**List of Exercises – Data Analytics:**

**Part A:**

1. **Introduction to power BI desktop and learn to create a workspace. (Create power BI account in www.powerbi.com -> create workspace->share workspace.**

* **Task 1: Sign in to the Power BI service**

In this task, you will sign in to the Power BI service.

Step 1: go to chrome browser

Step 2: Type **powerbi.com**

Step 3: Login using your account details (Student login & Password)

* **Task 2: Create a workspace**

In this task, you will create a workspace. You will also upgrade your account to Power BI Pro.

All content you develop in this course will be added to this workspace.

* To create a workspace, in the **Navigation** pane (located at the left), click **Workspaces**, and then click **Create a Workspace**.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

* When prompted to upgrade your account to Power BI Pro, click **Try Free**.

A yellow and red rectangle with black text

Description automatically generated

* When prompted to start the 60-day free Pro trial, click **Start Trial**.

A yellow and red rectangle with black text

Description automatically generated

* When the trial has been extended, click **Close**.
* Repeat the first step in this task to create the workspace.
* In the **Create a Workspace** pane (located at the right), in the **Workspace Name** box, enter a name for your workspace.

**Note: The name you enter must be unique within the tenant.**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

* To create the workspace, at the bottom of the pane, click **Save**.

A yellow rectangle with black text

Description automatically generated

* In the **Navigation** pane, notice that your workspace is open.

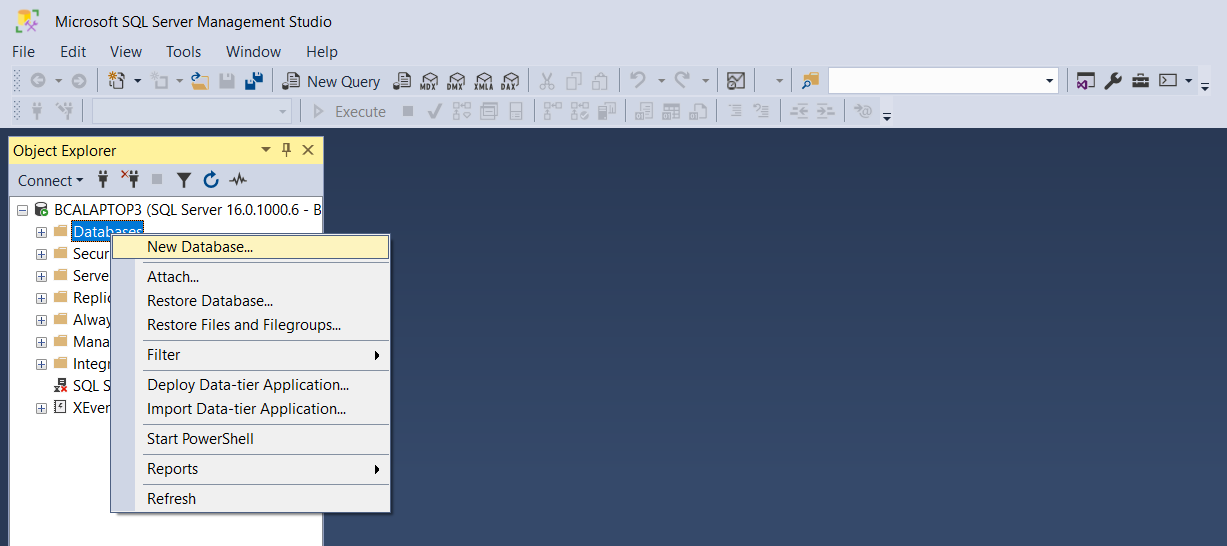
1. Prepare Data in Power BI desktop. (Prepare data using SSMS tool -> Then load data in Power BI desktop using get data from servers).

Task 1: Open SSMS Tool.

