

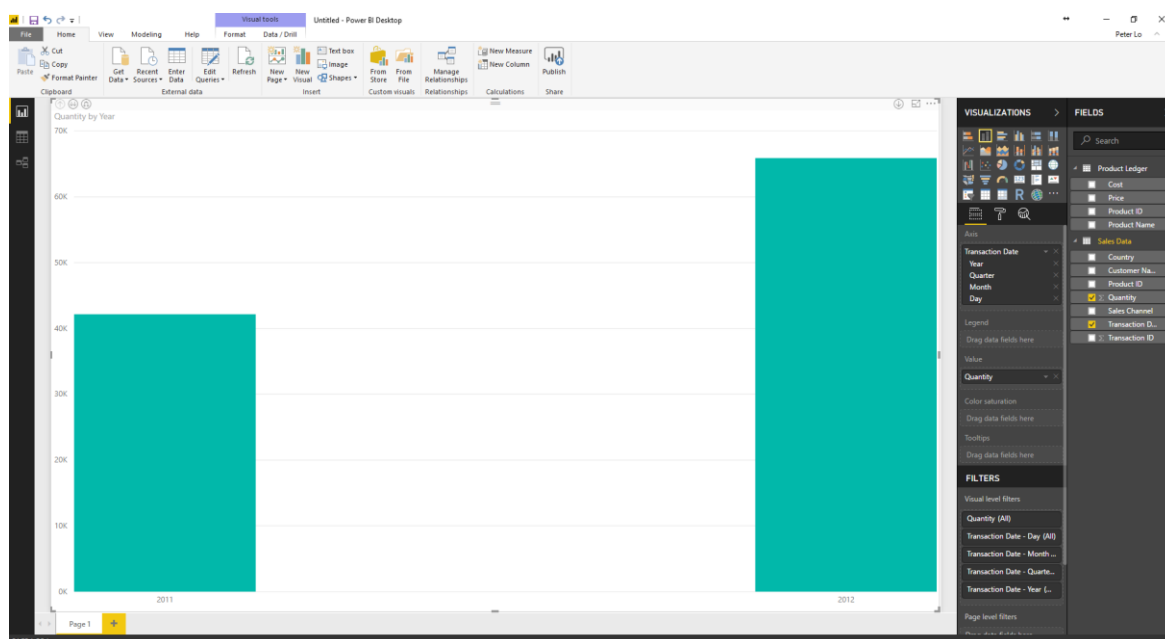
1. Drill Down

1. Import the file *sample-sales-data.xls* into Microsoft Power BI

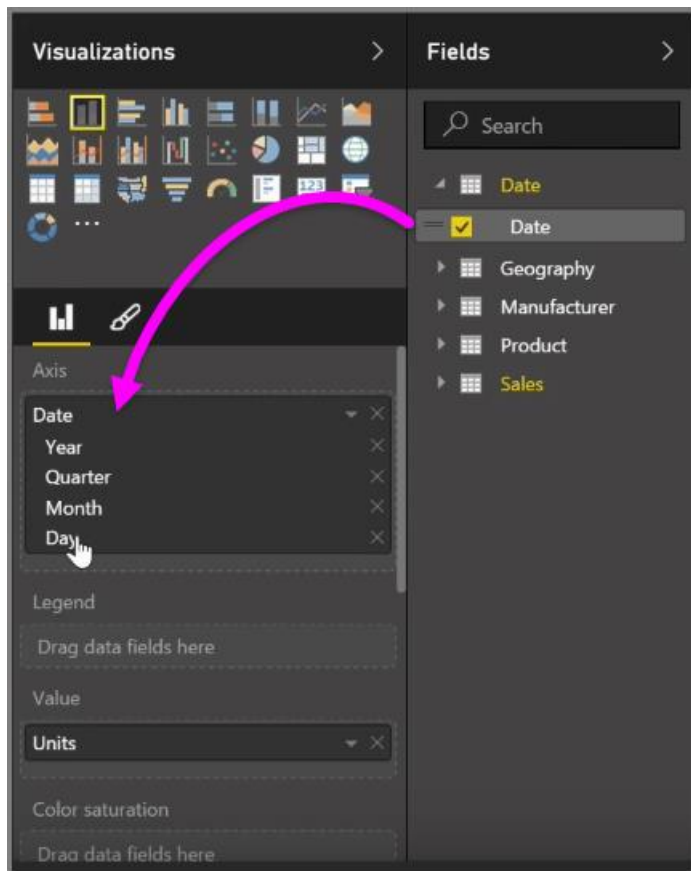
The screenshot shows the Microsoft Power BI Navigator window. On the left, under 'Display Options', the file 'sample-sales-data.xlsx [2]' is listed. Below it, 'Product Ledger' is unchecked, and 'Sales Data' is checked. On the right, the 'Sales Data' table is displayed with the following columns: Transaction ID, Customer Name, Country, and Product ID. The table contains 20 rows of data. At the bottom right, there are 'Load', 'Edit', and 'Cancel' buttons.

Transaction ID	Customer Name	Country	Product ID
23262	Candice Levy	Congo	SUPA:
23263	Xerxes Smith	Panama	DETA:
23264	Levi Douglas	Tanzania, United Republic of	DETA:
23265	Uriel Benton	South Africa	SUPA:
23266	Celeste Pugh	Gabon	PURA:
23267	Vance Campos	Syrian Arab Republic	PURA:
23268	Latifah Wall	Guadeloupe	DETA:
23269	Jane Hernandez	Macedonia	PURA:
23270	Wanda Garza	Kyrgyzstan	SUPA:
23271	Athena Fitzpatrick	Reunion	SUPA:
23272	Anjolie Hicks	Turks and Caicos Islands	DETA:
23273	Isaac Cooper	Netherlands Antilles	SUPA:
23274	Asher Weber	Macedonia	PURA:
23275	Ethan Gregory	Tuvalu	DETA:
23276	Hayes Rollins	Nepal	PURA:
23277	MacKenzie Moss	Oman	SUPA:
23278	Aphrodite Brennan	Malawi	SUPA:
23279	Angela Wise	Moldova	PURA:
23280	James Spencer	Burkina Faso	SUPA:
23281	Adria Kaufman	Bouvet Island	SUPA:
23282	Amir Alexander	Liberia	DETA:
23283	Lani Sweet	Vanuatu	SUPA:
23284	Clark Weaver	Palau	PURA:

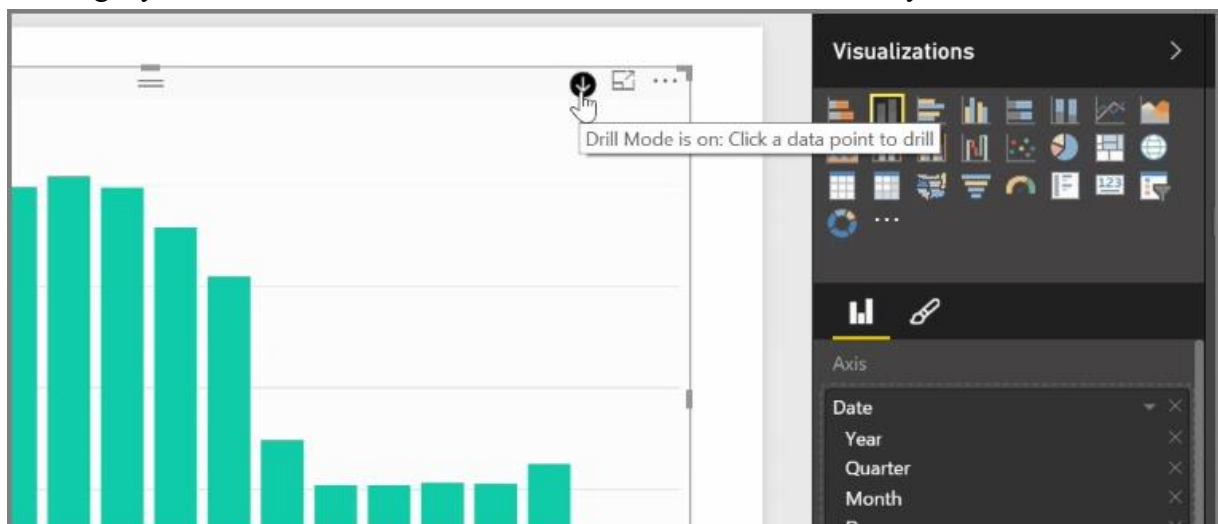
2. Create a *Quantity by Transaction Date* stacked column chart.



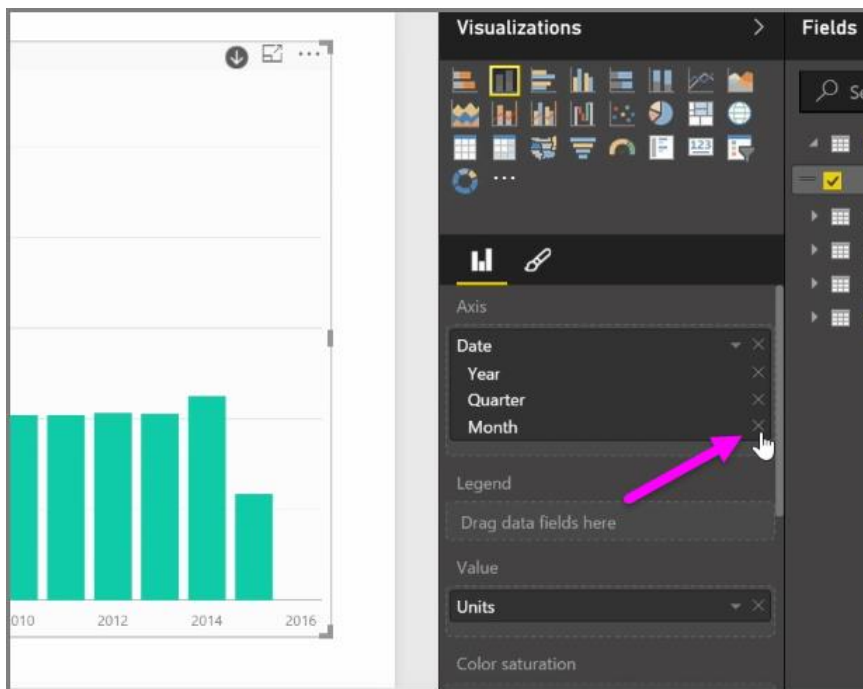
3. When you add a date field to a visual in the Axis field bucket, Power BI automatically adds a time hierarchy that includes Year, Quarter, Month and Day. By doing this, Power BI allows your visuals to have time-based interaction with those viewing your reports, by letting users drill-down through those different time levels



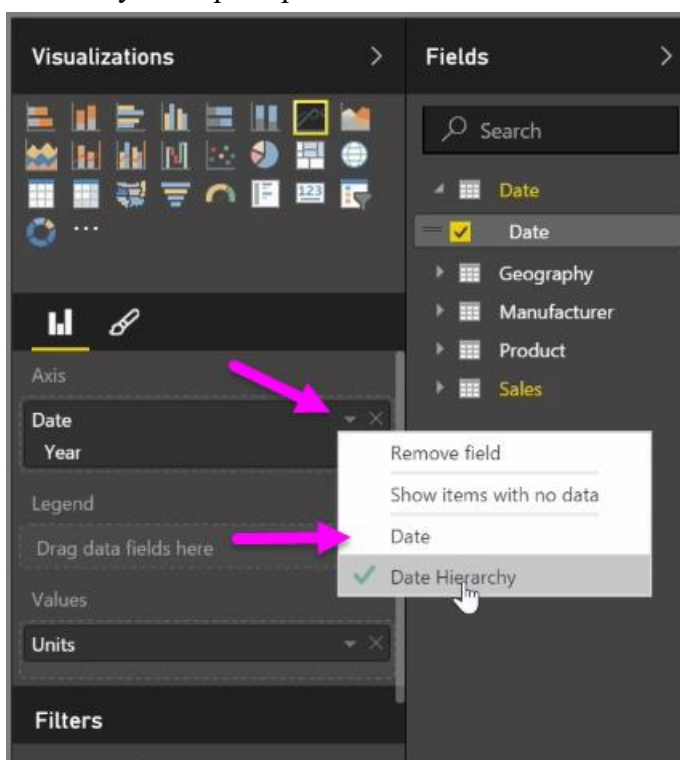
4. With a hierarchy in place, you can begin drilling down through the time hierarchy. For example, clicking a year in the chart drills down to the next level in the hierarchy.



