



# Power BI

## Data Dash using Power BI

### Day 1- Power Query

# Introduction

Today you will learn about various key features of the Power BI service. This is an introductory course intended to teach you how the author builds reports using the 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 modeling exercises in Lab 2. Additionally, the results of this lab will be the starting point for 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

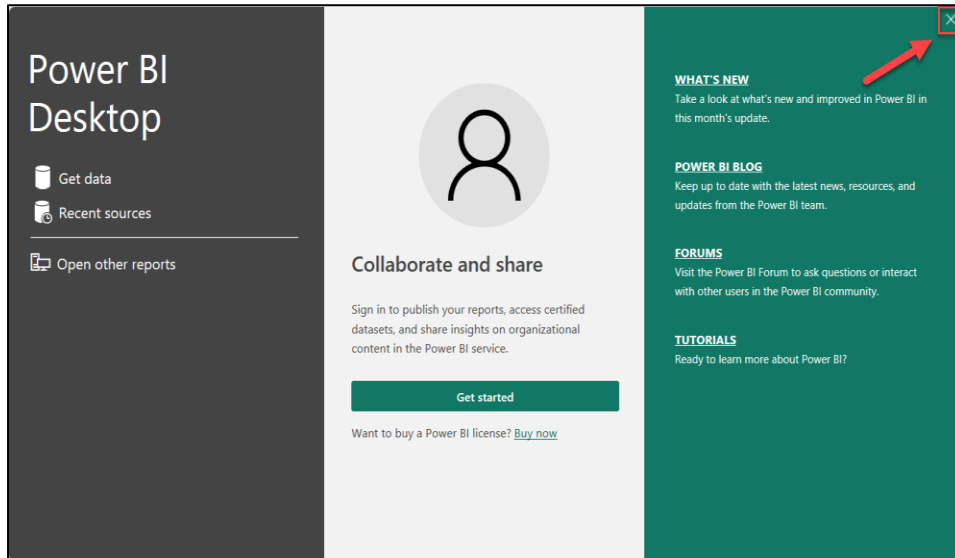
The dataset for this course contains sales data of VanArsdel and other competitors. We have seven years of transaction data by day, product, and zip code for each manufacturer. We are going to analyze data from seven countries.

USA sales data is in a CSV file located in the Usages 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.

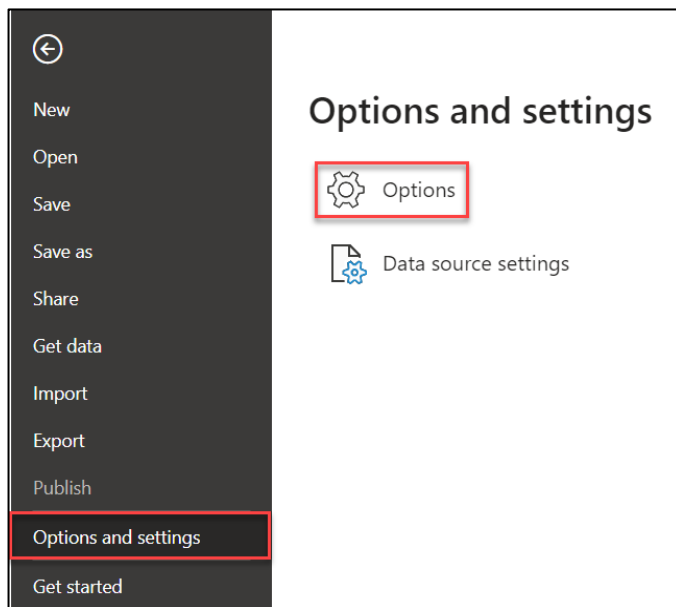
Product, Geography, and Manufacturer information is in a Microsoft Excel file called bi\_dimensions.xlsx in the USSales subfolder within the Data folder (/Data/USSales/).

1. If you don't already have the **Power BI Desktop** open, launch it now.
2. If you have not signed into the **Power BI Desktop**, select the **Get started** option.
3. **Sign in** using your Power BI credentials.
4. You will see the startup screen open. Select the **X** in the top right corner of the dialog box to close it.

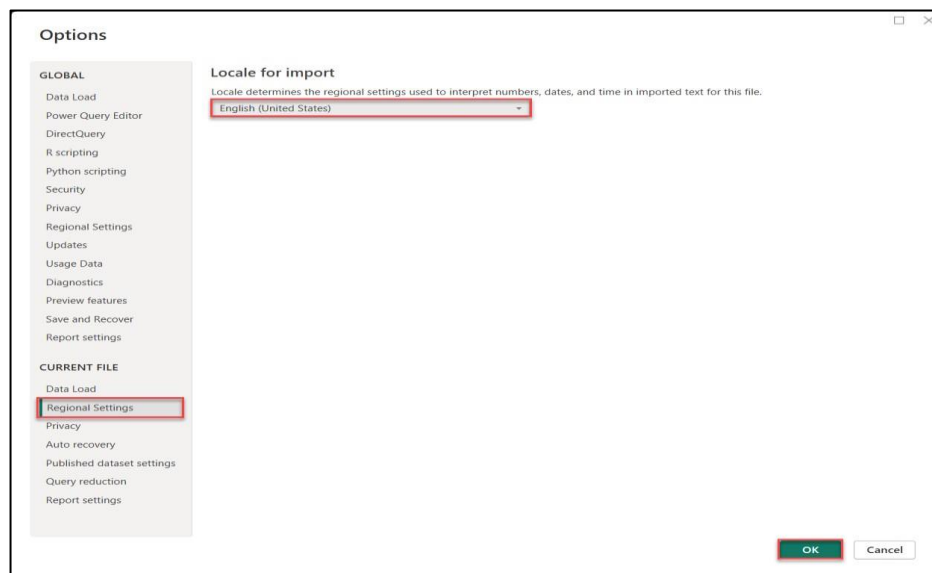


Next, let's set the **Locale** to US English to make it convenient in the rest of this lab.

5. From the ribbon, select **File**, then choose **Options and settings**. Then, select **Options**.



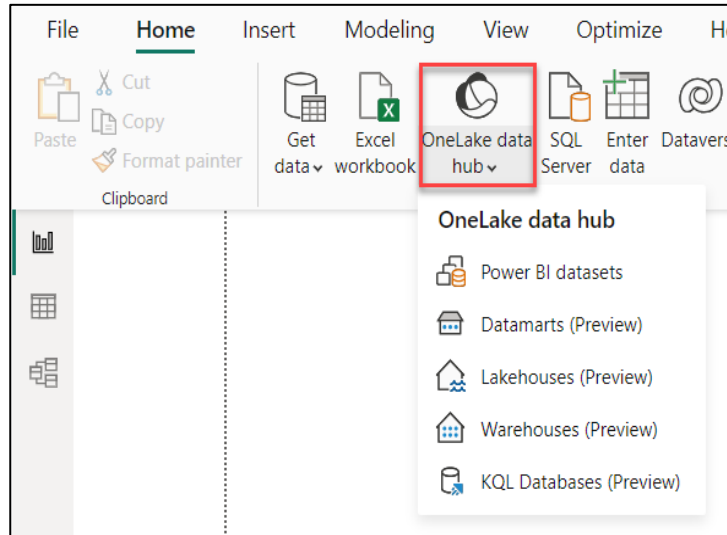
6. Within the pane to the left of the **Options** dialog box, select **Regional Settings** under **Current File**.
7. From the **Locale** drop-down, select **English (United States)**.
8. Then, select **OK** to close the dialog box.



The next step is to load data into the **Power BI Desktop**.

**Note:** Power BI Desktop has the capability to connect to 300+ data sources. The newest sources are part of Microsoft Fabric's OneLake data hub. You will not be using OneLake in today's class but to learn more read here:

<https://learn.microsoft.com/en-us/power-bi/fundamentals/fabric-get-started>

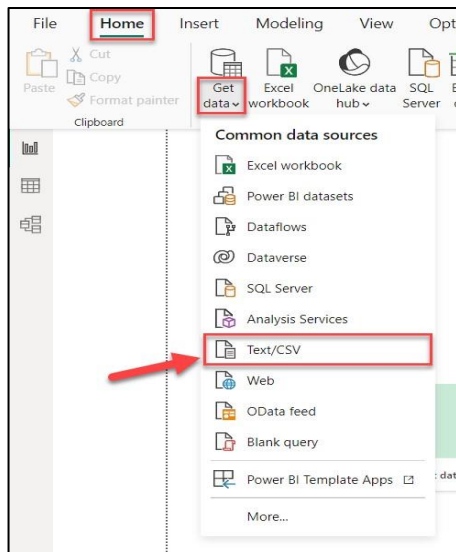


We are using CSV and Excel data files in this lab for simplicity. If you would like a full list of data sources, please visit this link: <https://docs.microsoft.com/en-us/power-bi/connect-data/desktop-data-sources>

Start by loading **USA Sales data**, which is in a CSV file.

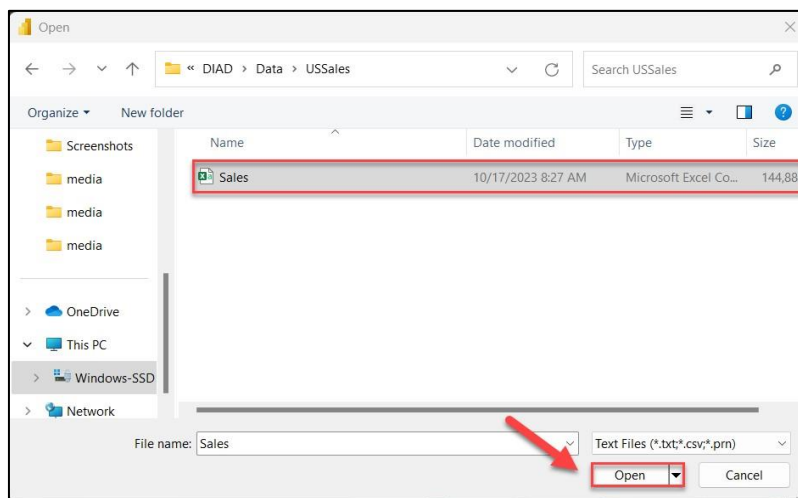
9. From the ribbon at the top of the screen, select the **Home** tab. Then, choose the **Get Data** drop-down (not the icon).

10. Select **Text/CSV** from the options list.



11. Browse to the **DIAD** folder (this folder may be called **diad-student-english** if you did not rename it in Lab 0), double-click **Data**, double-click the **USSales** folder, and then select the **Sales.csv** file.

12. Then, select the **Open** button.



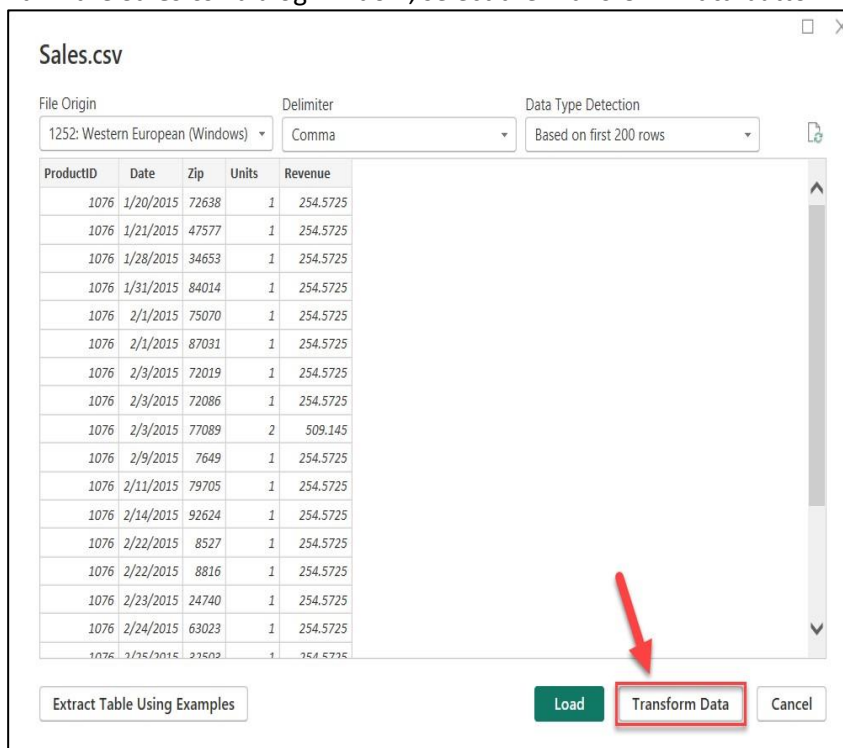
**Note:** If your folder appears empty then this likely means you forgot to unzip your class files. Navigate to your location where you saved the class files and unzip the files by right-clicking on the .zip file, then select **Extract All**.

