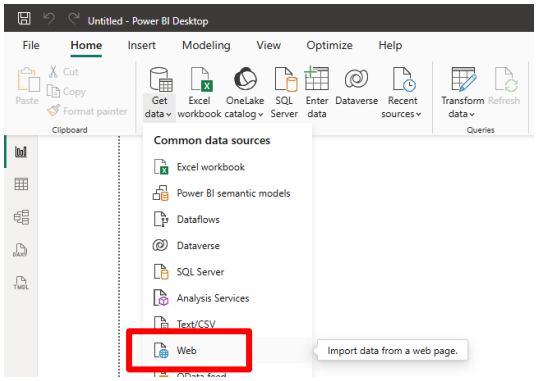
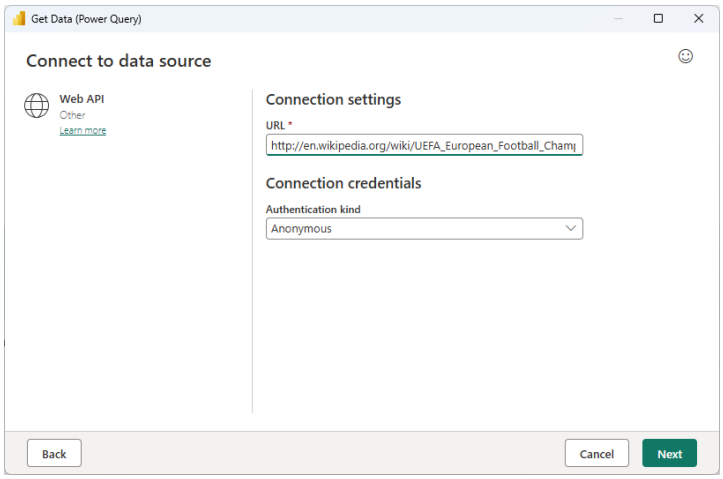
1. Create Report from Web Data Source

1.1 Add a Wikipedia Page Data Source

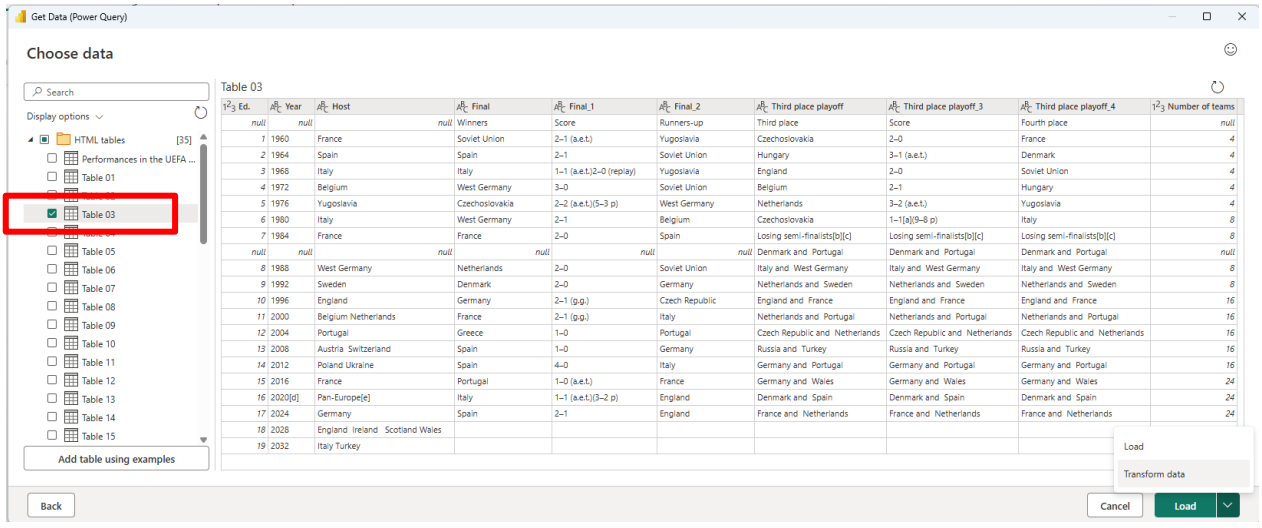
1. Select **Get data ➔ Web** in the **Home** ribbon tab.



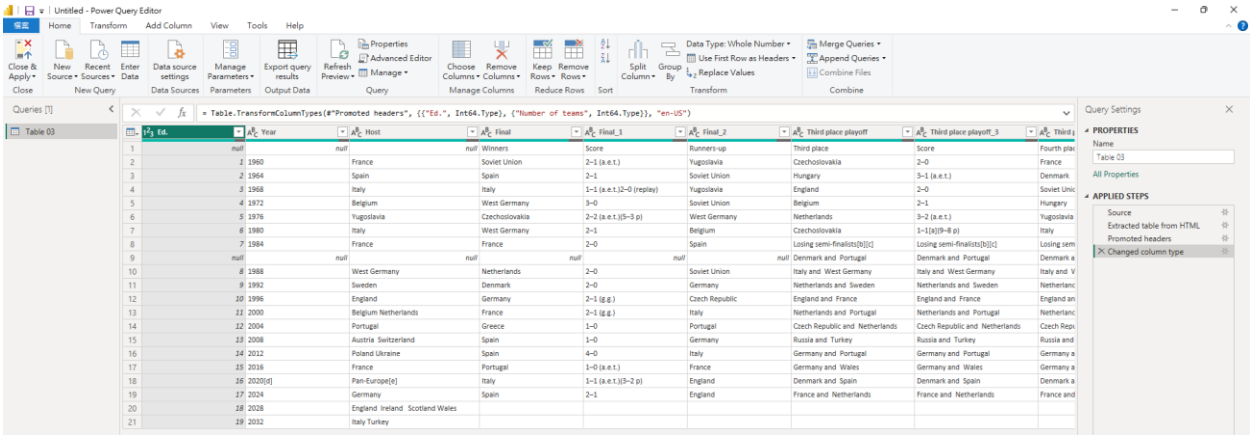
2. Type the address (http://en.wikipedia.org/wiki/UEFA_European_Football_Championship) in the URL to import a Tournament Summary table from the UEFA European Football Championship Wikipedia, and click **[Next]** button to continue



3. After establishing a connection to the web page, list of tables available on this Wikipedia page in the **Get Data (Power Query)** dialog. You can click on each of these tables to preview the data. In the left pane, select the *Table 03* table and select **[Load] ➔ [Transform Data]**.



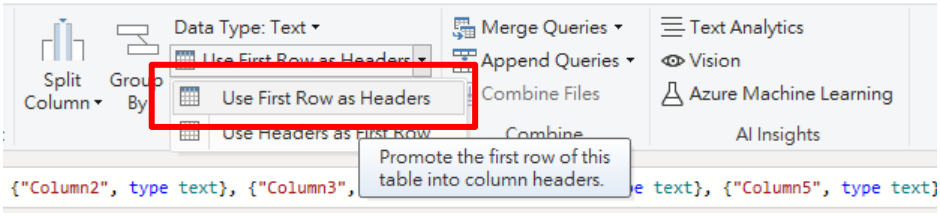
4. This will allow us to reshape this table before loading it to the report, since the data is not in the shape that we need for our analysis



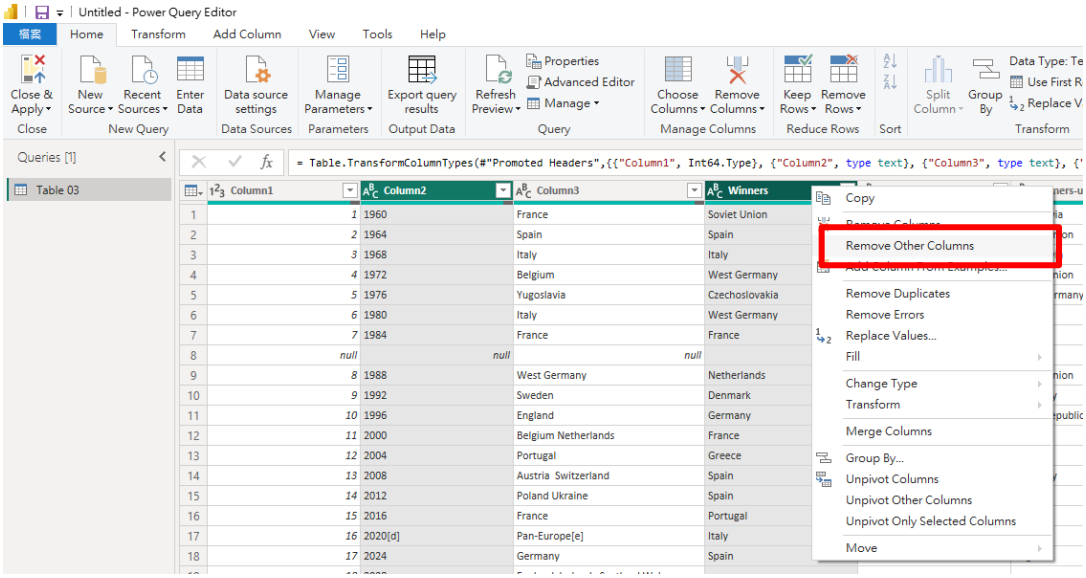
1.2 Data Sharping and Cleansing

1.2.1 Remove Other Columns to only Display columns of interest

1. Select **Use First row as Headers** in the **Home** ribbon tab.

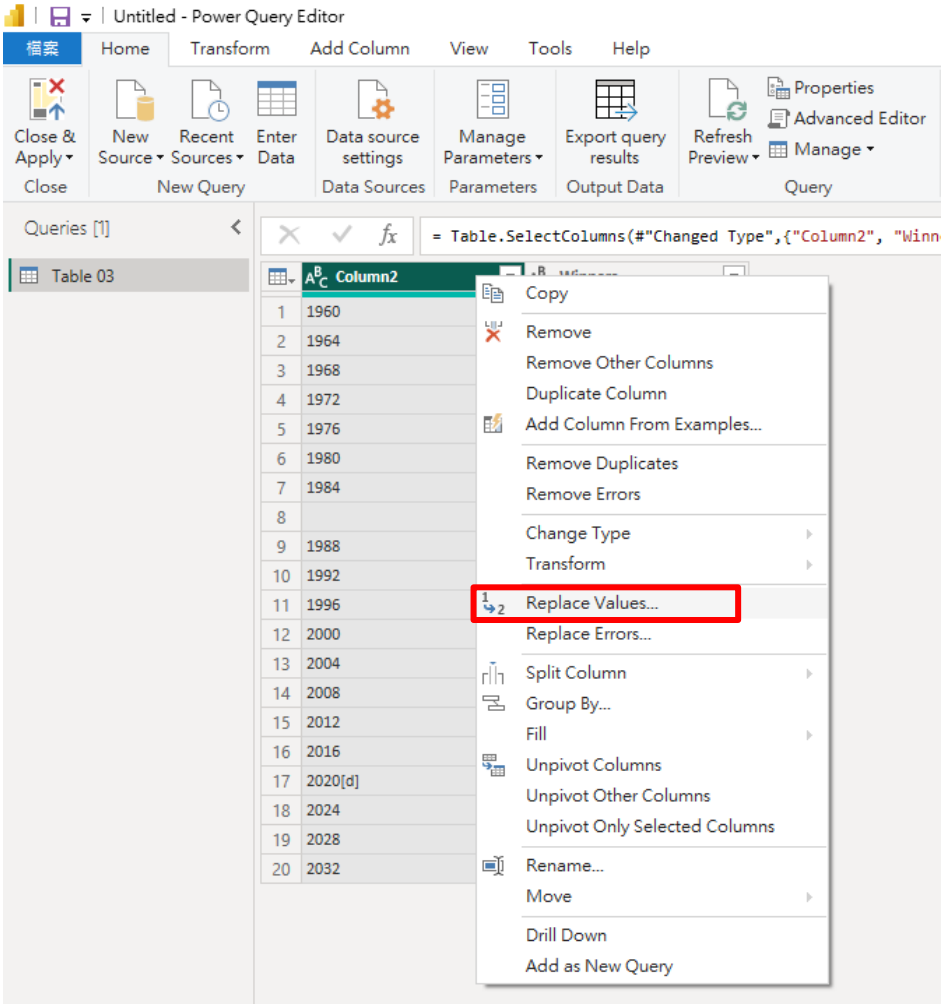


2. In the query preview grid, select the *Column2* and *Winners* columns. Right-click to select **Remove Other Columns** to remove the unselected columns.

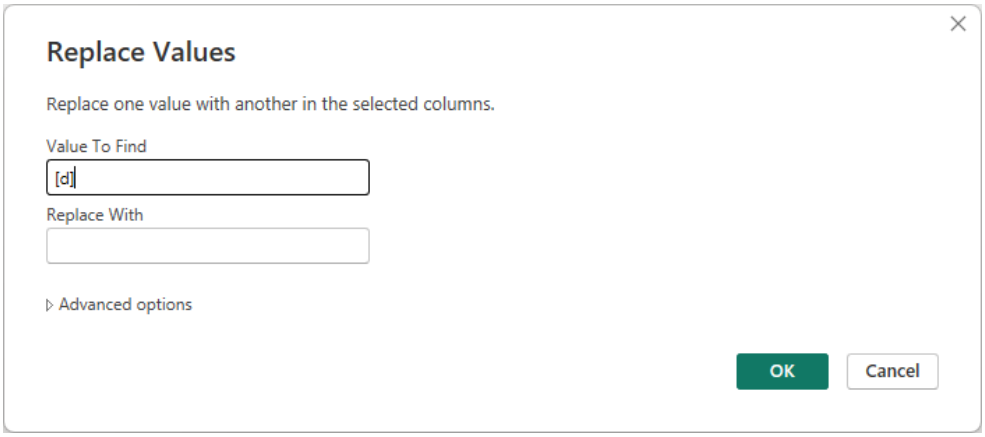


1.2.2 Clean up Values in Selected Column

1. Select the *Column2* column and right click to select **Replace Value..**

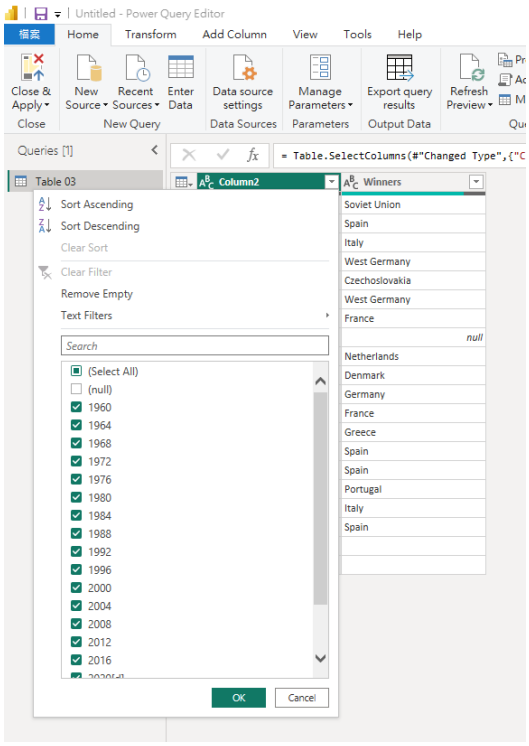


2. In the **Replace Values** dialog box, type *[d]* in the **Value to Find** text box and leave the **Replace With** text box empty. Click **[OK]** button to start.



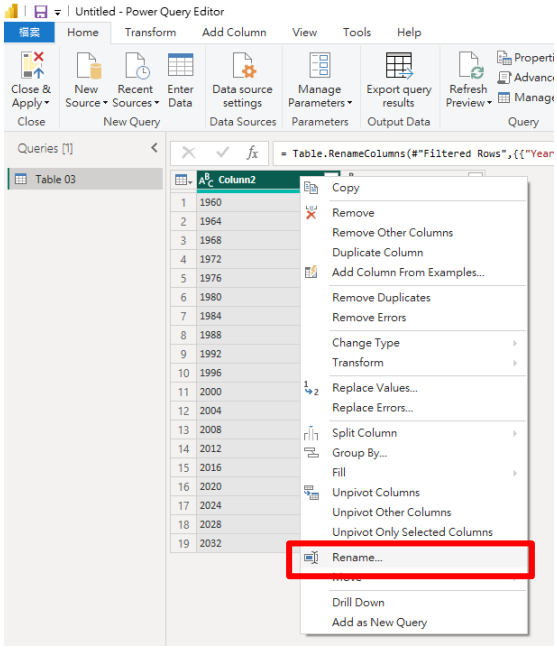
1.2.3 Filter Values in Column

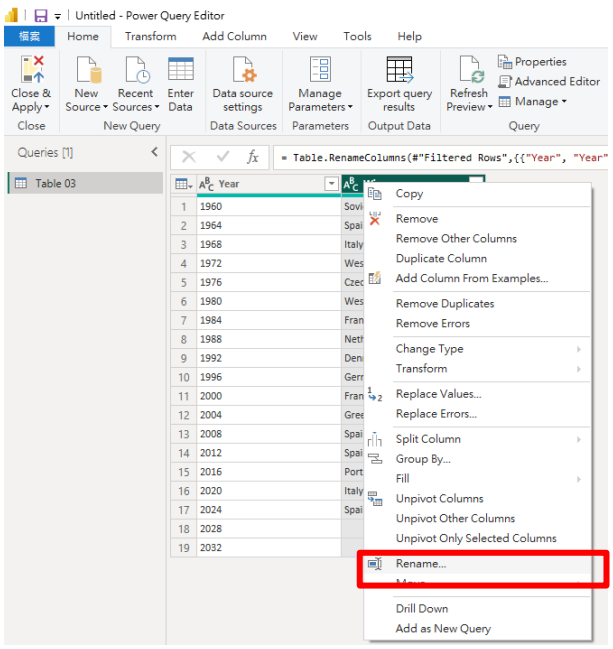
1. Click the filter drop down arrow on the *Column2* column.
2. In the Filter drop-down, clear the *null* option and click **[OK]**.



1.2.4 Rename Column

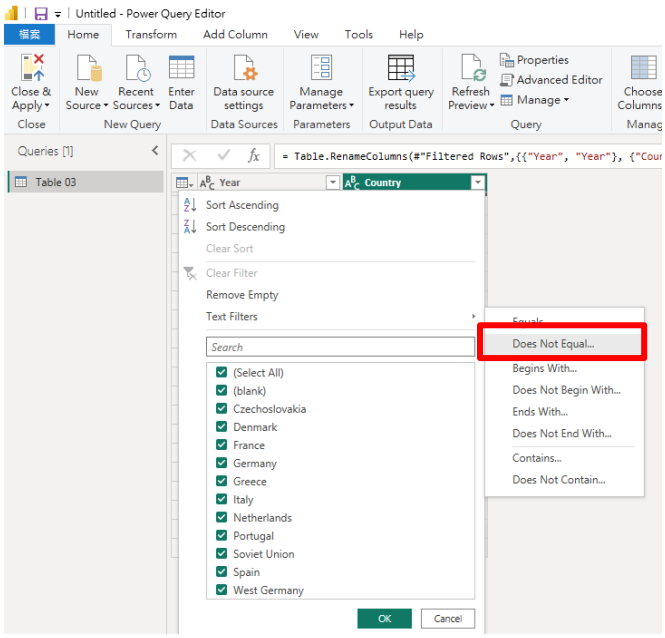
1. Select the *Column2* column and then right click to select **Rename**, change the name of this column to “Year”.
2. Select the *Winner* column and then right click to select **Rename**, change the name of this column to “Country”.



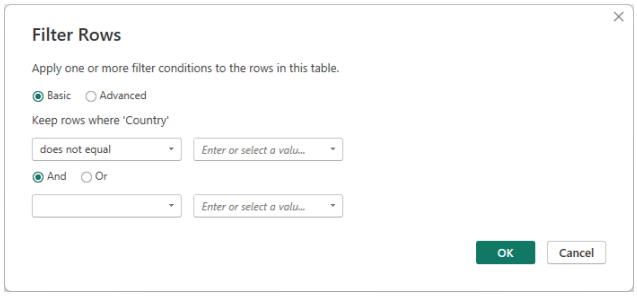


1.2.5 Filter out blank Values in Column

1. Right-click on one of the cells in the *Country* column that contain a null value.
2. Select **Text Filters ➔ Does not Equal** in the context menu. This creates a new filter step to remove rows with space values in the Country column.

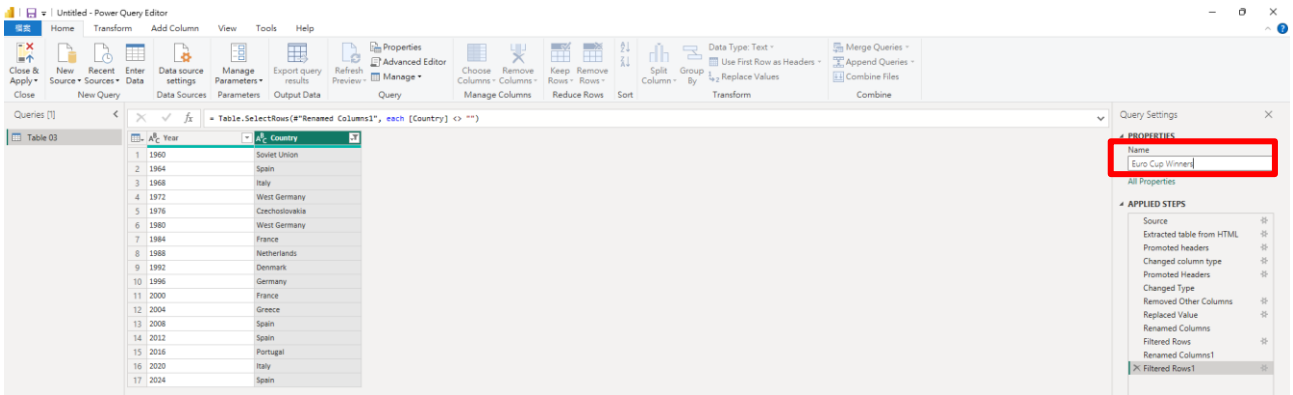


3. Press **[OK]** to start replacement.



1.2.6 Name a query

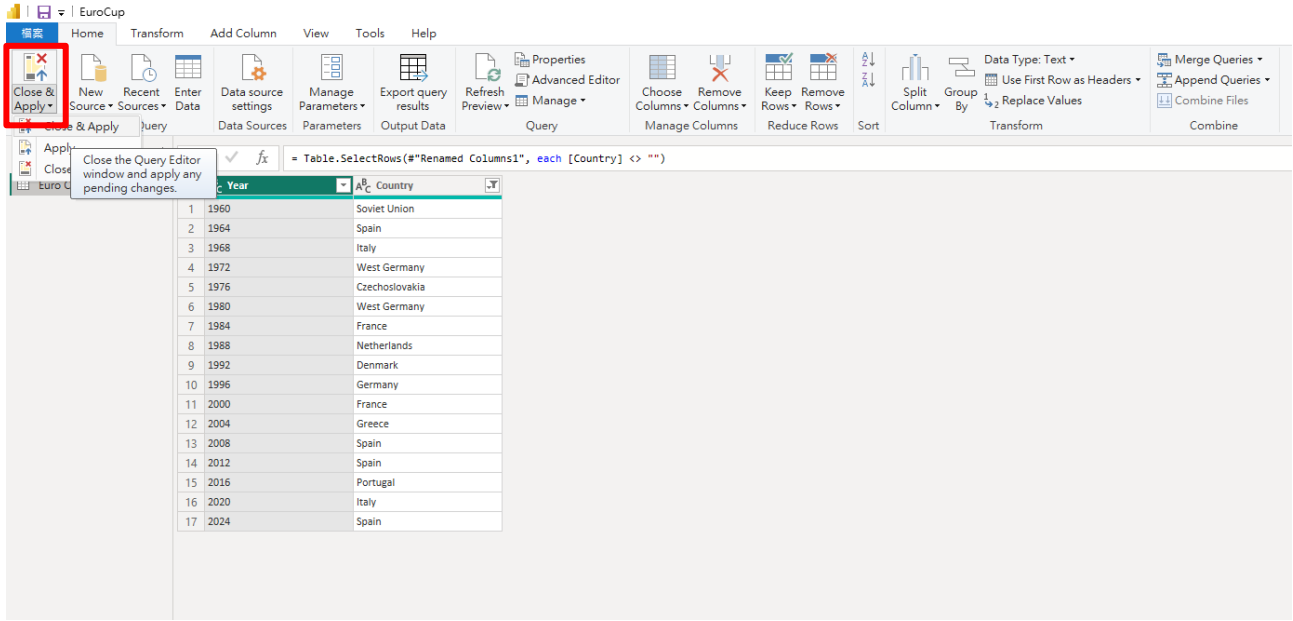
1. In the **Query Settings** pane, in the **Name** text box, enter **Euro Cup Winners**.



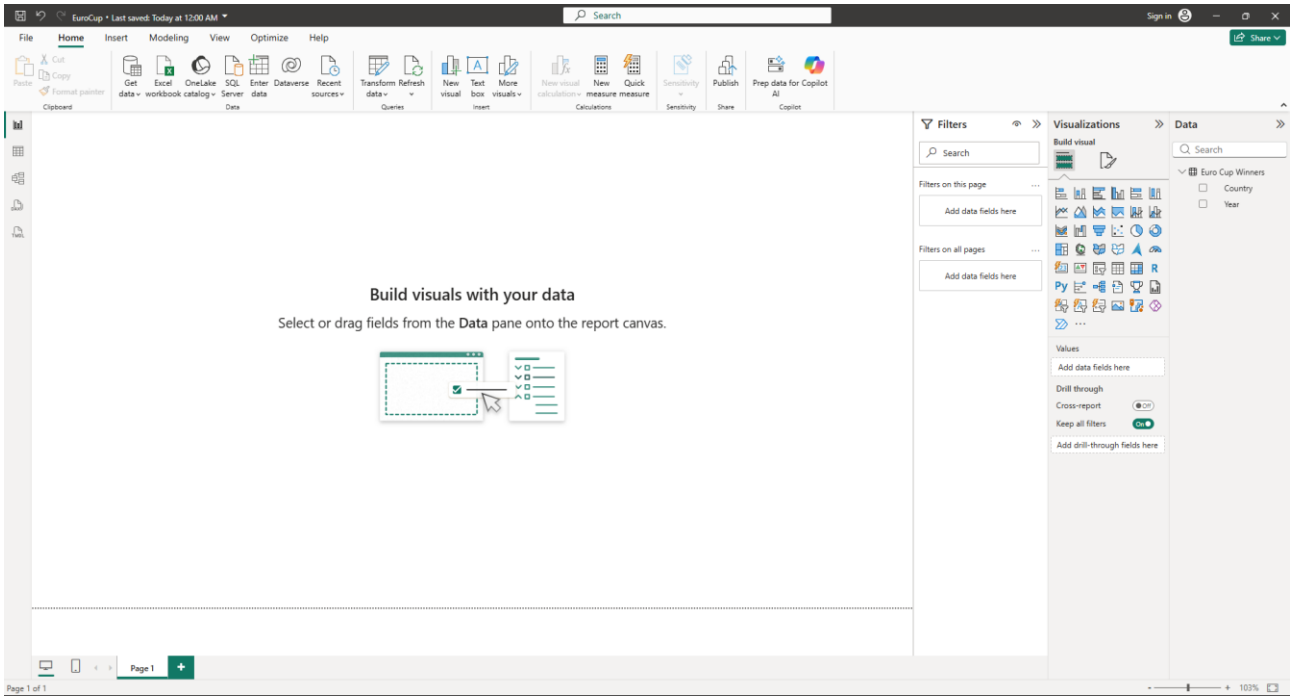
1.3 Create Visualizations using the Report View

1.3.1 Load the Query to Report

1. To load the query results to Power BI Desktop and create a report, we select **[Close & Apply]** from the **Home** ribbon.