5. In that automatically created hierarchy, you can also manage to which level your shared report allows people to drill. To do this, in the Visualizations pane, simply click the X beside the hierarchy that you want to remove. The deleted level is removed from the report, and drilling no longer displays that level.



6. If you need to get that level of the hierarchy back, just remove the date field, and then add it again from the Fields pane, and the hierarchy is once again created for you automatically. There may be times when you don't want the hierarchy to be used for a visual. You can control that by selecting the down-arrow button beside the Date field, and select Date rather than Date Hierarchy. That prompts Power BI to show the raw date values in the visual



7. You can also expand all data elements currently visible at once, rather than selecting a single quarter, or a single year. To do that, select the Drill all icon in the top left of the visual, which is a double-down arrow icon



2. DAX Function

2.1 Aggregation Functions

1. Open Power BI and get data from the “Financial Sample.xlsx”.
2. Create the following measure for SUM, AVERAGE, MAX and MIN function.

☐ ☒
SUM Demo = SUM(financials[Sales])

☐ ☒
AVERAGE DEMO = AVERAGE((financials[Sales]))

☐ ☒
MAX Demo = MAX(financials[Sales])

☐ ☒
MIN Demo = min(financials[Sales])

8. Create the following table to obtain the result

Product	Sales	SUM Demo	MAX Demo
Carretera	13,815,307.89	118,726,350.26	1,159,200.00
Montana	15,390,801.88		
Amarilla	17,747,116.06		
Velo	18,250,059.47		
VTT	20,511,921.02	AVERAGE DEMO 169,609.07	MIN Demo 1,655.08
Paseo	33,011,143.95		
Total	118,726,350.26		

2.2 Counting Functions

1. Create the following measure for COUNT and DISTINCTCOUNT function

☐ ☒
COUNT Demo = COUNT(financials[Product])

☐ ☒
DISTINCTCOUNT Demo = DISTINCTCOUNT(financials[Product])

2. Create the following table to obtain the result.

COUNT Demo	DISTINCTCOUNT Demo
700	6

2.3 Date Functions

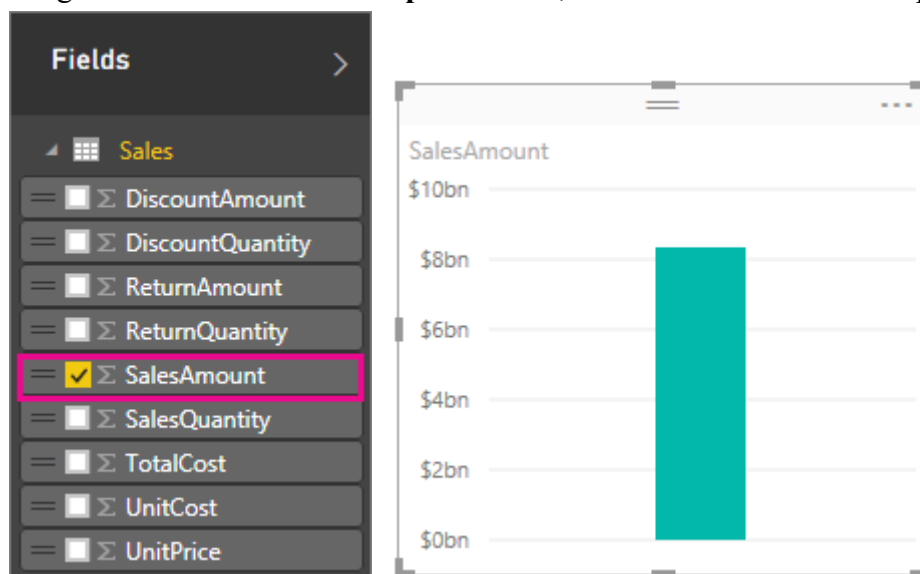
1. Find the end of month by using the EOMONTH function

☐ ☒
EOMONTH Demo = EOMONTH(TODAY(),0)

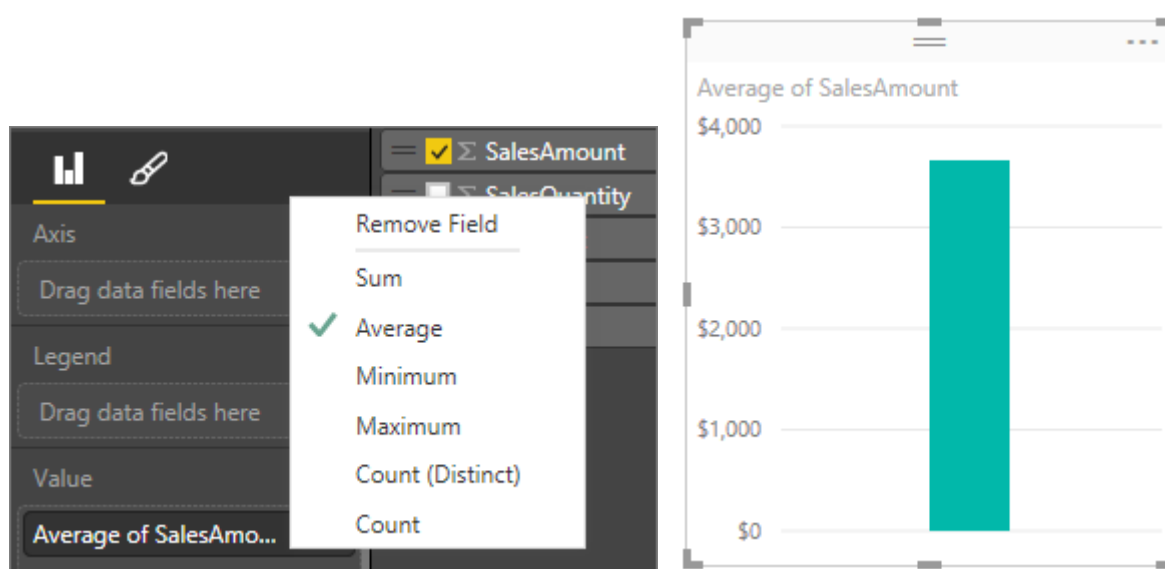
3. Create Custom Measures

3.1 Summarization

1. To complete the steps in this tutorial, you'll need to download the Contoso Sales Sample from <http://download.microsoft.com/download/4/6/A/46AB5E74-50F6-4761-8EDB-5AE077FD603C/Contoso%20Sales%20Sample%20for%20Power%20BI%20Desktop.zip>.
2. Unzip the file *Contoso Sales Sample for Power BI Desktop.zip*, and open *Contoso Sales Sample for Power BI Desktop.pbix*.
3. Drag *SalesAmount* onto the **Report** canvas, a new chart visualization appears.



4. In the **Value** area, select **Average** by clicking the down arrow next to *SalesAmount*. Our visualization changes to an average of all sales values in the *SalesAmount* field.

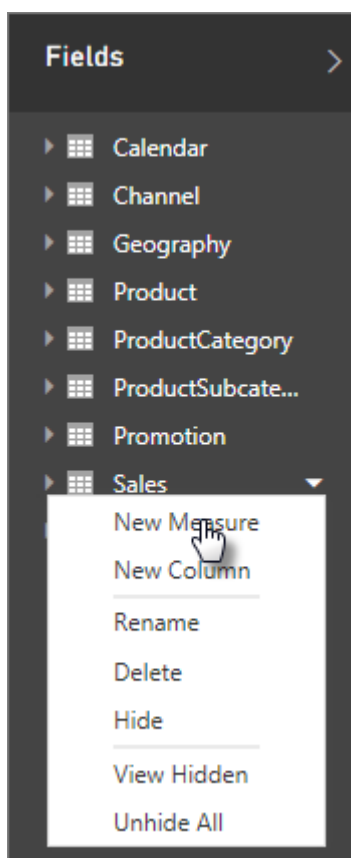


5. Drag the *RegionCountryName* field from the *Geography* table to our chart, sales amounts for each country are averaged and displayed.

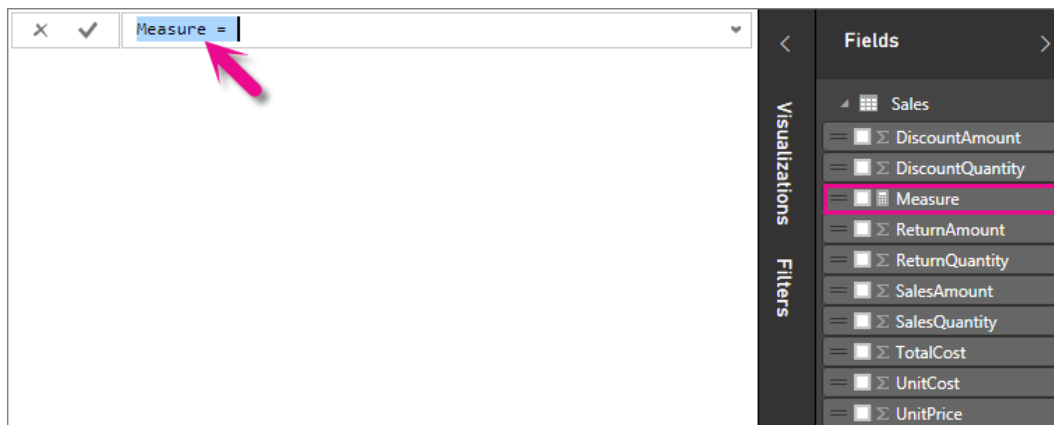


3.2 Creating Custom Measure – Net Sales

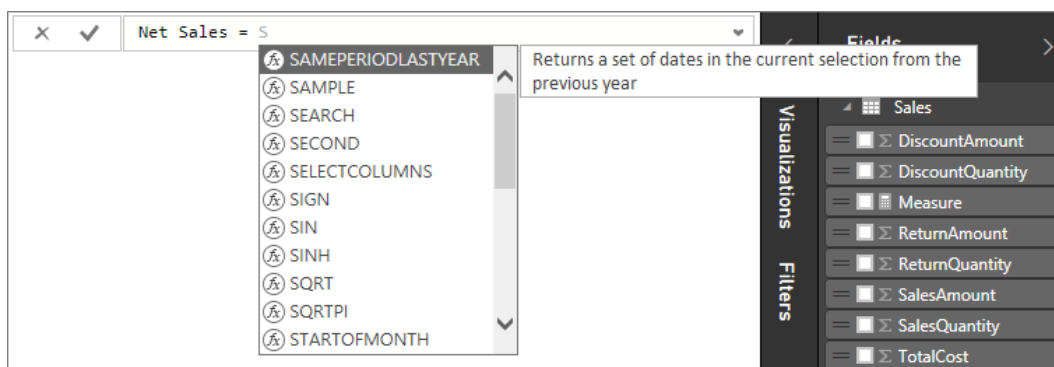
1. Click the down arrow on the *Sales* table in the field list, and then click **New Measure**. This will make sure our new measure is saved in the *Sales* table, where it will be easier to find.