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, Transform Data or Cancel.

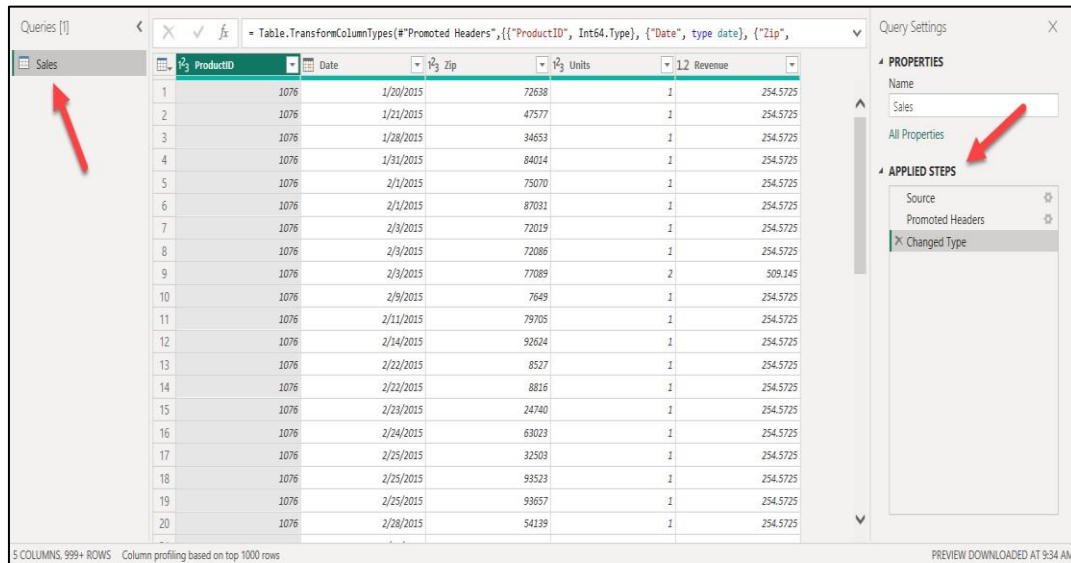
- **Load** adds the data from the source into Power BI Desktop for you to start creating reports.
- **Transform Data** allows you to perform data shaping operations such as merging columns, adding additional columns, changing data types of columns as well as bringing in additional data.
- **Cancel** returns you back to the main canvas.

12. Within the **Sales.csv** dialog window, select the **Transform Data** button.

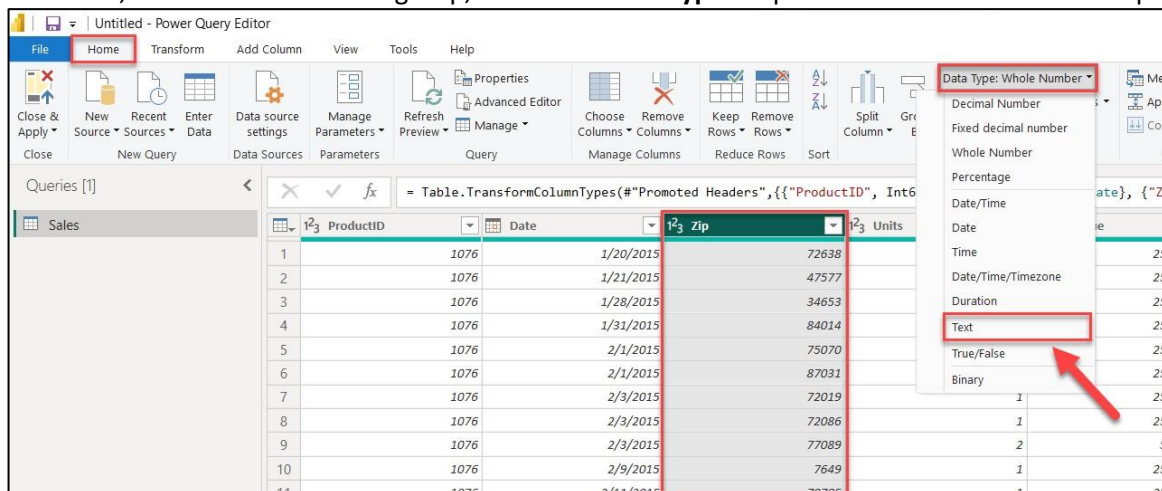


You should be in the Query Editor window as shown in the screenshot below. The Query Editor is used to perform data shaping operations. Notice that the sales file you connected shows as a query in the pane to the left of the screen. You can see a preview of the data in the center pane. Power BI predicts the data type of each field (based on the first 200 rows) as indicated by the icons to the right of each column

header. Within the pane to the right of the screen, steps that the Query Editor performs are recorded within the **APPLIED STEPS** section.



13. Notice that Power BI has set the **Zip** column 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, select the **Home** tab. From the menu at the top of the screen, within the **Transform** group, select the **Data Type** drop-down and choose the **Text** option.



14. The **Change Column Type** notification box opens. Select the **Replace current** button which overwrites Power BI's predicted data type.



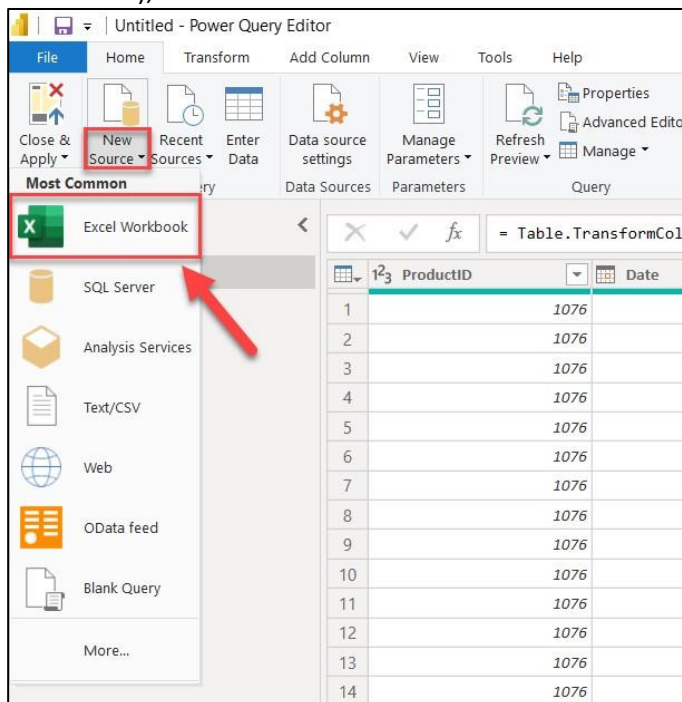


### IMPORTANT!

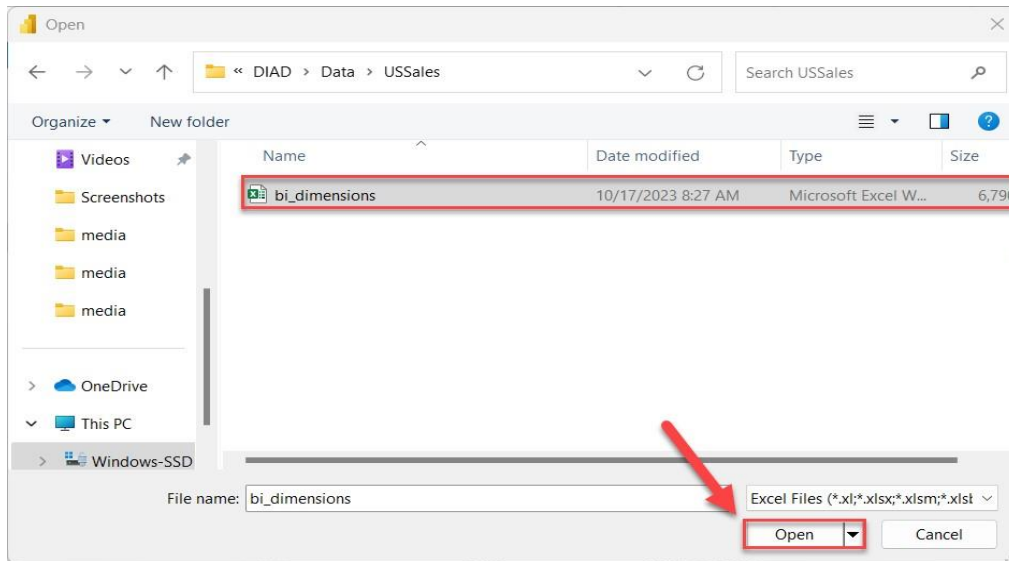
Missing these last two steps can impact your experience later.

Now, let's get the data that is in the Excel source file called **bi\_dimensions.xlsx**.

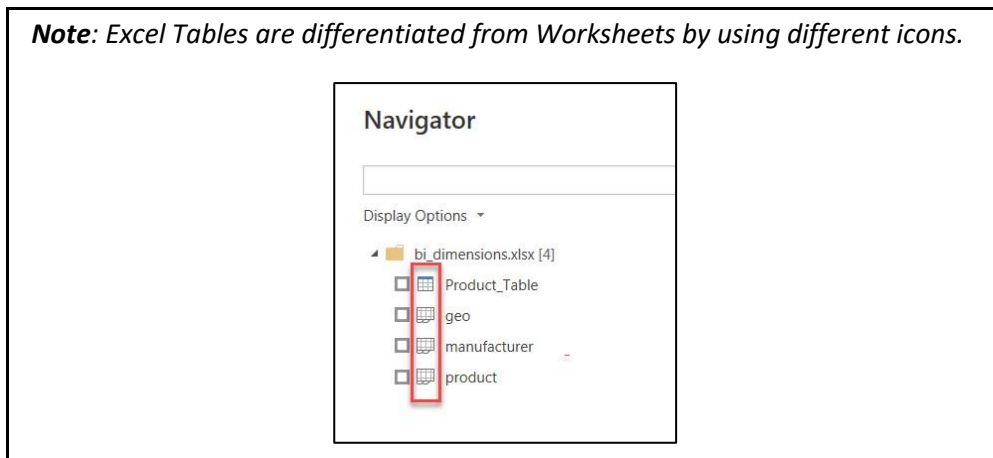
15. From the ribbon at the top of the screen, select the **Home** tab, choose the **New Source** drop-down (not the icon), and then select **Excel Workbook**.