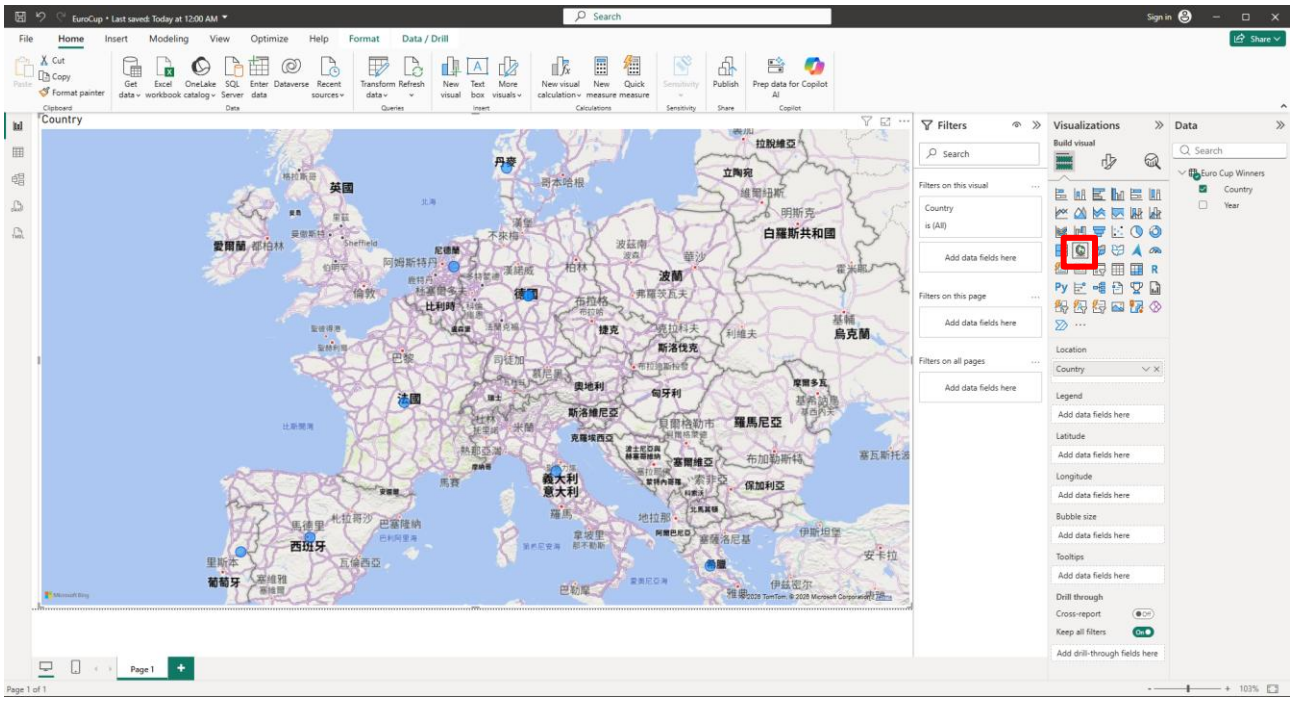


2. This will trigger evaluation of the query and load of the table output to the Report. In Power BI Desktop, select the **Report** icon to see Power BI Desktop in **Report** view. You can see the resulting table fields in the **Fields** pane at the right of the **Report** view.

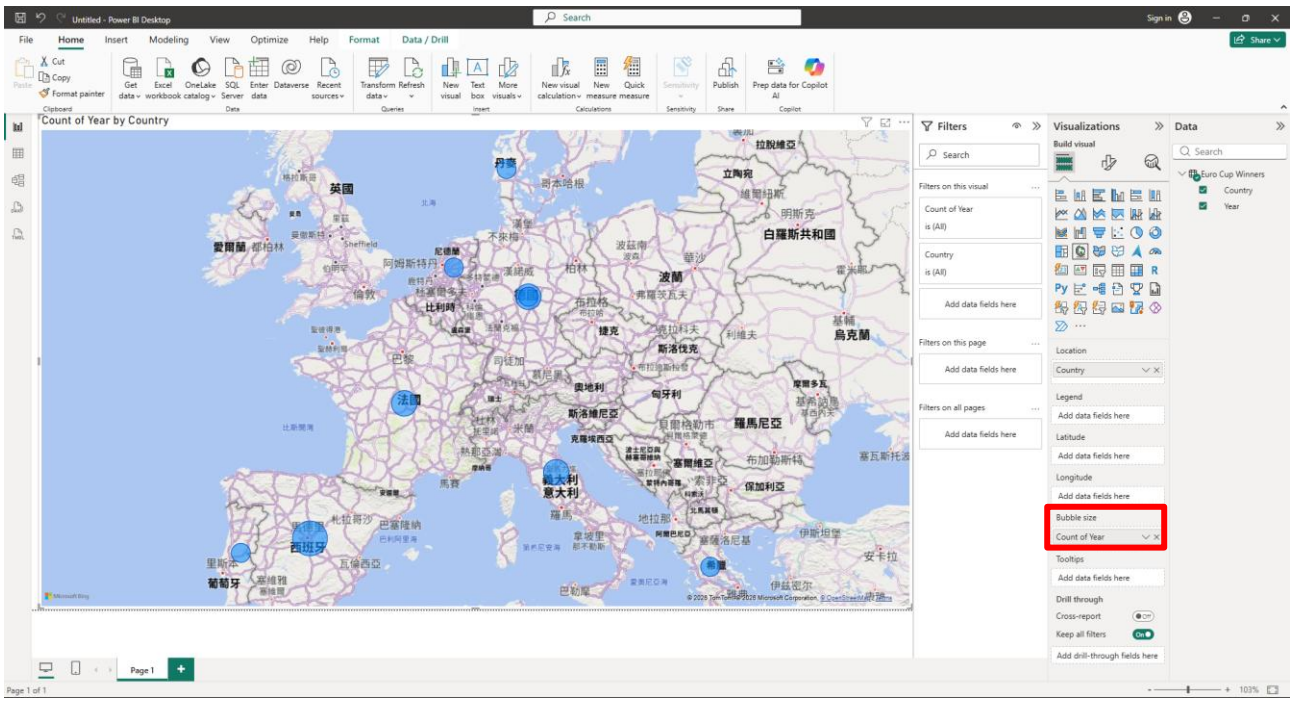


1.3.2 Create a Map Visualization

1. Drag the *Country* field and drop it in the **Report** canvas. This will create a new visualization in the **Report** canvas. In this case, since we have a list of countries, it will create a **Map** visualization.



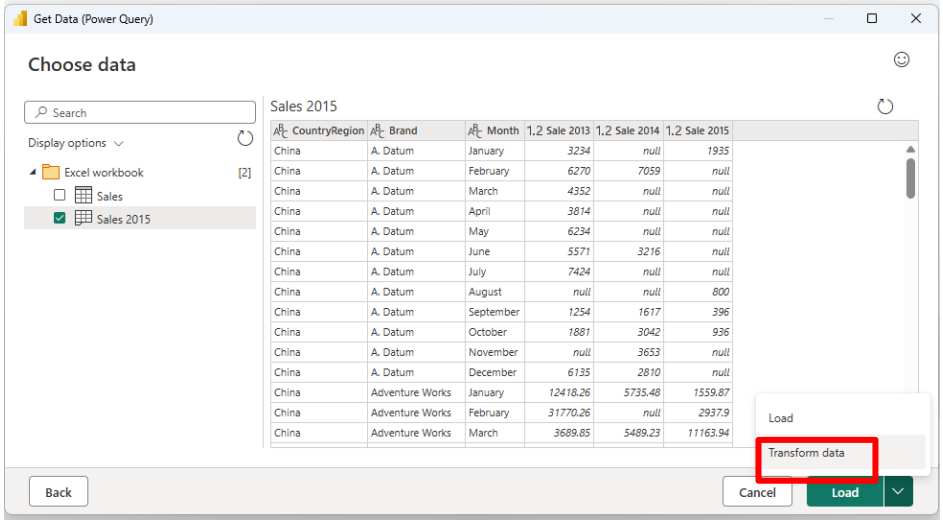
2. We can resize the visualization by dragging from one of the corners of the visualization up to the desired size. Note that currently all the points in the map have the same size. We want to change this so that countries with more Euro Cup tournaments won are represented with a larger point in the map. In order to do this, we can drag the *Year* field in the **Fields** list to the **Bubble Size** box in the lower half of the Fields pane.



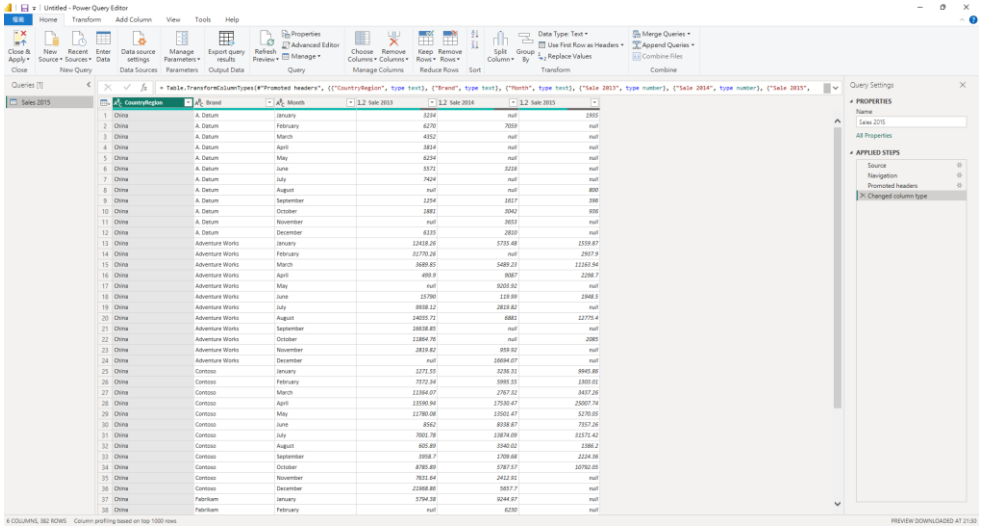
2. Merge Query for Analysis

2.1 Open First Excel Workbook

1. Select **Get Data ➔ Excel Workbook** in the **Home** ribbon tab,
2. In the **Get Data (Power Query)** dialog box, select the *2015 Sales.xlsx* file. Then in the **Navigator** pane, select the *Sales 2015* table and then select **[Load] ➔ [Transform Data]**.

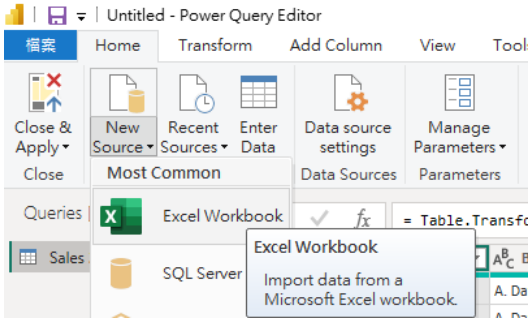


3. The file should be loaded successfully to Power BI

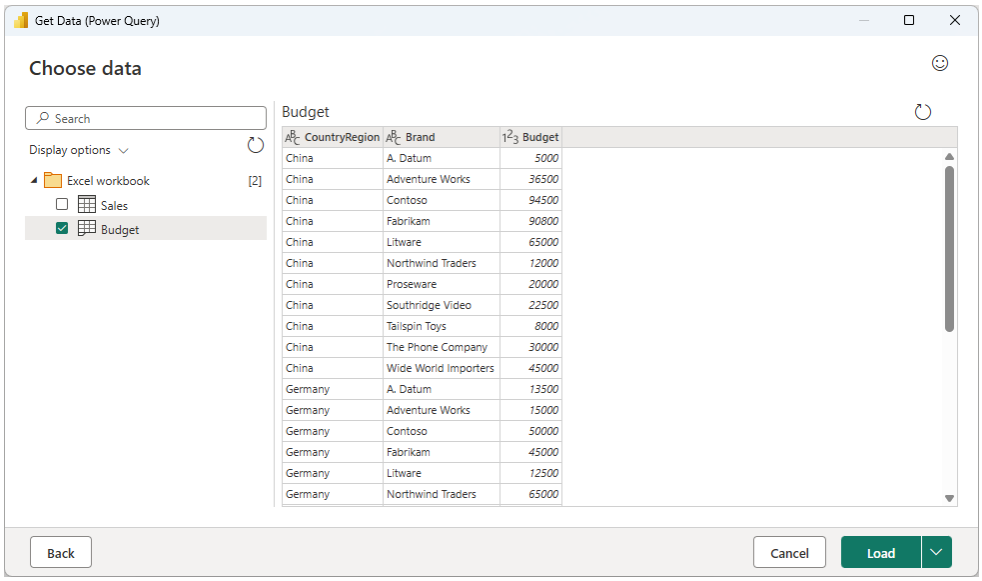


2.2 Open Second Excel Workbook

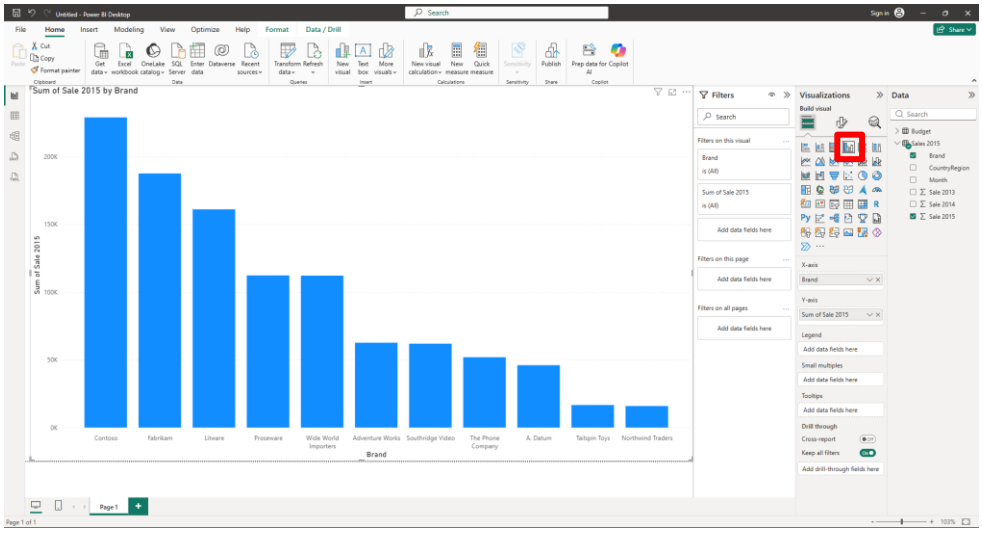
1. Select **New Source ➔ Excel Workbook** in the **Home** ribbon tab



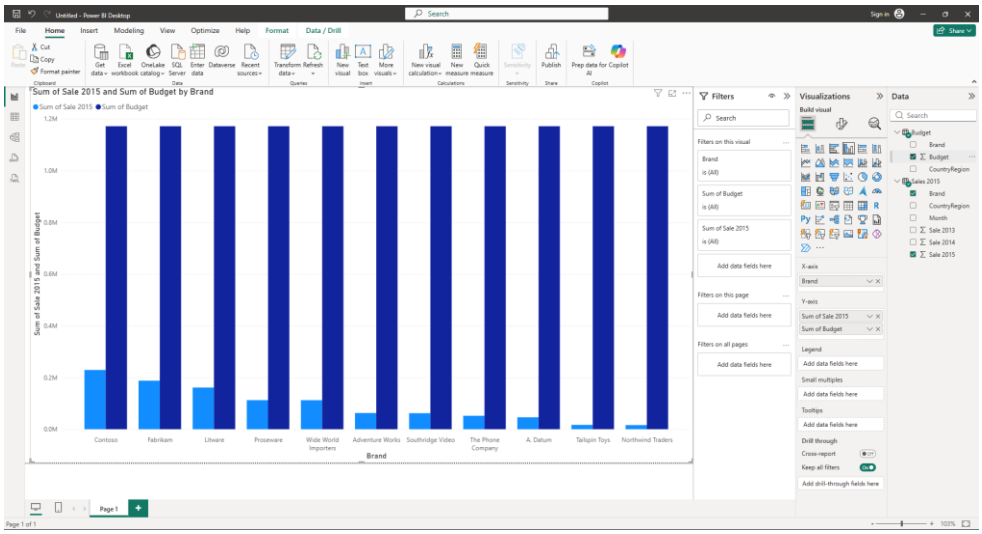
2. In the **Get Data (Power Query)** dialog box, select the *2015 Budget.xlsx* file. Then in the **Navigator** pane, select the **Budget** table and then select **[Load]**.



3. The file should be loaded successfully to the PowerBI. Select **[Close and Apply]** to confirm. Then create a **Clustered Column Chart** by selecting *Brand* and *Sales 2015* from *Sales 2015* Query.

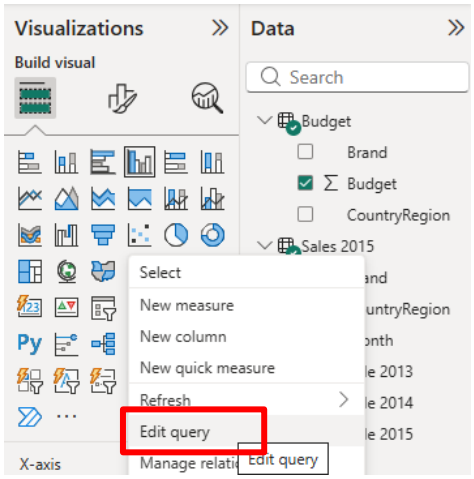


4. Select *Budget* from *Budget 2015*. However, the budget is failed to load.

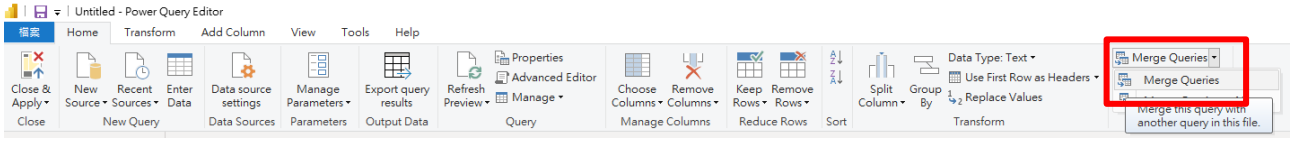


2.3 Merge Query

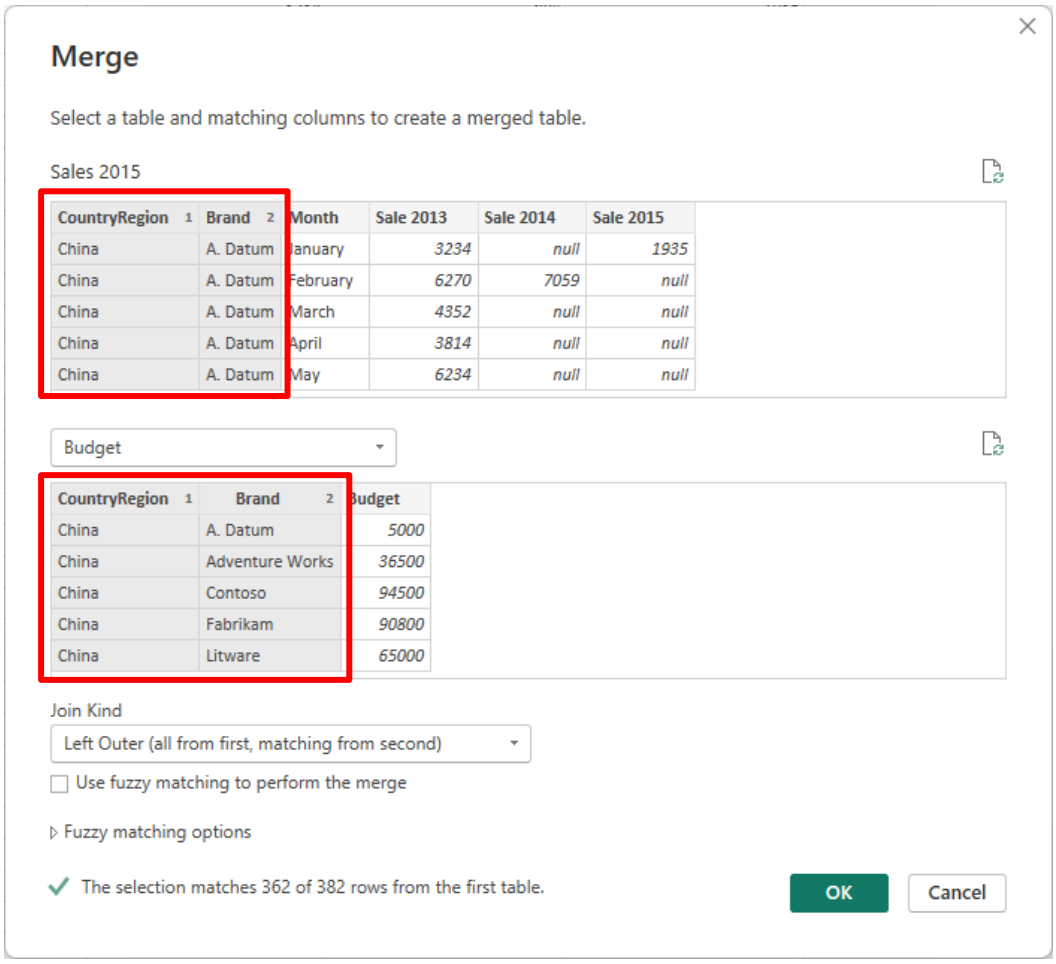
1. Right click the query *Sales 2015*, and then select **Edit query**.



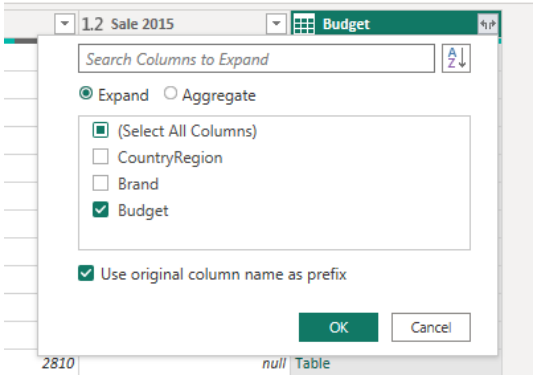
2. Select **Merge Query ➔ Merge Query** in **Home** tab in the Query Editor



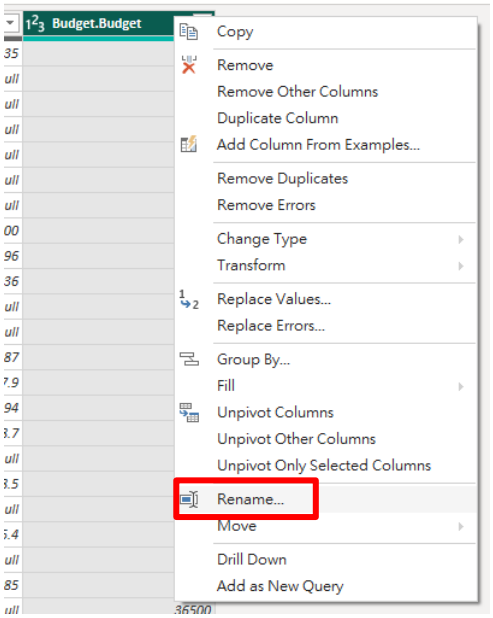
3. Select field *CountryRegion* and *Brand* in query *Sales 2015,* and also select field *CountryRegion* and *Brand* in query *Budget*. Then select **Left Outer (all from first, matching from second)** as **Join Kind** and press **[OK]**.



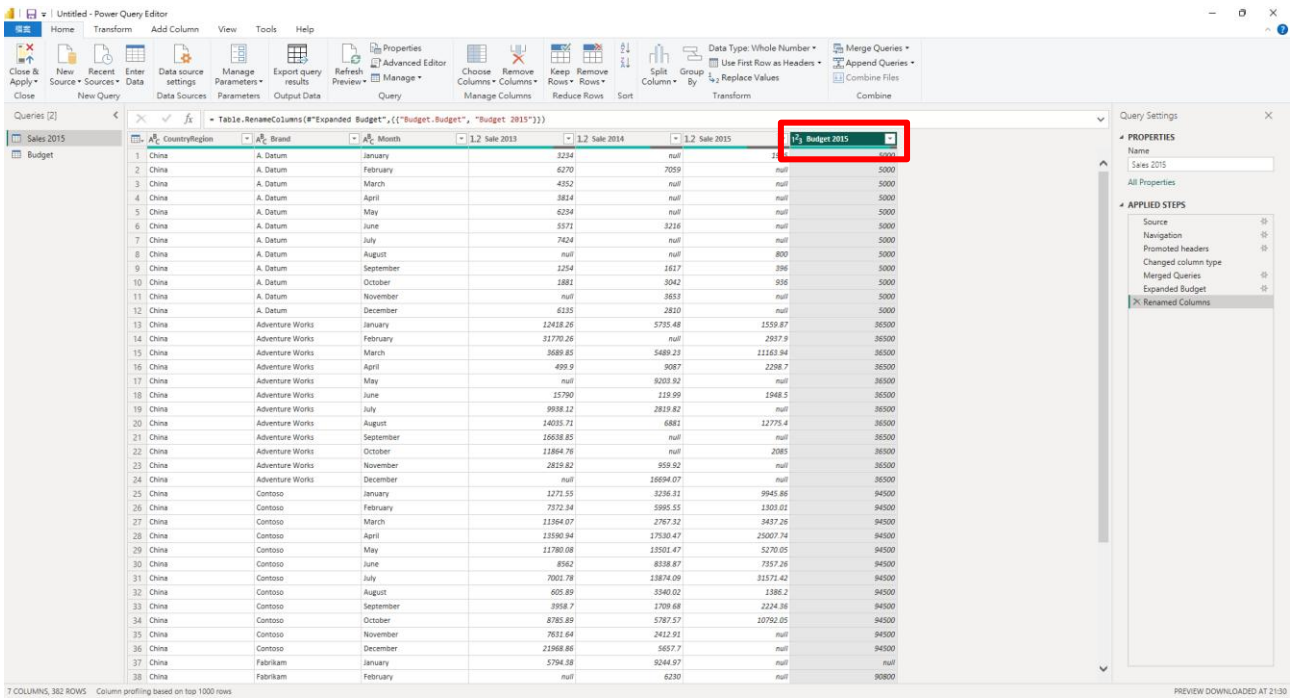
4. Expand to *Budget* column and select *Budget* field only, and press **[OK]**.



