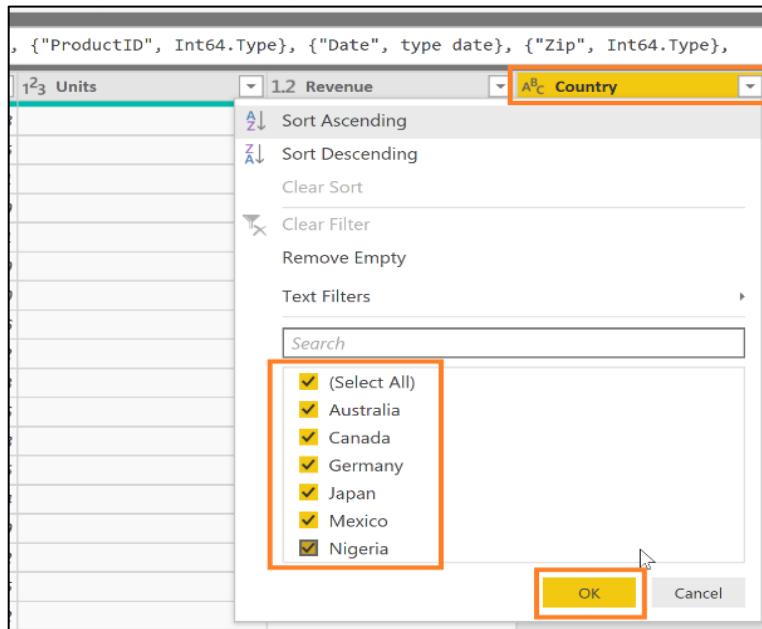
	Source.Name	ProductID	Date	Country
1	Australia.csv	1070		
2	Australia.csv	1070		
3	Australia.csv	1070		
4	Australia.csv	1070		
5	Australia.csv	1070		
6	Australia.csv	1070		
7	Australia.csv	1070		
8	Australia.csv	1070		
9	Australia.csv	1070		
10	Australia.csv	1070		
11	Australia.csv	1070		
12	Australia.csv	1070		
13	Australia.csv	1070		

45. Next, click the drop-down menu next to the **Country** column to see the unique values.  
 46. You will only see Australia as shown in the figure. By default, Power BI only loads the first 1000 rows. Click **Load more** to validate that you have data from the various countries included.

47.



You will see the countries Australia, Canada, Germany, Japan, Mexico, and Nigeria.



48. Click **OK**.

**Note:** You can perform various types of filters, sorting operations using the drop-down to verify the imported data.

## Power BI Desktop – Data Preparation

In this section, we will explore methods to [transform data in the data model](#). Transforming the data by renaming tables, updating data types, and appending tables together ensures that the data is ready to be used for reporting. In some instances, this means cleaning the data up so that similar sets of data

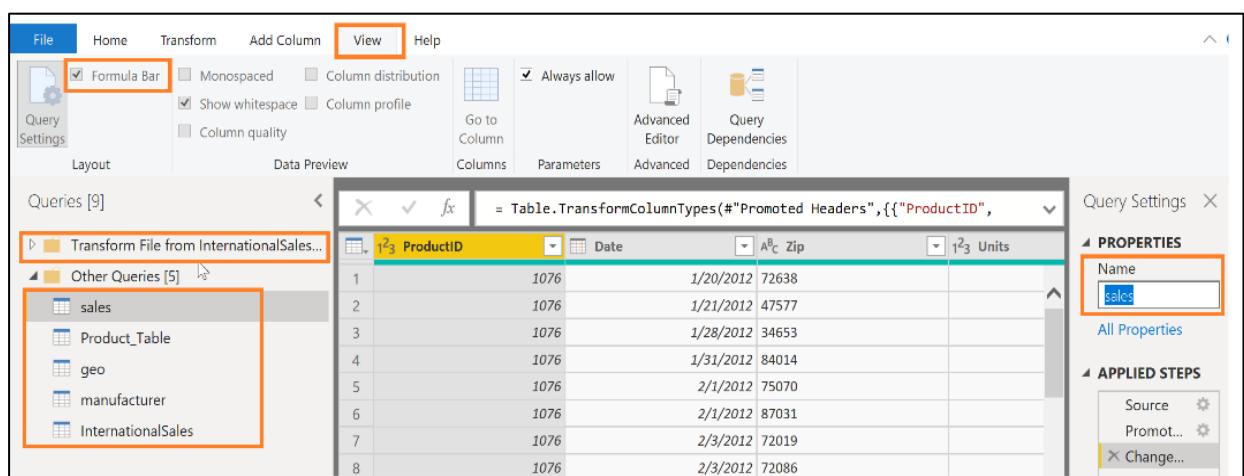
can be combined. In other instances, groups of data are renamed so that they are more easily recognized by end users and report writing is simplified.

## Power BI Desktop - Renaming tables

The Query Editor window should appear as shown below.

- If formula bar is disabled, you can turn on the formula bar from the **View** ribbon. This enables you to see the “M” code generated by each click on the ribbons.
- Click the options available on the ribbon, **Home**, **Transform**, **Add Column**, and **View**, to review the various features available.

1. Under the **Queries** panel, minimize the **Transform Files from InternationalSales** folder.
2. Click each query name in the **Other Queries** section.



The screenshot shows the Power BI Query Editor interface. The 'View' ribbon tab is selected, with the 'Formula Bar' checkbox checked. The 'Queries' pane on the left shows a folder 'Transform File from InternationalSales...' expanded, revealing five queries: 'sales', 'Product\_Table', 'geo', 'manufacturer', and 'InternationalSales'. The 'sales' query is selected. The main preview area shows a table with columns ProductID, Date, Zip, and Units. The 'Properties' pane on the right shows the 'Name' field set to 'sales'. The 'Applied Steps' pane shows a step named 'Change...'.