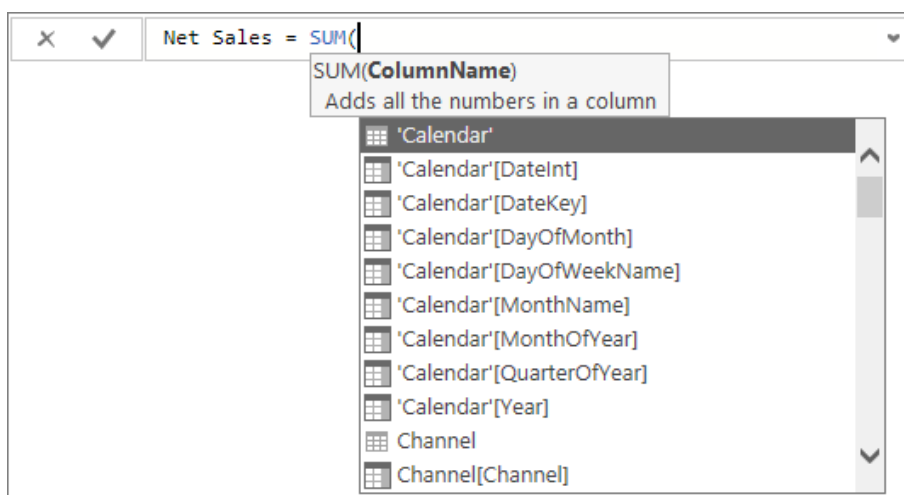
2. The formula bar appears along the top of the **Report** Canvas. This is where we can rename our measure and enter a DAX formula. Highlight *Measure* in the formula bar, and then type *Net Sales*.



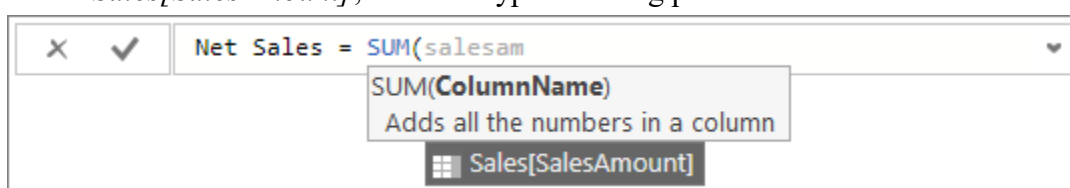
3. Select **SUM** by scrolling down, and then press **[Enter]**.



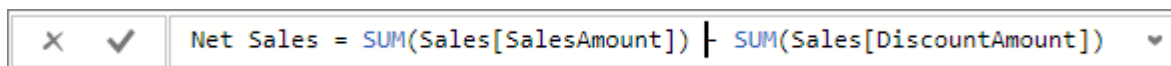
4. After we press **[Enter]**, an opening parenthesis appears along with another suggestion list of all of the available columns we can pass to the SUM function.



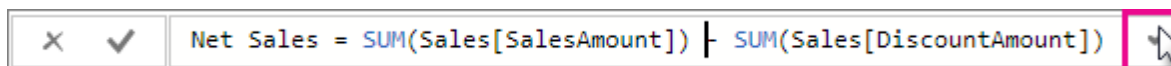
5. Select *Sales[SalesAmount]*, and then type a closing parenthesis



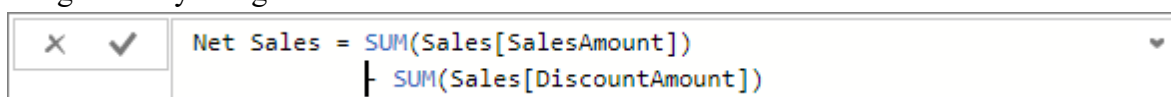
6. After the closing parenthesis for our first expression, type a space, and then a minus operator (-), followed by another space. Then enter another SUM function with the *Sales[DiscountAmount]* column as its argument.



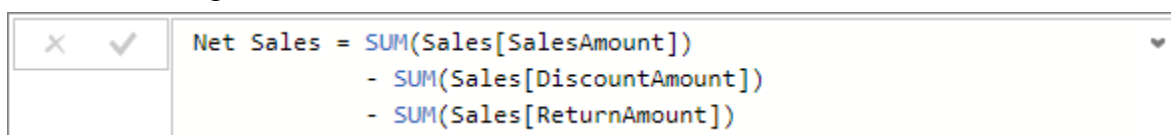
7. Click the down chevron on the right side of the formula bar.



8. 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.



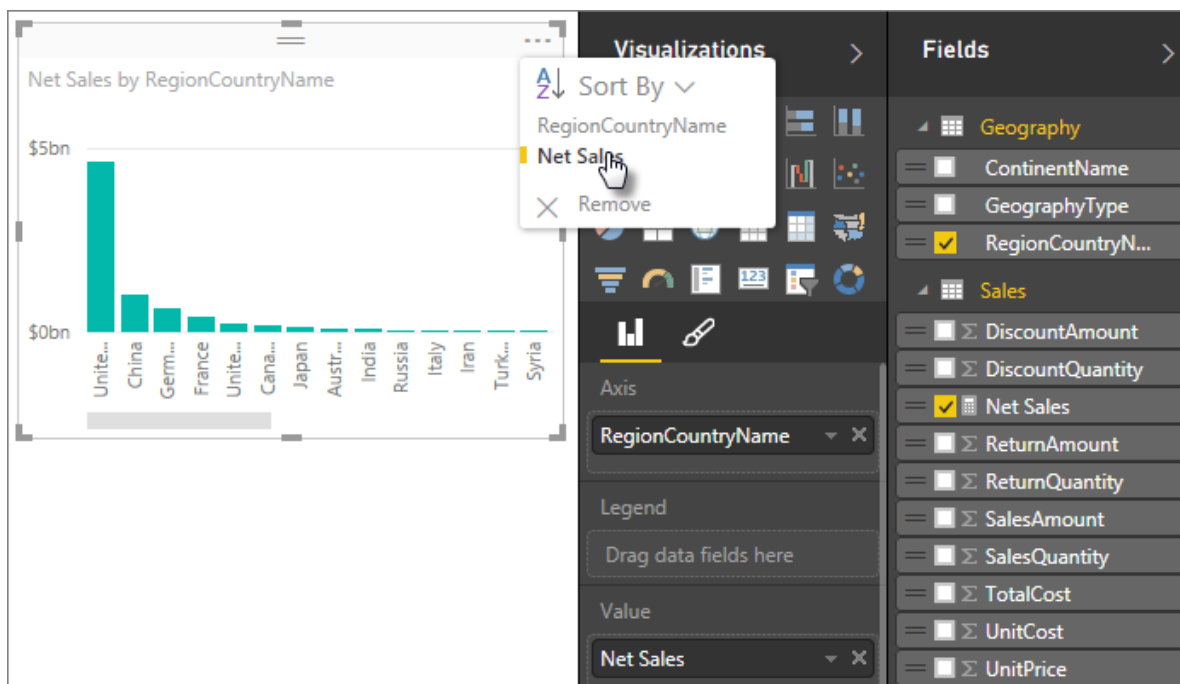
9. Add another minus operator followed by another SUM function and the *Sales[ReturnAmount]* column as its argument.



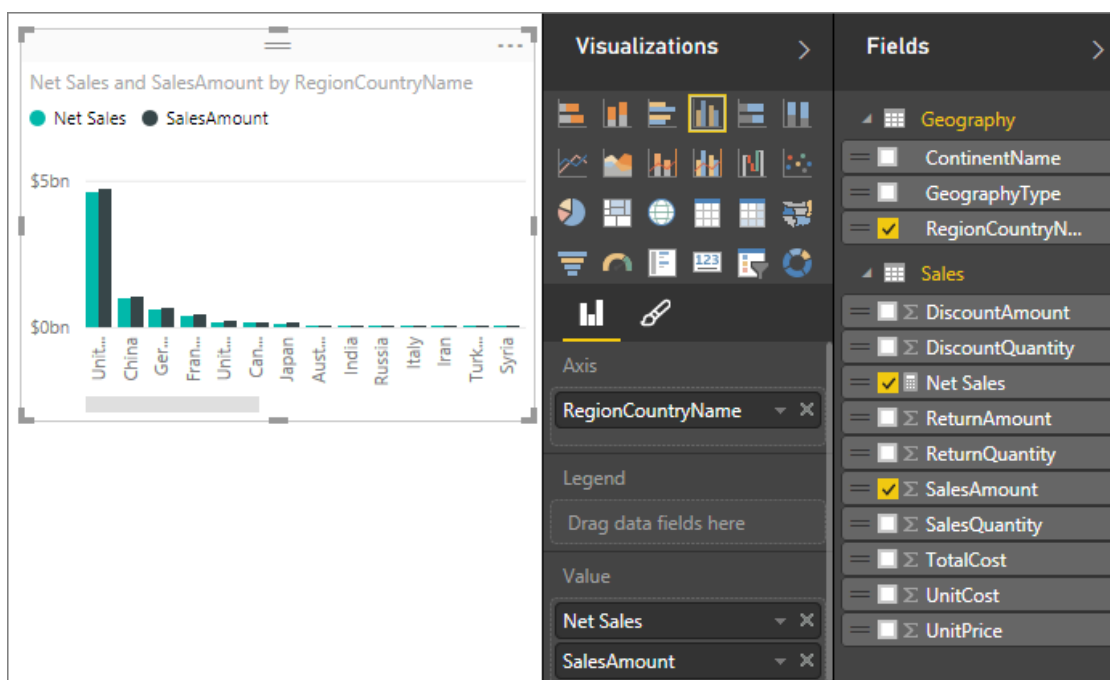
10. Press **[Enter]** in the formula bar to complete, the formula is validated and added to the field list in the *Sales* table.

3.3 Add New Measure to Report

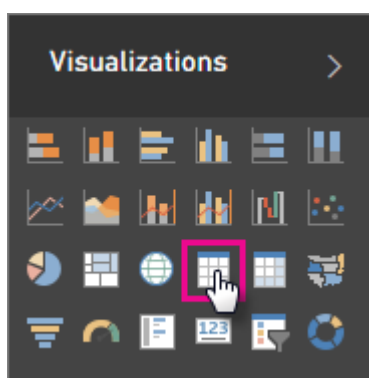
1. Drag the *Net Sales* measure from the *Sales* table onto the **Report** canvas. Now drag the *RegionCountryName* field from the **Geography** table into the chart.