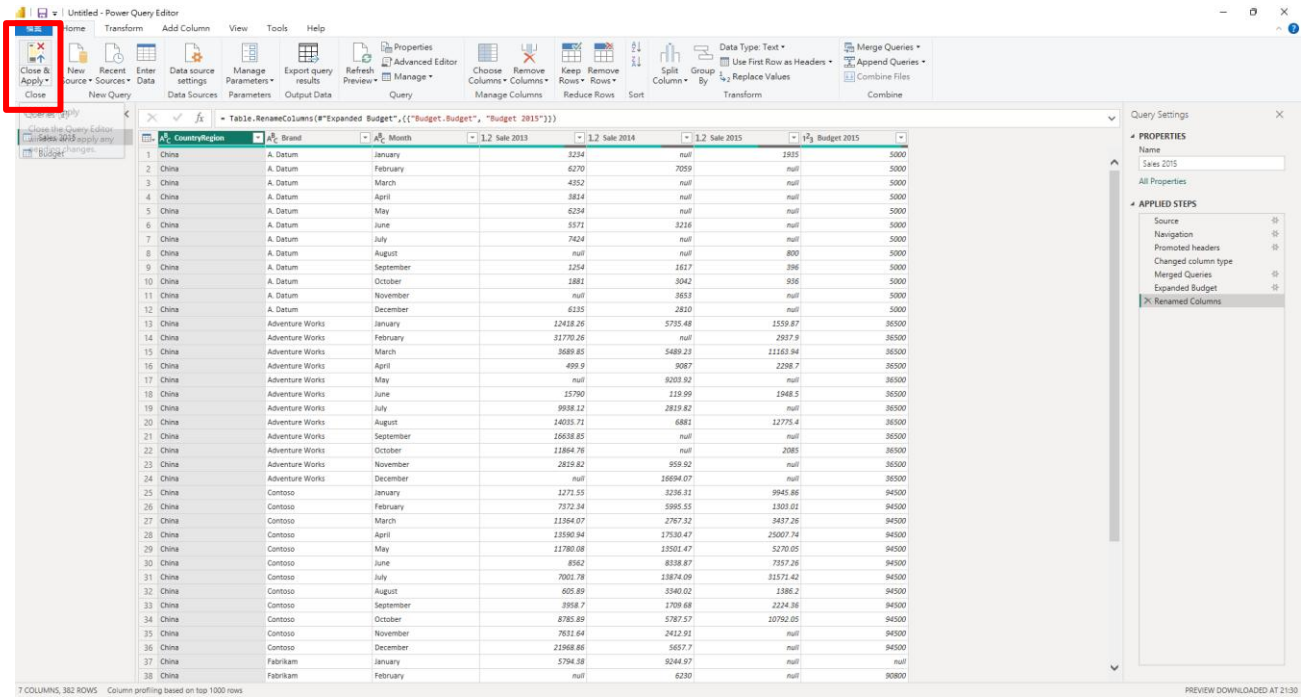
5. Right click the column *Budget.Budget* and select **Rename**.



6. Change the column name to **Budget 2015**.

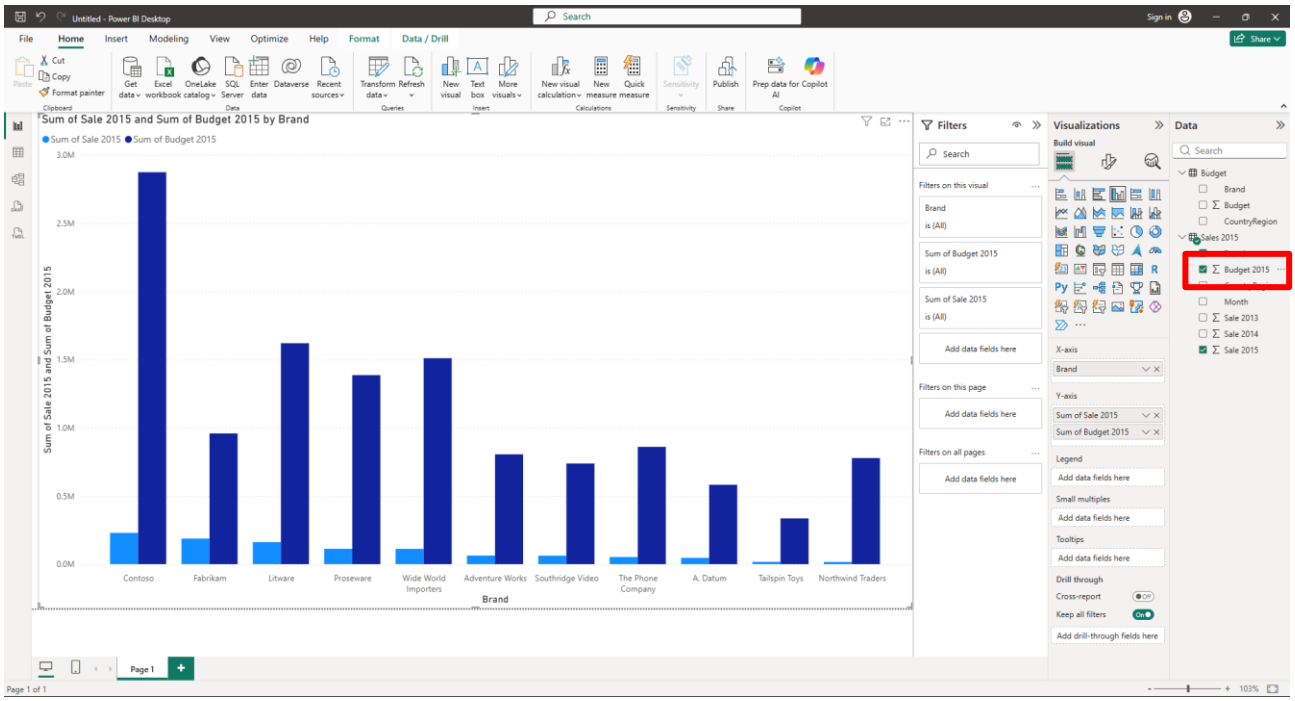


7. Press **[Close & Apply]**.



2.4 Create Combined Report

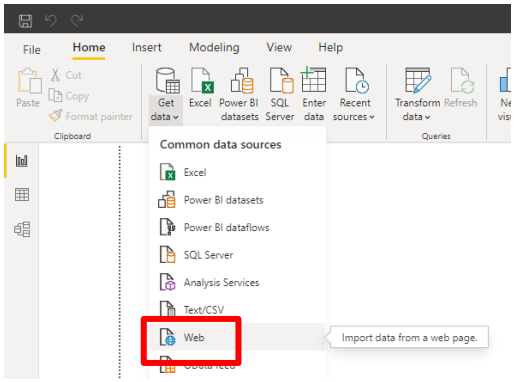
1. Replace the value from *Budget* ➔ *Budget* to *Sales* ➔ *Budget 2015* to the Sales Report



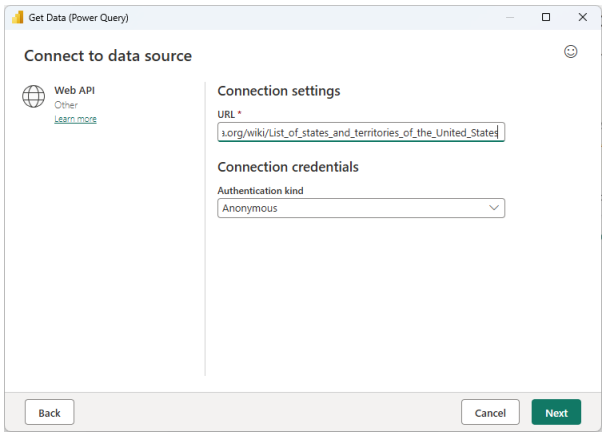
3. Add New Column from Example

3.1 Import Data from Web

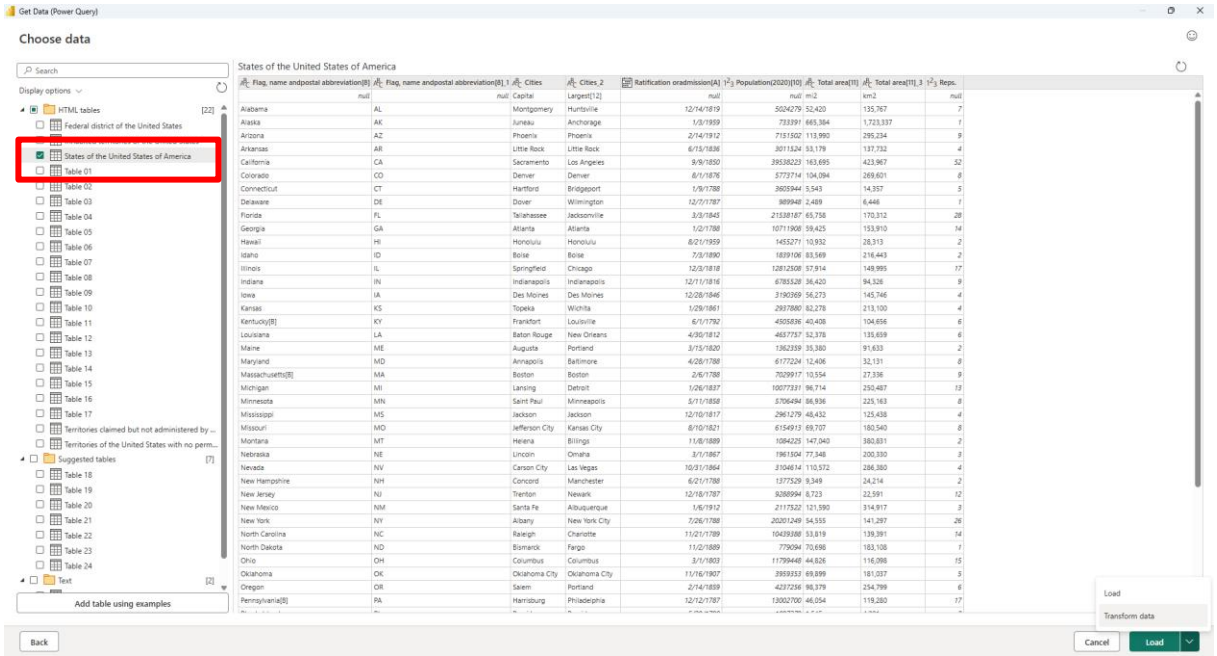
1. Select **Get Data ➔ Web** in the **Home** ribbon tab.



2. Type https://wikipedia.org/wiki/List_of_states_and_territories_of_the_United_States in the URL to import a Tournament Summary table from the UEFA European Football Championship Wikipedia, and click **[Next]** button to continue

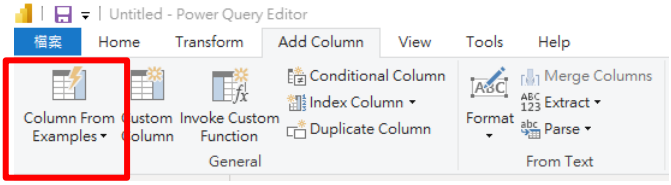


3. Select the *States of the United States of America* table and select **[Load] ➔ [Transform Data]**.

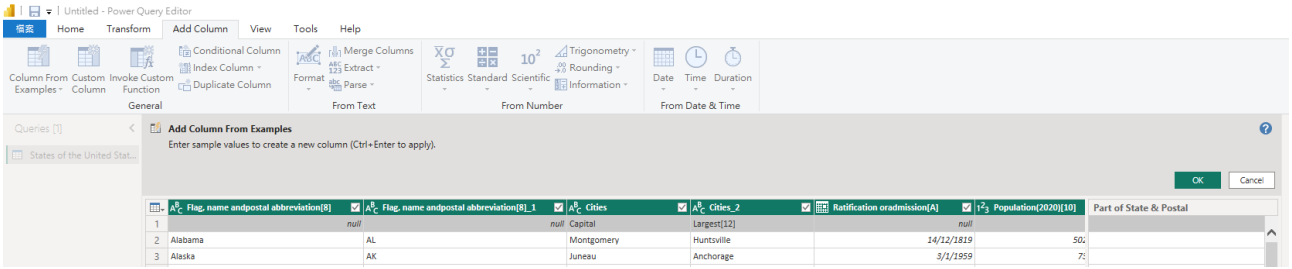


3.2 Add a New Column from Examples

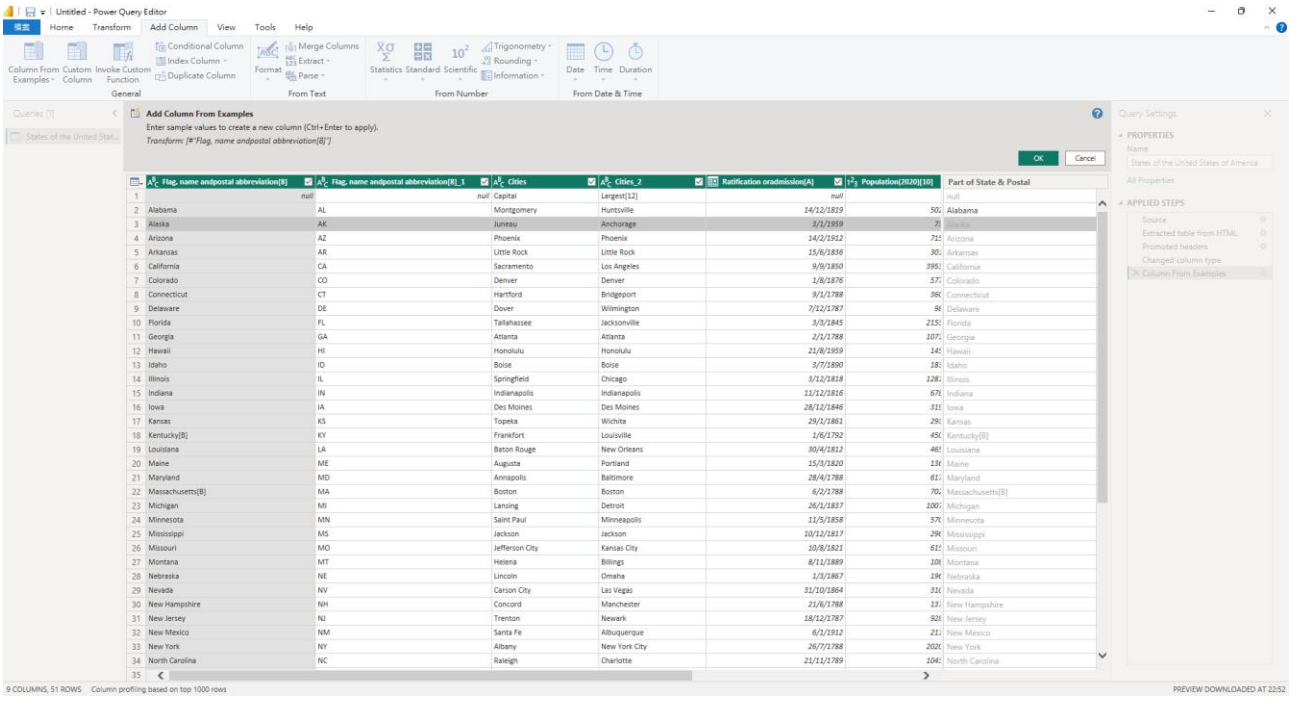
1. Once **Query Editor** is launched and you have some data loaded, you can get started adding a column from examples. To add a new column, select the **Add Column** tab on the ribbon and select **Column from Examples**.



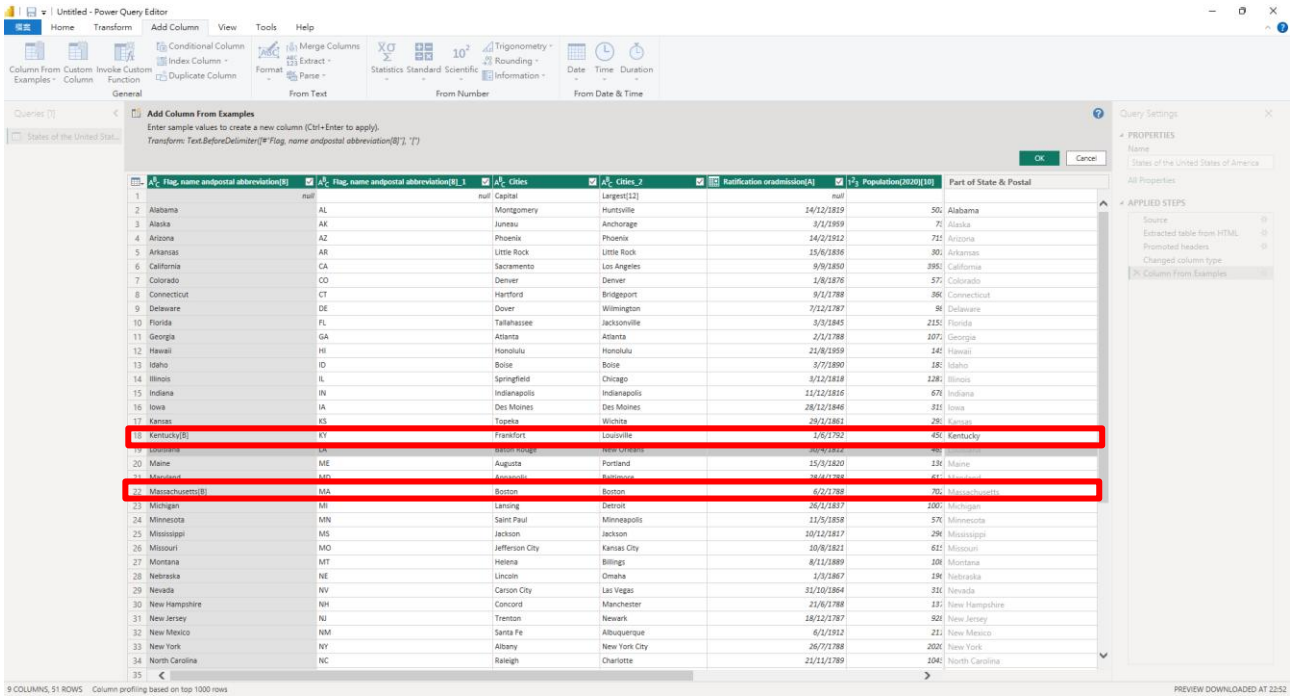
2. Double click **Column1** and change the column name to "*Part of State & Postal*"



3. Once you type your example in the new column, Power BI gives you a preview of how the column it's creating will appear, based on the transformations it detects. For example, we typed Alabama in the first row, corresponding to the Alabama value in the first column of the table. As soon as we hit **[Enter]** Power BI fills in the column based on that value.



4. Then we went to the row that included Kentucky[B] and deleted that last [B] portion and Power BI detected the change, and used the example to create a transformation. Press **[OK]** once you confirm your input.

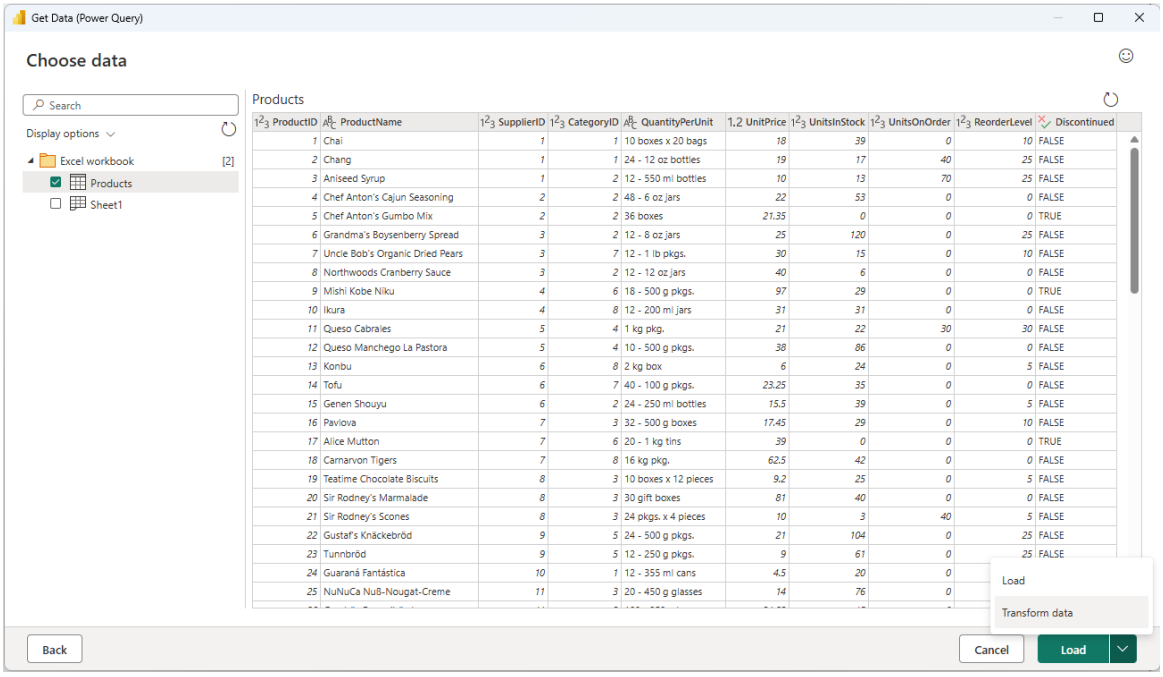


4. Create Custom Column

4.1 Import Product Data from Excel

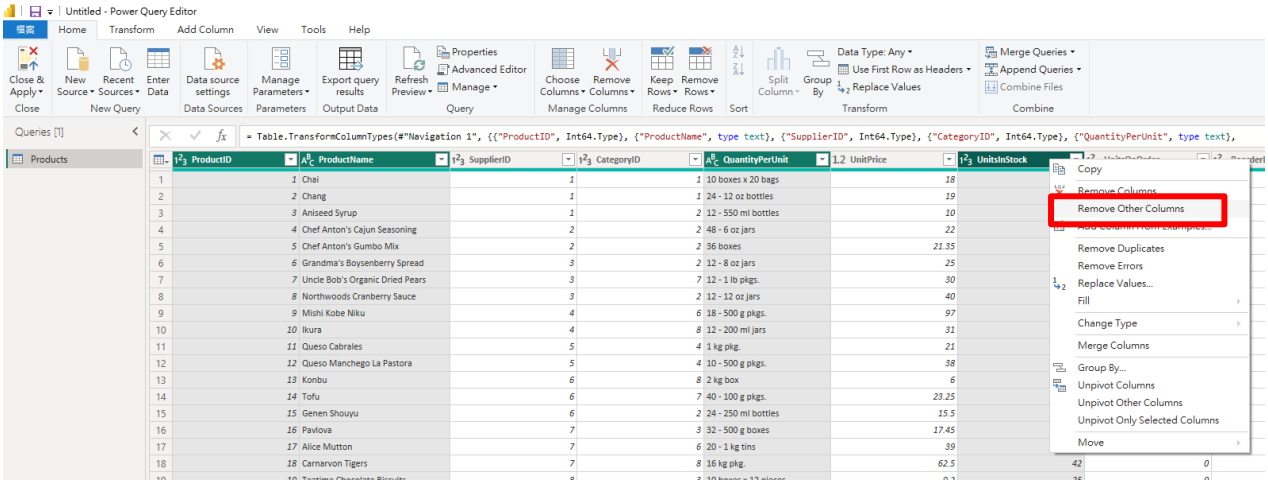
4.1.1 Connect to Excel Workbook

1. Select **Get Data ➔ Excel Workbook** in the **Home** ribbon tab
2. In the **Get Data (Power Query)** dialog box, select the *product.xlsx* file. Then in the **Navigator** pane, select the **Products** table and then select **[Transform Data]**.



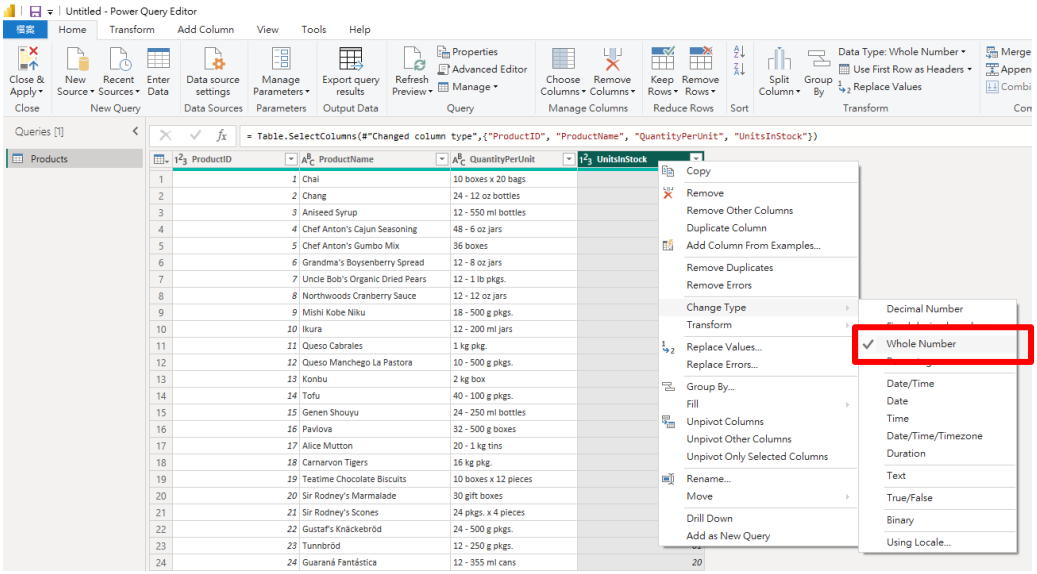
4.1.2 Remove other Columns to only Display Columns of Interest

1. In **Power Query Editor**, select the *ProductID*, *ProductName*, *QuantityPerUnit* and *UnitsInStock* columns.
2. Right-click on a column header and click **Remove Other Columns**.



4.1.3 Change the Data Type of Column

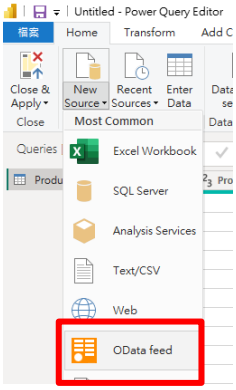
1. Select the *UnitsInStock* column.
2. Right-click on column header and select **Change Type ➔ Whole Number**.



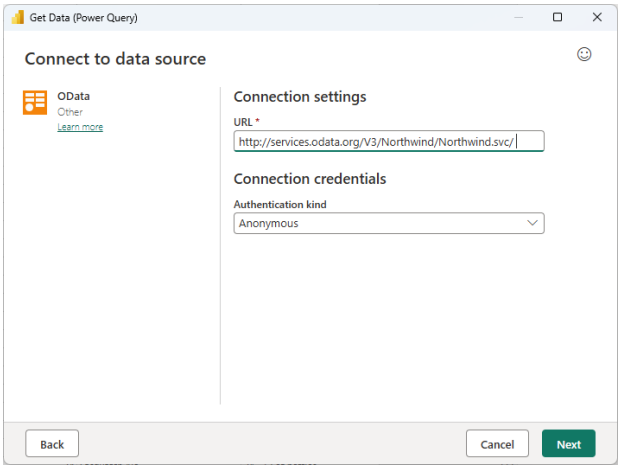
4.2 Import Order Data from OData Feed

4.2.1 Connect to OData Feed

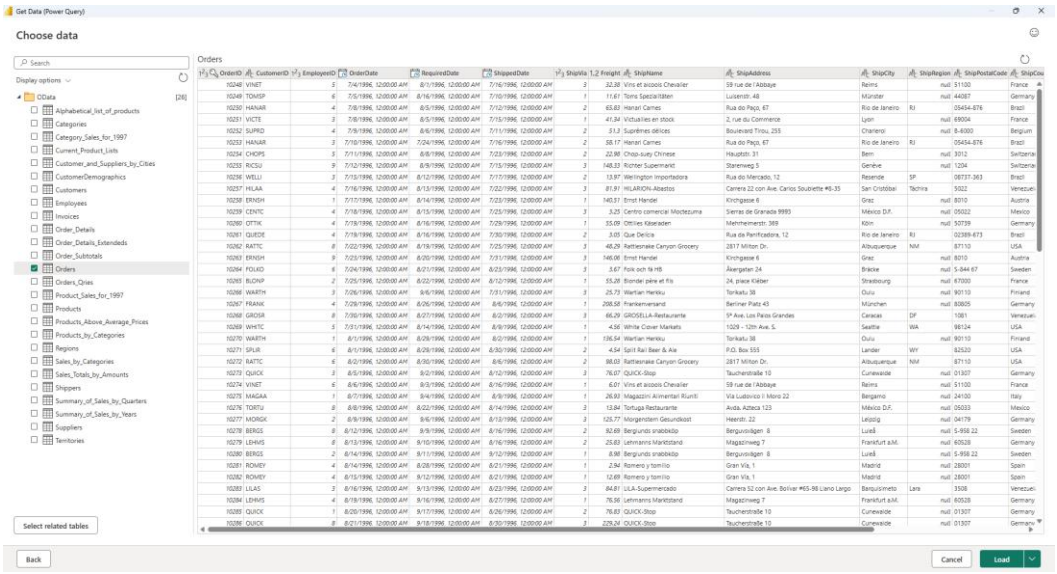
1. From the Home ribbon tab in Query Editor, select **New Source ➔ OData feed**.