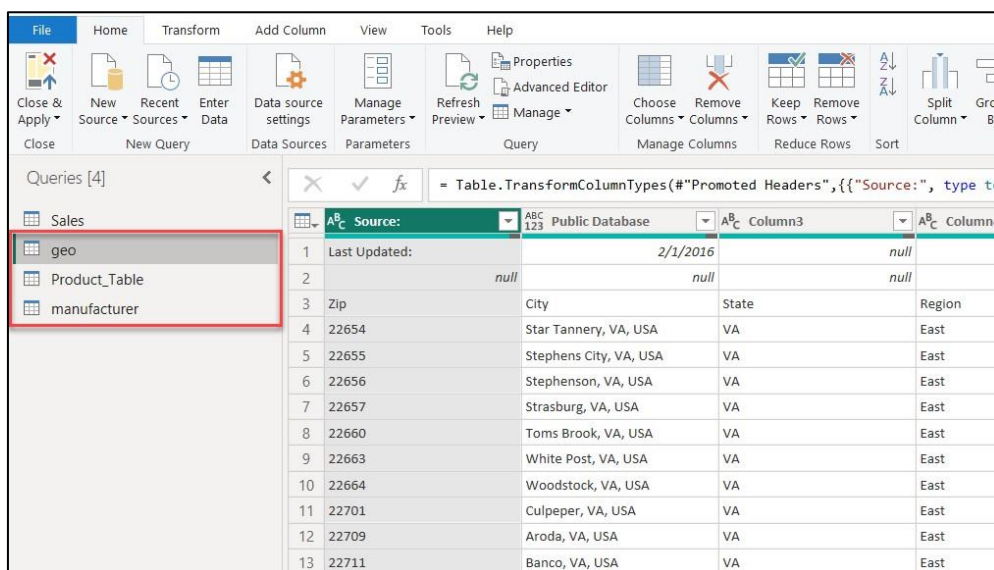
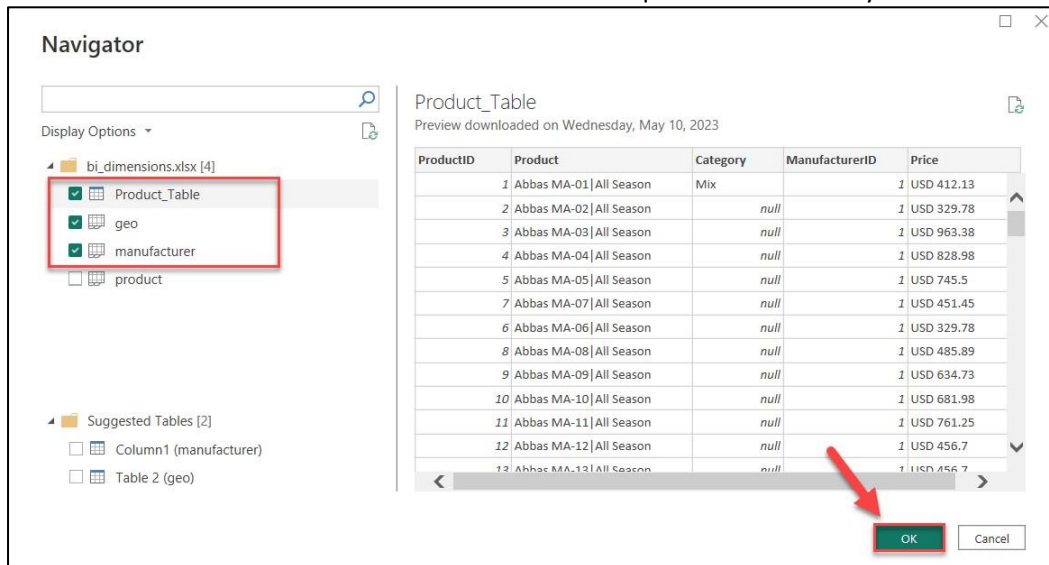
16. Browse to the **DIAD** folder, double-click **Data**, double-click the **USSales** folder, and then select **bi\_dimensions.xlsx**.
17. Then, select the **Open** button. The **Navigator** dialog box will appear.



The **Navigator** dialog will open. Within the list to the left of the dialog, you will see three sheets listed that are within the Excel workbook. It also lists the **Product\_Table** which is a pre-defined Excel table.



18. From the list to the left of the dialog, select the checkbox for **geo**. In the preview pane, notice that the first few rows are headers and are not part of the data. We will remove them shortly.
19. Select the checkbox for **manufacturer**. In the preview pane, notice that the last couple of rows are footers and are not part of the data. We will remove them shortly.
20. Select the checkbox for **Product\_Table**. Notice the different icon indicates that this data is stored in an Excel table.
21. Make sure that **Product\_Table**, **geo** and **manufacturer** are selected within the pane to the left, and then select **OK**. Notice that three sheets are added as queries in the Query Editor



## Power BI Desktop – Adding additional data

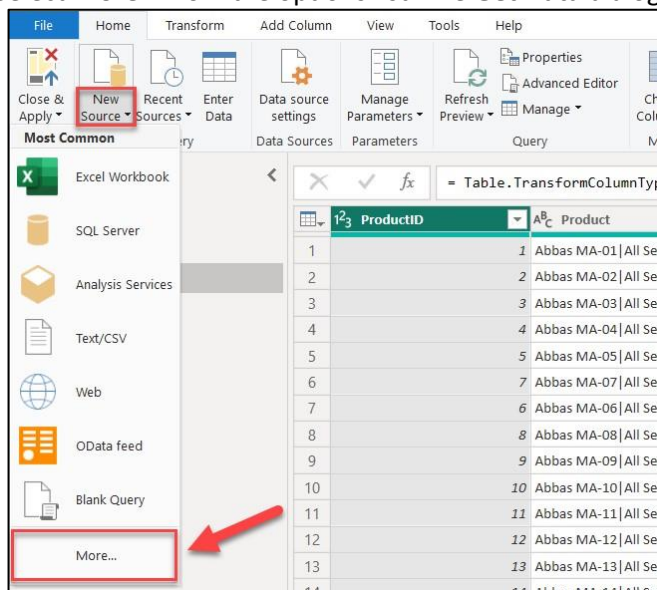
In this scenario, the international subsidiaries have agreed to provide their sales data so that the company's sales can be analyzed together. You've created a folder where they each put their data.

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

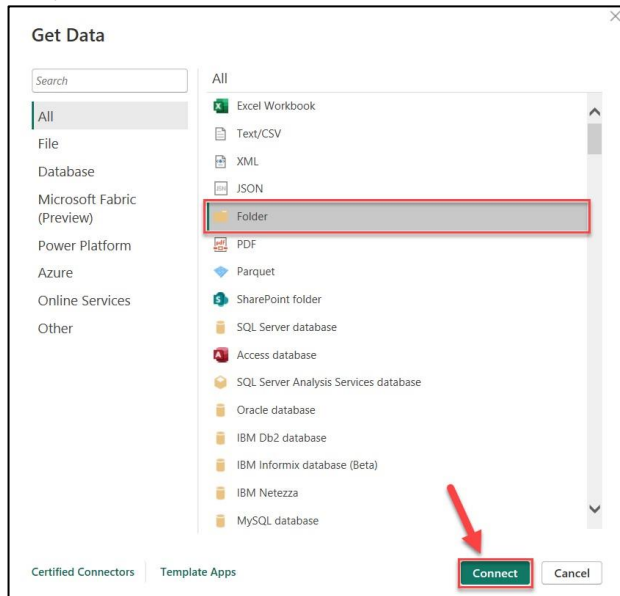
You can load the files one at a time, like you loaded the US Sales data, but Power BI provides an easier way to load all the files in a folder together at once.

22. From the **Home** tab of the Query Editor, select the **New Source** drop-down (*not the icon*).

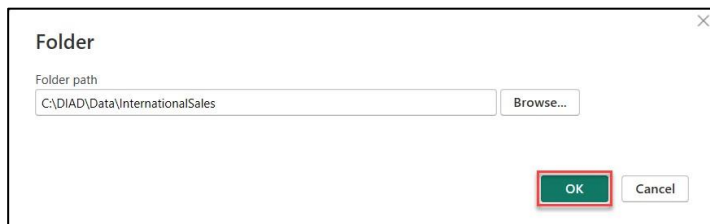
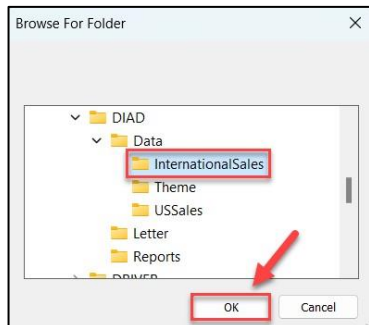
23. Select **More...** from the options list. The **Get Data** dialog will open.



24. Within the **Get Data** dialog box, select **Folder** from the **All** list.
25. Then, select the **Connect** button and the **Folder** dialog box will open.



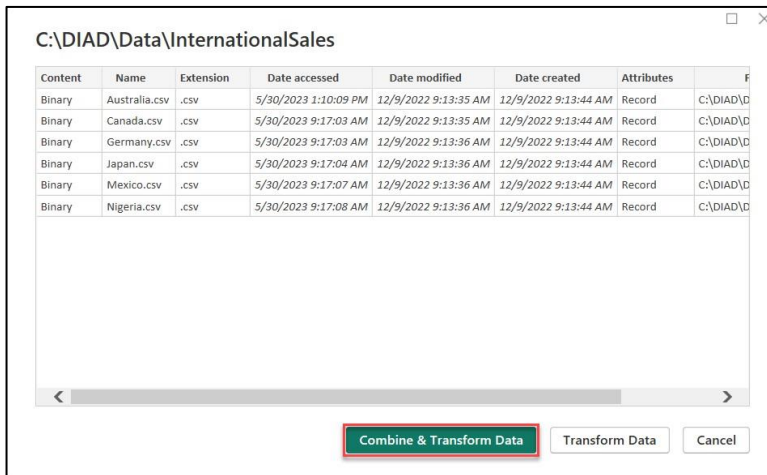
26. Within the Folder dialog box, select the **Browse...** button.
27. In the **Browse For Folder** dialog, navigate to the location where you **unzipped** the class files.
28. Open the **DIAD** folder, then open the **Data** folder. Select the **InternationalSales** folder.
29. Select **OK** to close the **Browse for Folder** dialog box. Then, select **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 selected folder dialog box will display the list of files within the folder.

30. Select the **Combine & Transform Data** button at the bottom of the dialog box.

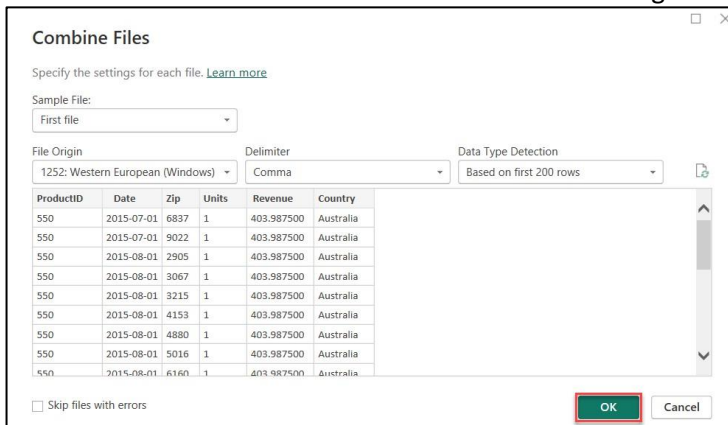


**Note:** The data in your file for **Date accessed**, **Date modified**, and **Date created** might be different than the dates displayed in the screenshot above.

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 default **Delimiter** option as **Comma**.

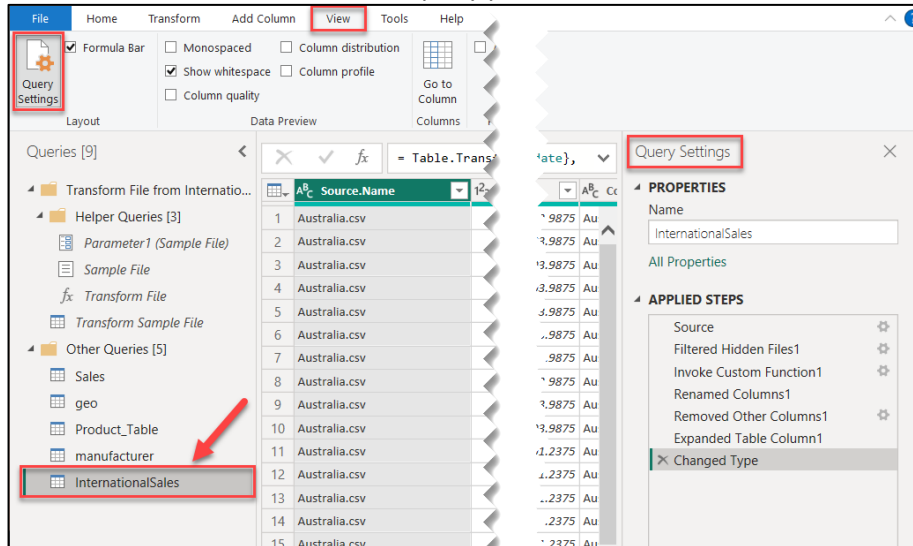
There is also an option to select each individual file in the folder (using **Example File** drop-down) to validate the format of the files.