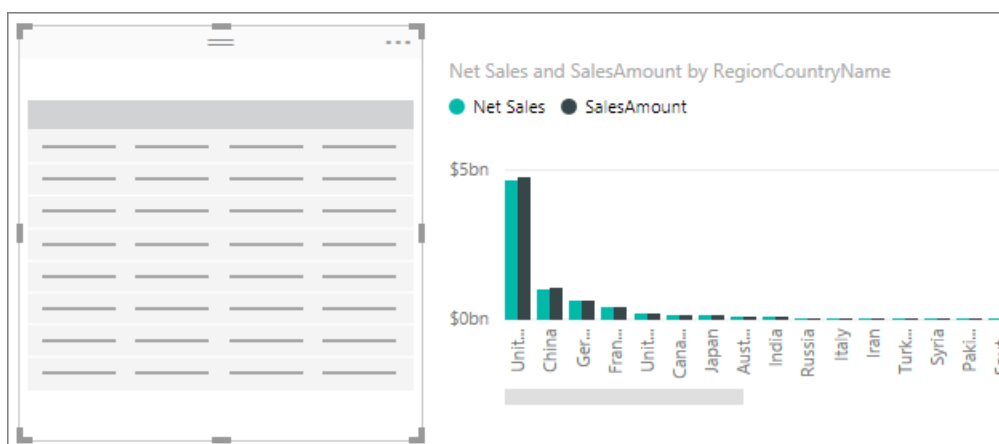
2. Drag the *SalesAmount* field into the chart, to see the difference between net sales and sales amount. We now really have two measures in our chart. *SalesAmount*, which was summed up automatically, and the *Net Sales* measure we created. In each case, the results were calculated in context of another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



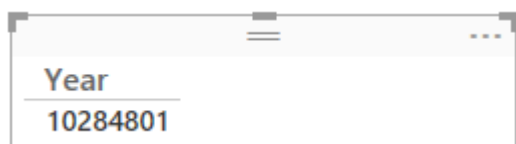
3. Click a blank area next to the chart, then in **Visualizations**, click on the **Table** visualization.



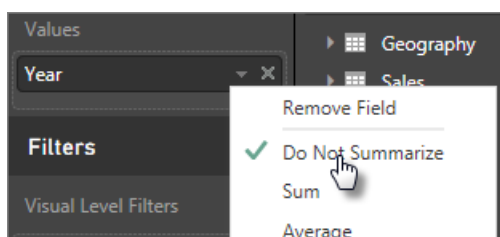
4. This creates a blank table visualization in the **Report** canvas.



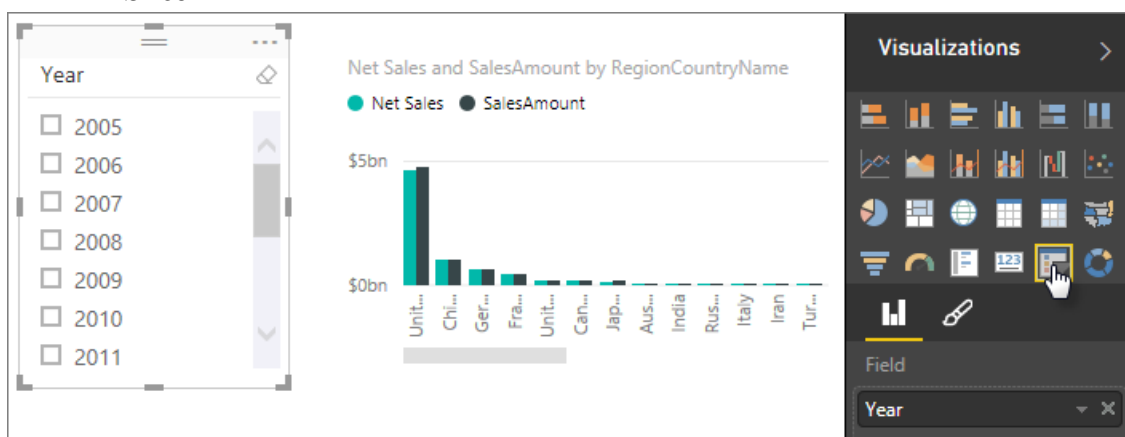
5. Drag the *Year* field from the *Calendar* table into the new blank table.



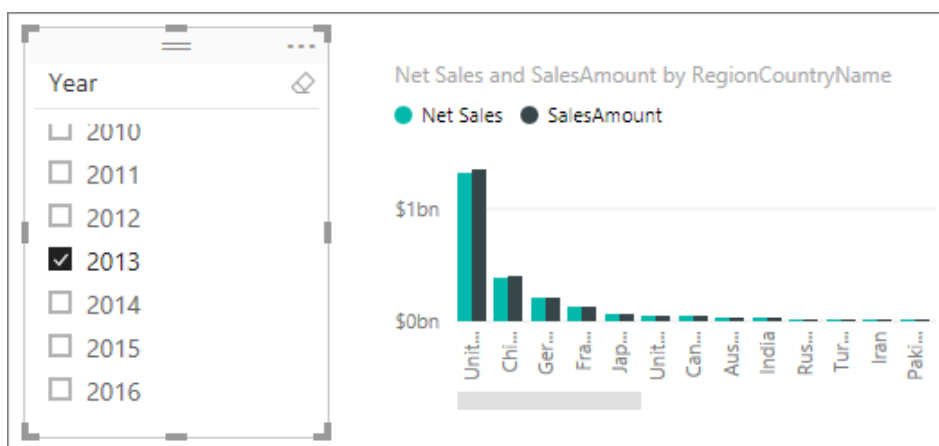
6. Because *Year* is a numeric field, Power BI Desktop summed up its values and gave us a chart. But, that doesn't do us much good as a Slicer. In Values, click the down arrow next to *Year*, and then click **Do Not Summarize**.



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

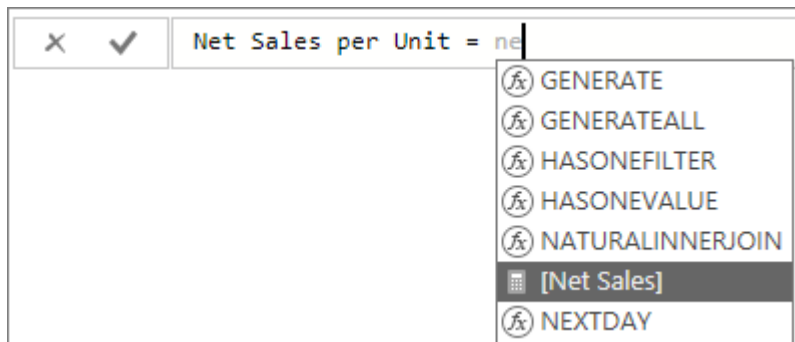


8. Now we have *Year* as a **Slicer**. We can select any individual or group of years and our report's visualizations will all be sliced accordingly.
9. Go ahead and click on *2013*. You'll see the chart change. Our *Net Sales* and *SalesAmount* measures are re-calculated, showing new results just for *2013*.

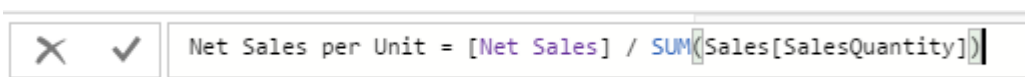


3.4 Create another Measure – Net Sales per Unit

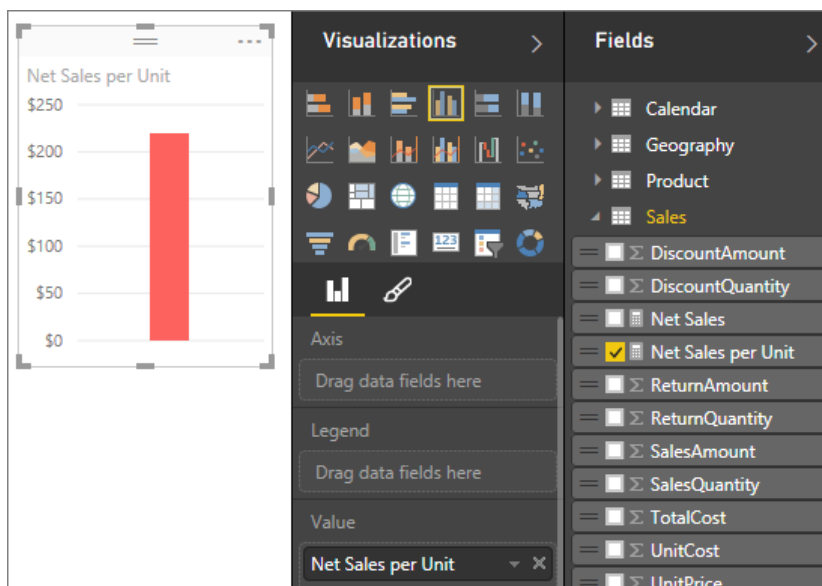
1. Create a new measure named *Net Sales per Unit* in the *Sales* table.
2. Begin typing *Net Sales*. The suggestion list will show what we can add. Select **[Net Sales]**.



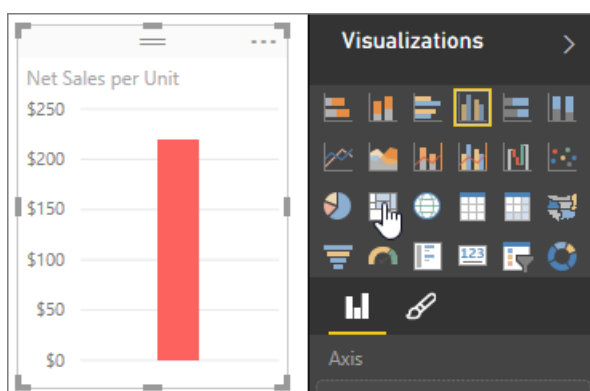
3. Right after *[Net Sales]*, enter a space, then a divide operator (/), then enter a SUM function, then type Quantity. The suggestion list shows all of the columns with Quantity in the name. Select *Sales[SalesQuantity]*. The formula should now look like this:



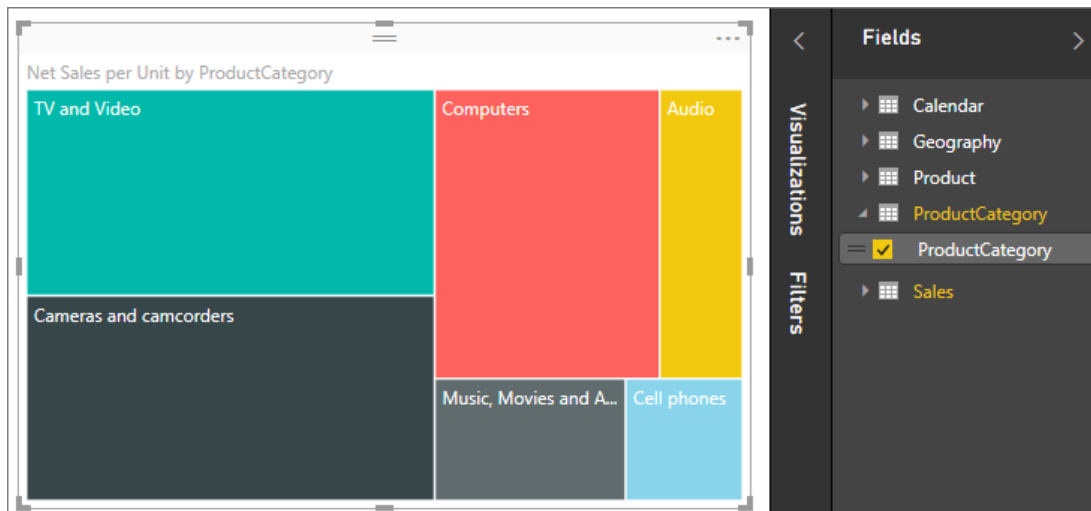
4. Drag the *Net Sales per Unit* measure onto a blank area in the report canvas.