2. Input http://services.odata.org/V3/Northwind/Northwind.svc/ in the **URL** field of the **Get Data (Power Query)** dialog box and press **[Next]** button.

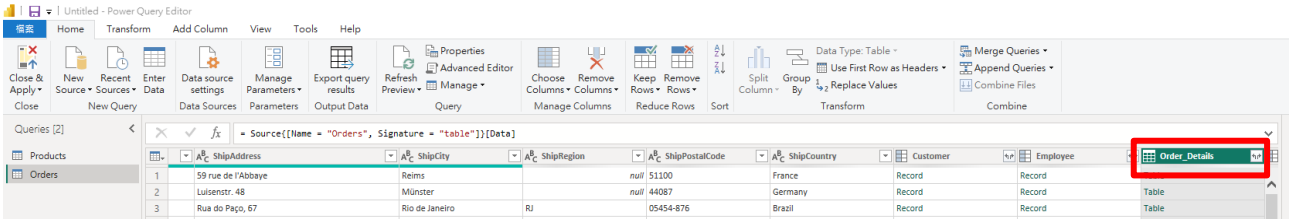


3. In the **Choose data** pane, select the *Orders* table, and then select **[Load]**.

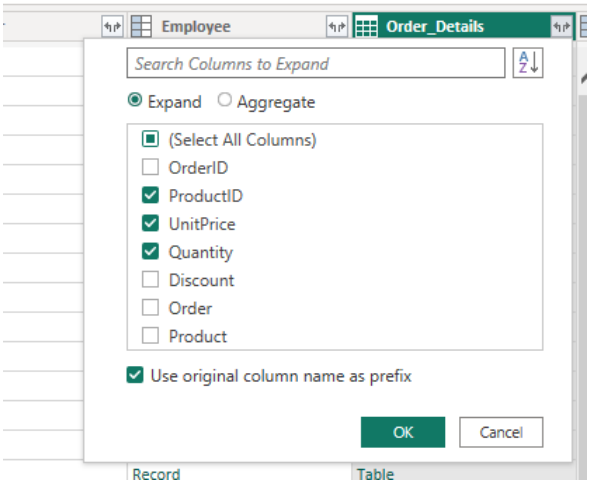


4.2.2 Expand the Order Details Table

1. In the Query View, scroll to the *Order_Details* column and select the expand icon .



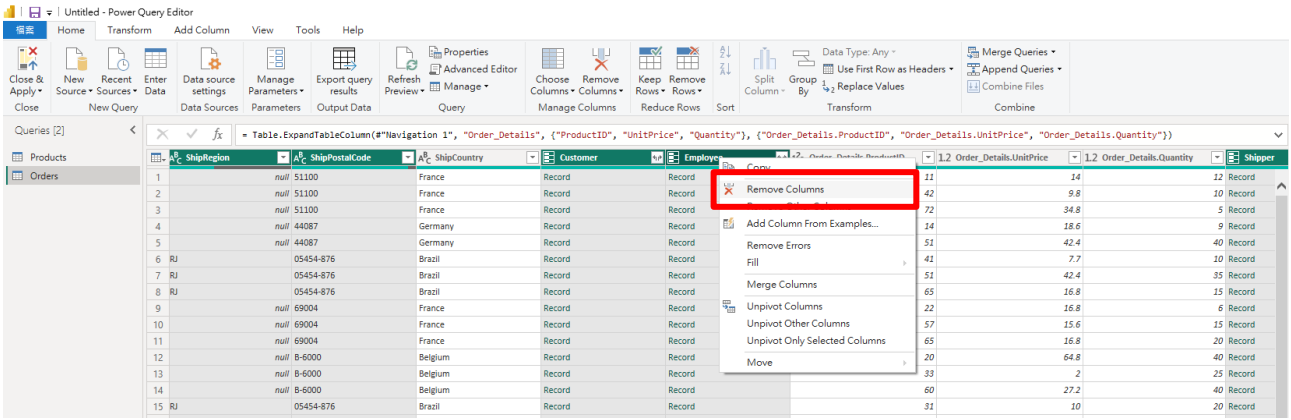
2. Select *ProductID*, *UnitPrice*, and *Quantity* in the Expand drop-down, and then click **[OK]**.



4.2.3 Remove Other Columns to only Display Columns of Interest

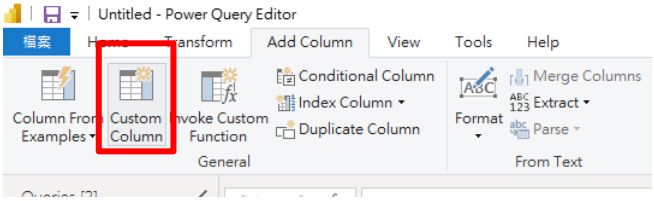
1. In the Query View, select all columns by clicking the first column (*OrderID*), and then press **[Shift]** to click the last column (*Shipper*).
2. Press **[Ctrl]** and click to unselect the following columns: *OrderDate*, *ShipCity*, *ShipCountry*, *Order_Details.ProductID*, *Order_Details.UnitPrice* and *Order_Details.Quantity*.

3. Right-click on any selected column header and click **Remove Columns**.

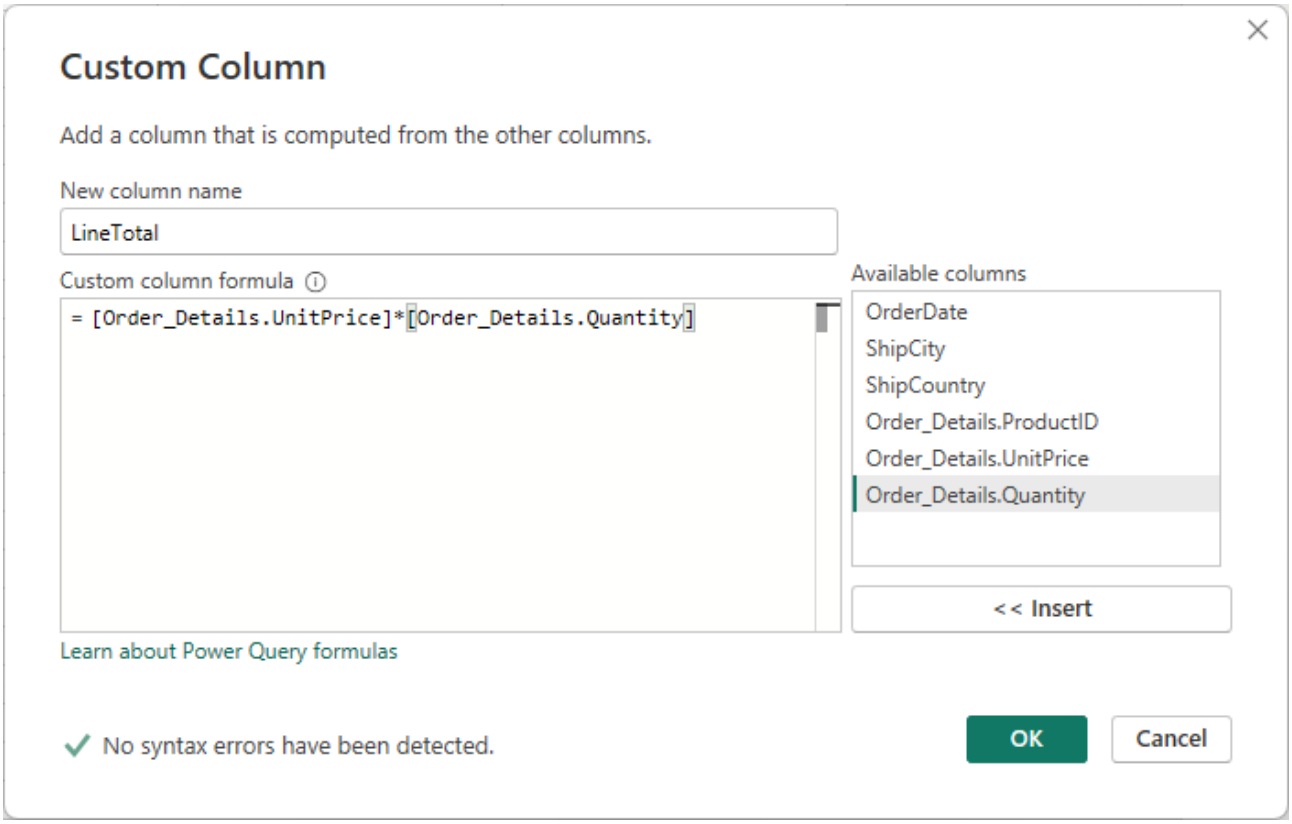


4.2.4 Calculate the Line Total for each Order_Details row

1. In the **Add Column** ribbon tab, click **Custom Column**.

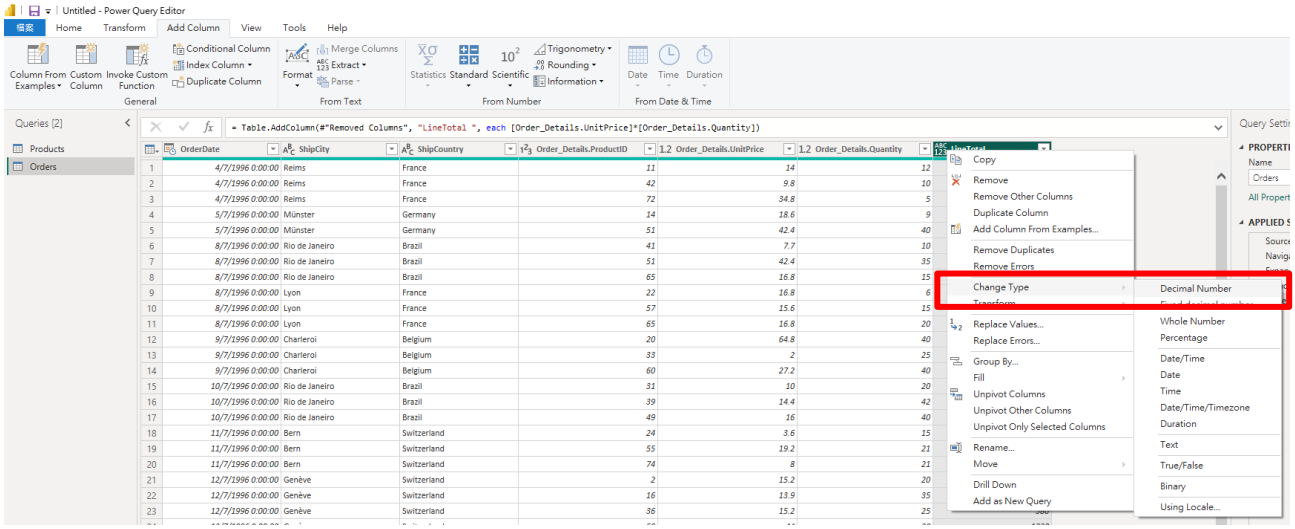


2. In the **Custom Column** dialog box, enter *[Order_Details.UnitPrice] * [Order_Details.Quantity]* in the **Custom column formula** textbox. Then enter *LineTotal* in the **New column name** textbox. Click **[OK]** to continue.



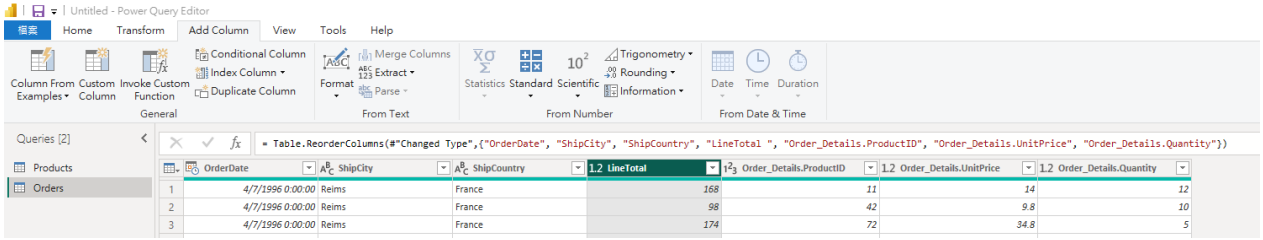
4.2.5 Set the Data Type of the LineTotal field

1. Right click the *LineTotal* column and select **Change Type ➔ Decimal Number**.

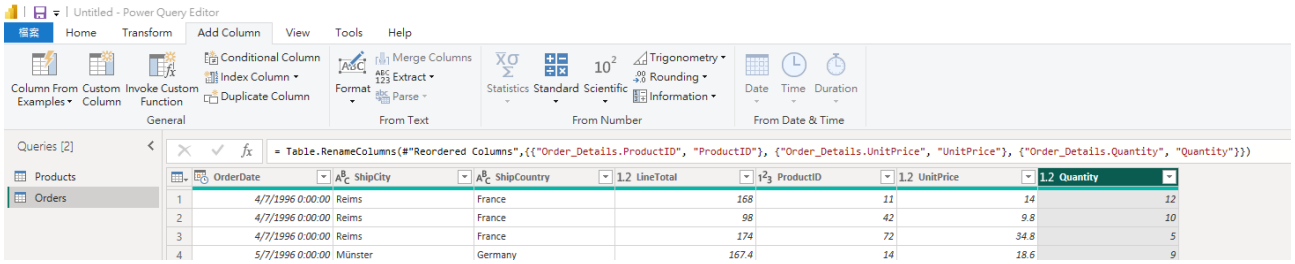


4.2.6 Rename and Reorder Columns in the Query

1. In **Power Query Editor**, drag the *LineTotal* column to the left, after *ShipCountry*.



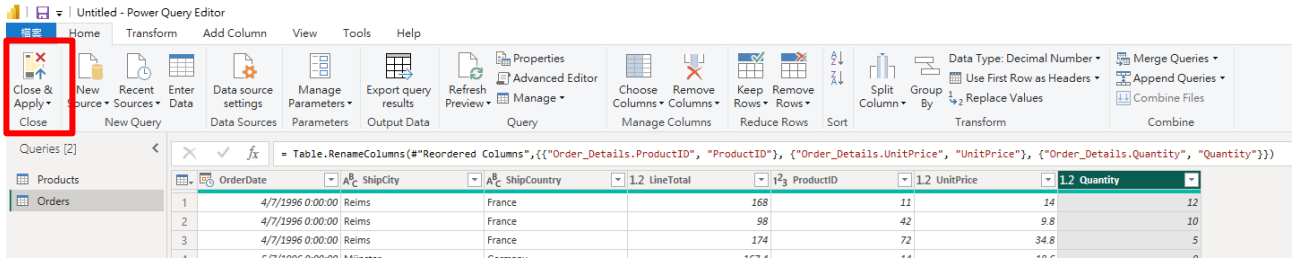
2. Rename the column name by removing the *Order_Details.* prefix from the *Order_Details.ProductID, Order_Details.UnitPrice* and *Order_Details.Quantity* columns, by double-clicking on each column header, and then deleting that text from the column name.



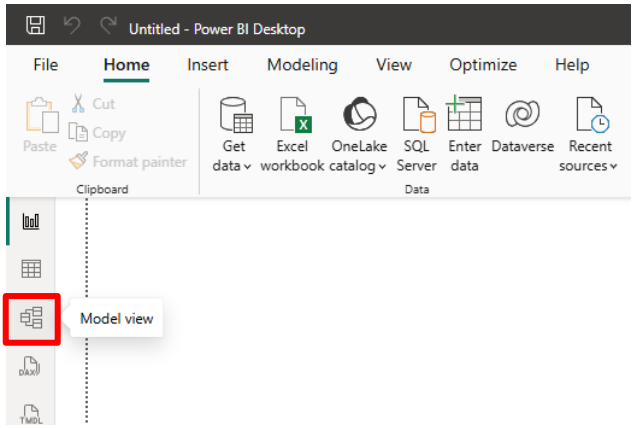
4.3 Combine the Products and Total Sales Queries

4.3.1 Confirm the Relationship between Products and Total Sales

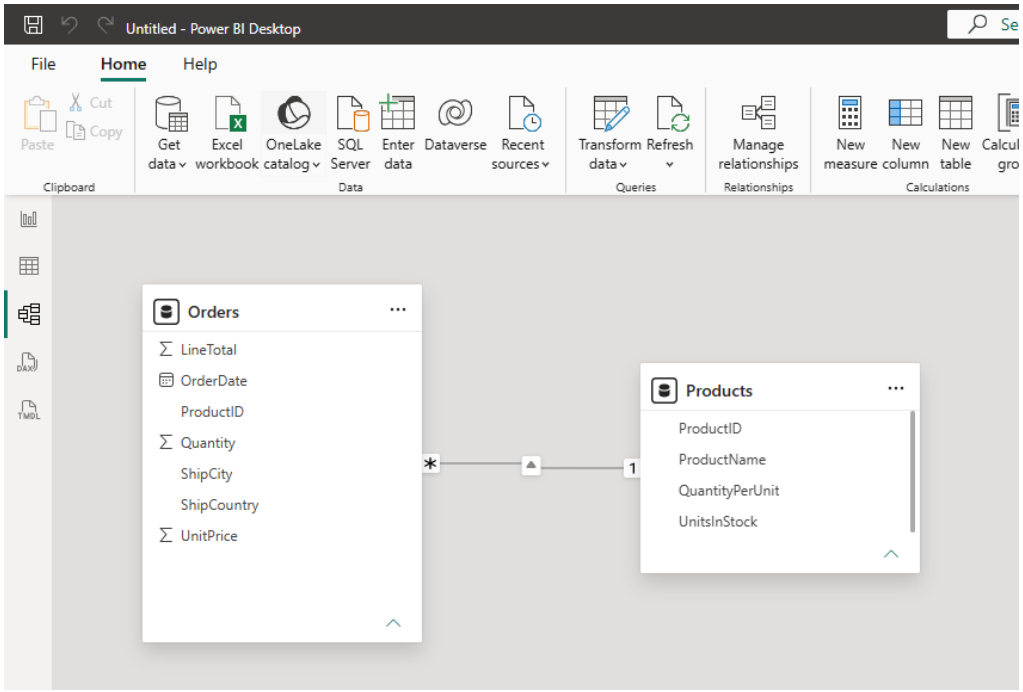
1. From the **Home** ribbon of **Query Editor**, select **[Close & Apply]** to load the model that we created in **Power Query Editor** into Power BI Desktop.



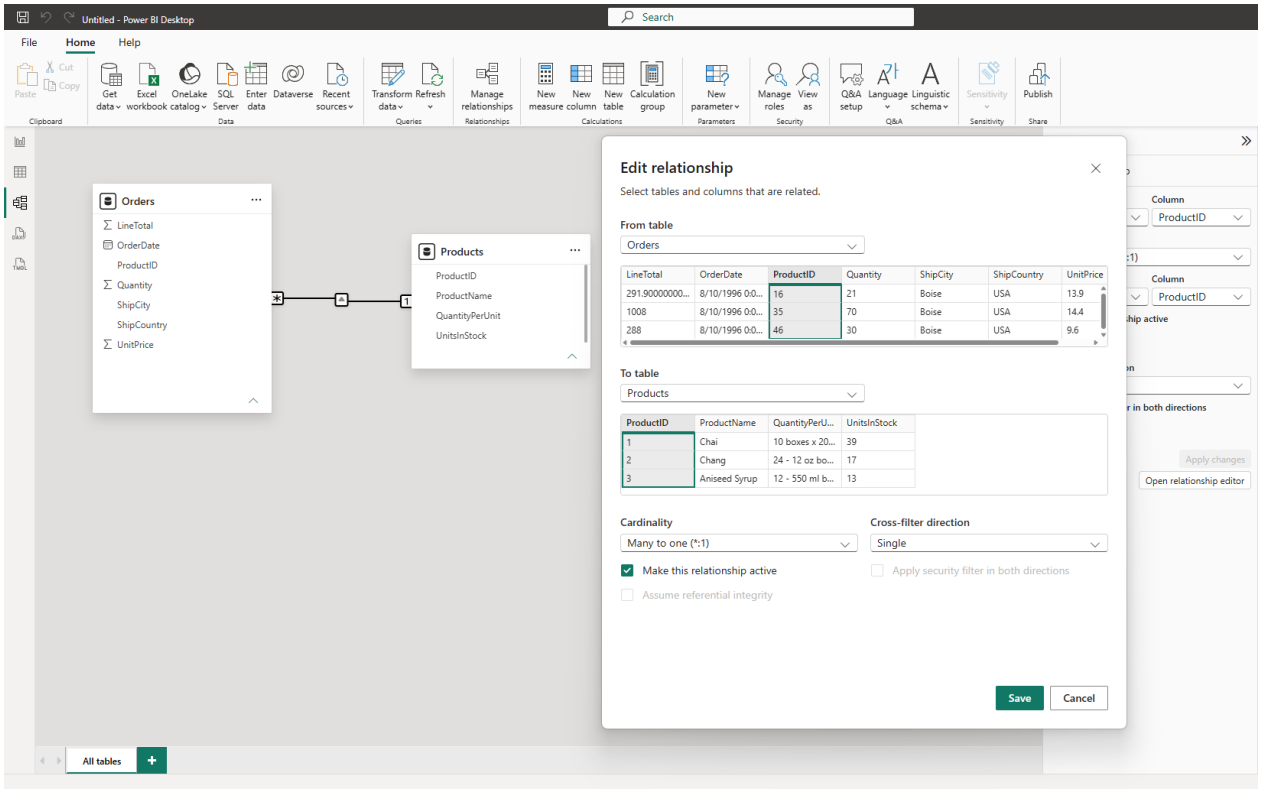
2. Once the data is loaded, Select **Model** view.



3. The relationship between the queries is visualized



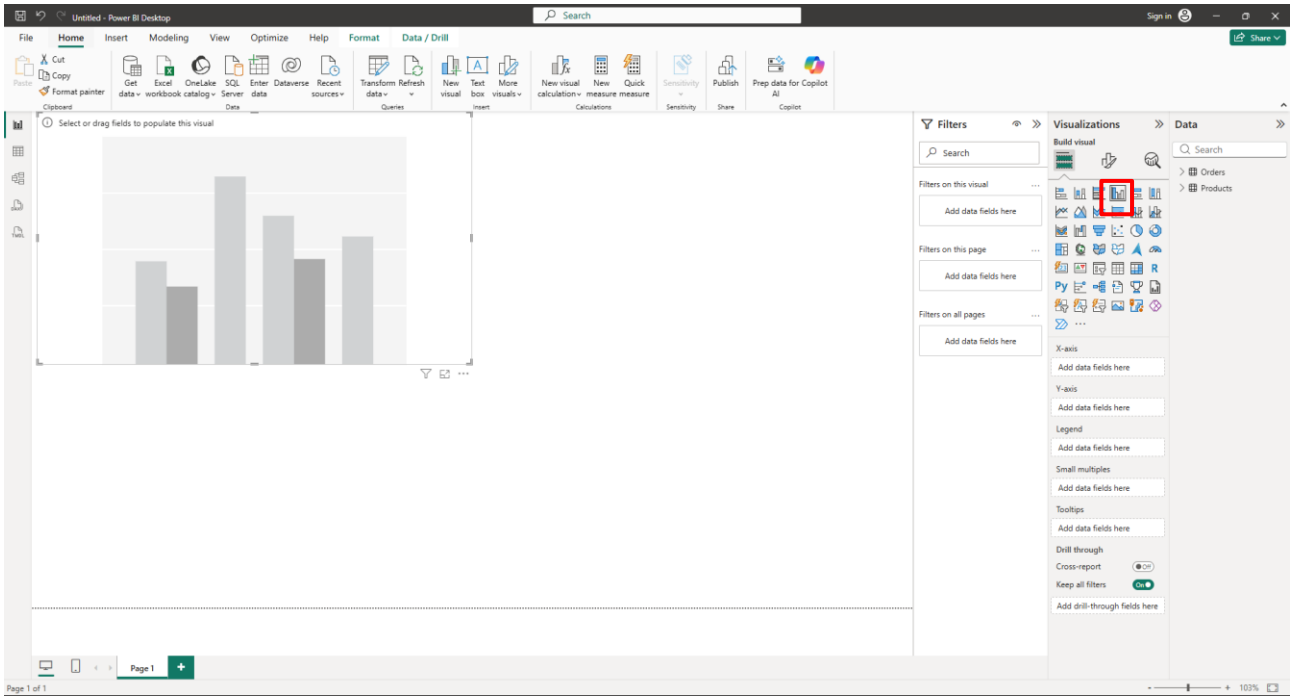
4. When you double-click the arrow on the line that connects them to queries, an **Edit Relationship** dialog appears. Since we don't need to make any changes, we'll just select **[Cancel]** to close the **Edit Relationship** dialog.