31. Select the **OK** button located at the bottom of the dialog window.



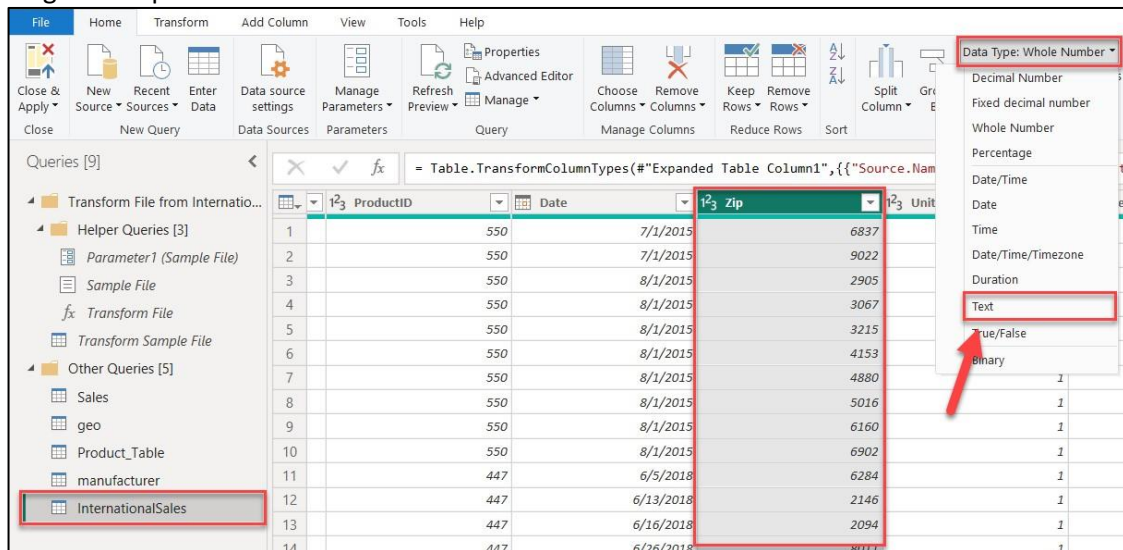
You will now be in the **Power Query Editor** window with a new query named **InternationalSales**.

32. If you do not see the **Queries** pane to the left of the screen, select the > (greater than) icon to expand and now view the Queries pane.
33. If you do not see the **Query Settings** pane on the right of the screen, select the **View** tab in the ribbon and choose **Query Settings** to view the pane.
34. Select **InternationalSales** from the query pane on the left.



Notice that the **Zip** column 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**.

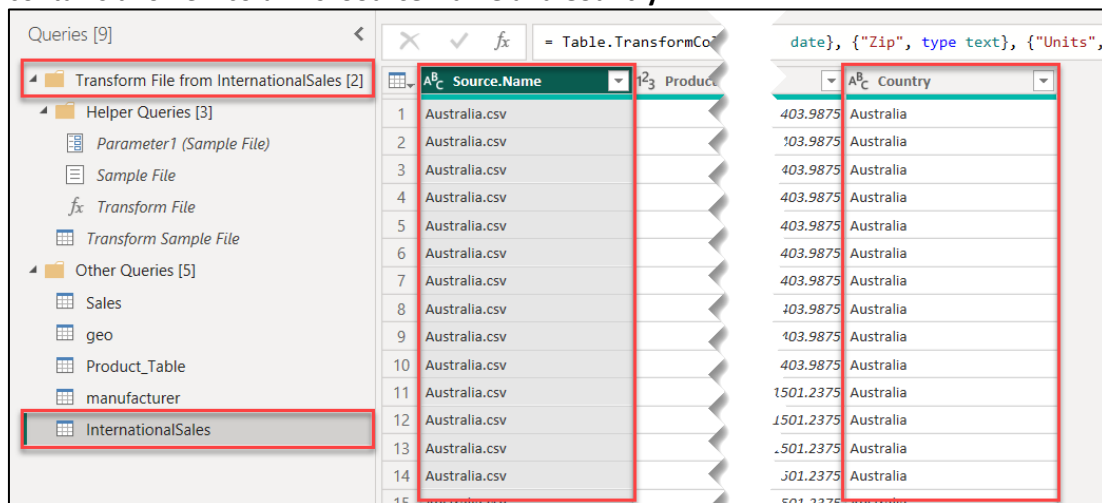
35. Select the **Zip** column within the **InternationalSales** query, and then change the **Data Type** to **Text** using the drop-down under the **Home** tab.



36. The **Change Column Type** dialog box will open. Select the **Replace Current** button when prompted.

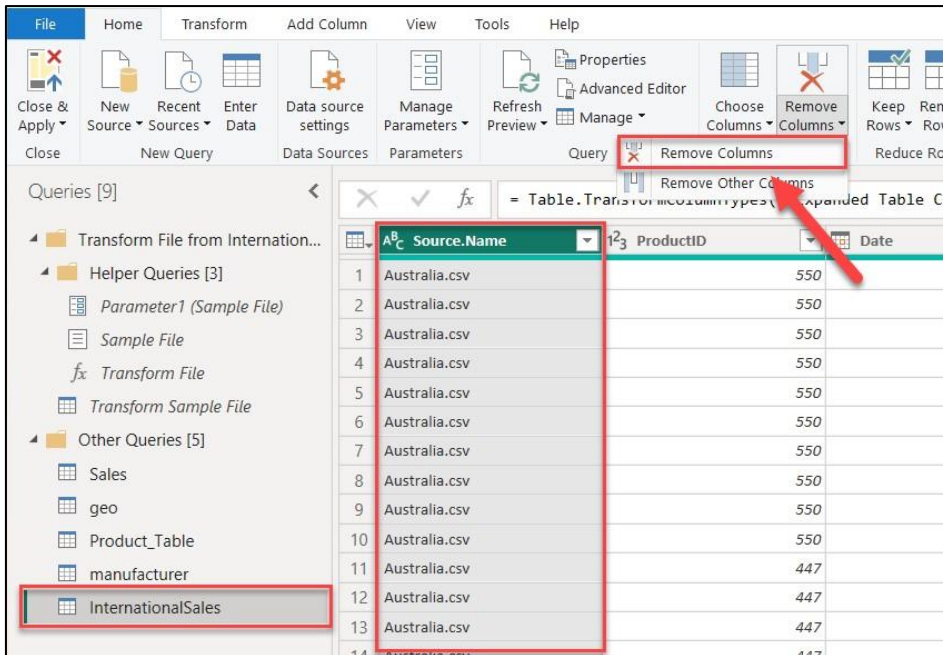
Within the **Queries** pane, notice that a **Transform File from the InternationalSales** folder is created. This contains the function used to load each of the files from the folder.

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

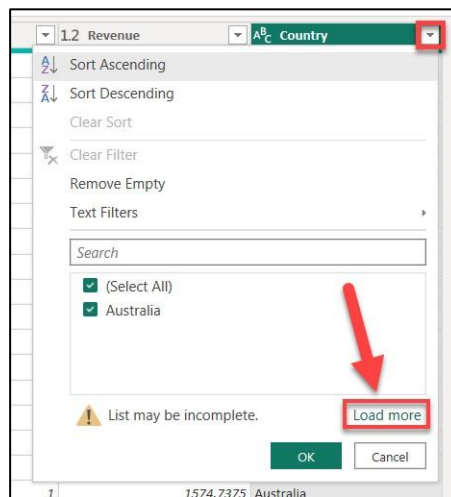


37. We do not need the **Source.Name** column within the **InternationalSales** query. Select the **Source.Name** column and from the ribbon, select the **Home** tab. Choose the **Remove Columns** dropdown (*not the icon*), and then select **Remove Columns** again.

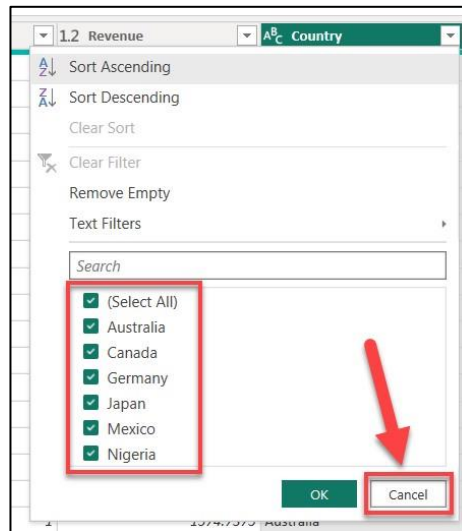




**Note:** You may find that Australia is the only country displayed. This is due to the **Power Query Editor** displaying only the first 1000 rows of any data source. To validate you have the data from all country files you can optionally select the drop-down menu next to the **Country** column, then select **Load more**.



You will now see that **Australia, Canada, Germany, Japan, Mexico, and Nigeria** are all selected.



If you did this optional step, select **Cancel**.

## Power BI Desktop – Renaming tables

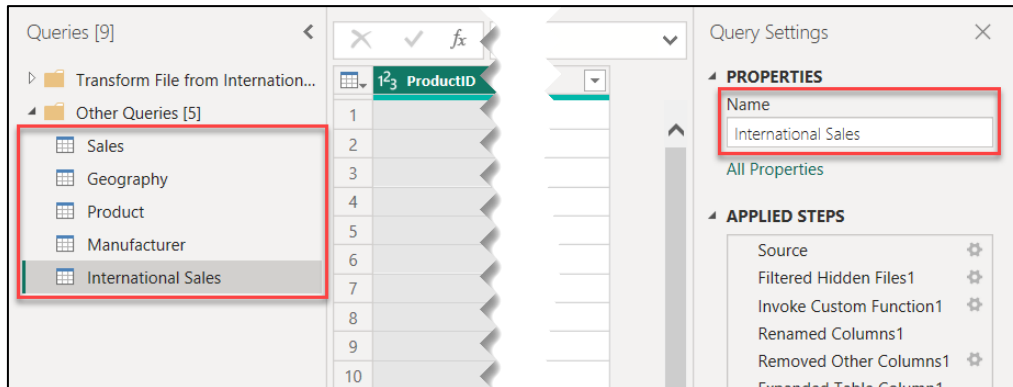
The Query Editor window should appear as shown below.

- 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 within the ribbons.
- Select the options available within the ribbon - **Home**, **Transform**, **Add Column**, and **View** - to review the various features available.

38. Within the **Queries** pane, minimize the folder called **Transform Files from InternationalSales**.

39. Next, **rename** the queries listed within the **Queries** pane. Using the text field within the **Properties** section of the **Query Settings** pane, use the new names listed below to change the name of each of the queries listed. After entering the new name within the text field, hit **Enter** on your keyboard to save the new name of the query.

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

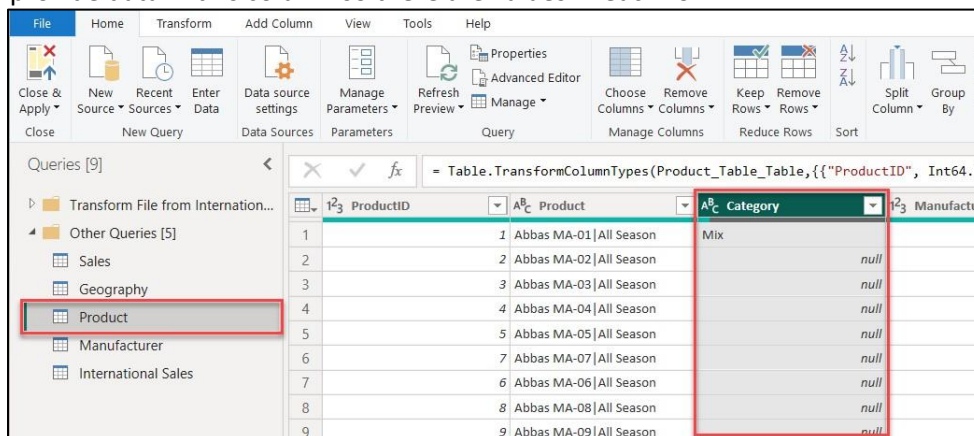


**Note:** It is a best practice to provide descriptive query and column names. These names are used in visuals and in the Q&A section of Power BI, which is covered in a later lab.