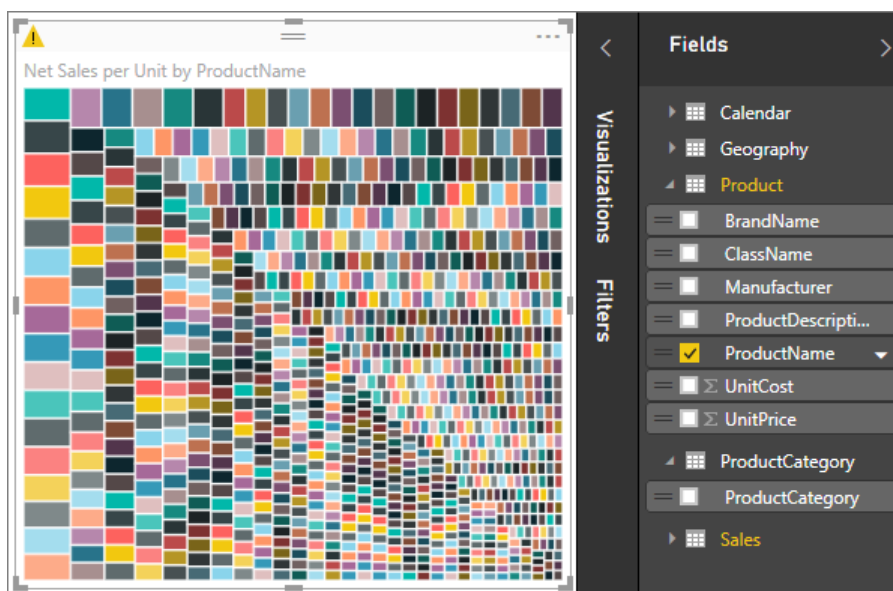
5. Change the chart visualization type to Tree Map.



6. Now drag the *ProductCategory* field from the *ProductCategory* table down into the Group area.



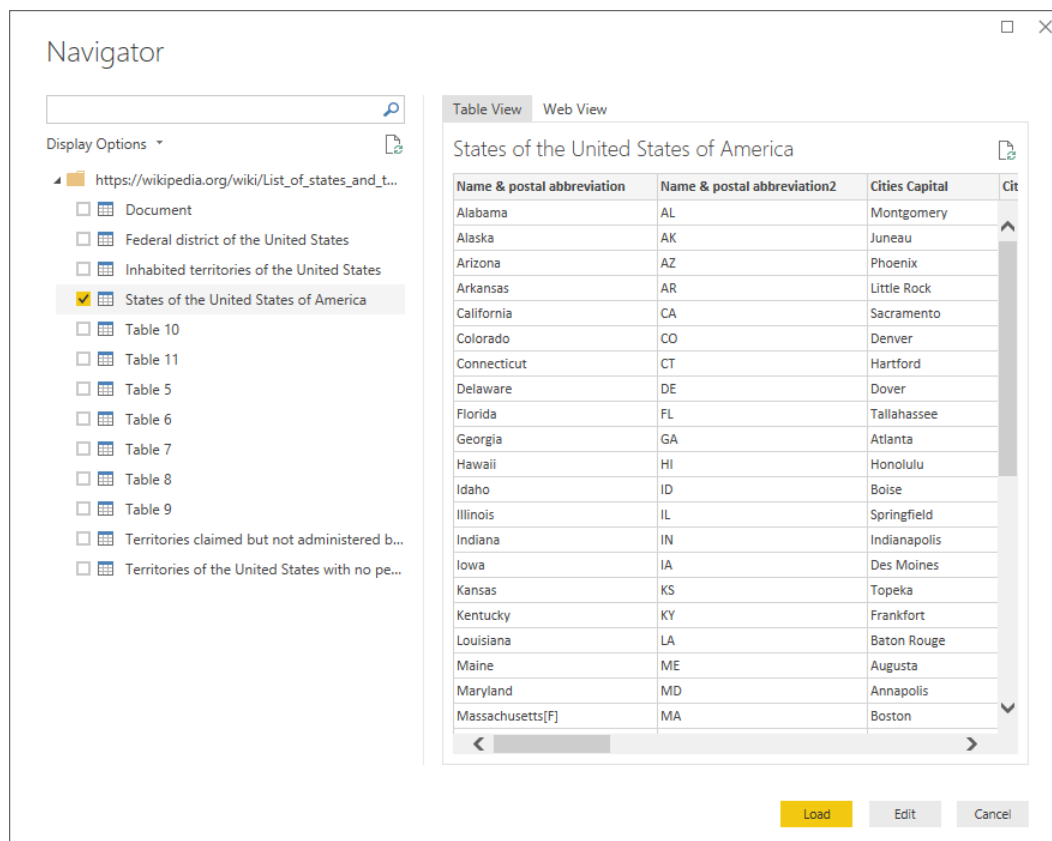
7. Remove the *ProductCategory* field, and then drag the *ProductName* field from the **Product** table down into the **Group** area instead.



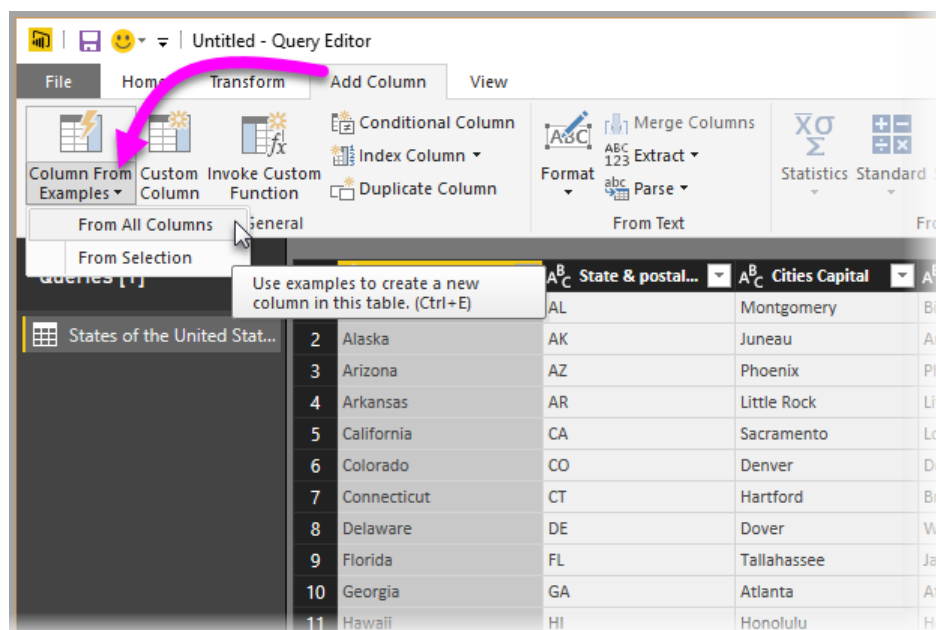
4. Add New Column from Example

4.1 Import Data from Web

1. To complete the steps in this tutorial, you need to import the data from https://wikipedia.org/wiki/List_of_states_and_territories_of_the_United_States

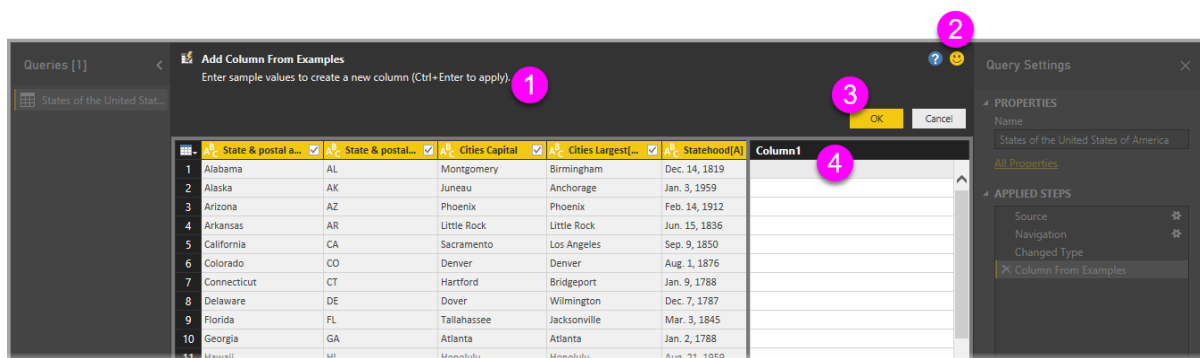


2. Once **Query Editor** is launched and you have some data loaded, you can get started adding a column from examples. To add a new column, in **Query Editor** select the **Add Column** tab on the ribbon and select **Column from Examples**.

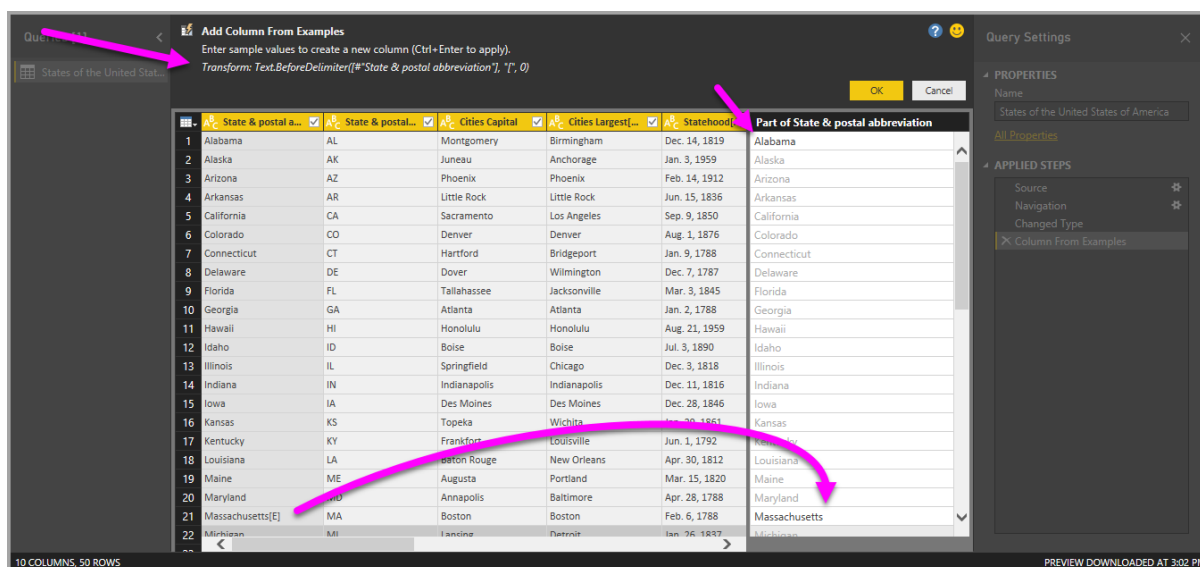


4.2 Add a New Column from Examples

1. This is called the **Add Columns From Examples** pane, and it consists of four primary areas:
 - 1) The **Command** bar which includes a brief description of the feature, or the transformation.
 - 2) The **Send Feedback** option to help Power BI improve this feature.
 - 3) The **[OK]** and **[Cancel]** buttons, which lets you commit your transformations and add the column, or cancel.
 - 4) The new column area, where you can type your sample values in any of the rows, relating to other columns in that row.