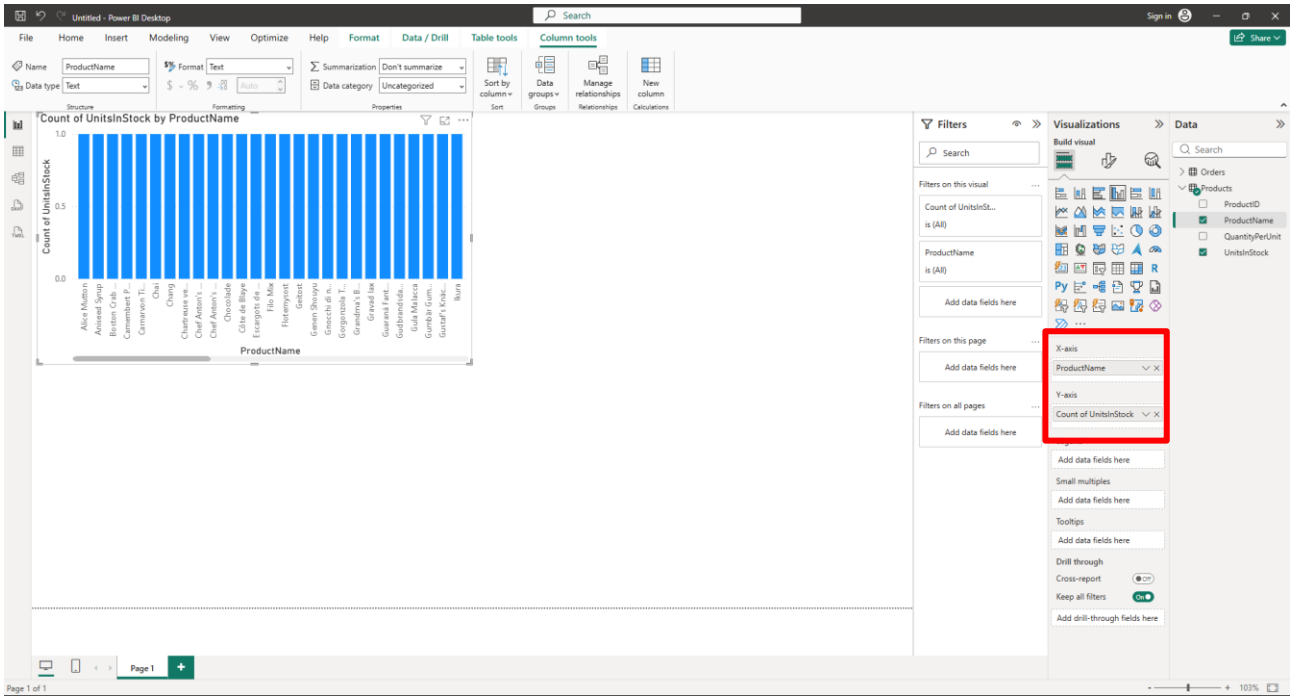
4.4 Build Visuals Using your Data

4.4.1 Clustered Column Chart (Units in Stock by Product)

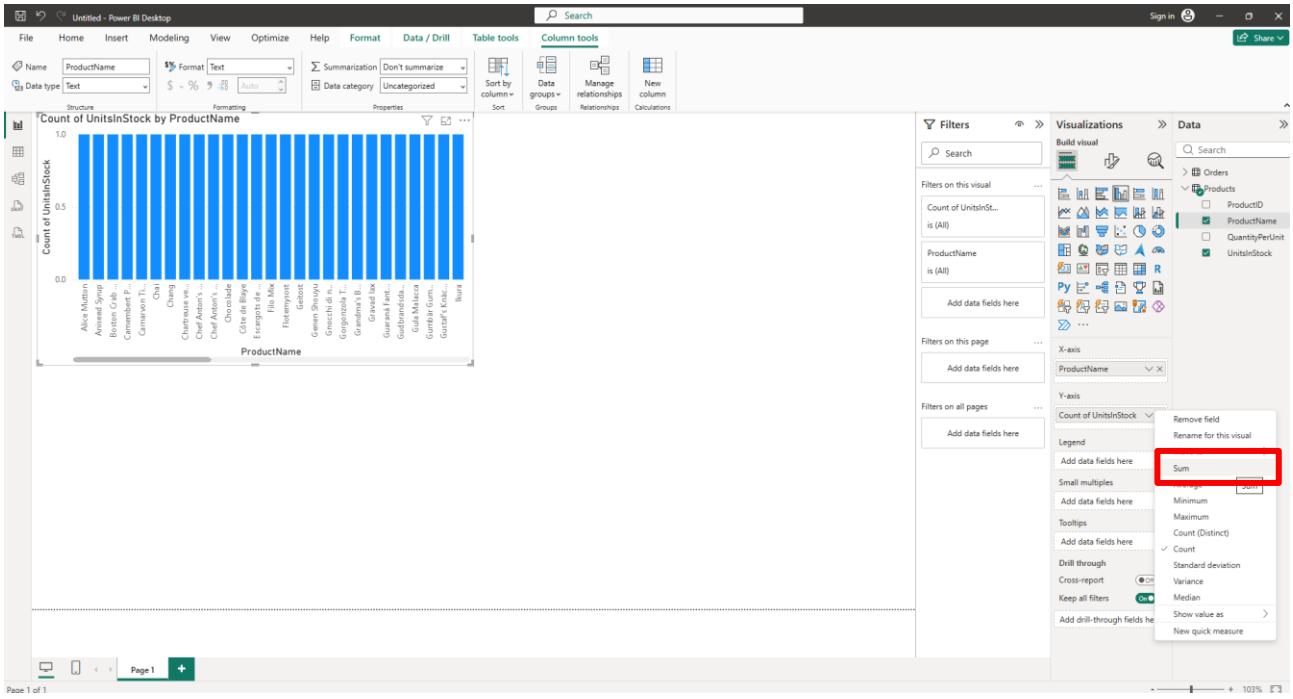
1. Add a **Clustered Column Chart** visual and then resize and reposition it.



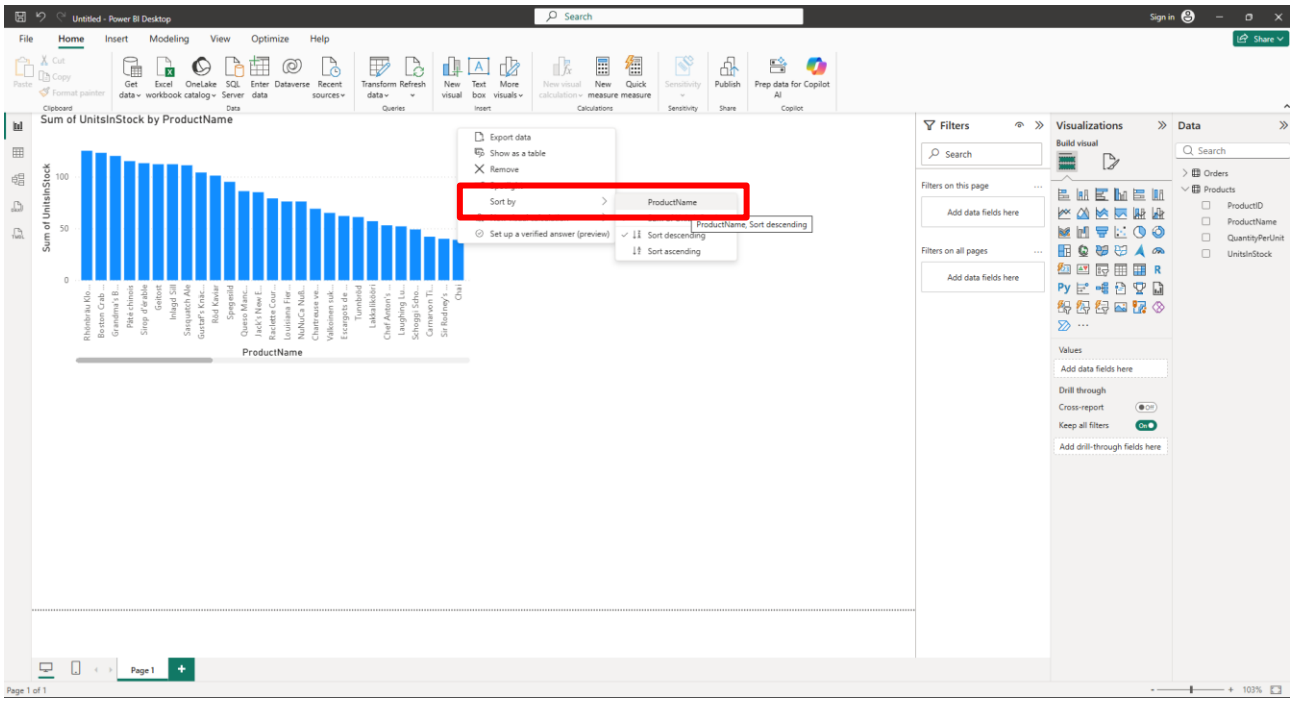
2. Add the following fields to the visual wells:
   - Assign *Product* ➔ *ProductName* to **X-axis**
   - Assign *Product* ➔ *UnitsInStock* to **Y-axis**



3. Change the *Count of UnitsInStock* in **Y-Axis** to **Sum**

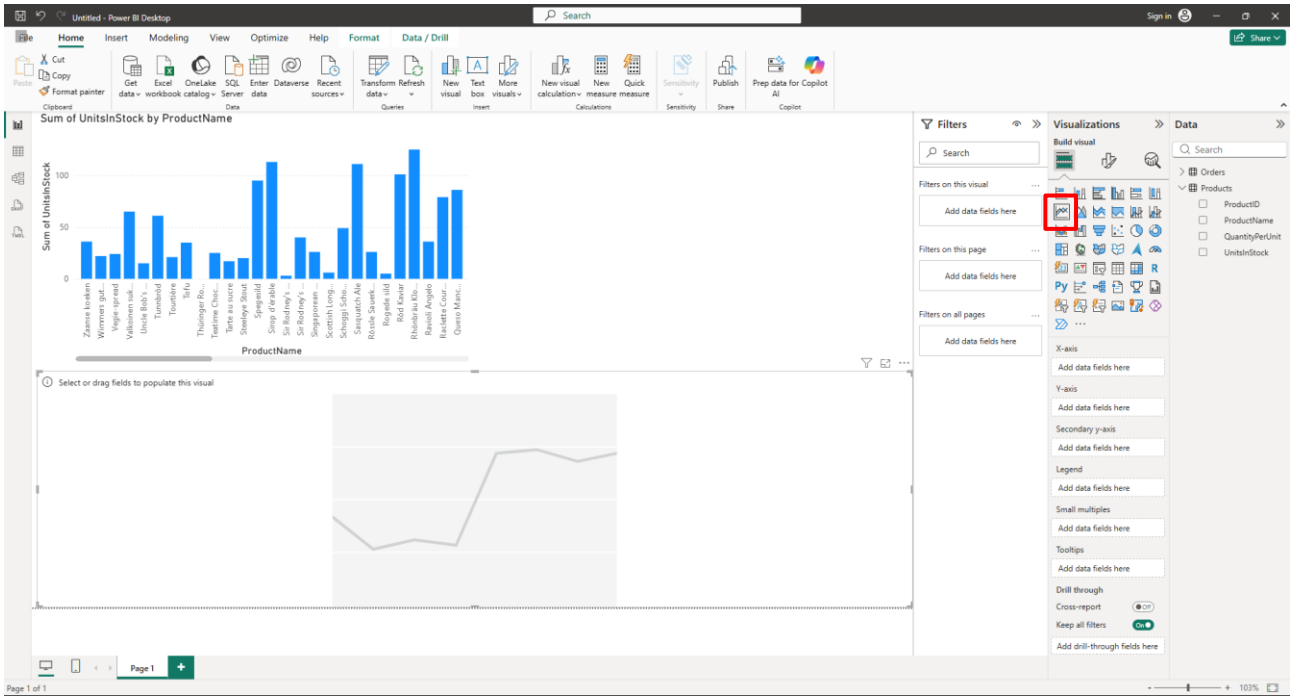


4. Select **Sort By** ➔ **ProductName** in ascending order using the skittles in the top right corner of the visualization.

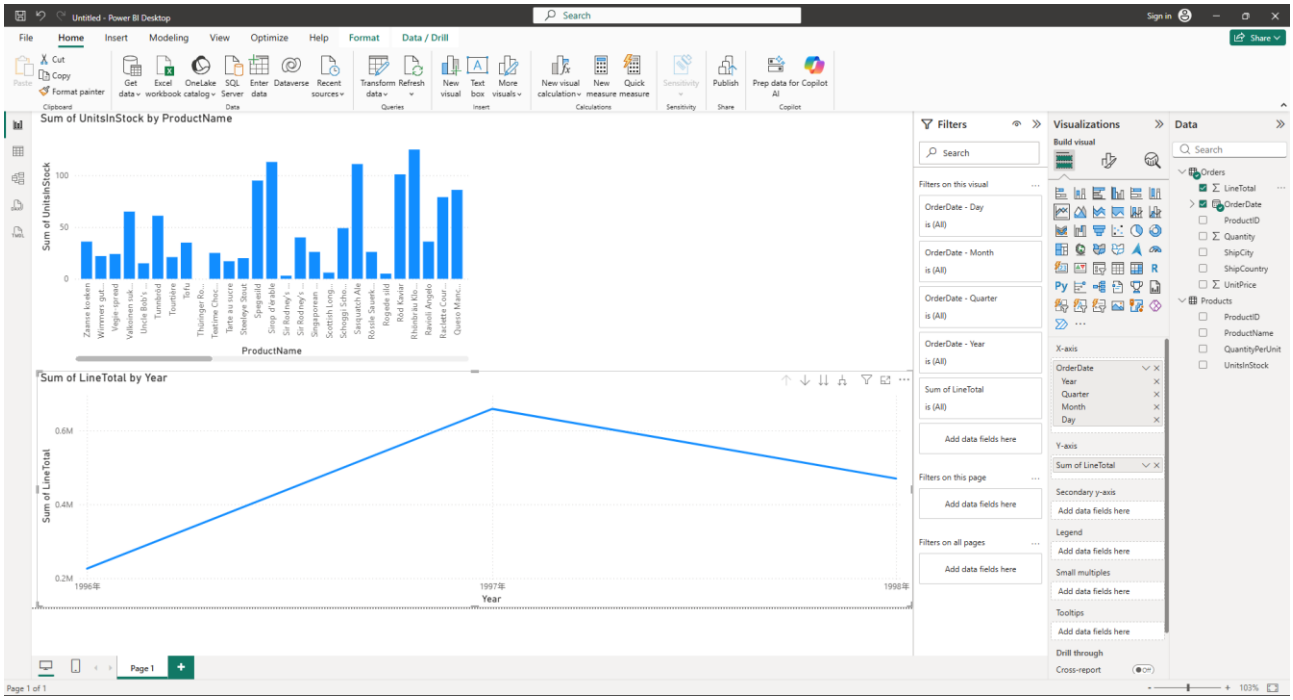


4.4.2 Line Chart (Total Sales by Year)

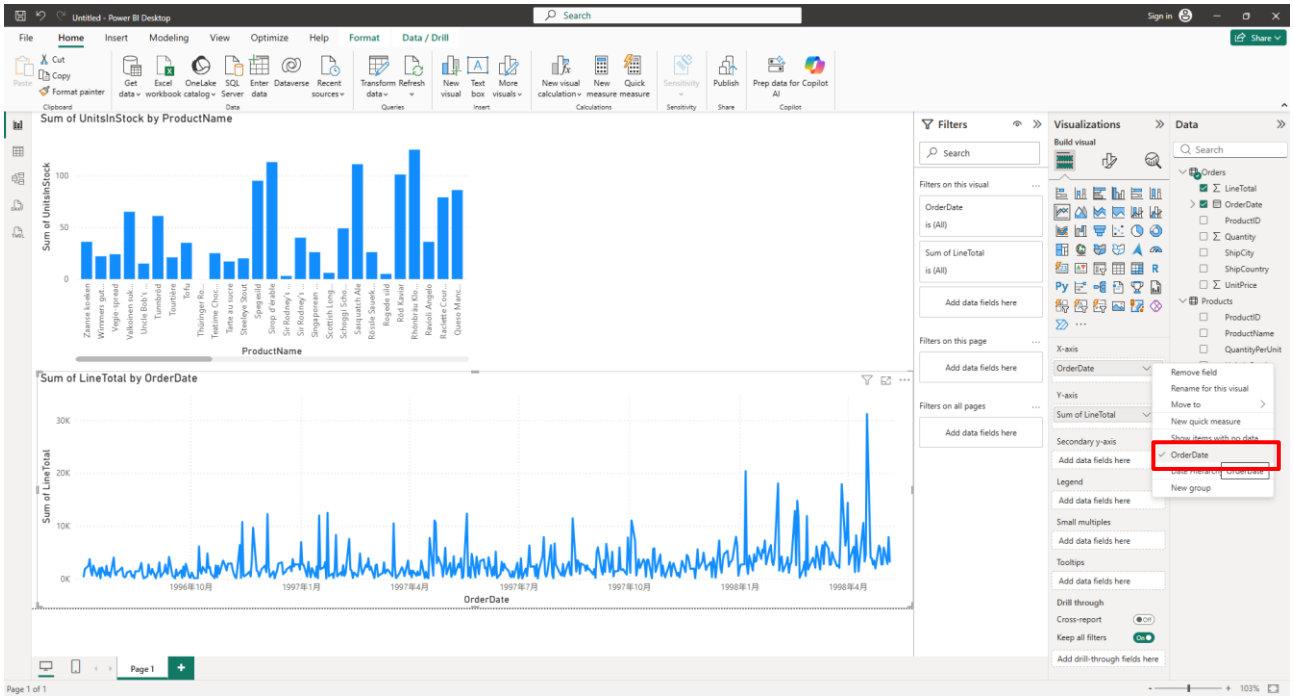
1. Add a **Line Chart** visual under the column chart, then resize and reposition it.



2. Add the following fields to the visual wells:
   - Assign *Order* ➔ *OrderDate* to **X-axis**
   - Assign *Order* ➔ *LineTotal* to **Y-axis**

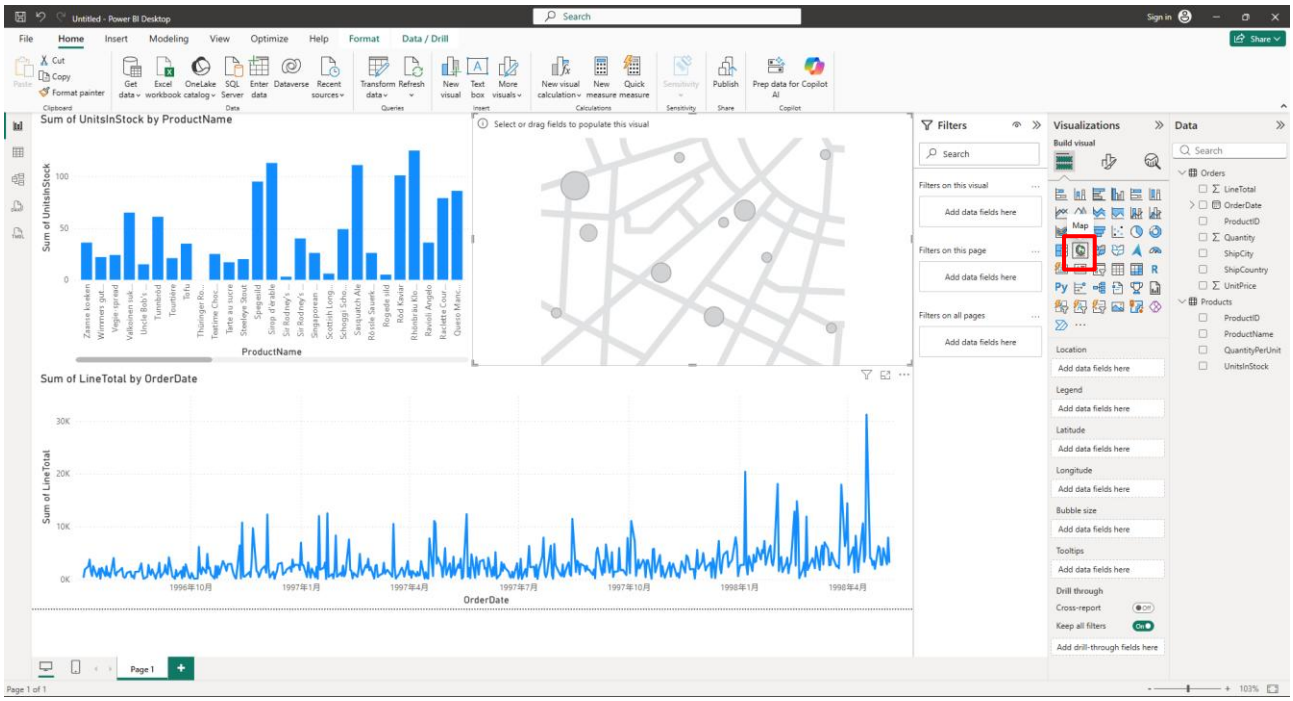


3. Select *OrderDate* in Axis and select **OrderDate**.

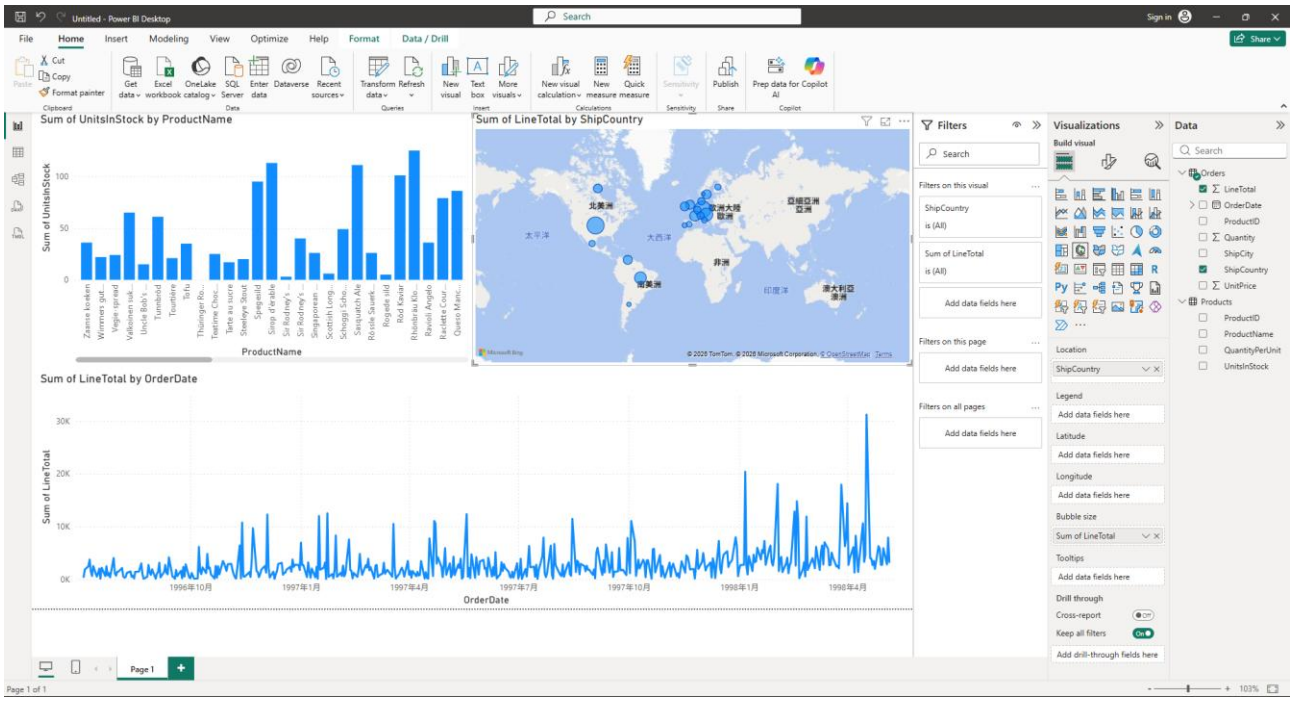


4.4.3 Map (Total Sales per Country)

1. Add a **Map** visual beside the column chart, then resize and reposition it.

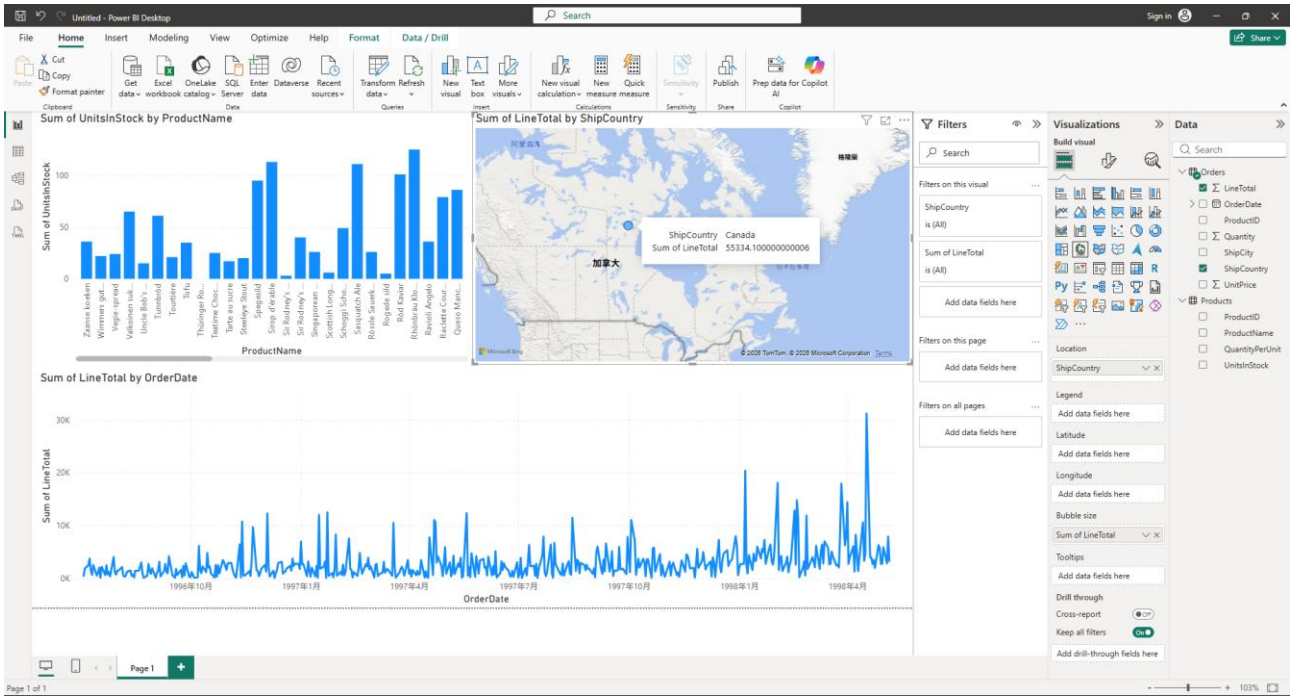


2. Add the following fields to the visual wells:
   - Assign *Order* ➔ *ShipCountry* to **Location**
   - Assign *Order* ➔ *LineTotal* to **Bubble Size**



4.4.4 Interact with Report Visuals

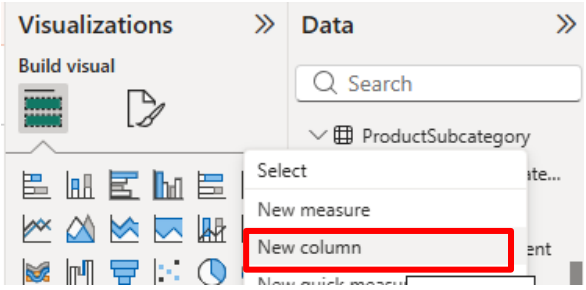
1. Click on the light blue circle centered in Canada. Note how the other visuals are filtered to show Stock (*ShipCountry*) and Total Orders (*LineTotal*) just for Canada.



5. Create Calculated Columns

5.1 Create a Custom Column

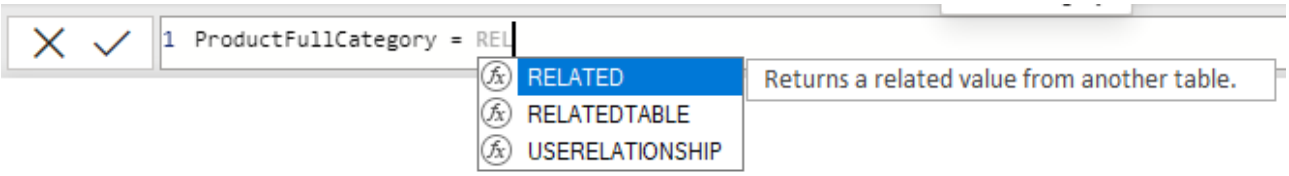
1. Download the Contoso Sales for Power BI sample designer file.
2. Select **File ➔ Open Report**, and open *Contoso Sales Sample for Power BI Desktop.pbix*
3. Click the down arrow on the *ProductSubcategory* table in the Fields list, and then click **New column**. This will make sure our new column is added to the *ProductSubcategory* table.