3. Navigate to **Query Settings**, and then the **Properties** section to rename the queries as shown below:

Initial Name	Final Name
sales	Sales
Product_Table	Product
geo	Geography
manufacturer	Manufacturer
InternationalSales	International Sales

**Note:** It is a best practice to provide descriptive query names and column names. These names are used in visuals and in the Q&A section, which is covered later in the lab.

## Power BI Desktop – Filling empty values

In our scenario, some of the data is not in the right format. Power BI provides extensive transformation capabilities to clean and prepare data to meet your needs. Let's start with the Product query.

Notice that the **Category** column has a lot of null values. Hover over the green/gray bar (known as the quality bar) below the column header. This allows you to easily identify errors and empty values in your data previews. It looks like there are values in the Category column only when the value changes. We need to provide data in this column so there are values in each row.

The screenshot shows the Power BI Desktop interface. On the left, the 'Queries [9]' pane is open, with the 'Product' query selected and highlighted with a red border. The main area displays a table with columns: ProductID, Product, Category, and Manufacturer. The Category column contains values like 'Abbas MA-01|All Season' and 'Mix', while the Manufacturer column contains 'null'. The ribbon at the top has the 'Transform' tab selected.

ProductID	Product	Category	Manufacturer
1	Abbas MA-01 All Season	Mix	null
2	Abbas MA-02 All Season		null
3	Abbas MA-03 All Season		null
4	Abbas MA-04 All Season		null
5	Abbas MA-05 All Season		null
6	Abbas MA-07 All Season		null
7	Abbas MA-06 All Season		null
8	Abbas MA-08 All Season		null

4. In the left panel, click the **Product** Query.
5. Click the **Category** column.
6. From the ribbon, click **Transform**, click **Fill**, and then click **Down**.

This screenshot shows the same Power BI Desktop interface after performing the steps in the previous list. The 'Category' column now contains the correct values for all rows, such as 'Abbas MA-01|All Season' for row 1 and 'Mix' for row 2. The 'Manufacturer' column still contains 'null' for all rows. The ribbon's 'Transform' tab is still selected, and the 'Fill Down' button is highlighted with a red border.

ProductID	Product	Category	Manufacturer
1	Abbas MA-01 All Season	Mix	null
2	Abbas MA-02 All Season		null
3	Abbas MA-03 All Season		null
4	Abbas MA-04 All Season		null
5	Abbas MA-05 All Season		null
6	Abbas MA-07 All Season		null
7	Abbas MA-06 All Season		null
8	Abbas MA-08 All Season		null

Notice how all the null values are filled with the appropriate Category values.

## Power BI Desktop – Splitting columns

In the Product query, notice the **Product** column. It looks like the product name and product segment are concatenated into one field with a pipe (|) separator. Let's split them into two columns. This will be useful when we build visuals, so we can analyze based on both fields.

7. From the left panel, click the **Product** Query.
  8. Click the **Product** column.
  9. From the ribbon, click **Transform**, click **Split Column**, and then click **By Delimiter**. The **Split Column by Delimiter** dialog box opens.
  10. In the dialog box, make sure that **Custom** is selected in the **Select or enter delimiter** drop-down menu.
- Note:** The **Select or enter delimiter** drop-down menu has some of the standard delimiters like comma, colon, and so on.
11. Notice that in the text area, there is a hyphen (-). Power BI assumes we want to split by hyphen. Remove the hyphen symbol and enter the pipe symbol (|) as shown in the screenshot.

## 12. Click OK.

**Note:** If the delimiter occurs multiple times, the **Split at** section provides the option to split only once (either left most or right most) or the option to split the column on each occurrence of the delimiter.

In this scenario, the delimiter occurs only once, therefore the Product column is split into two columns.

## Power BI Desktop – Renaming columns

Let's rename the columns.

13. Click the **Product.1** column, and then **right-click** next to the column name.

14. Click **Rename...** from the selection menu.

15. **Rename** the field to **Product**.

16. Following these steps, also rename **Product.2** to **Segment**.

## Power BI Desktop – Using Column From Examples to split columns

In the Product query, notice that the Price column has price and currency concatenated into one field. To do any calculations we only need the numeric value. Therefore, we need to split this field into two columns. We can use the split feature like earlier or we can use **Column From Examples**. **Column From Examples** is handy in scenarios where the pattern is more complex than simply a delimiter.

17. From the left panel, click the **Product** Query.
18. From the ribbon, click **Add Column**, click **Column From Examples**, and then click **From All Columns**.
19. In the **first row of Column1**, enter the first Price value, **412.13**, and click enter.
- Notice after you click enter, Power BI knows that you want to split Price column. The formula Power BI uses is displayed as well.
20. **Double-click** the column header **Text After Delimiter** to rename it.
21. **Rename** the column to **MSRP**.
22. Click **OK** to apply the changes.

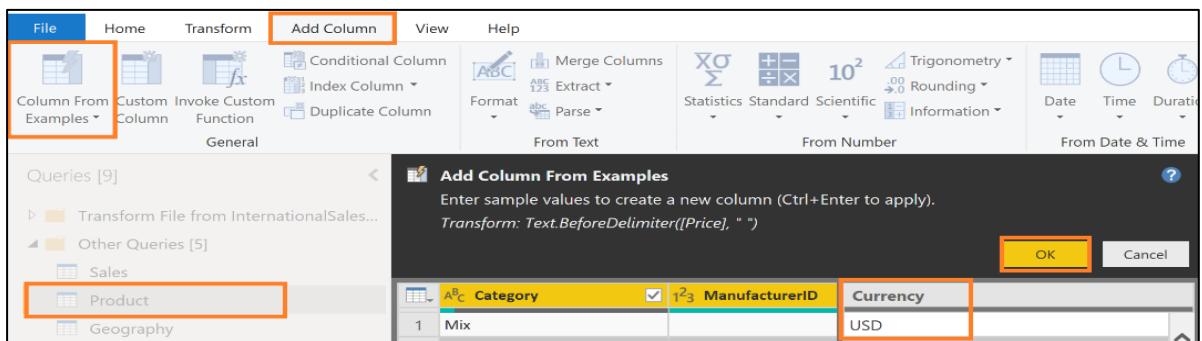
The screenshot shows the Power BI ribbon with the 'File' tab selected. Under the 'Add Column' tab, the 'Column From Examples' button is highlighted with a red box. A 'Queries [9]' pane on the left shows a 'Product' query selected. A 'Add Column From Examples' dialog is open, prompting for sample values to create a new column. The formula 'Transform: Text.AfterDelimiter([Price], " ")' is shown. A table preview shows the 'MSRP' column with the value '412.13'. The 'OK' button in the dialog is highlighted with a red box.