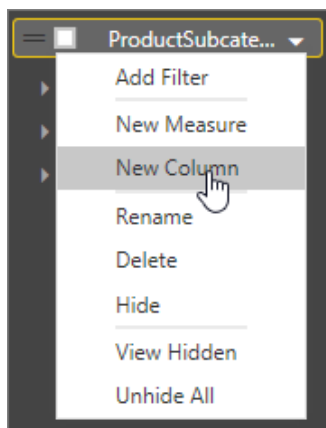
2. Change the column name to “Part of State & Postal” and type your example in the new column, Power BI gives you a preview of how the column it's creating will appear, based on the transformations it detects. For example, we typed Alabama in the first row, corresponding to the Alabama value in the first column of the table. As soon as we hit Enter Power BI fills in the column based on that value. Then we went to the row that included Massachusetts[F] and deleted that last [F] portion and Power BI detected the change, and used the example to create a transformation



5. Create Calculated Columns

5.1 Create a Custom Column

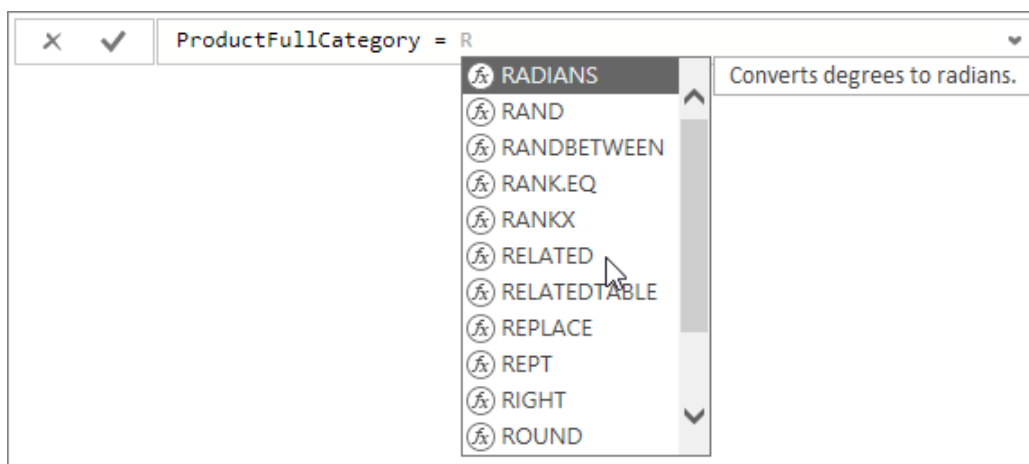
1. To complete the steps in this tutorial, you'll need to download the Contoso Sales Sample from <http://download.microsoft.com/download/4/6/A/46AB5E74-50F6-4761-8EDB-5AE077FD603C/Contoso%20Sales%20Sample%20for%20Power%20BI%20Desktop.zip>.
2. Unzip the file *Contoso Sales Sample for Power BI Desktop.zip*, and open *Contoso Sales Sample for Power BI Desktop.pbix* by Microsoft Power BI.
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.



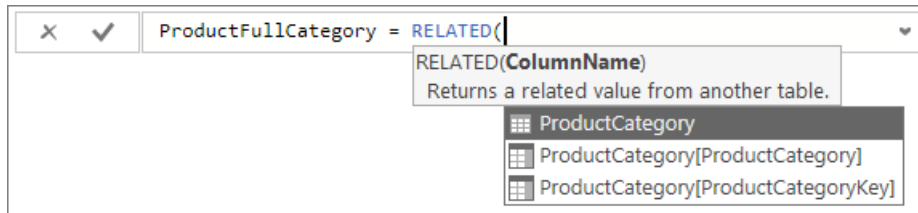
4. The formula bar appears along the top of the **Report** canvas. This is where we can rename our column and enter a DAX formula, type *ProductFullCategory*.



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.



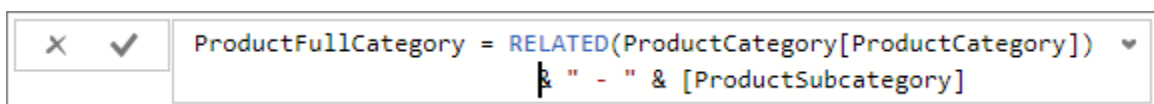
7. Select `ProductCategory[ProductCategory]`, and then type a closing parenthesis.



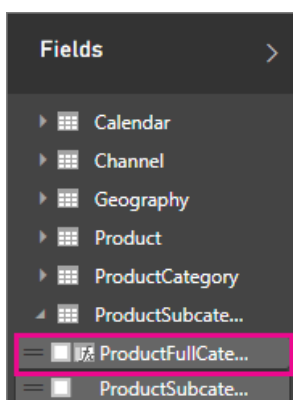
8. We want to add a dash symbol to separate each value, so after the closing parenthesis of the first expression, type a space, ampersand (&), quote, space, dash (-), another space, a closing quote, and then another ampersand.



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 field list in the `ProductSubcategory` table.



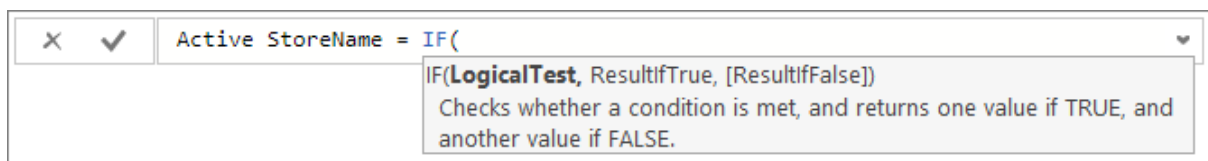
5.2 Add our New Column to Report

1. Drag the *ProductFullCategory* column from the *ProductSubcategory* table onto the Report canvas, and then drag the *SalesAmount* field from the Sales table into the chart.

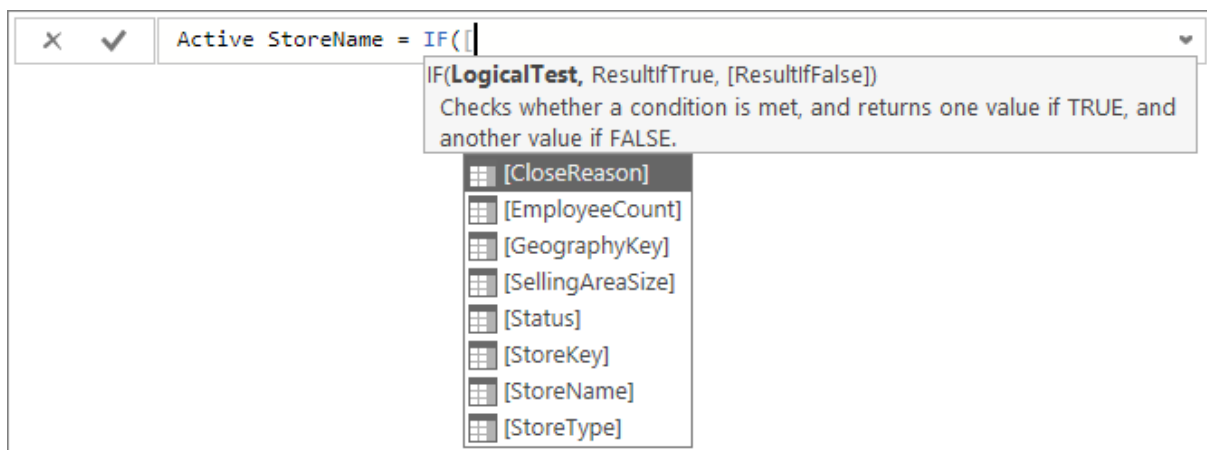
ProductFullCategory	SalesAmount
Audio - Bluetooth Headphones	\$41,907,488.91
Audio - MP4&MP3	\$65,190,616.40
Audio - Recording Pen	\$44,516,259.00
Cameras and camcorders - Camcorders	\$1,335,302,769.92
Cameras and camcorders - Cameras & Camcorders Accessories	\$51,643,775.83
Cameras and camcorders - Digital Cameras	\$365,082,489.93
Cameras and camcorders - Digital SLR Cameras	\$809,994,738.39
Cell phones - Cell phones Accessories	\$120,017,198.26
Cell phones - Home & Office Phones	\$48,333,160.98
Cell phones - Smart phones & PDAs	\$423,389,457.48
Cell phones - Touch Screen Phones	\$300,493,447.58

5.3 Create a Logical Column

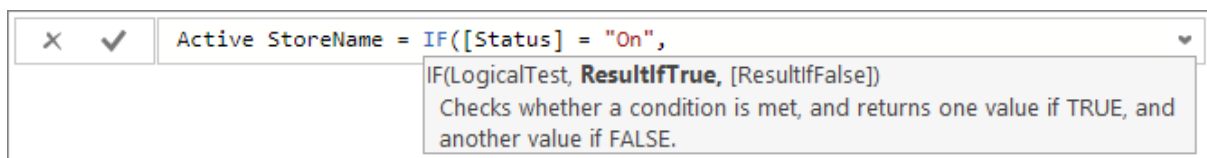
1. Create a new calculated column named *Active StoreName* in the *Stores* table. For this column, our DAX formula is going to check each stores status. If a stores status is On, our formula will return the stores name. If it's Off, it will have the name, "Inactive". Select **IF**.



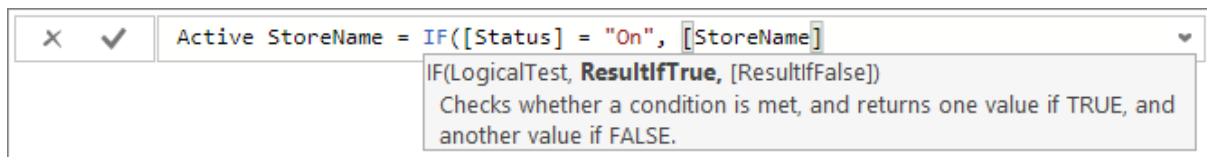
2. Type an opening bracket `[`, and select `[Status]` from the *Stores* table.



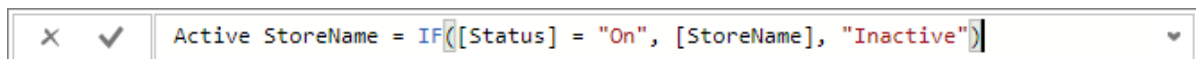
3. Right after `[Status]`, type `= "On"`, then enter a comma `,` to enter the second argument. The tooltip suggests we need to add the value for when the result is true.