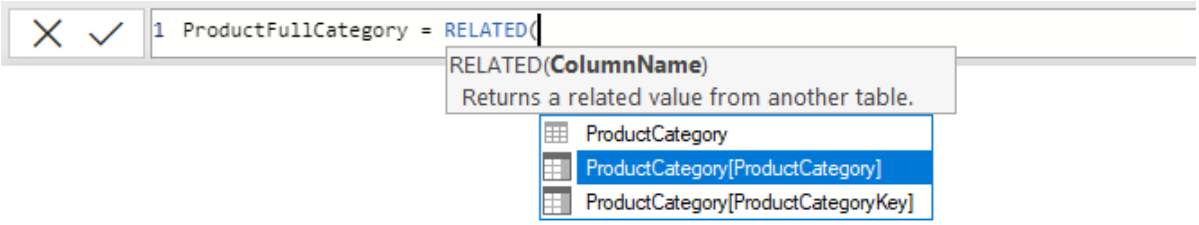
4. The formula bar appears along the top of the **Report** canvas. This is where we can rename our column and enter a DAX formula, type *ProductFullCategory*.



5. Select **RELATED** by scrolling down, and then pressing **[Enter]**.



6. An opening parenthesis appears along with another suggestion list of all of the available columns we can pass to the **RELATED** function. A description and details on what parameters are expected is also shown.



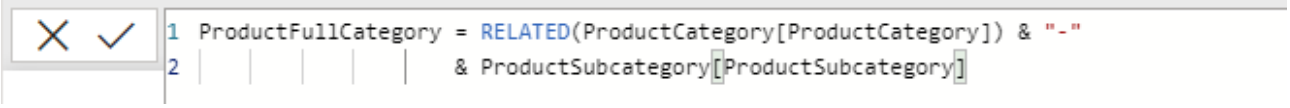
7. Select ProductCategory[ProductCategory], and then type a closing parenthesis.



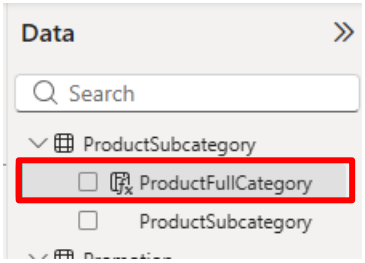
8. We want to add a dash symbol to separate each value, so after the closing parenthesis of the first expression, type a space, ampersand (&), quote, space, dash (-), another space, a closing quote, and then another ampersand.



9. Enter another opening bracket and then select the *[ProductSubcategory]* column to finish the formula. We didn't use another **RELATED** function in the second expression calling the *ProductSubcategory* column. This is because this column is already in the same table we're creating our new column in. We can enter *[ProductCategory]* with the table name (fully qualified) or without (non-qualified).

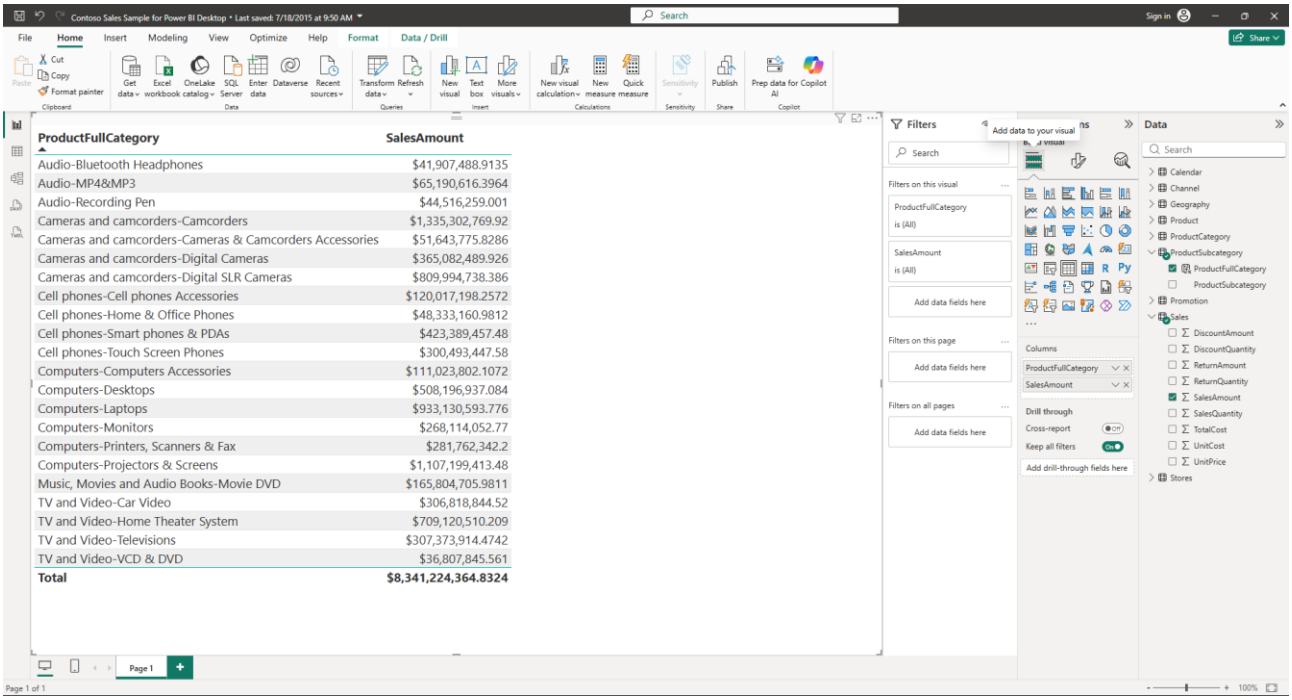


10. Complete the formula by pressing **[Enter]** or clicking on the checkmark in the formula bar. The formula is validated and added to the Data list in the *ProductSubcategory* table.



5.2 Add New Column to Report

1. Start on a blank report page, Drag *ProductSubcategory* ➔ *ProductFullCategory* onto the Report canvas, and then drag the *Sales* ➔ *SalesAmount* field into the table.

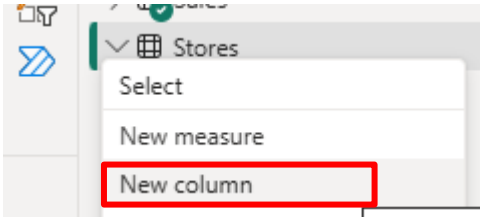


ProductFullCategory	SalesAmount
Audio-Bluetooth Headphones	$41,907,488.9135
Audio-MP4&MP3	$65,190,616.3964
Audio-Recording Pen	$44,516,259.001
Cameras and camcorders-Camcorders	$1,335,302,769.92
Cameras and camcorders-Cameras & Camcorders Accessories	$51,643,775.8286
Cameras and camcorders-Digital Cameras	$365,082,489.926
Cameras and camcorders-Digital SLR Cameras	$809,994,738.386
Cell phones-Cell phones Accessories	$120,017,198.2572
Cell phones-Home & Office Phones	$48,333,160.9812
Cell phones-Smart phones & PDAs	$423,389,457.48
Cell phones-Touch Screen Phones	$300,493,447.58
Computers-Computers Accessories	$111,023,802.1072
Computers-Desktops	$508,196,937.084
Computers-Laptops	$933,130,593.776
Computers-Monitors	$268,114,052.77
Computers-Printers, Scanners & Fax	$281,762,342.2
Computers-Projectors & Screens	$1,107,199,413.48
Music, Movies and Audio Books-Movie DVD	$165,804,705.9811
TV and Video-Car Video	$306,818,844.52
TV and Video-Home Theater System	$709,120,510.209
TV and Video-Televisions	$307,373,914.4742
TV and Video-VCD & DVD	$36,807,845.561
Total	**$8,341,224,364.8324**

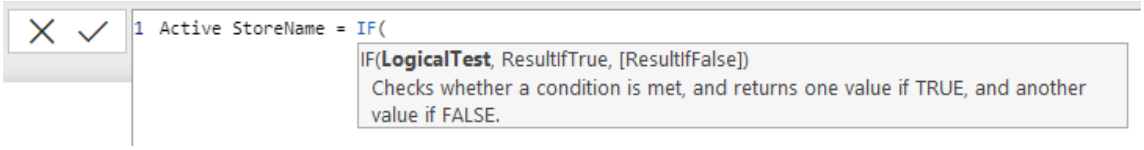
5.3 Create a Logical Column

Now we would like to create a new calculated column named *Active StoreName* in the *Stores* table. For this column, our DAX formula is going to check each stores status. If a stores status is On, our formula will return the stores name. If it's Off, it will have the name, "Inactive".

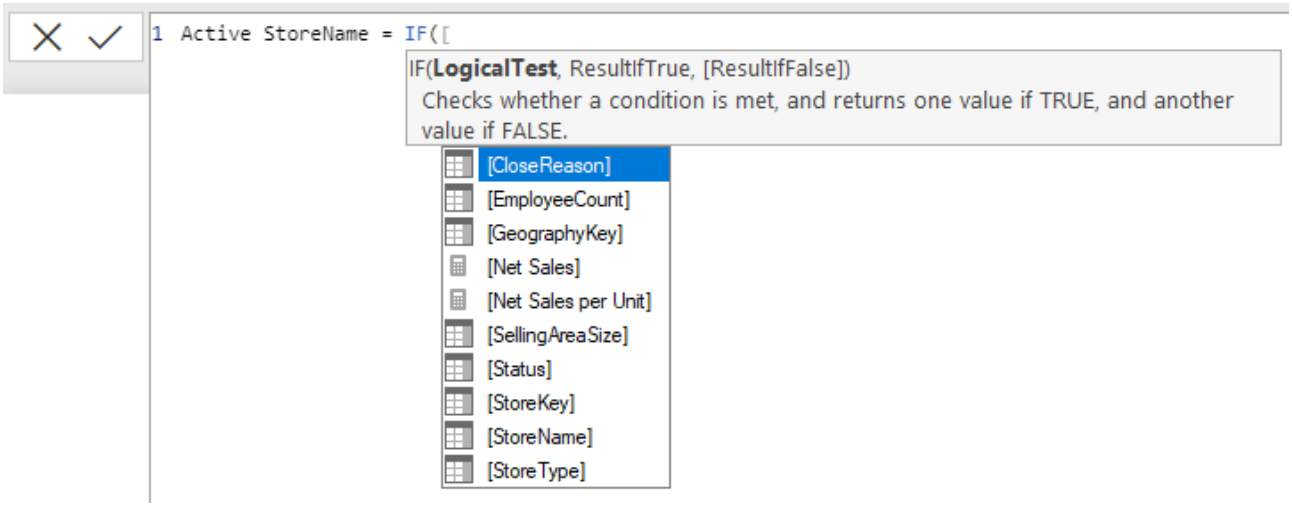
1. Click the down arrow on the *Stores* table in the Data list, and then click **New Column**.



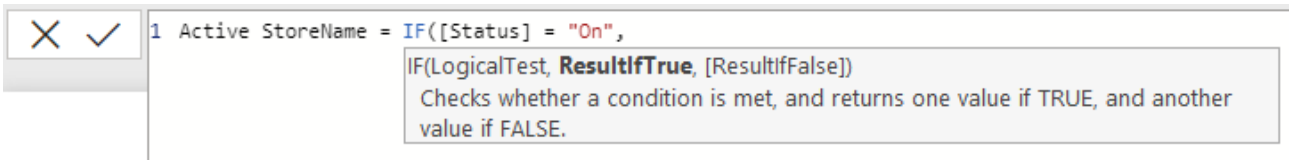
2. Select **IF** after typing the column name *Active StoreName*.



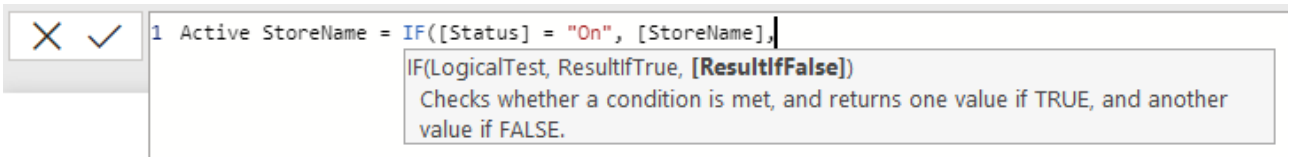
3. Type an opening bracket **[** , and select *[Status]*.



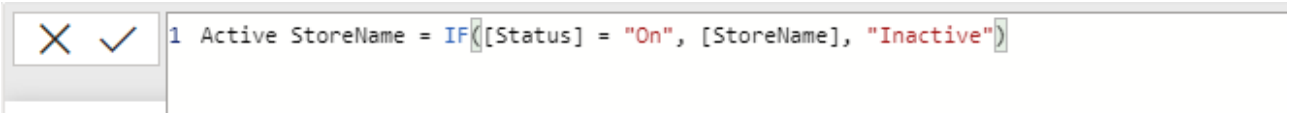
4. Right after *[Status]*, type = **"On"**, then enter a comma (**,**) to enter the second argument. The tooltip suggests we need to add the value for when the result is true.



5. If the store is On, we want to show the store's name. Type an opening bracket **[** and select the *[StoreName]* column, and then type another comma so we can enter our third argument.

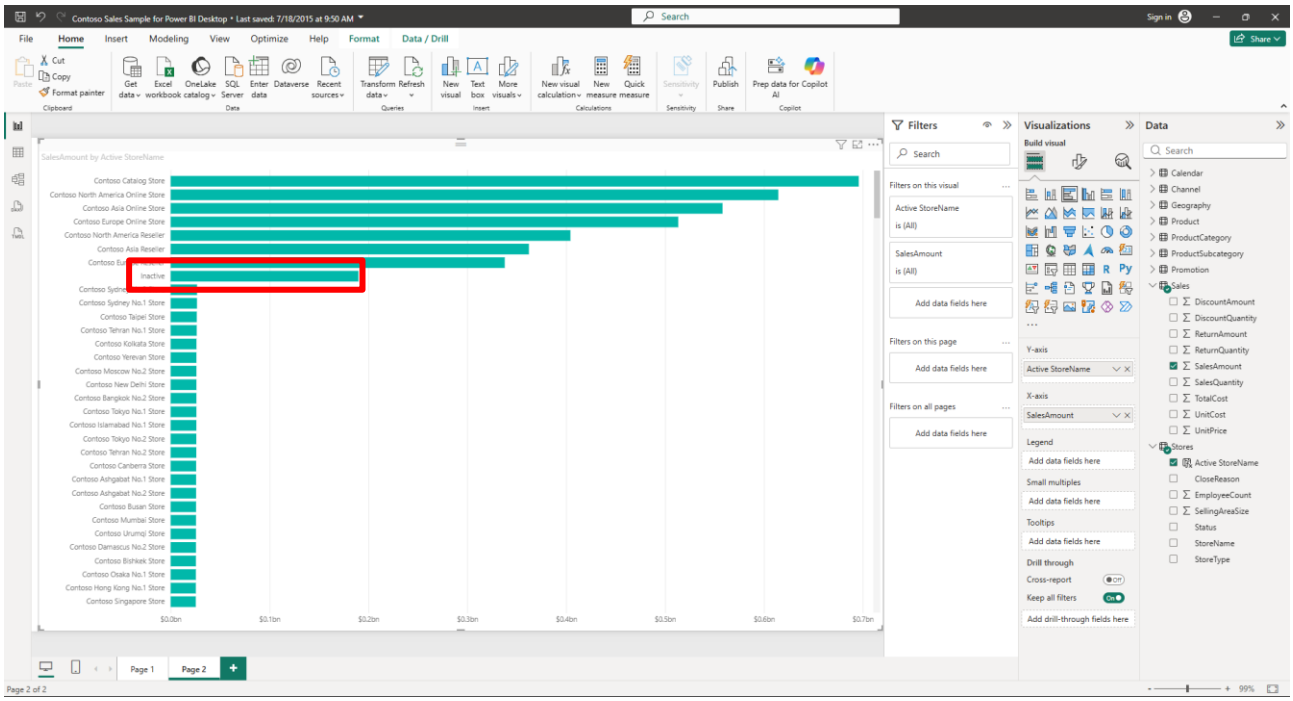


6. We need to add a value for when the result is false, in this case we want the value to be "**Inactive**". Complete the formula by pressing **[Enter]** in the formula bar. The formula is validated and added to the Data list in the *Stores* table.



7. Create a new page and add a **Clustered Bar Chart** visual, and add the following fields to the visual wells:
   - Assign *Store* ➔ *Active StoreName* to **Y-axis**
   - Assign *Sales* ➔ *Sales Amount* to **X-axis**

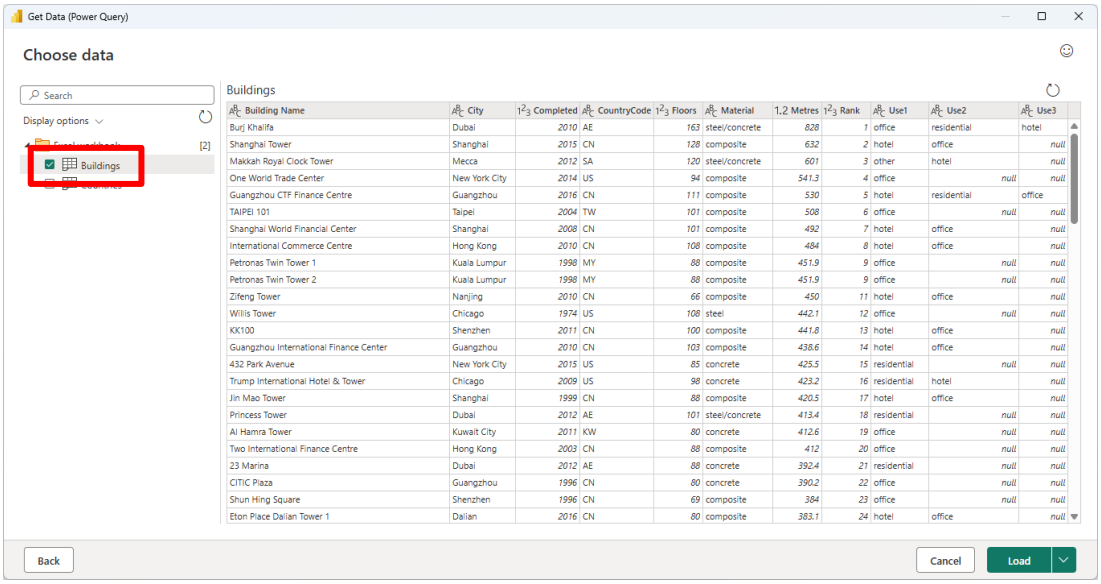
   In this chart, stores with a status of On are shown individually by name, but stores with a status of Off are grouped together and shown as Inactive.



6. Create Logical Column

6.1 Connect to Excel Workbook

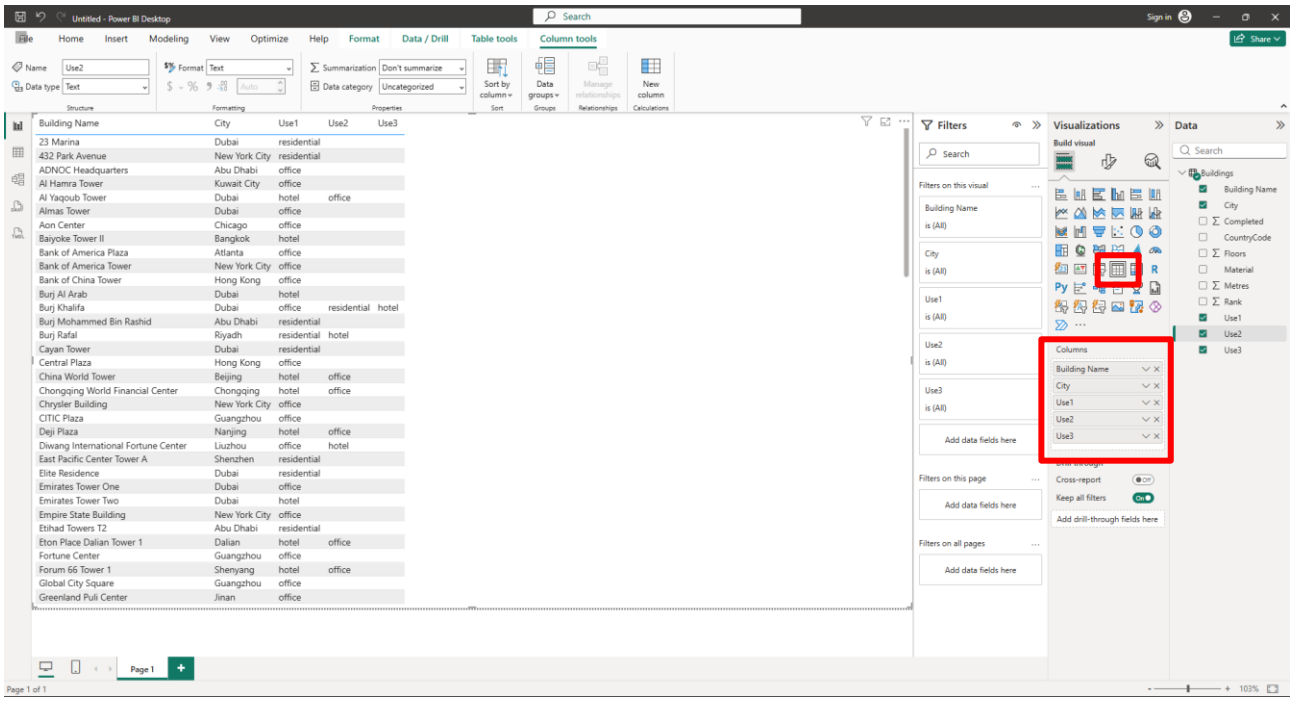
1. Select **Get Data ➔ Excel Workbook** in the **Home** ribbon tab,
2. In the **Get Data (Power Query)** dialog box, select the **Skyscrapers.xlsx** file. Then in the Choose data pane, select the **Buildings** table and then select **[Load]**.



Building Name	City	Completed	CountryCode	Floors	Material	Metres	Rank	Use1	Use2	Use3
Burj Khalifa	Dubai	2010	AE	163	steel/concrete	828	1	office	residential	hotel
Shanghai Tower	Shanghai	2015	CN	128	composite	632	2	hotel	office	null
Makkah Royal Clock Tower	Mecca	2012	SA	120	steel/concrete	601	3	other	hotel	null
One World Trade Center	New York City	2014	US	94	composite	541.3	4	office	null	null
Guangzhou CTF Finance Centre	Guangzhou	2016	CN	111	composite	530	5	hotel	residential	office
TAIPEI 101	Taipei	2004	TW	101	composite	508	6	office	null	null
Shanghai World Financial Center	Shanghai	2008	CN	101	composite	492	7	hotel	office	null
International Commerce Centre	Hong Kong	2010	CN	108	composite	484	8	hotel	office	null
Petronas Twin Tower 1	Kuala Lumpur	1998	MY	88	composite	451.9	9	office	null	null
Petronas Twin Tower 2	Kuala Lumpur	1998	MY	88	composite	451.9	9	office	null	null
Zifeng Tower	Nanjing	2010	CN	66	composite	450	11	hotel	office	null
Willis Tower	Chicago	1974	US	108	steel	442.1	12	office	null	null
KK100	Shenzhen	2011	CN	100	composite	441.8	13	hotel	office	null
Guangzhou International Finance Center	Guangzhou	2010	CN	103	composite	438.6	14	hotel	office	null
432 Park Avenue	New York City	2015	US	85	concrete	425.5	15	residential	null	null
Trump International Hotel & Tower	Chicago	2009	US	98	concrete	423.2	16	residential	hotel	null
Jin Mao Tower	Shanghai	1999	CN	88	composite	420.5	17	hotel	office	null
Princess Tower	Dubai	2012	AE	101	steel/concrete	413.4	18	residential	null	null
Al Hamra Tower	Kuwait City	2011	KW	80	concrete	412.6	19	office	null	null
Two International Finance Centre	Hong Kong	2003	CN	88	composite	412	20	office	null	null
23 Marina	Dubai	2012	AE	88	concrete	392.4	21	residential	null	null
CITIC Plaza	Guangzhou	1996	CN	80	concrete	390.2	22	office	null	null
Shun Hing Square	Shenzhen	1996	CN	69	composite	384	23	office	null	null
Eton Place Dalian Tower 1	Dalian	2016	CN	80	composite	383.1	24	hotel	office	null

6.2 Create Table Visual

1. Create a table for building name, city and use. There are 3 columns giving the uses to which each building is put. Add a **Table** visual, then add the following fields: *Building Name*, *City*, *Use1*, *Use2* and *Use3* to **Columns**.