- Notice that the **MSRP** field is of the data type text. It needs to be a decimal. Let's change it.
23. Click **ABC** in the **MSRP** column.
  24. From the menu, click **Fixed Decimal Number**.

The screenshot shows the Power BI ribbon with the 'Transform' tab selected. In the 'ABC' dropdown for the 'MSRP' column, the 'Fixed Decimal Number' option is highlighted with a red box. The 'Query Settings' pane on the right shows the 'Product' query settings. The 'APPLIED STEPS' pane is expanded, showing the steps: Source, Navigation, Changed Type, Split Column by Delimiter, Changed Type1, Renamed Columns, and Inserted Text After Delimiter. The 'Insert Text After Delimiter' step is highlighted with a red box.

Notice that all the steps we performed on the Product query are being recorded under **APPLIED STEPS** in the right panel.

- Now let's create a currency column in the same way.
25. With the Product query selected, from the ribbon, click **Add Column**, click **Column From Examples**, and then click **From All Columns**.
  26. In the first row of Column1 enter the first **Currency** value as USD and then click enter.

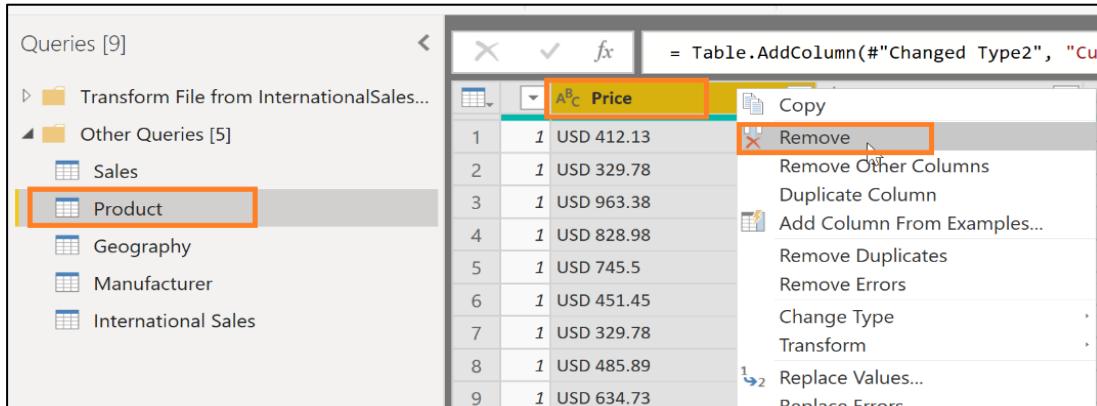


Notice that after you click enter, Power BI knows you want to split the Price column. The formula it uses is displayed as well.

27. Double click the column header **Text Before Delimiter** to rename it.
28. Rename the column to **Currency**.
29. Click **OK** to apply the changes.

Now that we have split **Price** column into the **MSRP** and **Currency** columns, we don't need the **Price** column. Let's remove it.

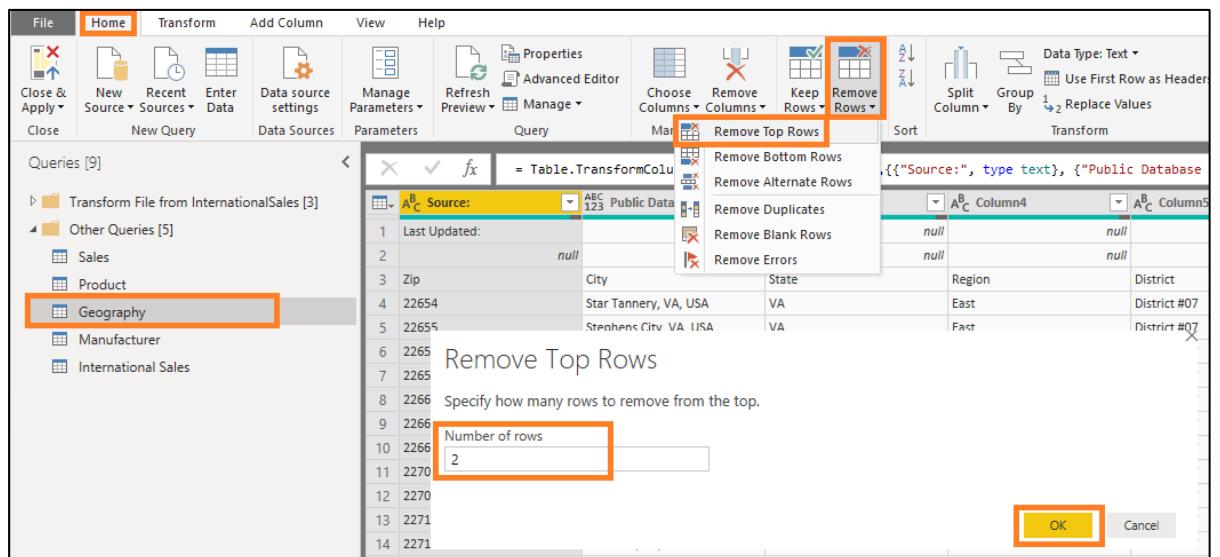
30. From the left panel, click the **Product** Query.
31. Right-click on the **Price** column.
32. Click **Remove**.