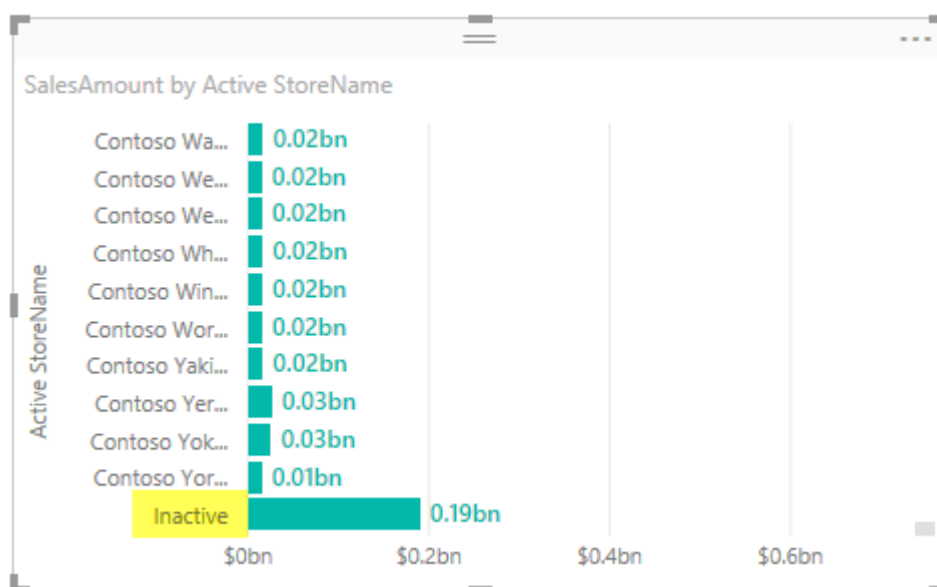
4. If the store is On, we want to show the store's name. Type an opening bracket [and select the *[StoreName]* column, and then type another comma so we can enter our third argument.



5. 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 field list in the *Stores* table.



6. Put the new *Active StoreName* column in visualizations. 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 Measure

6.1 Using IF in Measure

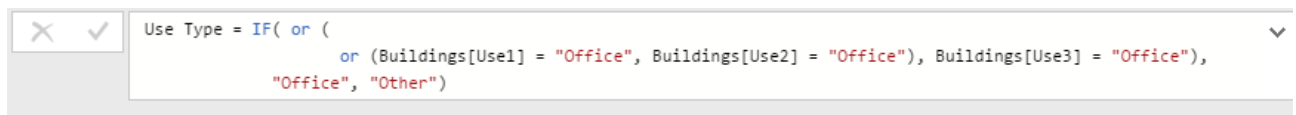
1. Create a new Power BI Desktop file, and load the **Buildings** worksheet from the Excel workbook “**Skyscrapers.xlsx**”. Then create a table for building name, city and use. There are 3 columns giving the uses to which each building is put.

Building Name	City	Use1	Use2	Use3
Zifeng Tower	Nanjing	hotel	office	
Yantai Shimao No. 1 The Harbour	Yantai	residential	hotel	office
Wuxi Suning Plaza 1	Wuxi	hotel	serviced apartments	office
Wuxi Maoye City - Marriott Hotel	Wuxi	hotel		
Wuxi International Finance Square	Wuxi	hotel	office	
Willis Tower	Chicago	office		
Wenzhou Trade Center	Wenzhou	hotel	office	

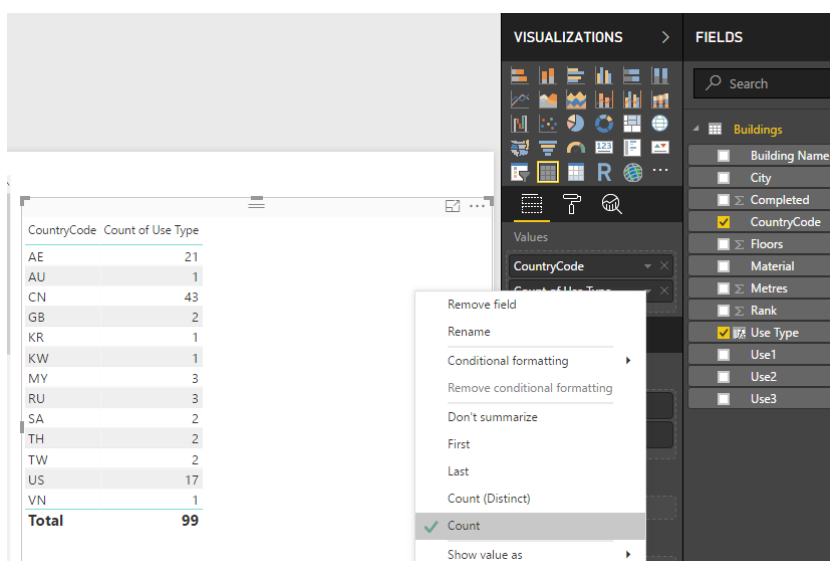
2. Use the IF function to create a new column to show for each office its use type:

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

3. 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")



4. Create a table by using *CountryCode* and *Use Type*, then 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 “**Percentage of Grant Total**”

CountryCode	Count of Use Type	%GT Count of Use Type
AU	1	1.01%
KR	1	1.01%
KW	1	1.01%
VN	1	1.01%
GB	2	2.02%
SA	2	2.02%
TH	2	2.02%
TW	2	2.02%
MY	3	3.03%
RU	3	3.03%
US	17	17.17%
AE	21	21.21%
CN	43	43.43%
Total	99	100.00%

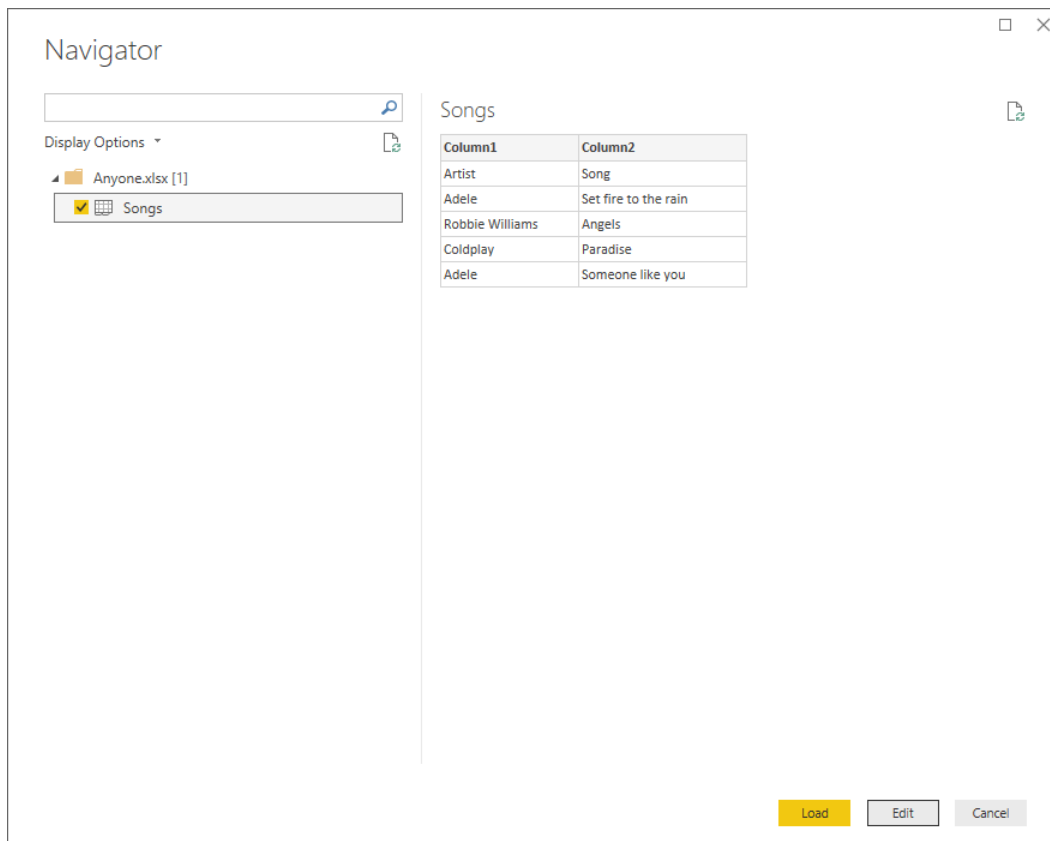
6. The final result should show like this:

CountryCode	Count of Use Type	%GT Count of Use Type
AU	1	1.01%
KR	1	1.01%
KW	1	1.01%
VN	1	1.01%
GB	2	2.02%
SA	2	2.02%
TH	2	2.02%
TW	2	2.02%
MY	3	3.03%
RU	3	3.03%
US	17	17.17%
AE	21	21.21%
CN	43	43.43%
Total	99	100.00%

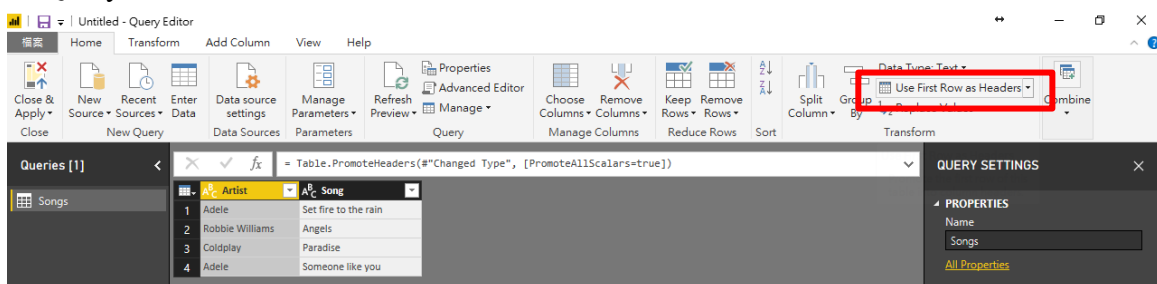
7. Parameter

7.1 Define Data Source by Parameter

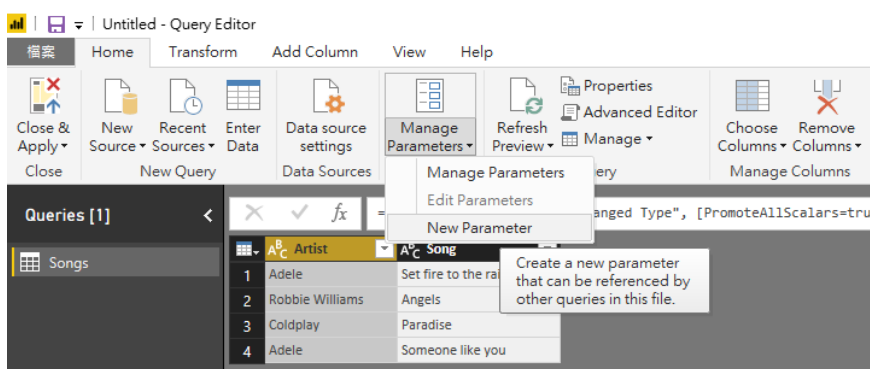
1. Select **Get Data** → **Excel** in the **Home** ribbon tab. In the **Open File** dialog box, select the *Anyone.xlsx* file. Then in the **Navigator** pane, select the **Song** table and then select **[Edit]**.