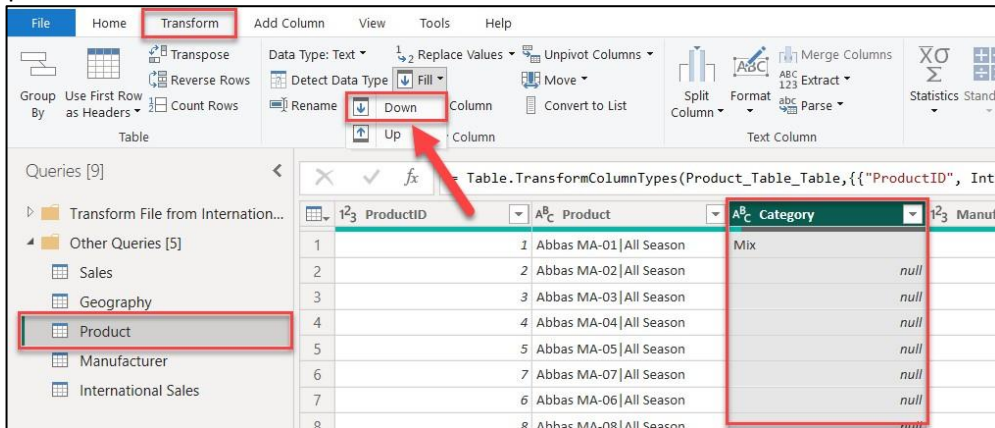
## Power BI Desktop – Filling empty values

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 by selecting the **Product** query from the **Queries** pane.

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.



40. With the **Product** query selected from the **Queries** pane, select the **Category** column.
41. From the ribbon, select the **Transform** tab, choose the **Fill** drop-down, and then select the **Down** option.



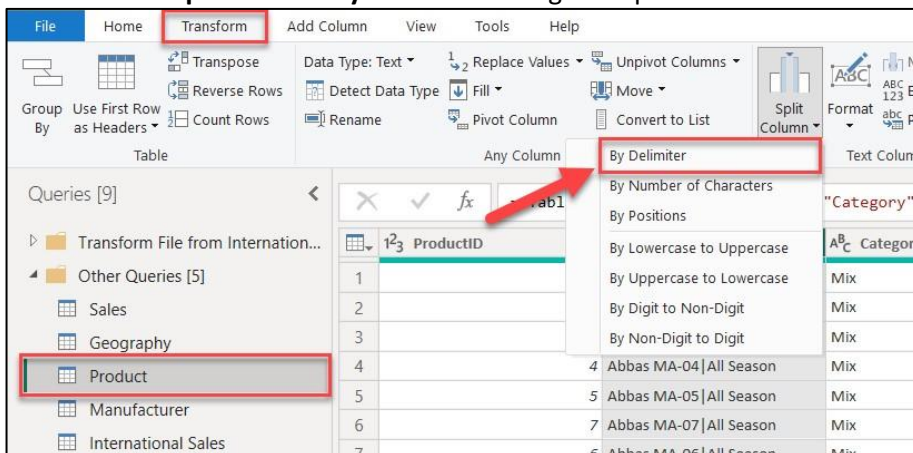
Notice how all the null values are filled with the appropriate Category values.

**Note:** The fill down operation takes a column and traverses through the values in it to fill any null values in the next rows until it finds a new value. This process continues on a row-by-row basis until there are no more values in that column.

## 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.

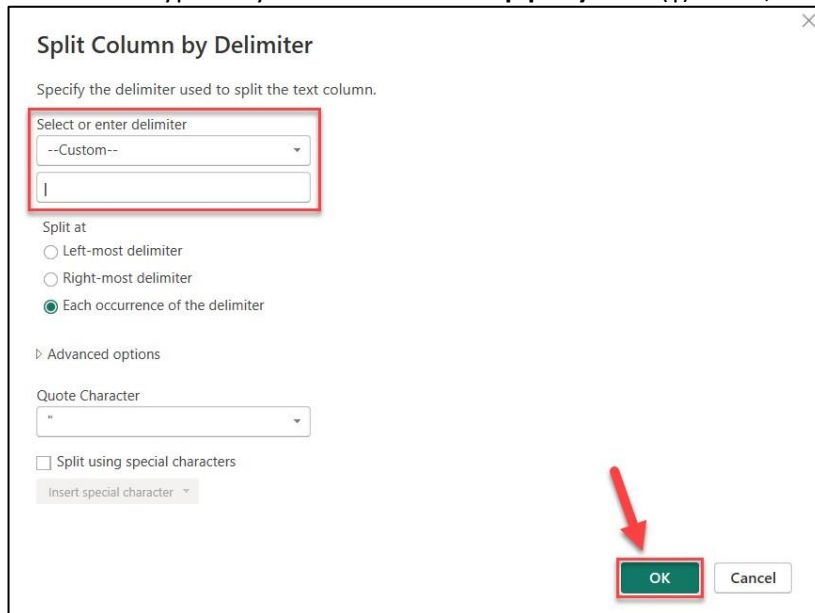
42. From the **Queries** pane to the left, ensure that the **Product** query is selected.
43. Select the **Product** column from the query table.
44. From the ribbon, select the **Transform** tab, expand the **Split Column** drop-down, and then select **By Delimiter**. The **Split Column by Delimiter** dialog box opens.



45. Within the dialog box, ensure 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.

46. Notice that within the text box, there is a **hyphen** (-). Power BI assumes we want to split by hyphen. **Remove** the hyphen symbol and enter the **pipe symbol** (|). Then, select **OK**.

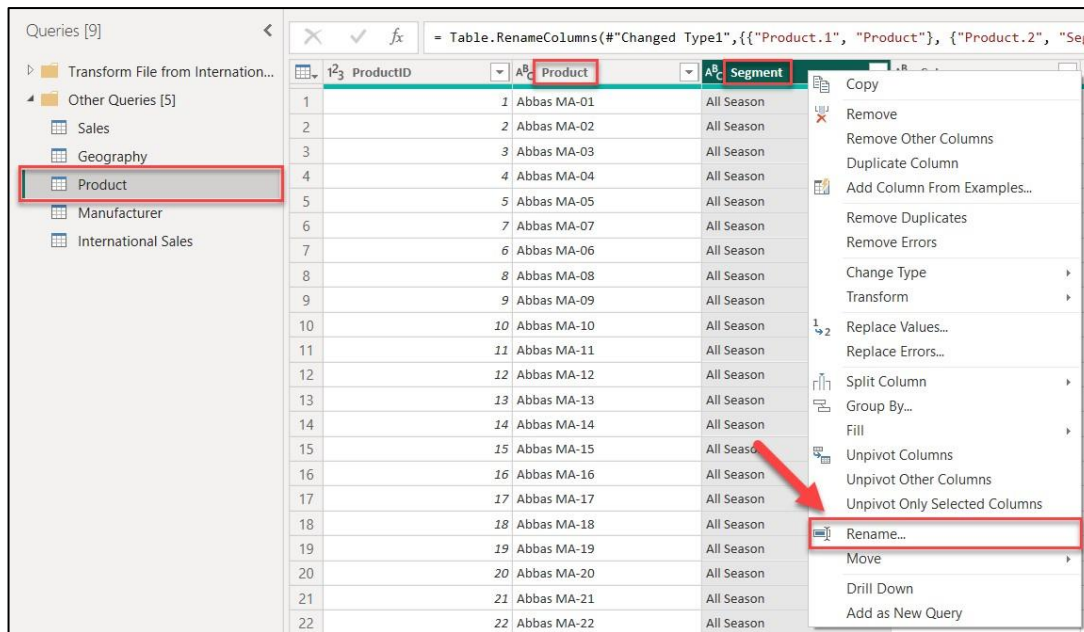


**Note:** If the delimiter occurs multiple times, the **Split at** section provides the option to split only once (either left most or right most) or the option to split the column on each occurrence of the delimiter. In this scenario, the delimiter occurs only once, therefore the Product column is split into two columns.

## Power BI Desktop – Renaming columns

Let's rename the columns now to something more user friendly.

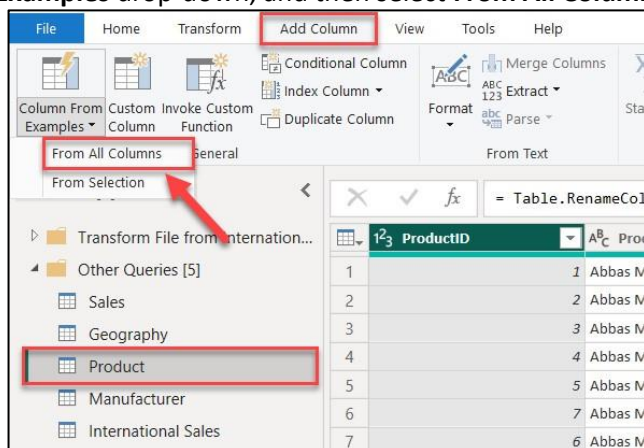
47. Select the **Product.1** column, and then right-click next to the column name.  
48. Choose **Rename...** from the options menu.  
49. **Rename** the field to **Product**.  
50. Use the same steps to rename **Product.2** to **Segment**.



## 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.

51. From the **Queries** pane to the left of the screen, ensure that the **Product** query is selected.
52. From the ribbon at the top of the screen, select the **Add Column** tab, choose the **Column From Examples** drop-down, and then select **From All Columns**.



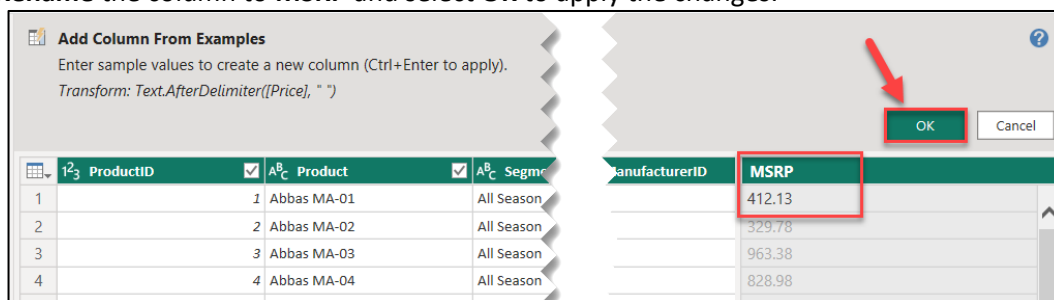
53. Within the **first row** of the newly added **Column1**, enter the first **Price** value, **412.13**, and hit **Enter** on your keyboard.

Notice after you hit Enter, Power BI knows that you want to split the **Price** column. The formula Power BI uses is displayed as well.

**Note:** A common mistake that can occur here is the **Column From Example** feature may attempt to auto-type **USD 412.13** with the Intellisense feature. **DO NOT** accept this autotyped value.

54. Double-click the column header of the newly added column within the query table.

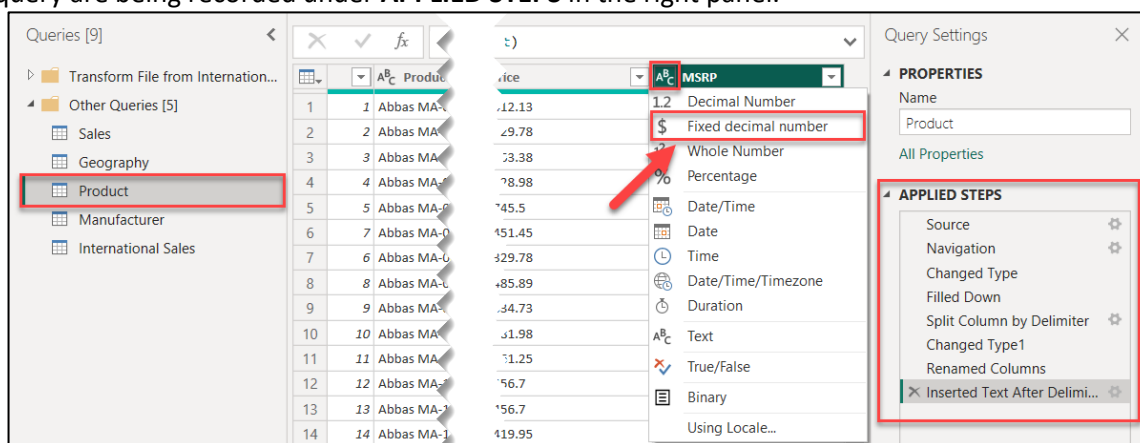
55. **Rename** the column to **MSRP** and select **OK** to apply the changes.