### Power BI Desktop – Removing unwanted rows

In the **Geography** query, notice that the first two rows are informational. They are not part of the data. Similarly, in the Manufacturer query, the last couple of rows are not part of the data. Let's remove them so we have a clean dataset.

33. In the left panel, click the **Geography** query.
34. From the ribbon, click **Home**, click **Remove Rows**, and then click **Remove Top Rows**.
35. The **Remove Top Rows** dialog box opens. Enter **2** in the text box since we want to remove the top informational data row and the blank second row.
36. Click **OK**.



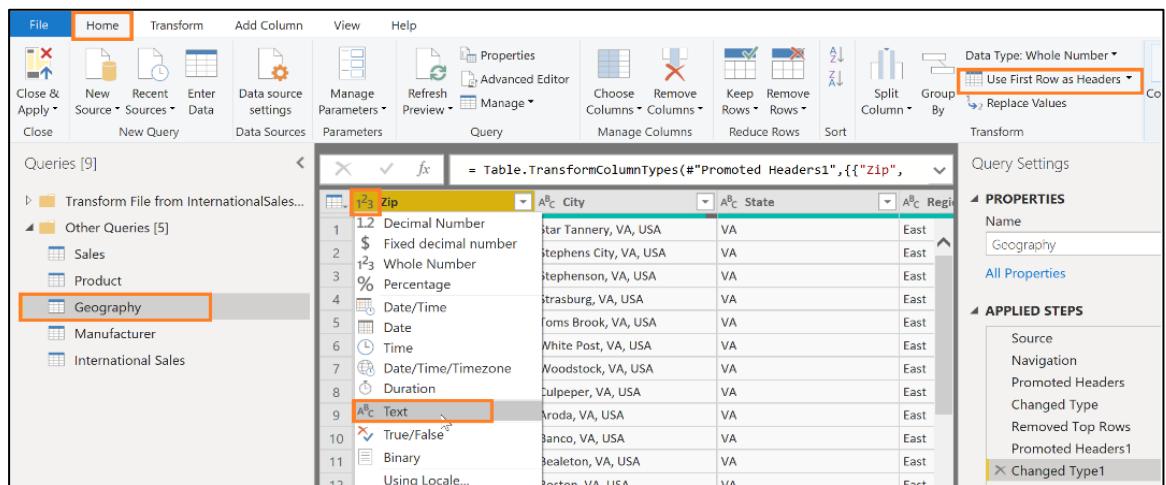
Notice the first row in the Geography query is now the column header. Let's make it a header.

37. With **Geography** query selected in the left panel, from the ribbon click **Home**, and then click **Use First Row as Headers**.

With that step, Power BI will predict the data type of each field again.

Notice that the column **Zip** was changed to the number data type. Let's change it to text as we did earlier. If we don't, we will see errors when we load the data.

38. Click **123** next to the Zip Column. From the dialog box, click **Text**.
39. Click **Replace Current** in the **Change Column Type** dialog box.



40. From the left panel, click the **Manufacturer** query. Notice the bottom three rows are not part of the data. Let's remove them.

41. From the ribbon, click **Home**, click **Remove Rows**, and then click **Remove Bottom Rows**.

42. The Remove Bottom Rows dialog box opens. Enter 3 in the Number of rows text box.

43. Click OK.

The screenshot shows the Power BI Desktop interface. The ribbon is at the top with 'Home' selected. In the center, there's a 'Transform' tab with various tools like 'Transpose', 'Unpivot Columns', and 'Pivot Column'. Below the ribbon is a 'Queries [9]' pane on the left, which lists 'Transform File from InternationalSales [3]', 'Other Queries [5]' (including Sales, Product, Geography, Manufacturer, International Sales), and 'Table.TransformColumn1'. The main area shows a table with columns 'Column1' and 'Column2'. The 'Manufacturer' query is highlighted with an orange box. A 'Remove Bottom Rows' dialog box is open over the table, with the 'Number of rows' input field set to '3'. The 'OK' button is highlighted with an orange box.

## Power BI Desktop – Transposing data

44. From the left panel, click the **Manufacturer** Query. Notice that the **ManufacturerID**, **Manufacturer**, and **Logo** data is laid across in rows. Also notice that the header is not useful. We need to transpose the table to meet our needs.

45. From the ribbon click **Transform** and then click **Transpose**.

The screenshot shows the Power BI Desktop interface with the 'Transform' ribbon tab selected. The 'Transpose' icon is highlighted with an orange box. The 'Queries [9]' pane on the left shows the 'Manufacturer' query highlighted with an orange box. The main area displays a table with columns 'Column1' and 'Column2'. The data in 'Column1' includes 'ManufacturerID', 'Manufacturer', and 'Logo'. The data in 'Column2' includes 'Abbas' and a URL. The 'Transpose' tool has been applied, changing the row-based data into column-based data.

Notice that this transposes the data into columns. Now we need the first row to be the header.

46. From the ribbon click **Home** and then click **Use First Row as Headers**.

Notice that now the **Manufacturer** table is laid out the way we need it with a header and values along columns.

Also notice that on the right panel under **APPLIED STEPS** you will see the list of transformations and steps that have been applied. You can navigate through each change made to the data by clicking on the step. Steps can also be deleted by clicking on the X that appears to the left of the step. The properties of each step can be reviewed by clicking on the gear to the right of the step.

### Power BI Desktop – Appending queries

To analyze the Sales of all countries, it is convenient to have a single **Sales** table. To do this, you need to append all the rows from the **International Sales** query to the **Sales** query.