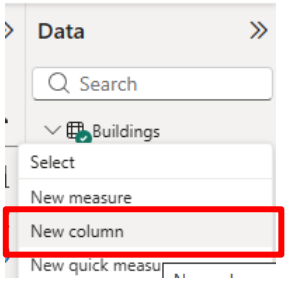
Building Name	City	Use1	Use2	Use3
23 Marina	Dubai	residential		
432 Park Avenue	New York City	residential		
ADNOC Headquarters	Abu Dhabi	office		
Al Hamra Tower	Kuwait City	office		
Al Yaqoub Tower	Dubai	hotel	office	
Almas Tower	Dubai	office		
Aon Center	Chicago	office		
Baiyoke Tower II	Bangkok	hotel		
Bank of America Plaza	Atlanta	office		
Bank of America Tower	New York City	office		
Bank of China Tower	Hong Kong	office		
Burj Al Arab	Dubai	hotel		
Burj Khalifa	Dubai	office	residential	hotel
Burj Mohammed Bin Rashid	Abu Dhabi	residential		
Burj Rafal	Riyadh	residential	hotel	
Cayan Tower	Dubai	residential		
Central Plaza	Hong Kong	office		
China World Tower	Beijing	hotel	office	
Chongqing World Financial Center	Chongqing	hotel	office	
Chrysler Building	New York City	office		
CITIC Plaza	Guangzhou	office		
Deji Plaza	Nanjing	hotel	office	
Diwang International Fortune Center	Liuzhou	office	hotel	
East Pacific Center Tower A	Shenzhen	residential		
Elite Residence	Dubai	residential		
Emirates Tower One	Dubai	office		
Emirates Tower Two	Dubai	hotel		
Empire State Building	New York City	office		
Etihad Towers T2	Abu Dhabi	residential		
Eton Place Dalian Tower 1	Dalian	hotel	office	
Fortune Center	Guangzhou	office		
Forum 66 Tower 1	Shenyang	hotel	office	
Global City Square	Guangzhou	office		
Greenland Puli Center	Jinan	office		

6.3 Using IF in Column

We would like to use the IF function to create a new column to show for each office its use type:

Rule	Use type
If any of the 3 **Use** columns is **office**	**Office**
Otherwise	**Other**

1. Click the down arrow on the *Buildings* table in the Data list, and then click **New column**.

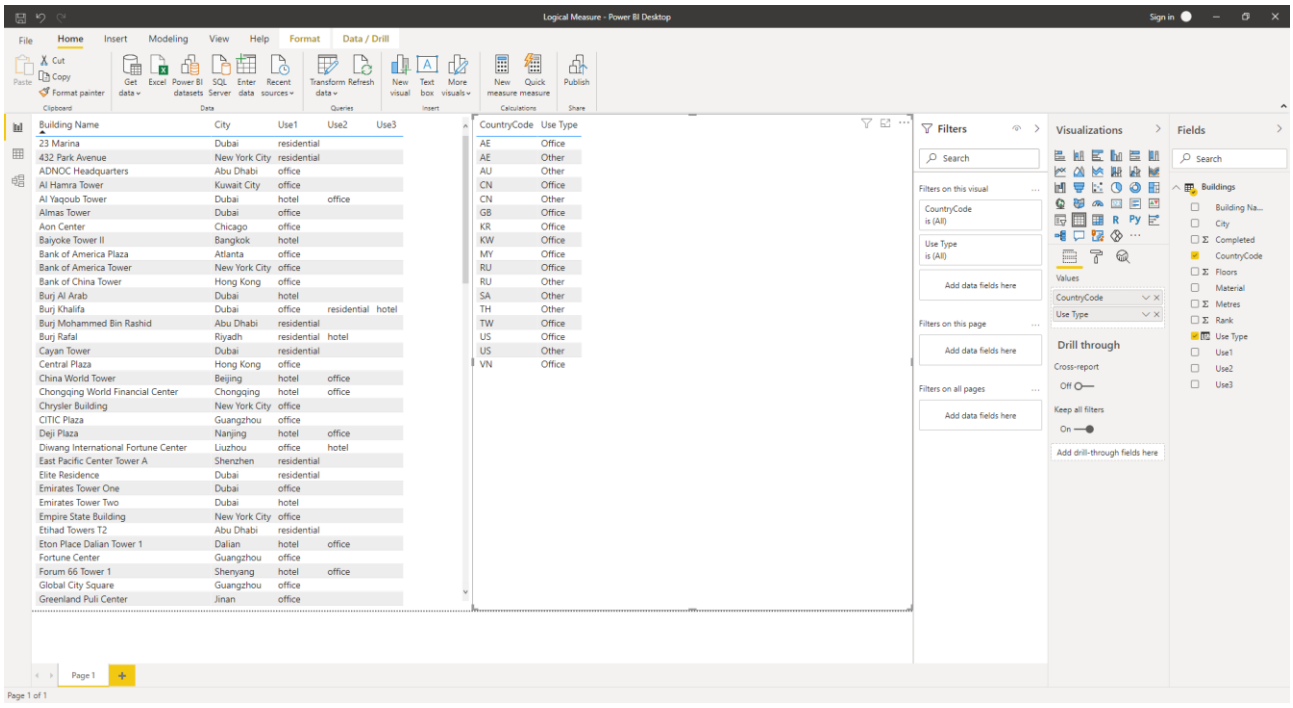


2. Name this new column as Use Type. In order to perform this checking, IF and OR function is used in the function.

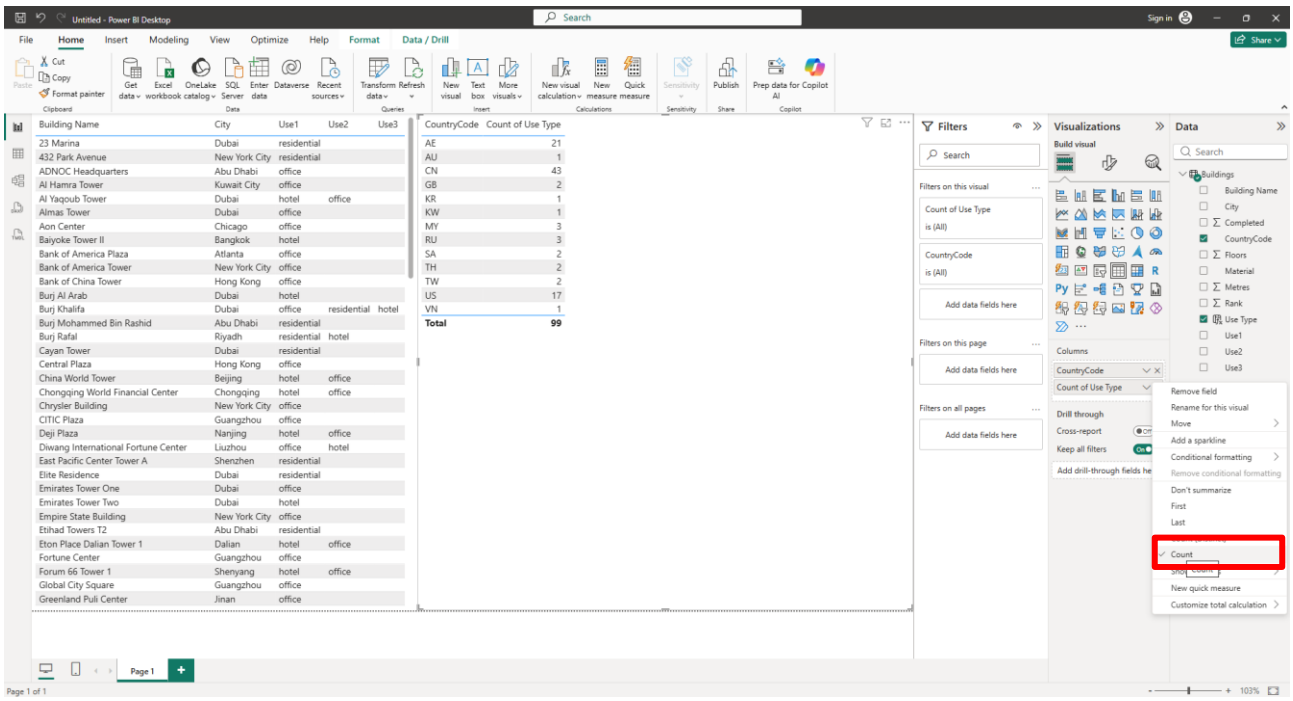
   Use Type = IF(or (or (Buildings[Use1] = "Office", Buildings[Use2] = "Office"), Buildings[Use3] = "Office"), "Office", "Other")

```
1  Use Type = IF( or (
2                  or (Buildings[Use1] = "Office", Buildings[Use2] = "Office"),
3                  Buildings[Use3] = "Office"),
4              "Office", "Other")
```

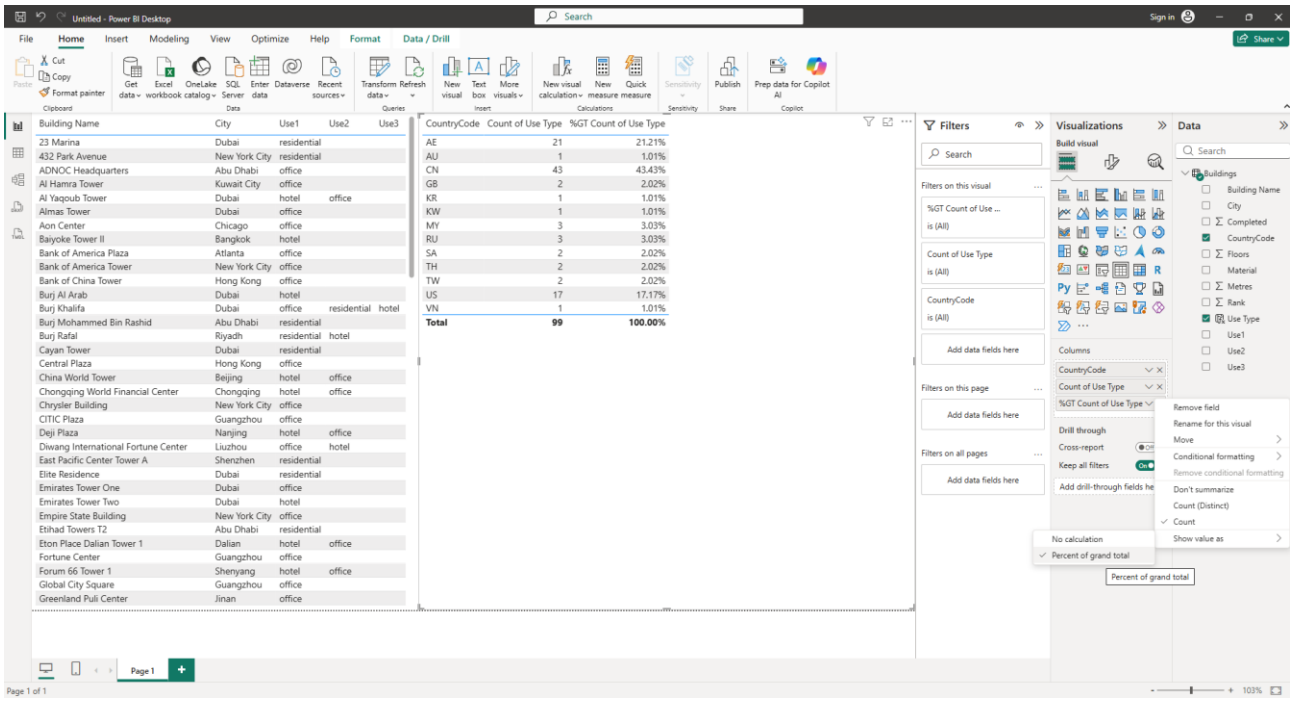
3. Create a table by using *CountryCode* and *Use Type*.



4. Change the calculate method for *Use Type* to **Count**



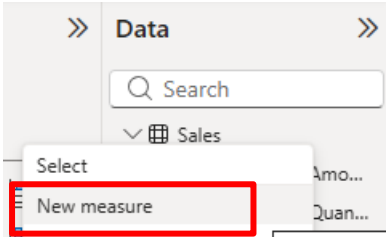
5. Drag the *Use Type* to values again, then change the calculate method for *Use Type* to **Count** and **show value as** to **Percentage of Grant Total**.



7. Create Custom Measures

7.1 Creating Custom Measure – Net Sales

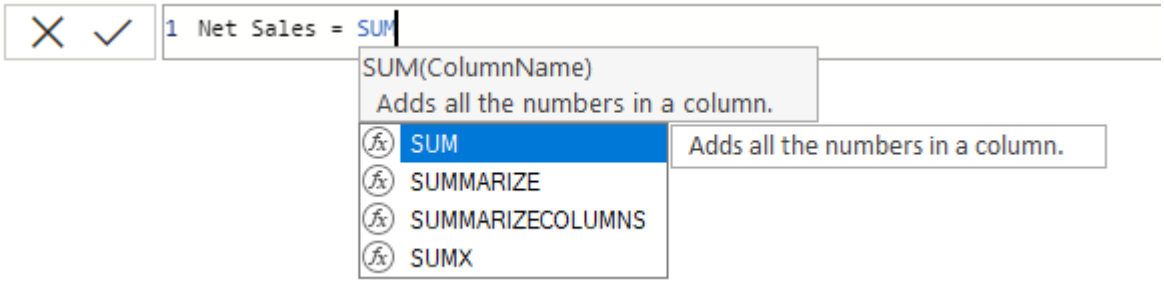
1. Download the Contoso Sales for Power BI sample designer file.
2. Select **File ➔ Open Report**, and open *Contoso Sales Sample for Power BI Desktop.pbix*
3. Click the down arrow on the *Sales* table in the Data list, and then click **New Measure**. This will make sure our new measure is saved in the *Sales* table, where it will be easier to find.



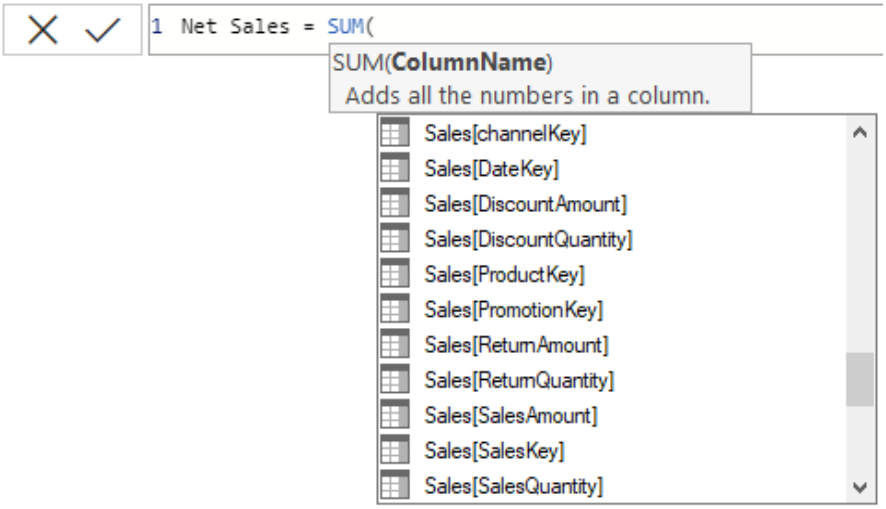
4. The formula bar appears along the top of the **Report** Canvas. This is where we can rename our measure and enter a DAX formula. Highlight *Measure* in the formula bar, and then type ***Net Sales***.



5. Select **SUM** by scrolling down, and then press **[Enter]**. An opening parenthesis appears along with another suggestion list of all of the available columns we can pass to the SUM function.



6. Select *Sales[SalesAmount]*, and then type a closing parenthesis



7. After the closing parenthesis for our first expression, type a space, and then a minus operator (-),

followed by another space. Then enter another SUM function with the *Sales[DiscountAmount]* column as its argument.

```
1 Net Sales = SUM(Sales[SalesAmount]) - SUM(Sales[DiscountAmount])
```

8. Click the down chevron on the right side of the formula bar.

```
1 Net Sales = SUM(Sales[SalesAmount]) - SUM(Sales[DiscountAmount])
```

9. We can enter new parts to our formula on a new line by pressing **[Alt]** + **[Enter]**. We can also move things over by using **[Tab]**.

```
1 Net Sales = SUM(Sales[SalesAmount])
2             - SUM(Sales[DiscountAmount])
```

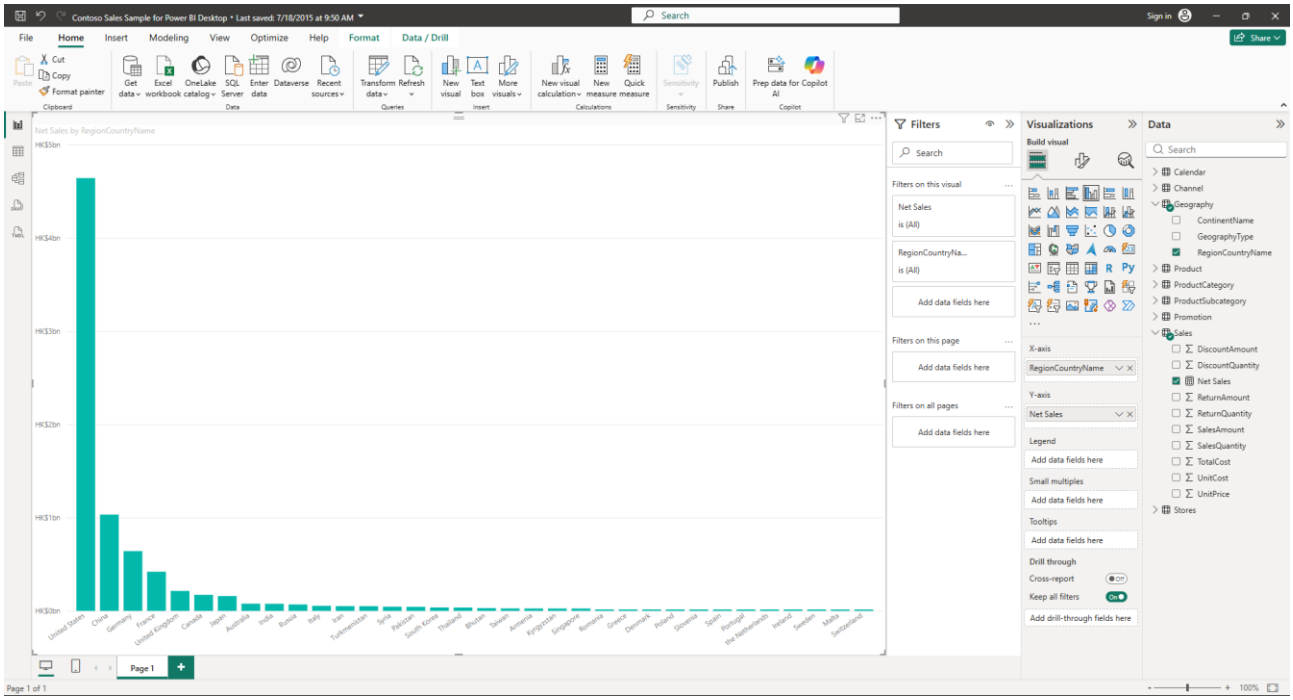
10. Add another minus operator followed by another SUM function and the *Sales[ReturnAmount]* column as its argument.

```
1 Net Sales = SUM(Sales[SalesAmount])
2             - SUM(Sales[DiscountAmount])
3             - SUM(Sales[ReturnAmount])
```

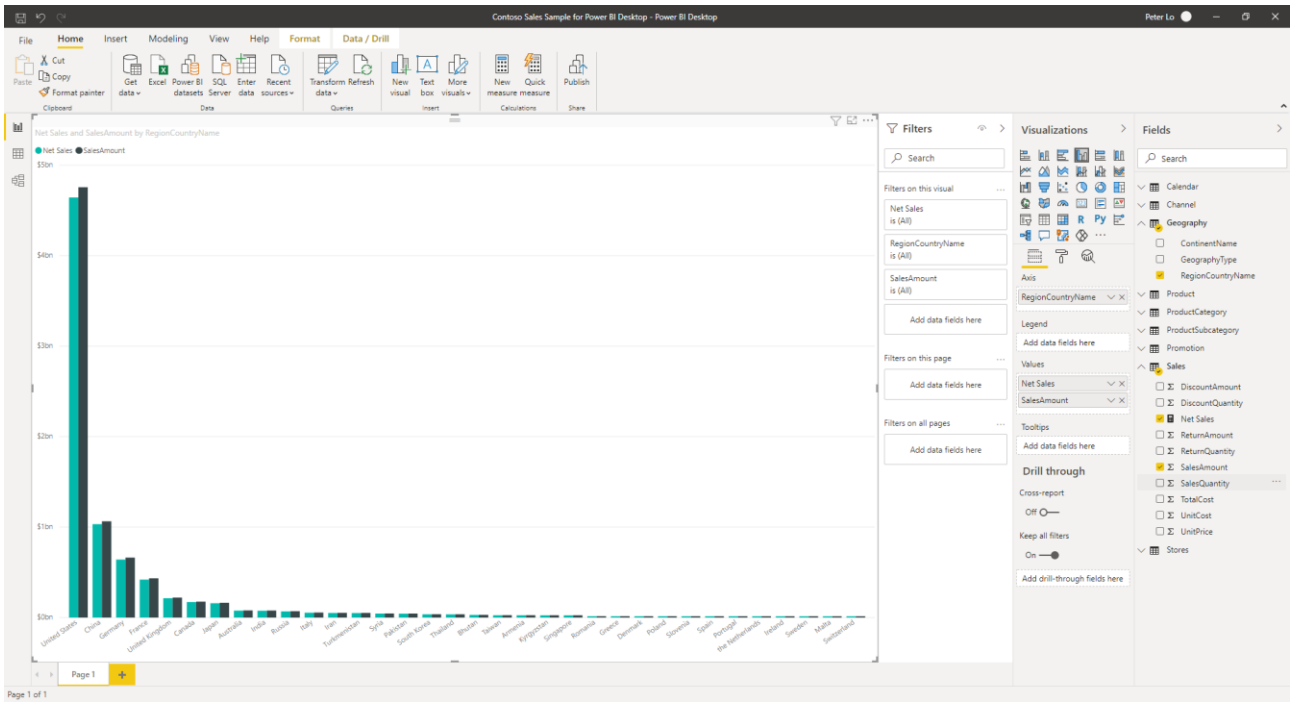
11. Press **[Enter]** in the formula bar to complete, the formula is validated and added to the Data list in the *Sales* table.

7.2 Add New Measure to Report

1. Add a **Clustered Bar Chart** visual and add the following fields to the visual wells:
   - Assign *Geography* ➔ *RegionCountryName* to **X-axis**
   - Assign *Sales* ➔ *Net Sales* to **Y-axis**

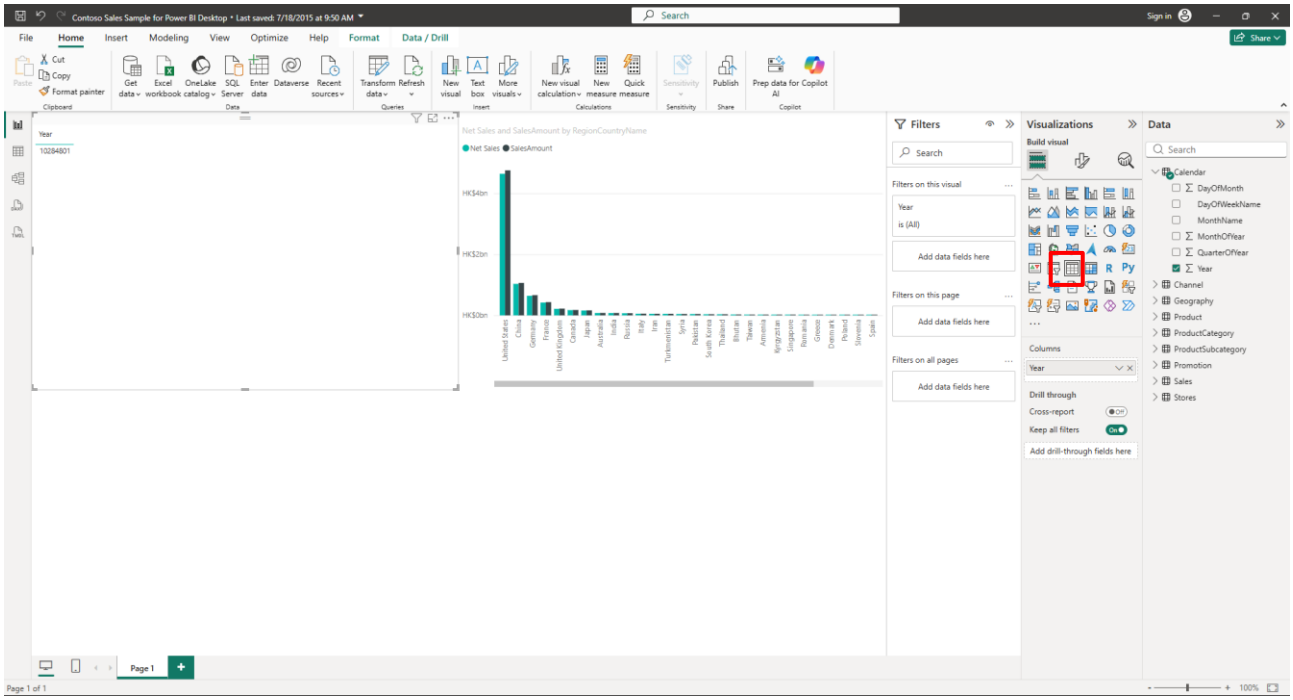


2. Drag the *Sales* ➔ *SalesAmount* field into the chart, to see the difference between net sales and sales amount. We now really have two measures in our chart. *SalesAmount*, which was summed up automatically, and the *Net Sales* measure we created. In each case, the results were calculated in context of another field we have in the chart, the countries in *RegionCountryName*.

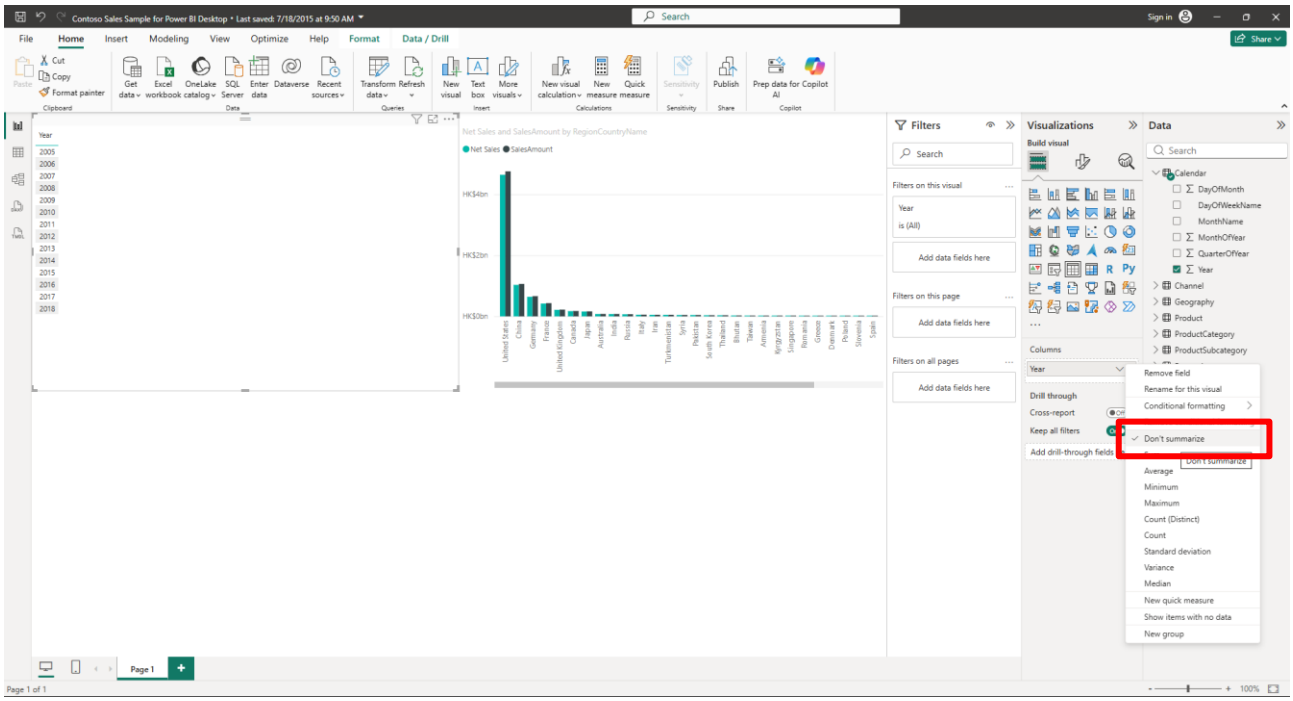


3. Add a **Table** visual and add the following fields to the visual wells.
   - Assign *Calendar* ➔ *Year* to **Columns**

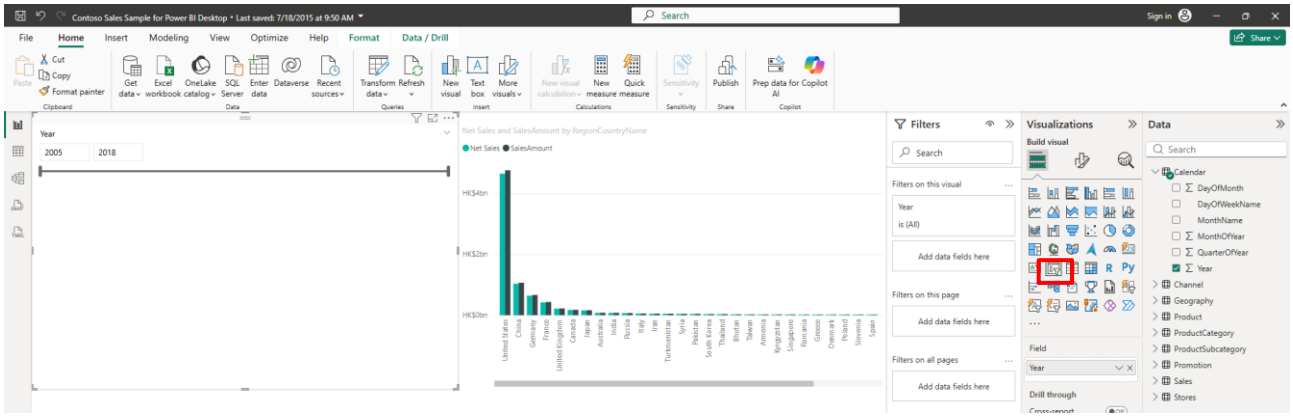
Because *Year* is a numeric field, Power BI Desktop summed up its values.