2. In Query Editor, select **Use First Row as Headers**.



3. Select **Manage Parameter** → **New Parameters**



4. Create the first parameter “**FilePath**” to store the folder path to where your Excel files are stored

Parameters

New

FilePath

Name

FilePath

Description

The folder path to where your Excel files are stored

☒ Required

Type

Any

Suggested Values

Any value

Current Value

C:\Temp\

OK Cancel

5. Create the second parameter “**Playlist**” to the name of the Excel workbook to load (including file extension)

Parameters

New

FilePath Playlist

Name

Playlist

Description

The name of the Excel workbook to load (including file extension)

☒ Required

Type

Any

Suggested Values

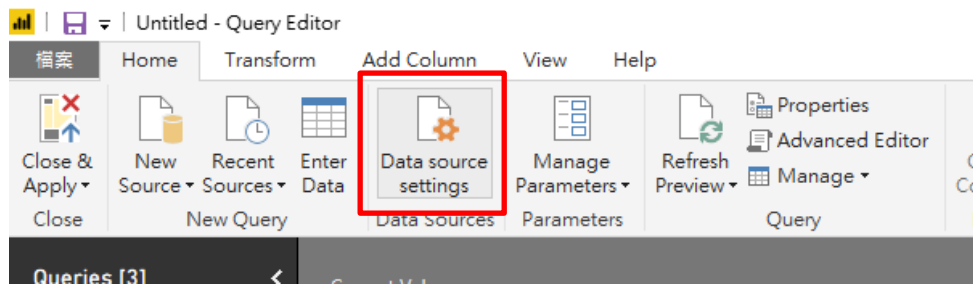
Any value

Current Value

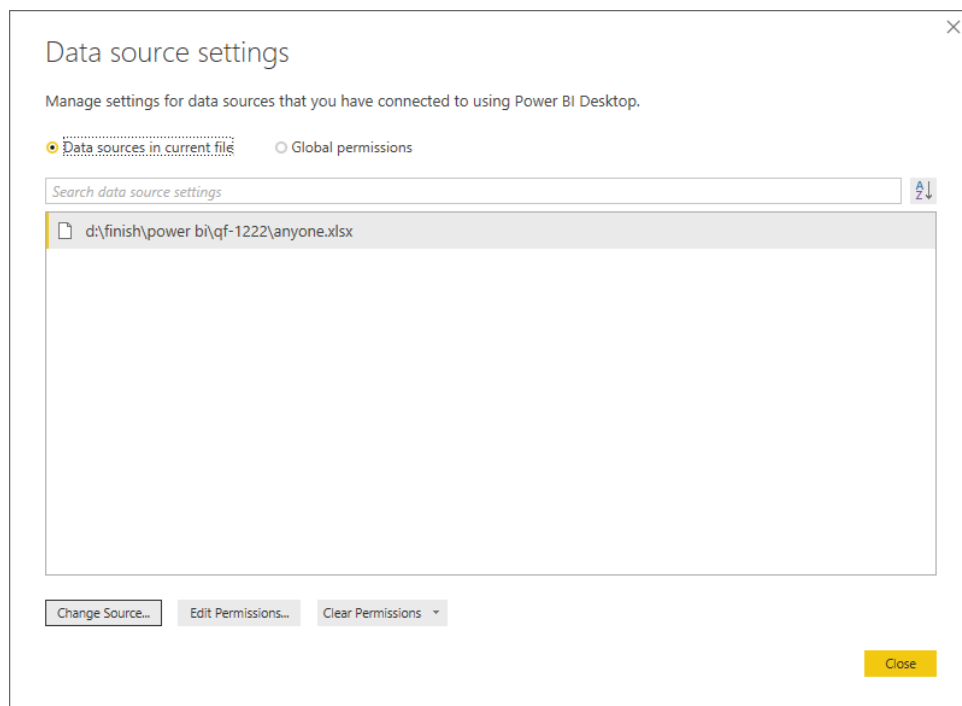
Anyone.xlsx

OK Cancel

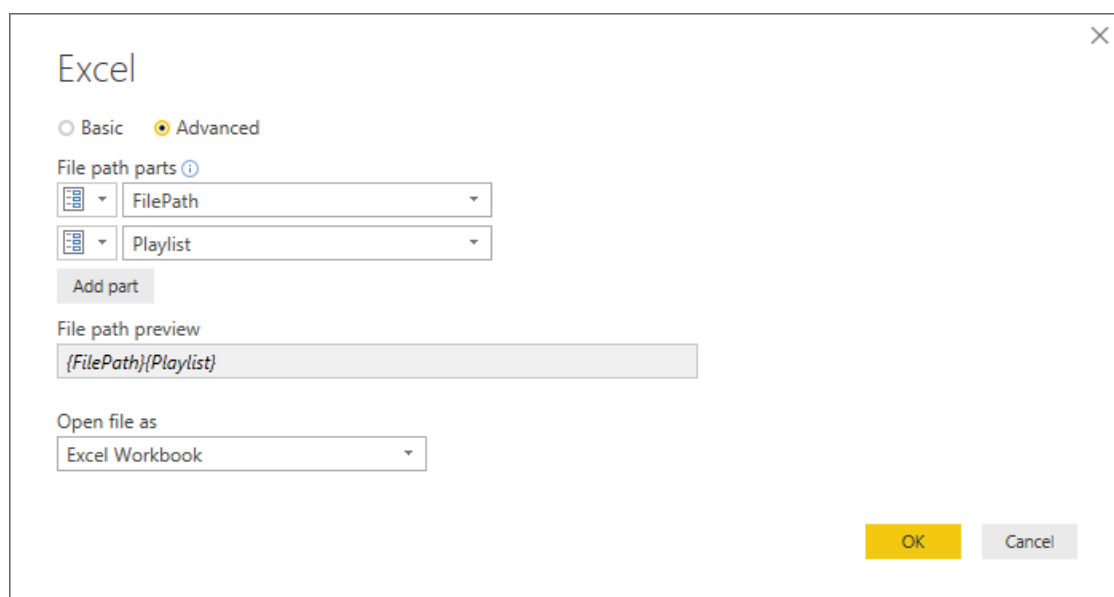
6. Select **Data Source Setting**.



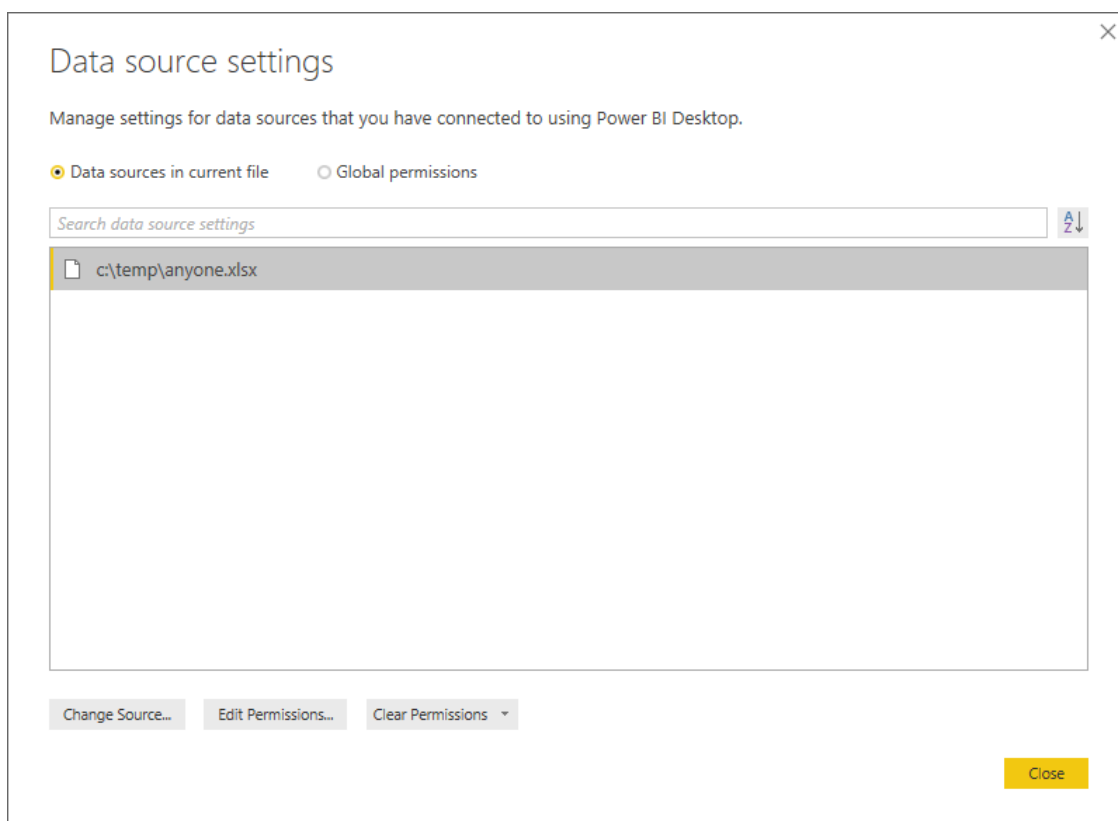
7. Select **[Change Source]**



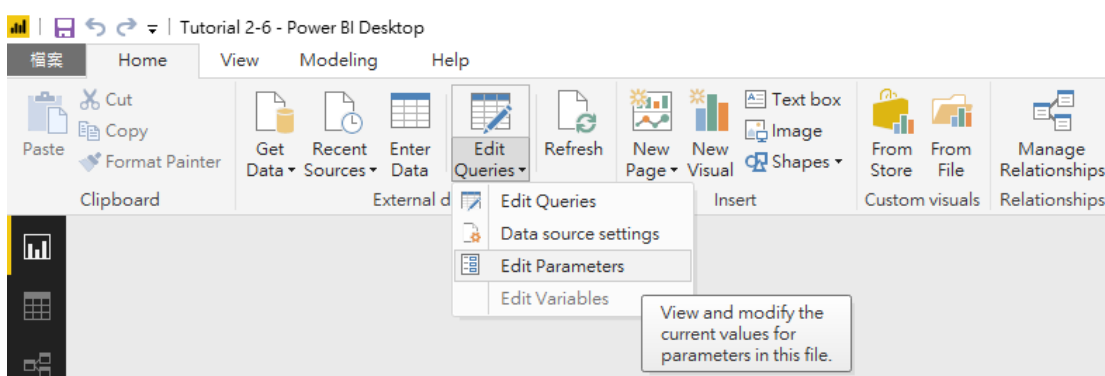
8. Change the file path part to parameter and press **[OK]** to confirm



9. Press **[Close]** button to leave the Data source setting dialog. Then click **[Close and Apply]**.



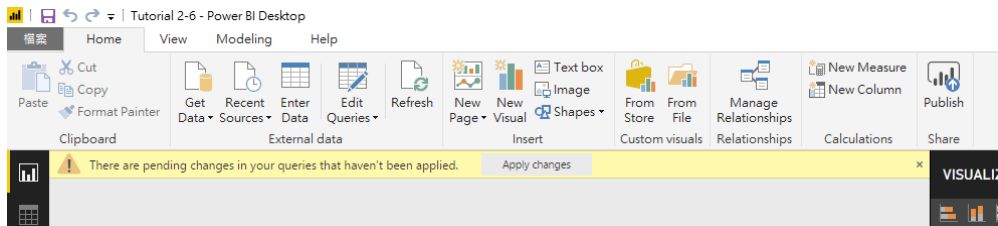
10. Select **Edit Queries** → **Edit Parameters**.



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



12. Press [Apply Change] button to reload the data.



13. The data from your specified file are loaded.