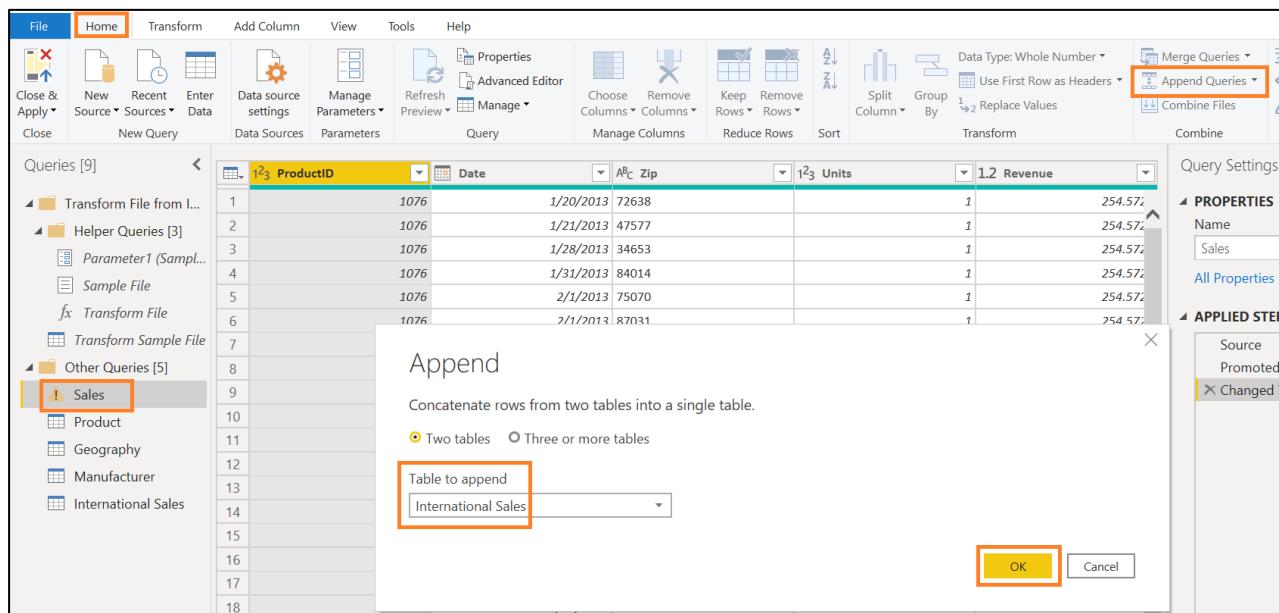
47. Click **Sales** in the Queries window in the left panel as shown above.

48. From the ribbon click **Home** and then click **Append Queries**.

The **Append** dialog box opens. There is an option to append **Two tables** or **Three or more tables**.

Leave **Two tables** selected since we are appending just two tables.

49. Click **International Sales** from the drop-down and then click **OK**.



You will now see a new column in the **Sales** table called **Country**. Since the International **Sales** query had the additional column for **Country**, Power BI Desktop added the column to the **Sales** table when it loaded the values from the **International Sales** query.

You will see **null values** in the **Country** column by default for the **Sales** table rows because that column did not exist for the table with USA data. We will now add the value “**USA**” as a data shaping operation.

#### 50. From the ribbon click **Add Column** and then click **Conditional Column**.

The screenshot shows the Power BI Desktop ribbon with the 'Add Column' tab selected. Within the 'Conditional Column' dropdown, the 'Conditional Column' option is highlighted with a red box. The 'Queries [9]' pane on the left shows the 'Sales' query selected. The main area displays a table with columns: Zip, Units, Revenue, and Country. The 'Country' column is highlighted with a yellow box. A formula bar above the table shows the formula: = Table.Combine({#"Changed Type", #"International Sales"}). The 'OK' button at the bottom right of the dialog is highlighted with a yellow box.

51. In the **Add Conditional Column** dialog box, enter the name of the column as “**CountryName**”.

52. Click **Country** from the **Column Name** drop-down menu.

53. Click **equals** from the **Operator** drop-down menu.

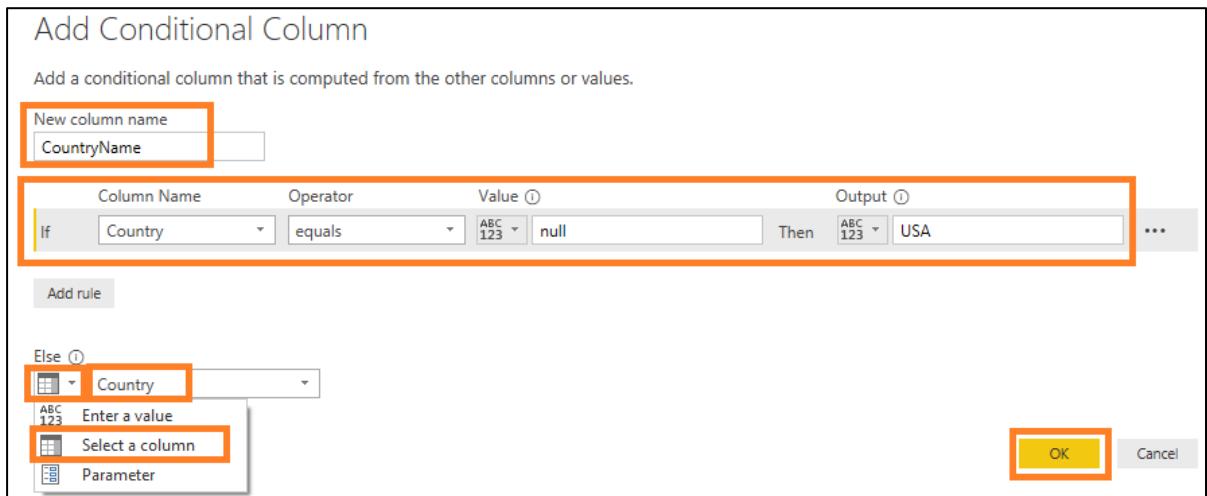
54. Enter **null** in the **Values** text.