Notice that the **MSRP** field has a Data Type of **Text**. The Data Type that it needs to be is decimal. Let's change it.

56. Select the **ABC** icon to the left of the **MSRP** column header.

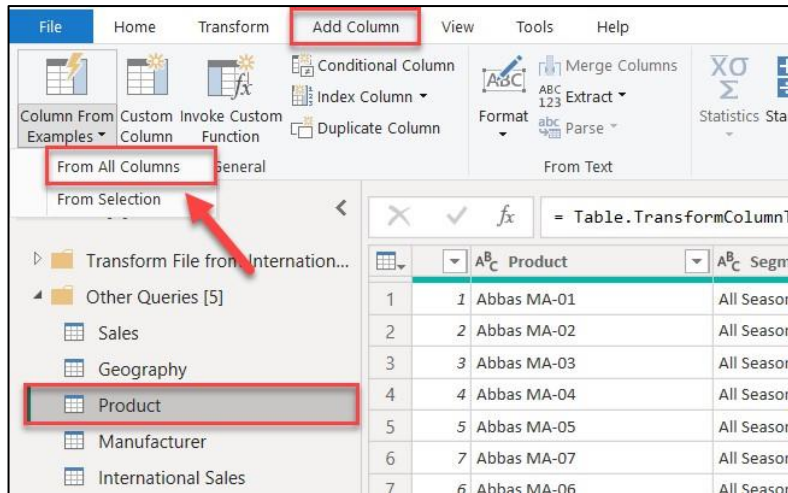
57. From the menu, select **Fixed Decimal Number**. Notice that all the steps we performed on the Product query are being recorded under **APPLIED STEPS** in the right panel.



Now let's create a **Currency** column in the same way.

58. With the **Product** query selected, from the ribbon, select the **Add Column** tab, choose the **Column From Examples** drop-down, and then select **From All Columns**.

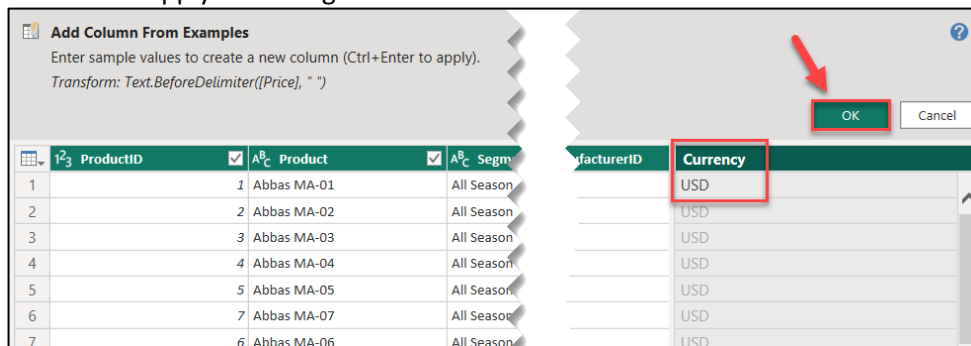




59. Within the **first row** of the newly added **Column1** enter the first **Currency** value as **USD** and then hit **Enter** on your keyboard.

60. **Rename** the column header from **Column1** to **Currency**.

61. Select **OK** to apply the changes.



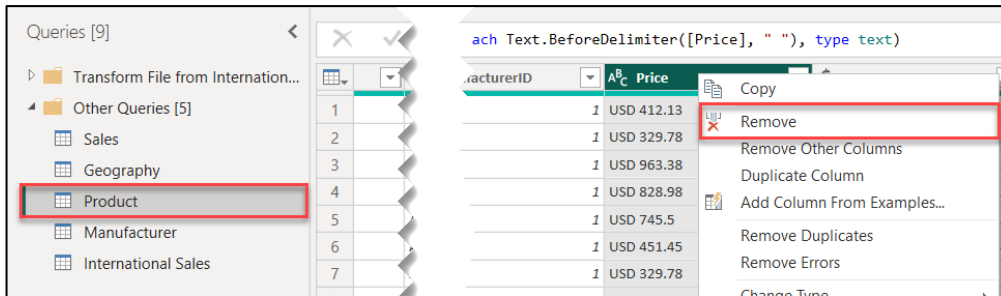
Notice that after you hit **Enter**, Power BI knows you want to split the **Price** column. The formula it uses is displayed above as well.

Now that we have split the **Price** column into the **MSRP** and **Currency** columns, we no longer need the original **Price** column. Let's remove it.

62. Ensure that you are still viewing the **Product** query. Right-click on the **Price** column.

63. Select **Remove** from the options menu.



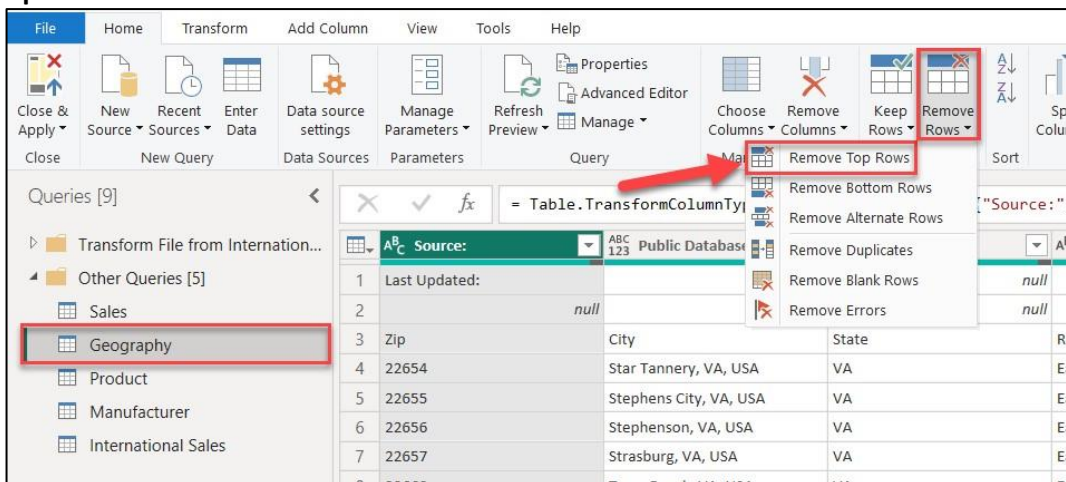


## Power BI Desktop – Removing unwanted rows

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 to work with.

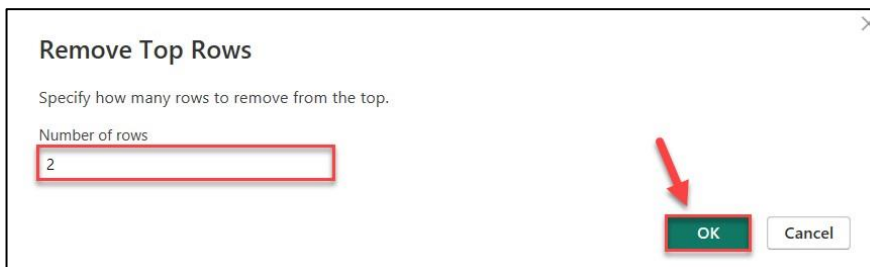
64. Within the Queries pane to the left of the screen, select the **Geography** query.

65. From the ribbon, select the **Home** tab, choose the **Remove Rows** drop-down, and then select **Remove Top Rows**.



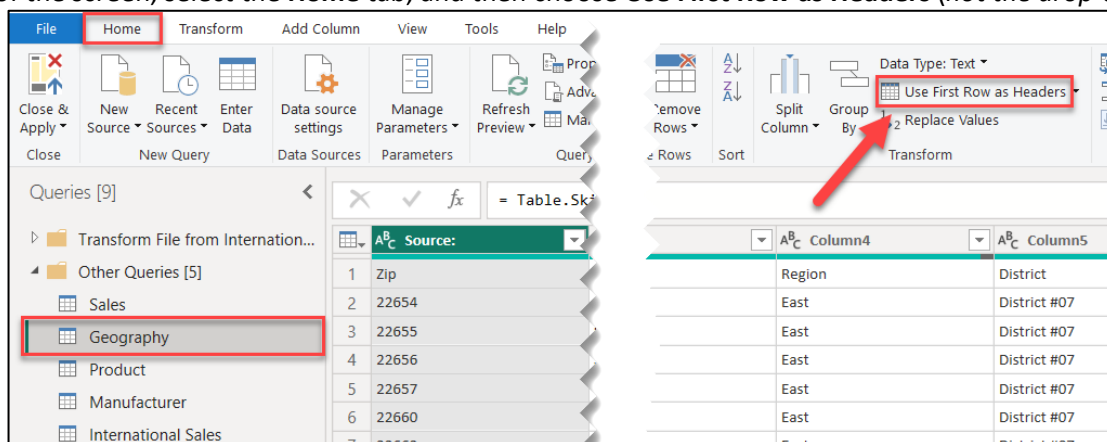
66. The **Remove Top Rows** dialog box opens. Enter **2** in the text box since we want to remove the top informational data row and the blank second row.

67. Then, select **OK**.



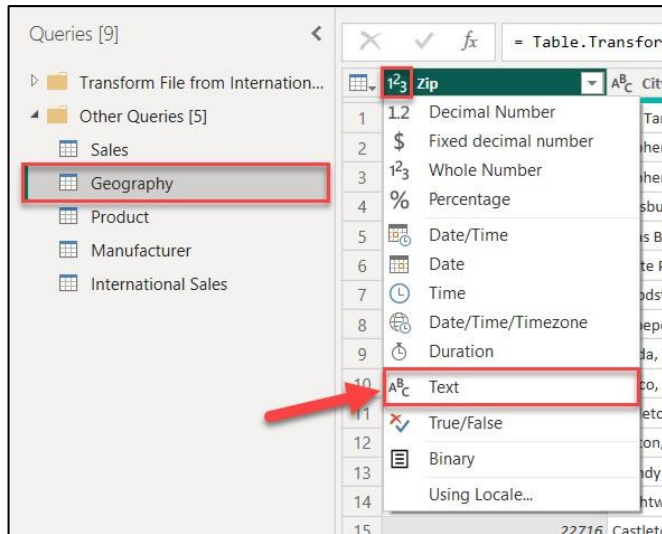
Notice the first row in the Geography query contains the column headers. Let's move them into the column header position.

68. Ensure that the **Geography** query is still selected within the Queries pane. From the ribbon at the top of the screen, select the **Home** tab, and then choose **Use First Row as Headers** (*not the drop-down*).



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** again as we did earlier. If we don't, we will see errors when we load the data.

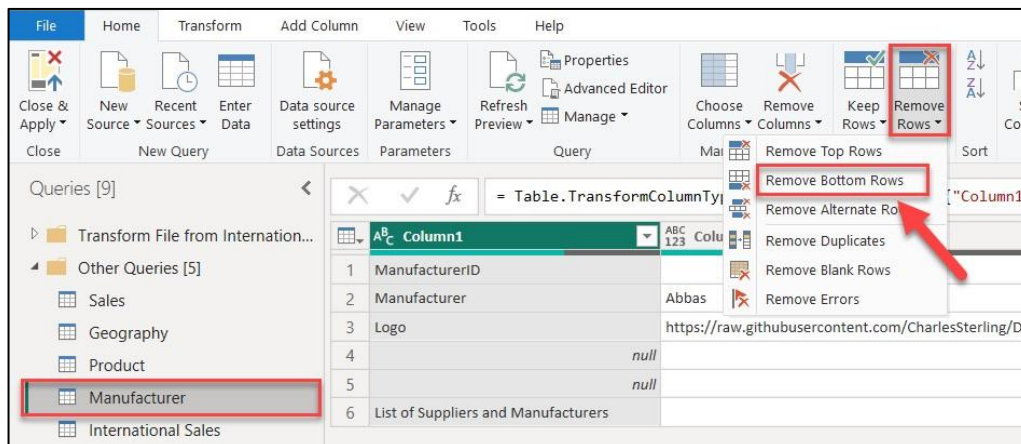
69. Select the **data type** icon to the left of the **Zip** column header. From the options menu, select **Text**.