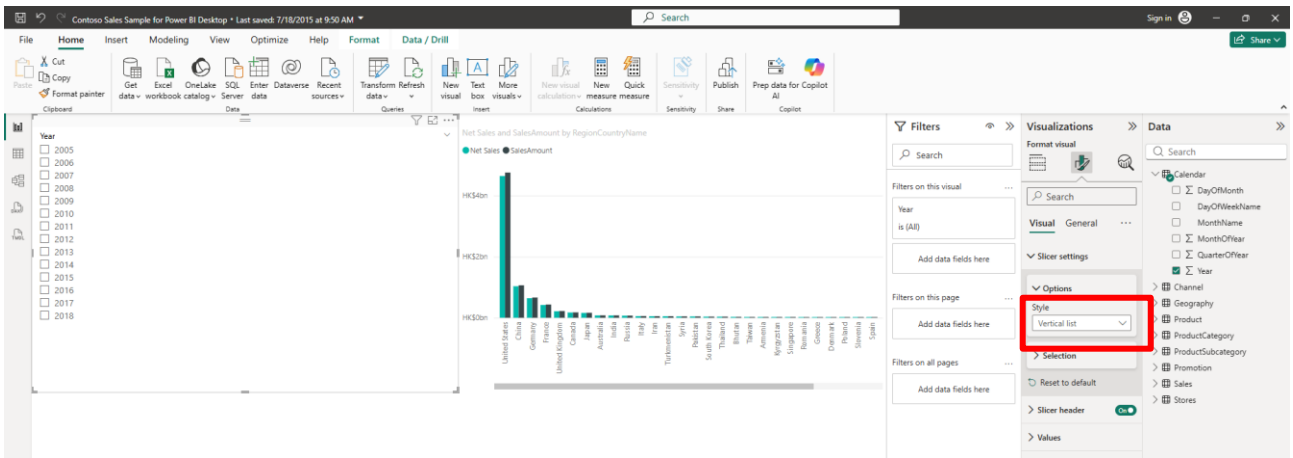
4. But that doesn't do us much good as a Slicer. Click the down arrow next to *Year* in **Values**, and then click **Don't Summarize**.



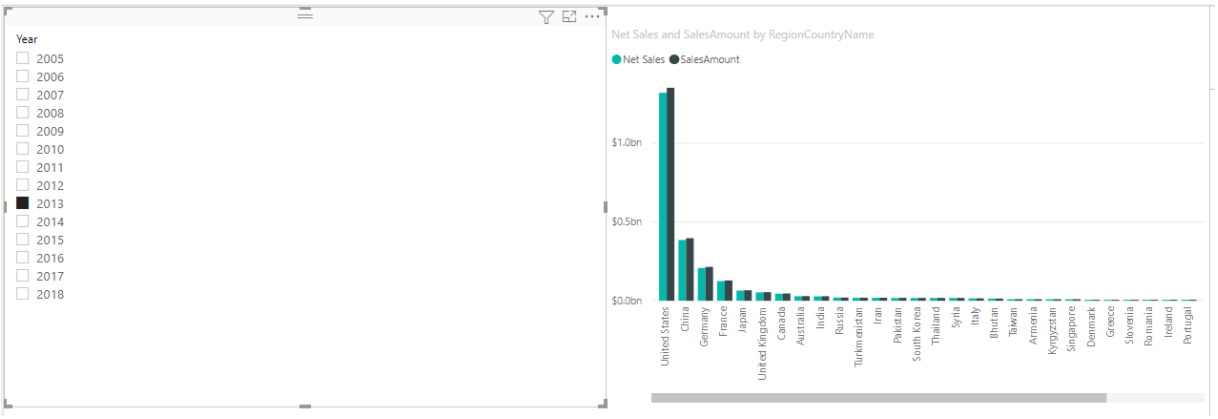
5. Now we can change the *Year* field in the table visualization into a **Slicer**. In **Visualizations**, click the **Slicer** visualization.



6. Change the *Year* Slicer to a **Vertical List**. We can select any individual or group of years and our report's visualizations will all be sliced accordingly.

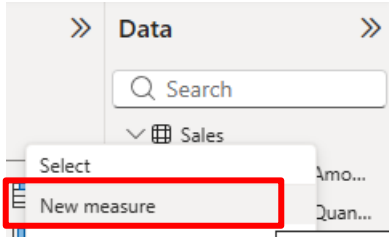


7. Go ahead and click on *2013*. You'll see the chart change. Our *Net Sales* and *SalesAmount* measures are re-calculated, showing new results just for *2013*.

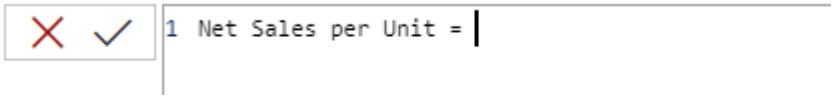


7.3 Create Another Measure – Net Sales per Unit

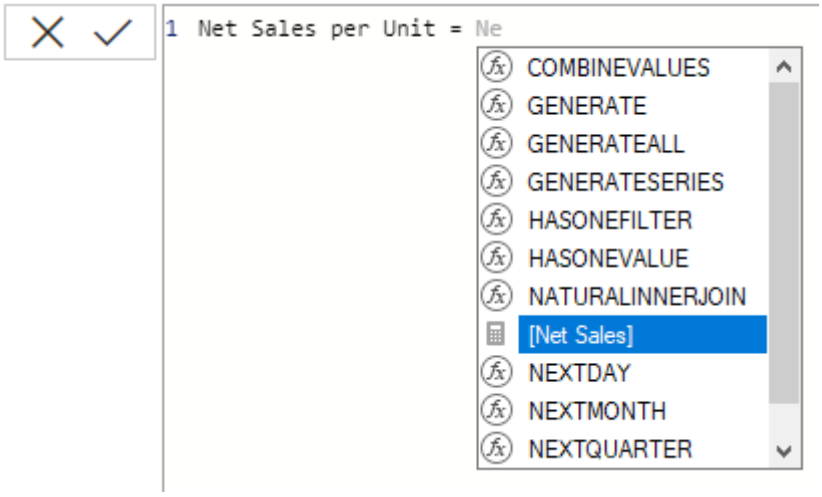
1. In order to create another new measure, click the down arrow on the *Sales* table in the Data list, and then click **New Measure**.



2. Highlight Measure in the formula bar, and then rename as *Net Sales per Unit*.



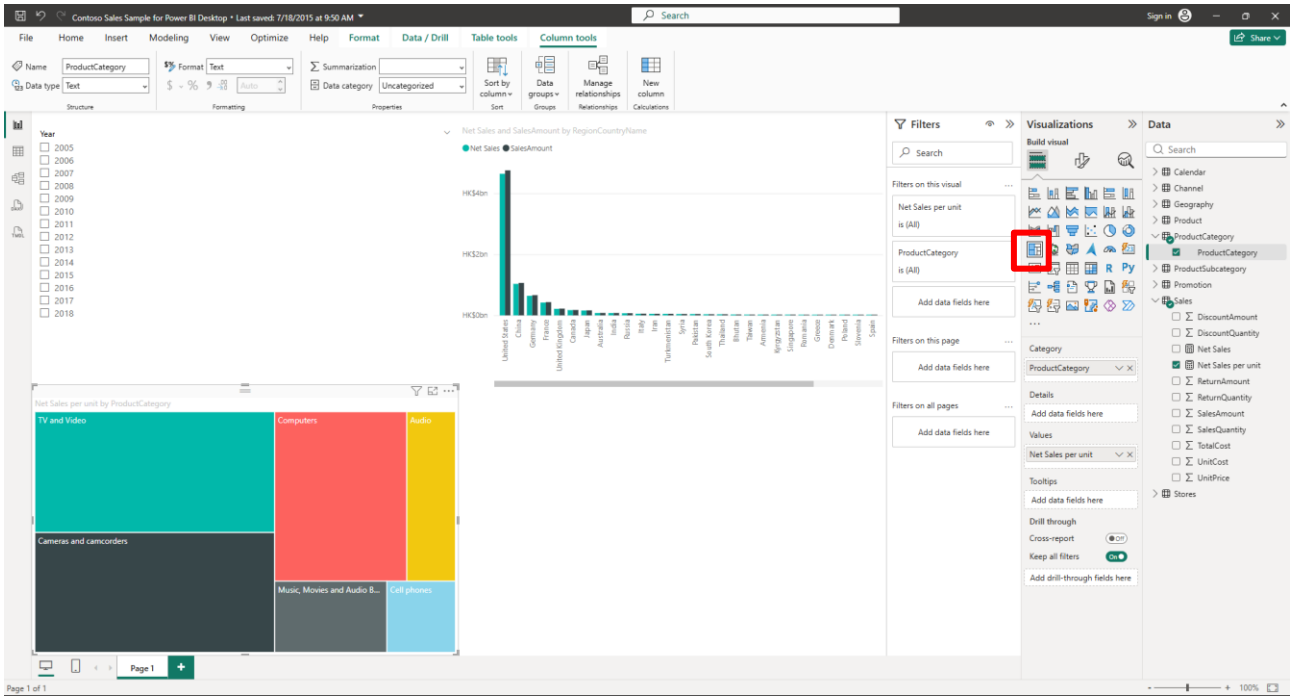
3. Begin typing *Net Sales*. The suggestion list will show what we can add. Select *[Net Sales]*.



4. Right after *[Net Sales]*, enter a space, then a divide operator (/), then enter a SUM function, then type Quantity. The suggestion list shows all the columns with Quantity in the name. Select *Sales[SalesQuantity*]. The formula should now look like this:

```
1 Net Sales per Unit = [Net Sales] / Sum(Sales[SalesQuantity])
```

5. Add a **Treemap** visual and add the following fields to the visual wells:
   - Assign *Sales* ➔ *Net Sales per Unit* to **Values**
   - Assign *ProductCategory* ➔ *ProductCategory* to **Category**



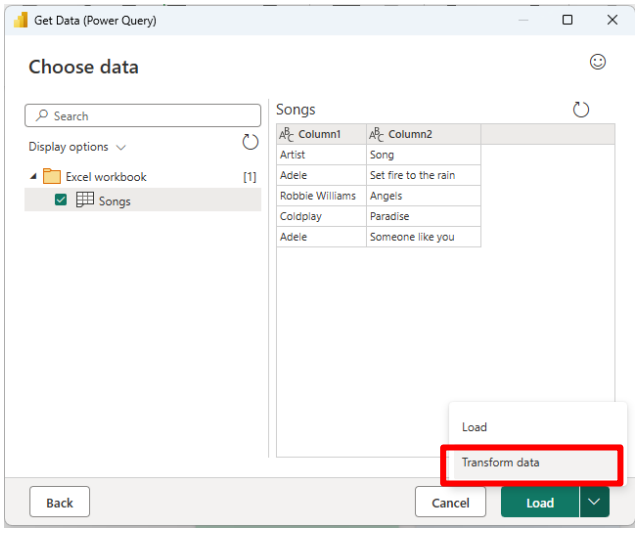
8. Parameter

A parameter serves to easily store and manage a value that can be reused. Parameters give you the flexibility to dynamically change the output of your queries depending on their value, and can be used for:

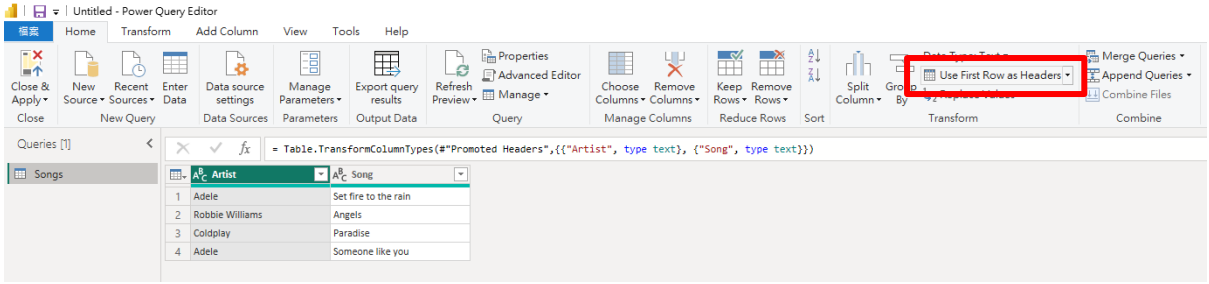
- Changing the argument values for transforms and data source functions
- Inputs in custom functions

You can easily manage your parameters inside the Manage Parameters window. You can get to the Manage Parameters window by selecting the Manage Parameters option inside Manage Parameters in the Home tab.

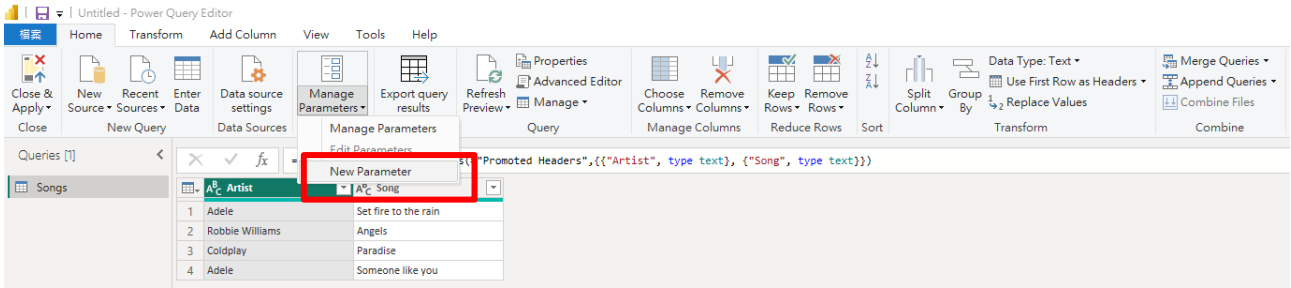
1. Select **Get Data ➔ Excel Workbook** in the **Home** ribbon tab. In the **Open File** dialog box, select the *Anyone.xlsx* file. Then in the **Choose Data** pane, select the **Songs** worksheet and click **[Load] ➔ [Transform Data]** button.



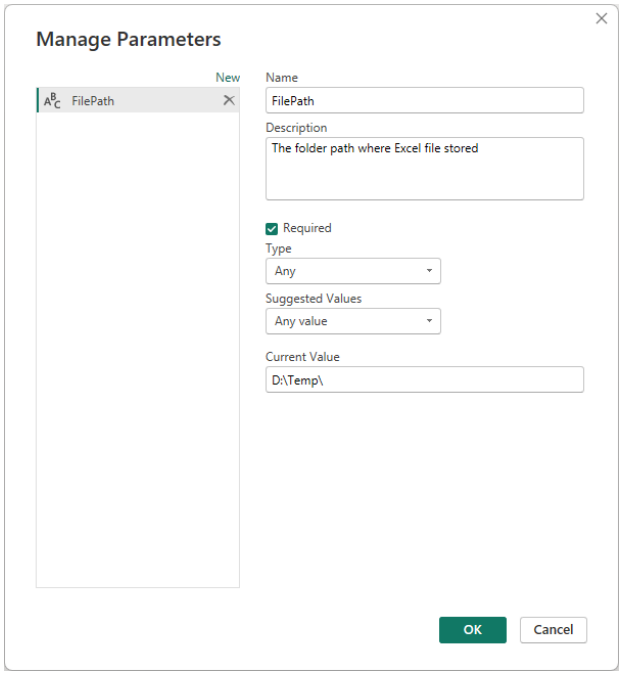
2. In Power Query Editor, select **Use First Row as Headers** in **Home** tab.



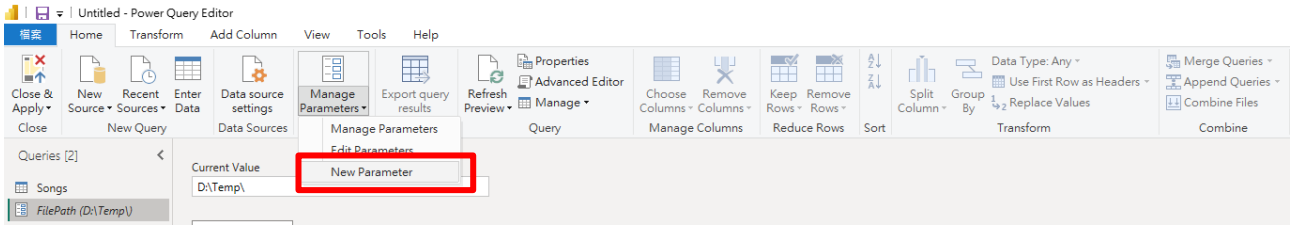
3. Select **Manage Parameter ➔ New Parameters** in **Home** tab.



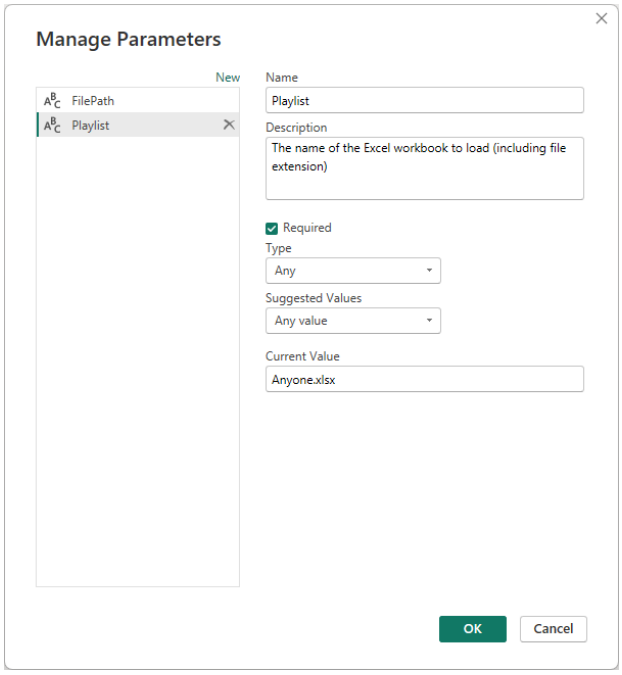
4. Create the first parameter "**FilePath**" to store the folder path where your Excel files are stored, then press **[OK]** to continue.



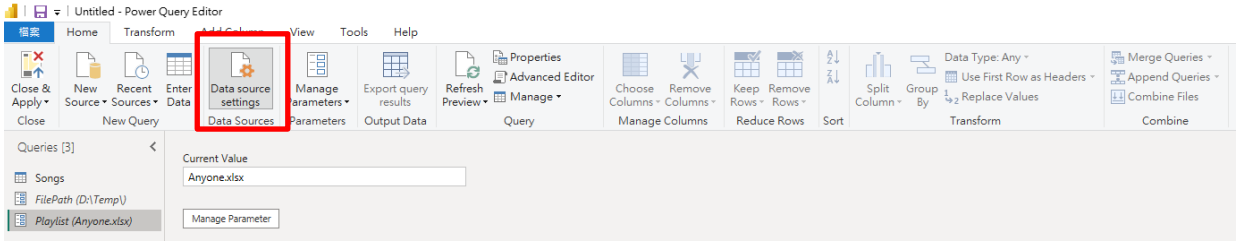
5. The parameter is created successfully, click **Manage Parameter ➔ New Parameters** in **Home** tab again to create another parameter.



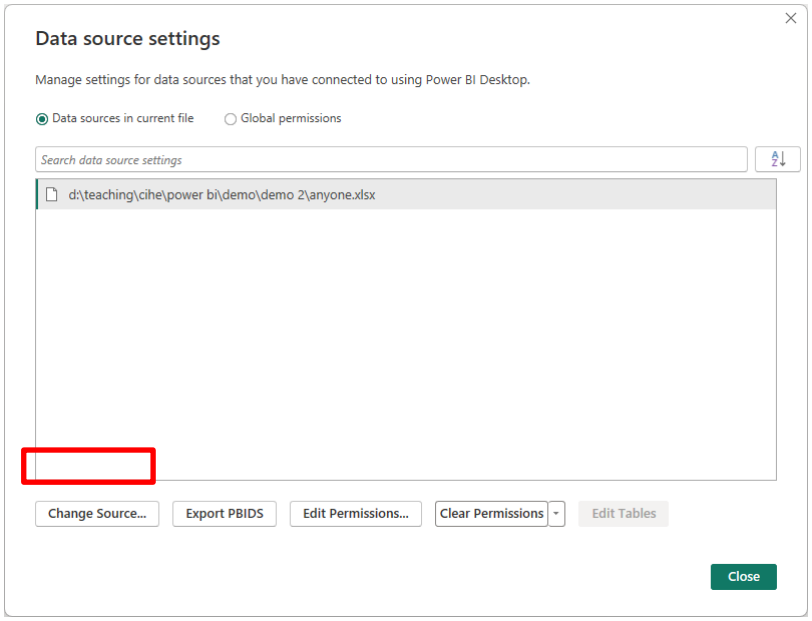
6. Create the second parameter "**Playlist**" to the name of the Excel workbook to load (including file extension)



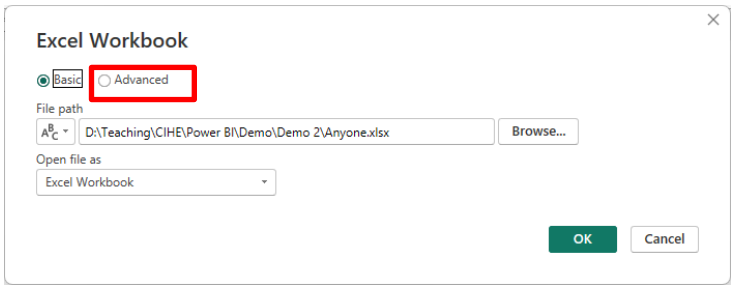
7. Select **Data Source Setting** in Home tab.



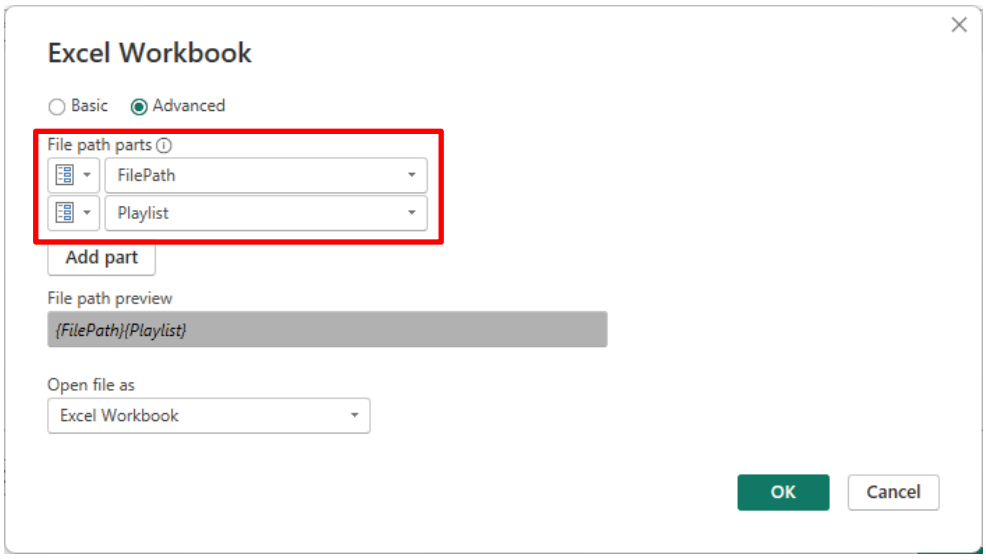
8. Select **[Change Source]**



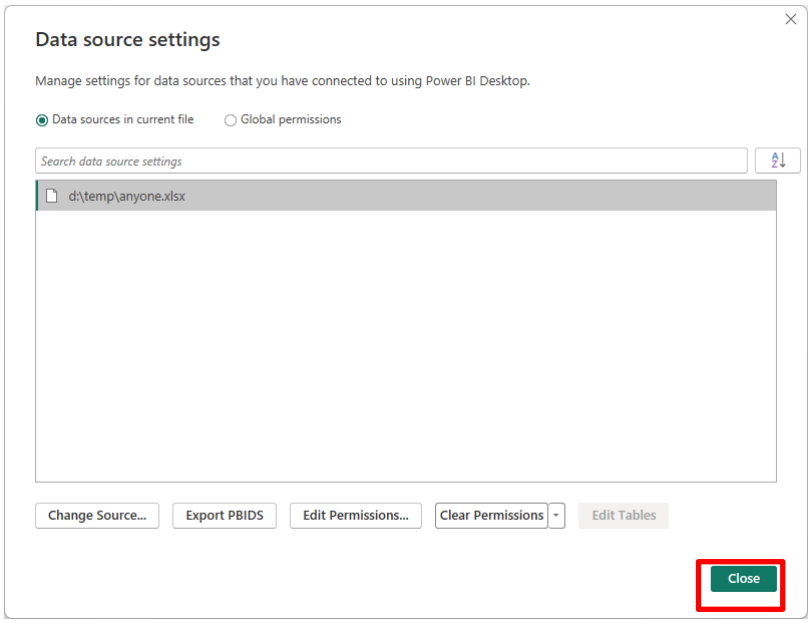
9. Select **[Advanced]**.



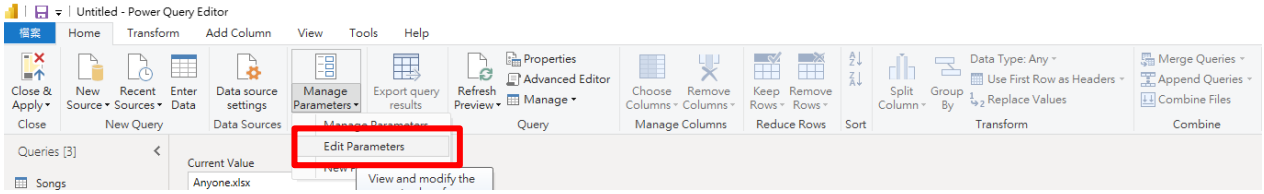
10. Change the **file path part** to parameter (*FilePath* and *Playlist*) and press **[OK]** to confirm



11. Press **[Close]** button to leave the Data source setting dialog.



12. Select **Edit Queries ➔ Edit Parameters**.



13. Input the FilePath and Playlist and press **[OK]**.



14. If you browse the Queries Songs, you should be able to see the content from your new file. Press **[Refresh Preview]** to reload when necessary.