55. Enter **USA** in the **Output** text.

56. Click the drop-down menu under **Else** and then click the **Select a column** option.

57. Click **Country** from the column drop-down menu.

58. Click **OK**.



This reads: if current Country value equals null then the value should be USA otherwise use the current Country value

59. You will see the **CountryName** column in the Query editor window.

The screenshot shows the Power Query Editor interface. On the left, the 'Queries [9]' pane shows a tree structure with 'Sales' selected. In the main area, a table is displayed with columns: '1', '2', '3', 'Units', '1.2 Revenue', 'Source.Name', 'Country', and 'CountryName'. The 'CountryName' column contains values like 'null', 'USA', 'null', etc. On the right, the 'Query Settings' pane shows 'APPLIED STEPS' with a step labeled 'Added Conditional Column'. The 'Country' column header is highlighted with an orange border.

The original **Country** column is only required as a temporary column. It is not required in the final table for analysis and can be removed.

60. Right-click on the **Country** column and click **Remove** as shown in the figure.

The screenshot shows the Power Query Editor interface. The 'Country' column header is highlighted with an orange border. A context menu is open over the column, with the 'Remove' option highlighted with an orange border. Other options in the menu include 'Remove Other Columns', 'Duplicate Column', 'Add Column From Examples...', 'Remove Duplicates', 'Remove Errors', and 'Change Type'.

We can now rename the **CountryName** column to **Country**.

61. Right-click on the **CountryName** column and rename it to **Country**.

62. Using **Home** then **Data Type** or by selecting the data type next to the column header, change the **data type** of the **Country** column to **Text**.

63. Using **Home** then **Data Type** or by selecting the data type next to the column header, change the **data type** of the **Revenue** column to **Fixed Decimal Number** because it is a currency field.

When the data is refreshed, it will process through all the “Applied Steps” that you have created.

The newly named **Country** column will have names for all countries, including the USA. You can validate this by clicking on the drop-down menu next to the **Country** column to see the unique values.

64. At first, you will only see USA data. Click **Load more** to validate you have data from all seven countries.

65. Click **OK** to close this filter.

The screenshot shows the Microsoft Power BI Data Editor interface. On the left, the 'Queries [9]' pane lists 'Transform File from International...', 'Helper Queries [3]', 'Other Queries [5]', and the 'Sales' query, which is currently selected and highlighted with a red box. In the main area, a table view shows columns 'Apc Zip', 'Revenue', and 'Country'. The 'Country' column has a dropdown arrow icon, and its dropdown menu is open, also highlighted with a red box. The menu lists several countries with checkboxes: '(Select All)', '(blank)', Australia, Canada, Germany, Japan, Mexico, Nigeria, and USA, with 'USA' checked. On the right side, there are 'Query Settings' and 'APPLIED STEPS' panes. The 'APPLIED STEPS' pane shows a list of steps: Source, Promoted Headers, Changed Type, Appended Query, Added Conditional ..., Renamed Columns, Removed Columns, Renamed Columns1, Changed Type1, Filtered Rows, and 'Filtered Rows1' (which is expanded to show it's applied to the 'Sales' query). Buttons 'OK' and 'Cancel' are at the bottom right of the dropdown menu.

Typically, when exploring data, we load a subset of data. There are multiple ways to do this. From the ribbon, you can click **Home**, click **Keep Rows**, and then click **Keep Top Rows**. Another way to load a subset of data is to navigate to **Home**, then click **Keep Rows** and then click **Keep Bottom Rows**. A third method is to click **Home**, click **Keep Rows**, and then click **Keep Range of Rows**. You can use any of these options to filter down to a subset of data.

Our dataset has data from 2013 to 2020. For our analysis we want to start with the last three years of data (2017-2020). We don't yet know how many rows will result. We can filter by year to get the subset.

66. Click the arrow next to **Date** in the **Sales** Query.

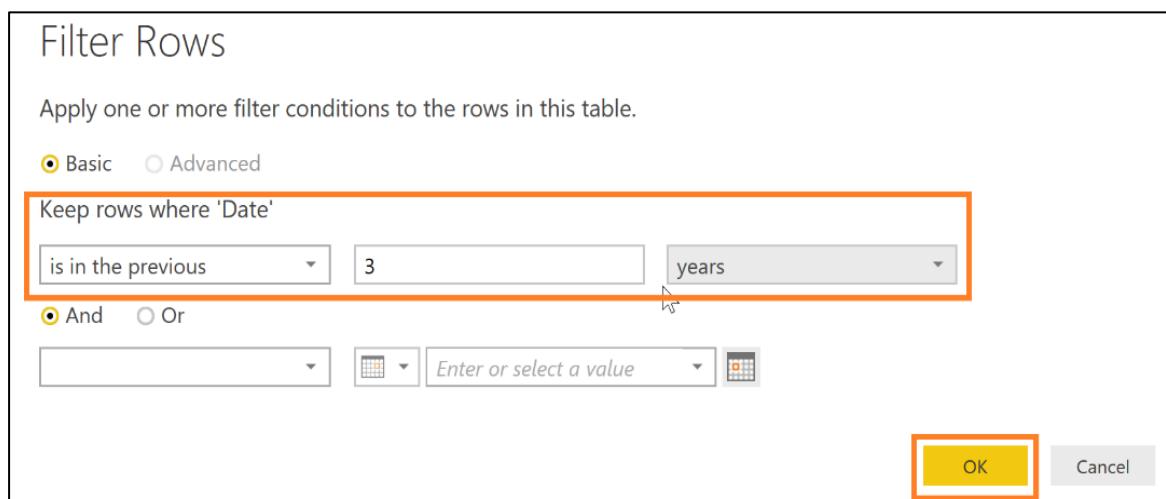
67. Click **Date Filters** and then click **In the Previous...**