70. Select **Replace Current** in the **Change Column Type** dialog box.

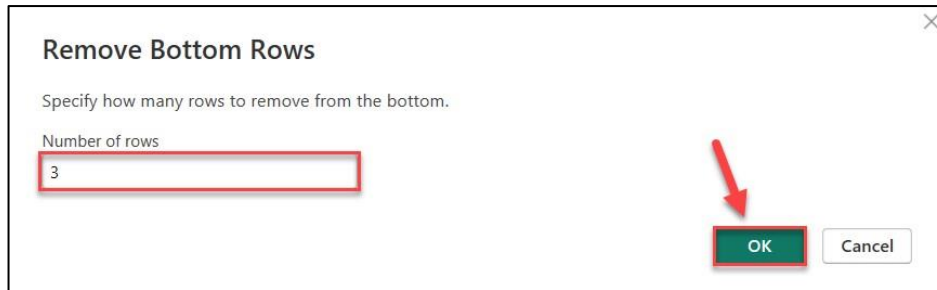
71. From the **Queries** pane, select the **Manufacturer** query. Notice the bottom three rows are not part of the data. Let's remove them.

72. From the ribbon, select the **Home** tab, choose the **Remove Rows** drop-down, and then select **Remove Bottom Rows**.



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

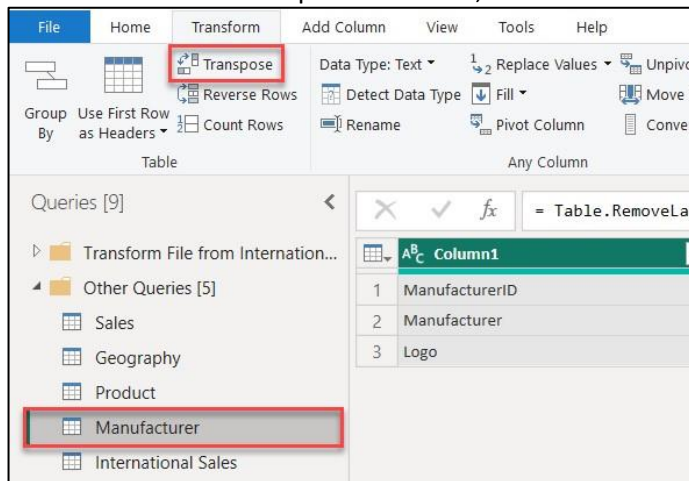
74. Then, select **OK**.



## Power BI Desktop – Transposing data

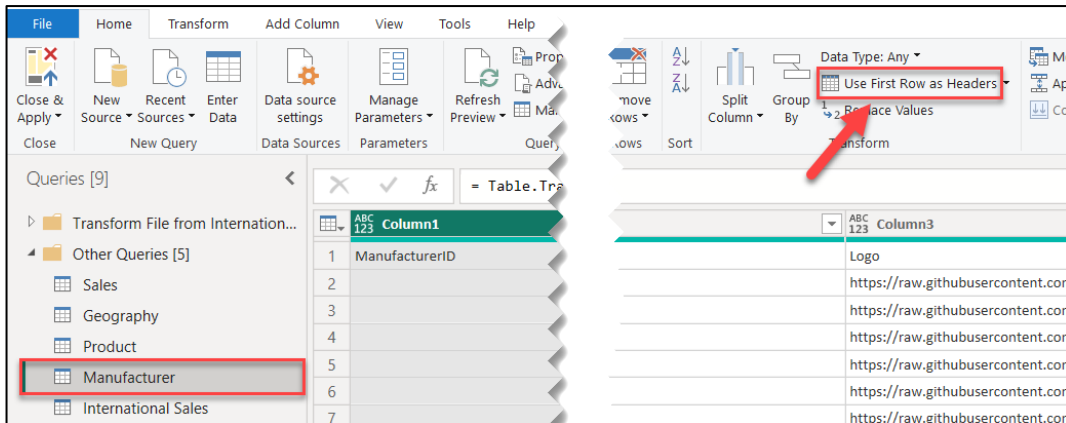
75. From the Queries pane to the left of the screen, select the **Manufacturer** query. Notice that the **ManufacturerID**, **Manufacturer**, and **Logo** data are laid across in rows. Also notice that the header is not useful. We need to transpose the table to meet our needs.

76. From the ribbon at the top of the screen, select the **Transform** tab, then choose **Transpose**.

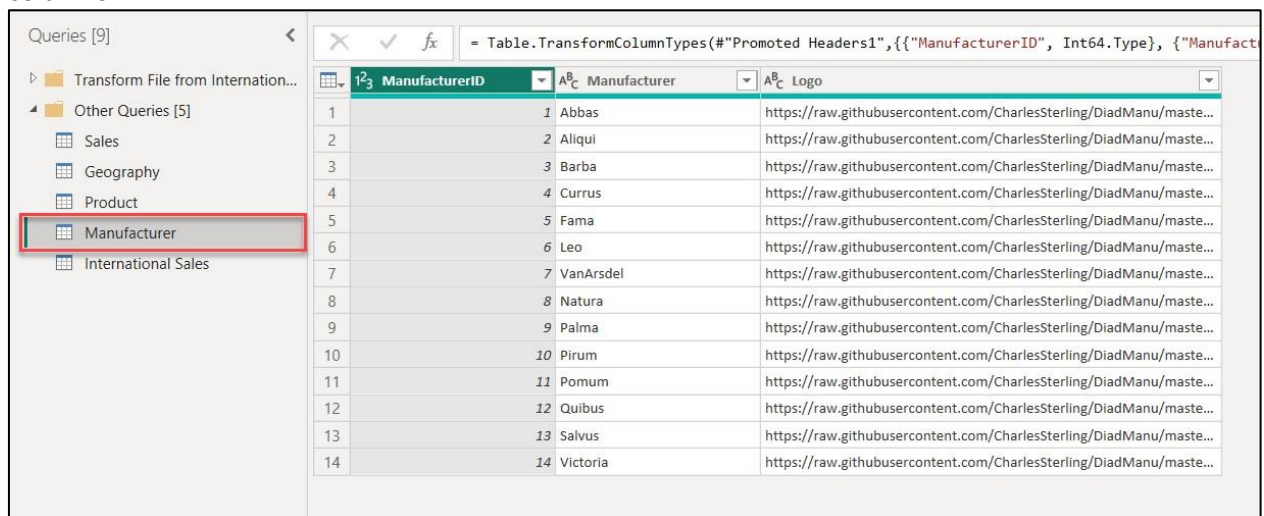


Notice that this transposes the data into columns. Now we need the first row to be the header.

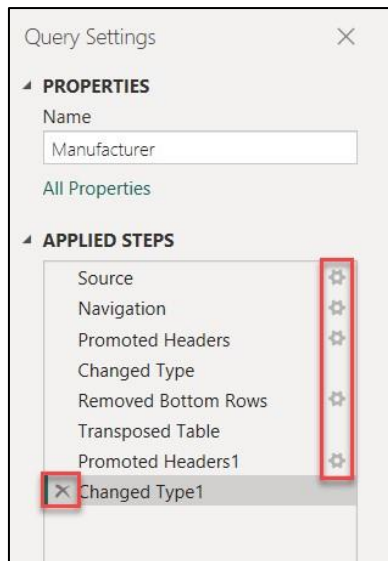
77. From the ribbon at the top of the screen, select the **Home** tab, and then choose the **Use First Row as Headers** button (*not the drop-down*).



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



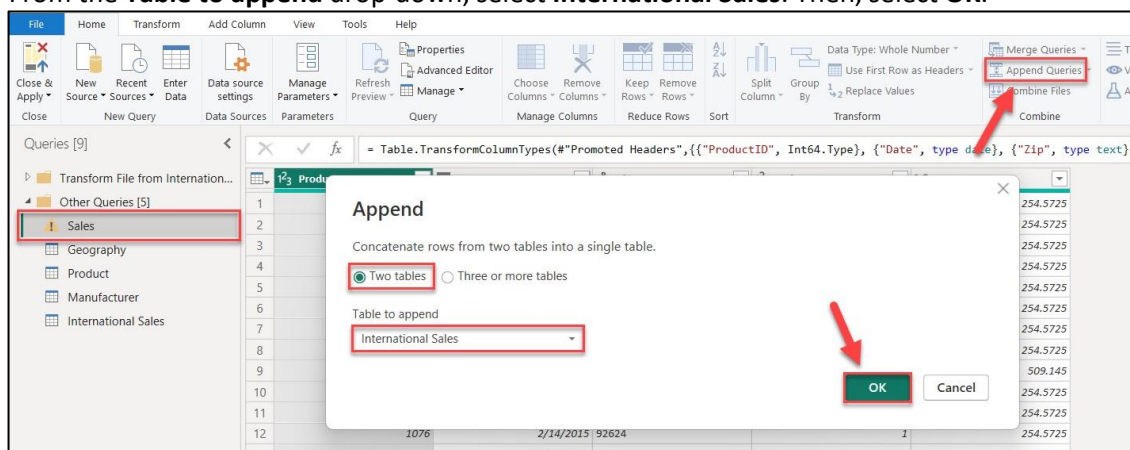
Also, notice that with the **Query Settings** pane, under **APPLIED STEPS**, you will see the list of transformations and steps that have been applied. You can navigate through each change made to the data by selecting the step. Steps can also be deleted by choosing the **X** that appears to the left of the step. The properties of each step can be reviewed by selecting 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.

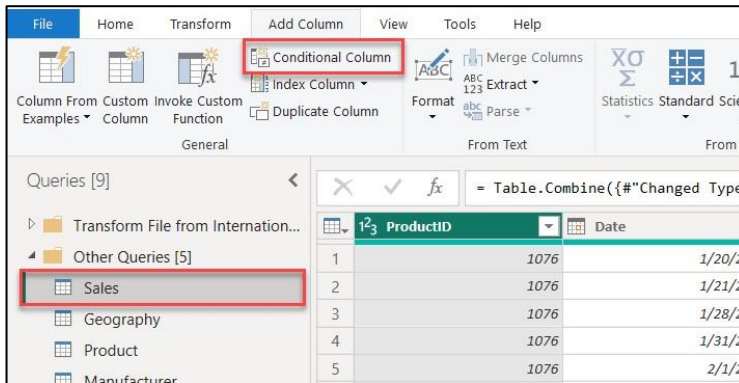
78. Within the **Queries** pane to the left of the screen, select the **Sales** query.
79. From the ribbon at the top of the screen, select the **Home** tab, and then choose **Append Queries** button (*not the drop-down*).
80. The **Append** dialog box opens. There is an option to append **Two tables** or **Three or more tables**. Leave **Two tables** selected since we are appending just two tables.
81. From the **Table to append** drop-down, select **International Sales**. Then, select **OK**.




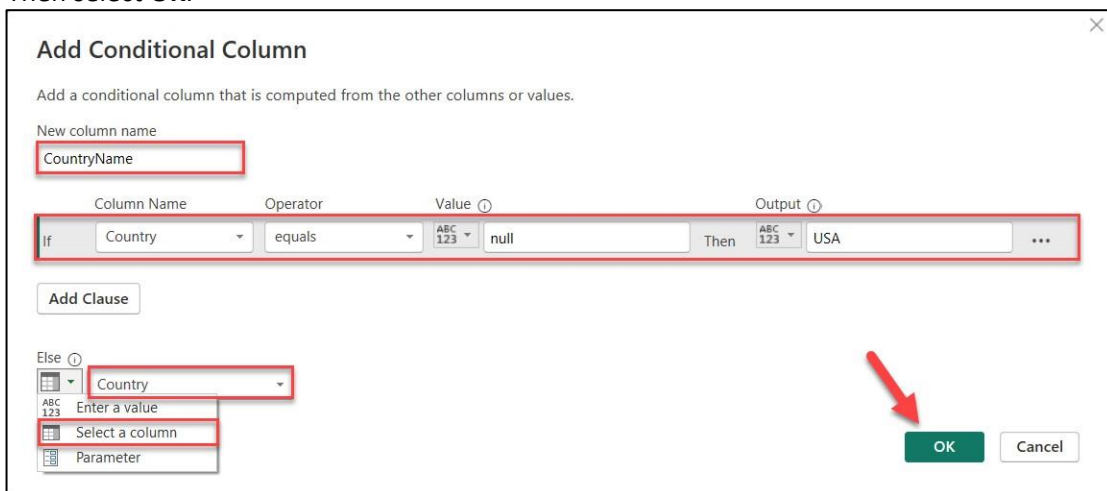
You will now see a new column in the **Sales** table called **Country**. Since the **International Sales** query had the additional column for **Country**, the Power Query Editor added the **Country** column to the newly updated **Sales** table when it loaded the values from the **International Sales**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

You may also notice that there are **null** values within the **Country** column by default for the **Sales** table rows. This is because that column did not exist for the table with USA data. We will now add the value **USA** as a data shaping operation.