The screenshot shows the Power BI Data Editor interface. On the left, the 'Queries [9]' pane is open, showing a hierarchy of queries. The 'Sales' query is selected and highlighted with a red box. In the main area, the 'Date' column of the 'Sales' table is selected and highlighted with a yellow box. A context menu is open over the 'Date' column, with various options like 'Sort Ascending', 'Sort Descending', 'Clear Sort', 'Clear Filter', 'Remove Empty', and 'Date Filters'. The 'Date Filters' option is highlighted with a red box. A sub-menu for 'Date Filters' is open, showing options like 'Equals...', 'Before...', 'After...', 'Between...', 'In the Next...', 'In the Previous...', 'Is Earliest', 'Is Latest', and 'Is Not Earliest'. The 'In the Previous...' option is also highlighted with a red box.

68. The **Filter Rows** dialog box opens. Enter **3** in the text box next to **is in the previous**.

69. Click **years** from the drop-down menu.

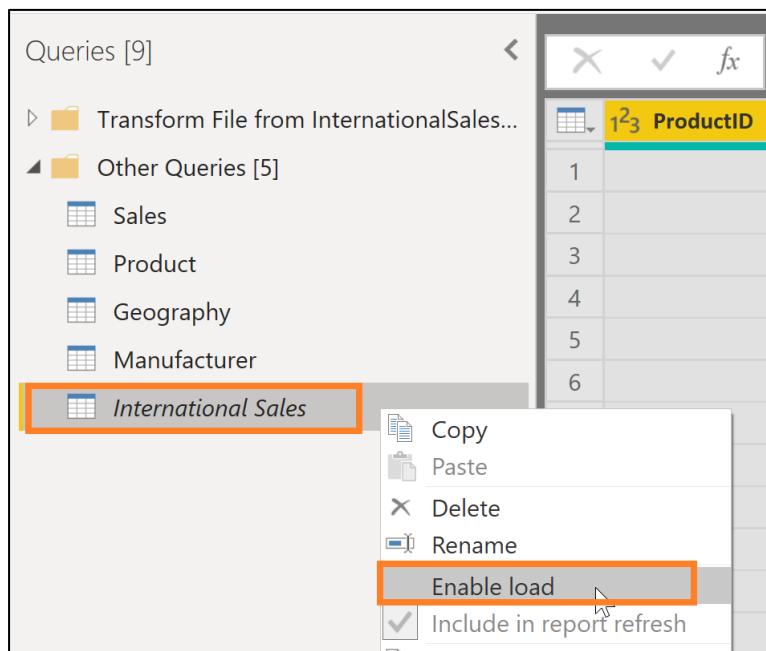
70. Click **OK**.



Now that the International Sales data is appended to the Sales query, we don't need the International Sales table to load into the data model. Let's prevent the International Sales table from loading into the data model.

71. From the Queries panel on the left, click the **International Sales** query.

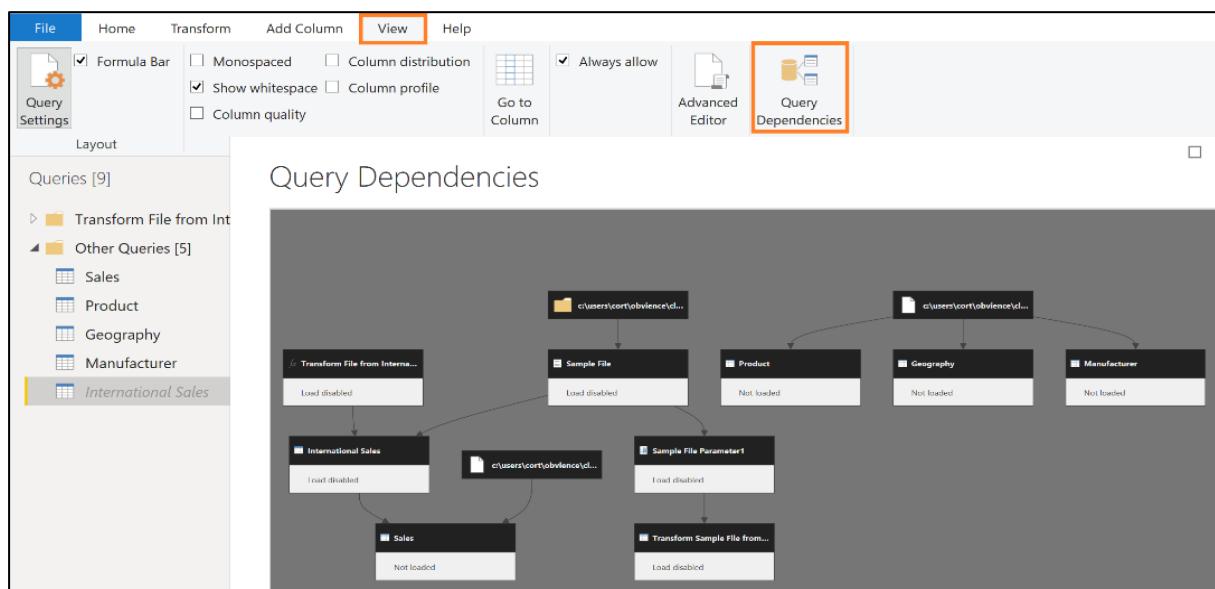
72. Right-click and then click **Enable Load**. This will disable loading International Sales.



**Note:** The appropriate data from the International Sales table will load into the Sales table each time the model is refreshed. By removing the International Sales table, we are preventing duplicate data from loading into the model and increasing its file size. In some instances, storing very large amounts of data affects the data model performance.

### 73. From the ribbon click **View** and then click **Query Dependencies**.

This opens the **Query Dependencies** dialog box. The dialog box shows the source of each query and its dependencies. For example, we see that the Sales query has a CSV file source and a dependency on the International Sales query. This is a useful information to share knowledge with your team members.