82. From the ribbon at the top of the screen, select the **Add Column** tab, and then choose the **Conditional Column** button.



83. In the **Add Conditional Column** dialog box, enter the name of the column as **CountryName**.
84. Select **Country** from the **Column Name** drop-down menu.
85. Choose **equals** from the **Operator** drop-down menu.
86. Enter **null** in the **Value** text box.
87. Enter **USA** in the **Output** text box.
88. Select the  drop-down menu under **Else** and then choose the **Select a column** option.
89. Choose **Country** from the column drop-down menu.
90. Then select **OK**.

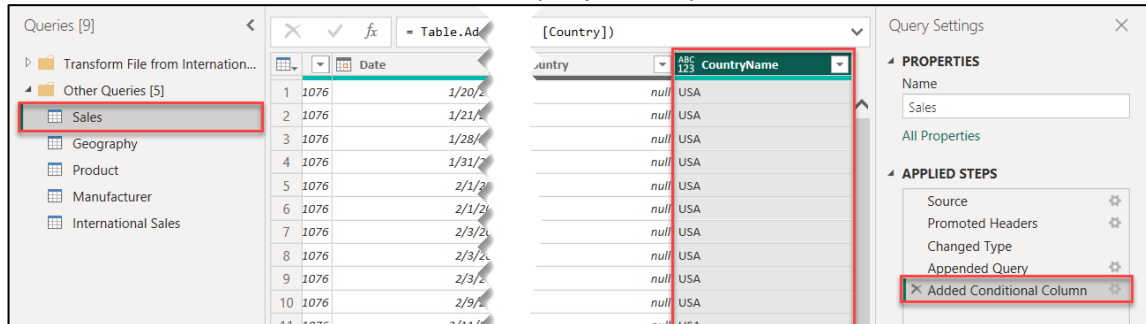


*This reads: If the current Country value is equal to null, then the value should return USA; otherwise, if the value is not null, then use the current Country value.*



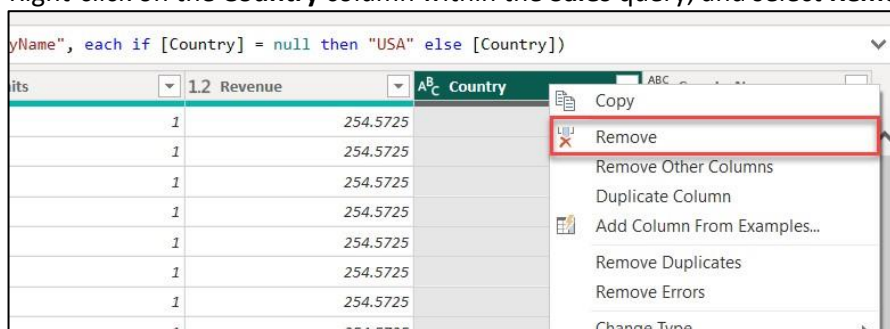
**Note:** A common mistake on the previous step is that the **Else** may not being set correct. Please double check your **Else** part of the conditional column matches the screenshot above.

91. You will see the **CountryName** column in the Query editor window. Notice that within the **APPLIED STEPS** list, it has added to the list the action you just completed.



The original **Country** column containing the null values is no longer needed and can be removed from the final table for analysis.

92. Right-click on the **Country** column within the **Sales** query, and select **Remove** from the options menu.



With this column now removed, we can now **rename** the **CountryName** column to **Country**.

93. Right-click on the **CountryName** column within the **Sales** query, and **rename** it to **Country**.  
94. Select the **Data Type** icon to the left of the **Country** column header and change the **Data Type** to **Text**.  
95. Next, select the **Data Type** icon to the left of the **Revenue** column header and change the **Data Type** to **Fixed decimal number**. We do this because it is a currency field.



text}, {"Revenue", Currency.Type}})

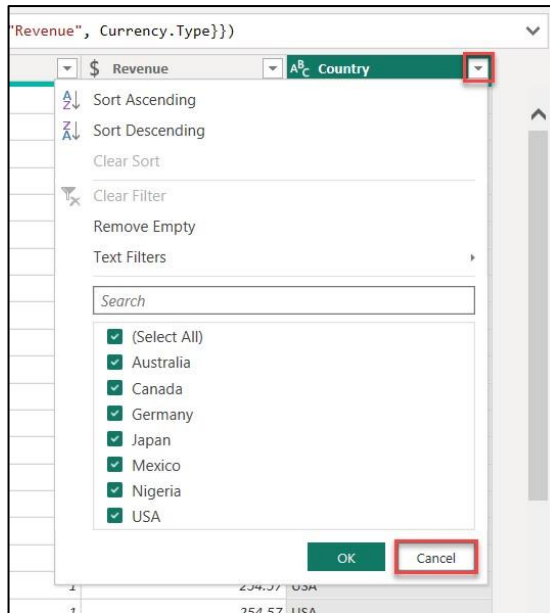
	\$ Revenue	A <sup>B</sup> <sub>C</sub> Country
1	254.57	USA
1	254.57	USA
1	254.57	USA
1	254.57	USA
1	254.57	USA

**Note:** The difference between a Fixed decimal number and a Decimal number is related to the length and precision of the decimal places. <https://learn.microsoft.com/en-us/power-bi/connectdata/desktop-data-types#number-types>

When the data is refreshed, it will process through all the **APPLIED STEPS** that you have created.

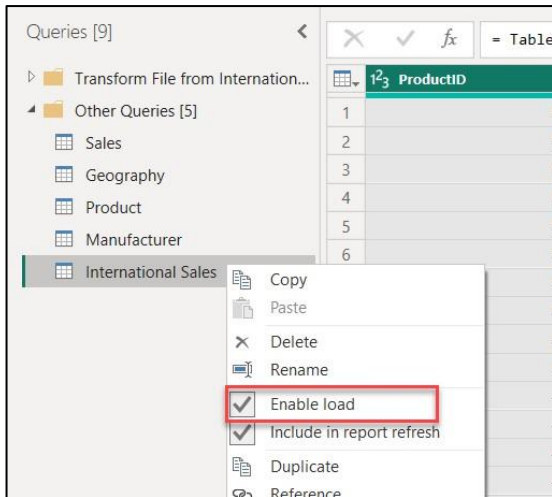
The newly named **Country** column will have names for **all countries**, including the USA. You can validate this by selecting the drop-down menu next to the **Country** column to see the unique values.

96. At first, you will only see USA data. Select the **drop-down arrow** to the right of the **Country** column header. Select **Load more** to validate your data from all seven countries.
97. Select **Cancel** to close this filter. You **do not** need to apply this filter to the data.



Now that the **International Sales** data is appended to the **Sales** query, in order to avoid duplicating data we should suppress the **International Sales** table from loading into the data model.

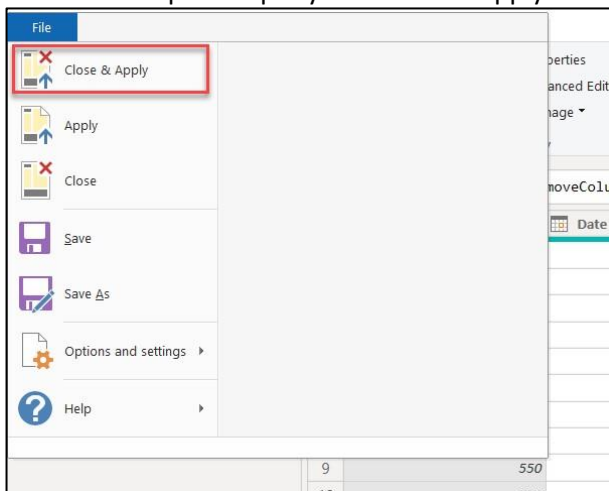
98. From the **Queries** pane to the left of the screen, select the **International Sales** query.
99. Right-click on the **International Sales** query within the Queries pane, and then choose **Enable Load** to **deselect** this setting. This will disable loading of the International Sales query into the data model. *(You should see the name of this query become italicized in the Queries pane after deselecting the Enable load option)*



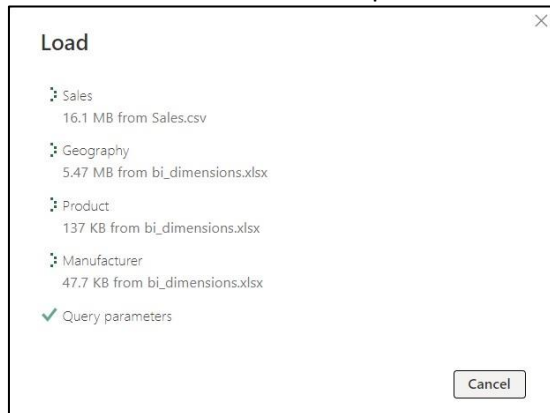
**Note:** The appropriate data from the International Sales table will load onto the Sales table each time the model is refreshed. By removing the International Sales table, we are preventing duplicate data from loading into the model and increasing its file size. In some instances, storing very large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

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.

100. From the ribbon at the top of the screen, select the **File** tab, then choose **Close & Apply**. This will close out the power query window and apply all changes.



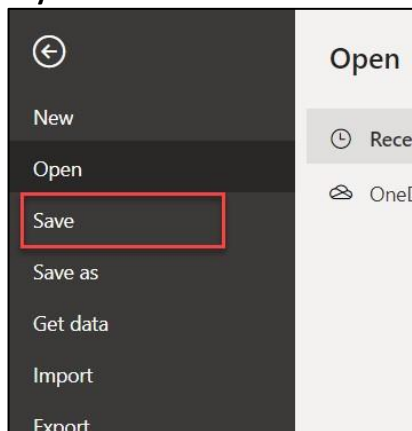
All the data will be loaded in memory in the Power BI Desktop. You will see the progress dialog box with the number of rows being loaded in each table as shown in the Figure. Once the load completes the results of this Power BI Desktop file will be used in Lab 02.






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

101. Once the data has finished loading, select the **File** tab from the ribbon at the top of the screen.

102. Then, from the options menu to the left, select **Save** to save the file. Name the file as **MyFirstPowerBIModel**. Save the file within the **DIAD Reports (\DIAD\Reports)** folder.



103. Within the **navigation pane** to the left of the screen, select the **Data**  icon to view the data that was loaded. If you need to return to the **Power Query editor** again, navigate to **Home**  **Transform Data**  **Transform data**.

File

Home

Help

Table tools

Paste

Cut

Copy

Get data

Excel workbook

OneLake data hub

SQL Server

Enter data

Dataverse

Recent sources

Transform data

Refresh

Manage relationships

Clipboard

✕

✓

68274

68275

68276

71512

71513

71515

Zip

City

State

Region

District

Country

Transform data

Transform data

Data source settings

Edit parameters

Edit variables

	Zip	City	State	Region	District	Country
	68274		Oaxaca		Oaxaca de Juarez	Mexico
	68275		Oaxaca		Oaxaca de Juarez	Mexico
	68276		Oaxaca		Oaxaca de Juarez	Mexico
	71512		Oaxaca		Ocotlan de Morelos	Mexico
	71513		Oaxaca		Ocotlan de Morelos	Mexico
	71515		Oaxaca		Ocotlan de Morelos	Mexico