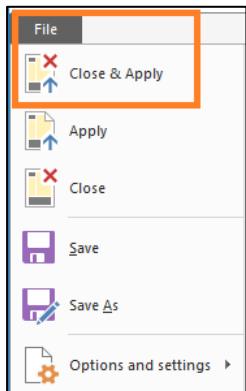


### 74. Click **Close** in the dialog box.

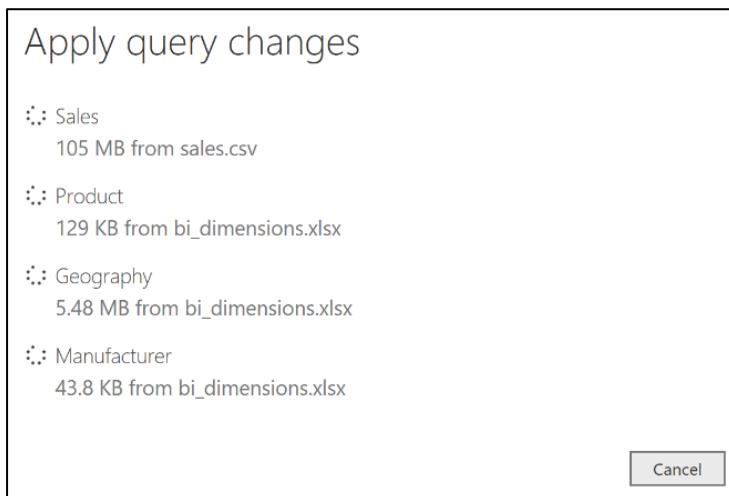
Note that you can zoom in and out of the **Query Dependencies** view as needed.

You have now successfully completed import and data shaping operations and are ready to load the data into the Power BI Desktop data model to visualize the data.

75. Click **File** and then click **Close & Apply**. This will close out the power query window and apply all changes



All the data will be loaded in memory in the Power BI Desktop. You will see the progress dialog box with the number of rows being loaded in each table as shown in the Figure.

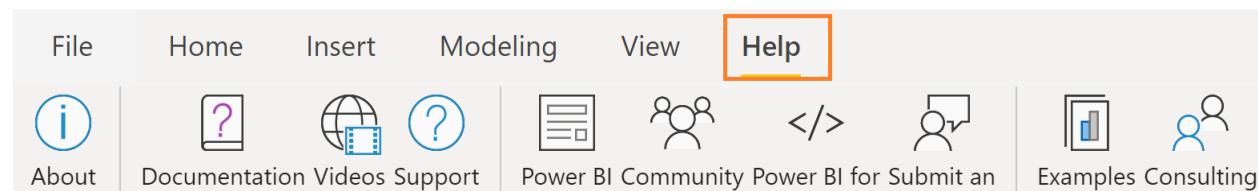


**Note:** It may take several minutes to load all the tables.

76. Click **File** and then click **Save** to save the file after the data loading is complete. Name the file as “**MyFirstPowerBIModel**”. Save the file in the DIAD Reports (**\DIAD\Reports**) folder.

# References

Dashboard in a Day introduces you to some of the key functions available in Power BI. In the ribbon of the Power BI Desktop, the Help section has links to some great resources.



Here are a few more resources that will help you with your next steps with Power BI.

- Getting started: <http://powerbi.com>
- Power BI Desktop: <https://powerbi.microsoft.com/desktop>
- Power BI Mobile: <https://powerbi.microsoft.com/mobile>
- Community site <https://community.powerbi.com/>
- Power BI Getting started support page:  
<https://support.powerbi.com/knowledgebase/articles/430814-get-started-with-power-bi>
- Support site <https://support.powerbi.com/>
- Feature requests <https://ideas.powerbi.com/forums/265200-power-bi-ideas>
- New ideas for using Power BI [https://aka.ms/PBI\\_Comm\\_Ideas](https://aka.ms/PBI_Comm_Ideas)
- Power BI Courses <http://aka.ms/pbi-create-reports>

© 2020 Microsoft Corporation. All rights reserved.

By using this demo/lab, you agree to the following terms:

The technology/functionality described in this demo/lab is provided by Microsoft Corporation for purposes of obtaining your feedback and to provide you with a learning experience. You may only use the demo/lab to evaluate such technology features and functionality and provide feedback to Microsoft. You may not use it for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this demo/lab or any portion thereof.

COPYING OR REPRODUCTION OF THE DEMO/LAB (OR ANY PORTION OF IT) TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION IS EXPRESSLY PROHIBITED.

THIS DEMO/LAB PROVIDES CERTAIN SOFTWARE TECHNOLOGY/PRODUCT FEATURES AND FUNCTIONALITY, INCLUDING POTENTIAL NEW FEATURES AND CONCEPTS, IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION FOR THE PURPOSE DESCRIBED ABOVE. THE TECHNOLOGY/CONCEPTS REPRESENTED IN THIS DEMO/LAB MAY NOT REPRESENT FULL FEATURE FUNCTIONALITY AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. WE ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH FEATURES OR CONCEPTS. YOUR EXPERIENCE WITH USING SUCH FEATURES AND FUNCTIONALITY IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

**FEEDBACK.** If you give feedback about the technology features, functionality and/or concepts described in this demo/lab to Microsoft, you give to Microsoft, without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your feedback in them. These rights survive this agreement.

MICROSOFT CORPORATION HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE DEMO/LAB, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, OUTPUT THAT DERIVES FROM USE OF DEMO/ LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE DEMO/LAB FOR ANY PURPOSE.

## **DISCLAIMER**

This demo/lab contains only a portion of new features and enhancements in Microsoft Power BI. Some of the features might change in future releases of the product. In this demo/lab, you will learn about some, but not all, new features.