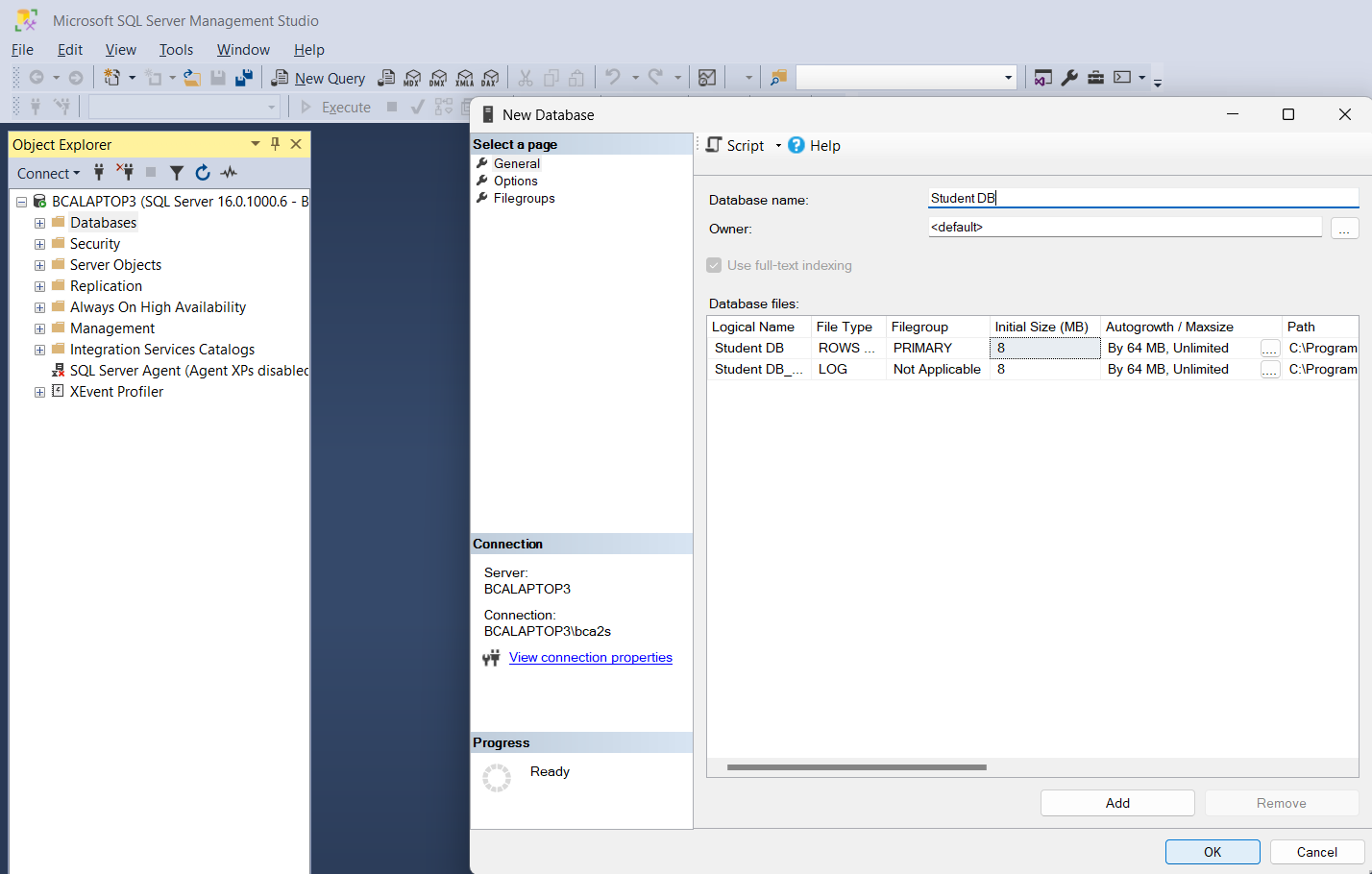
A computer screen shot of a computer

Description automatically generated

**Create New Database: Right click on Database and click on New Database.**



**Give Database Name and click on Ok.**



Student Database will be created.

**Task 2:**

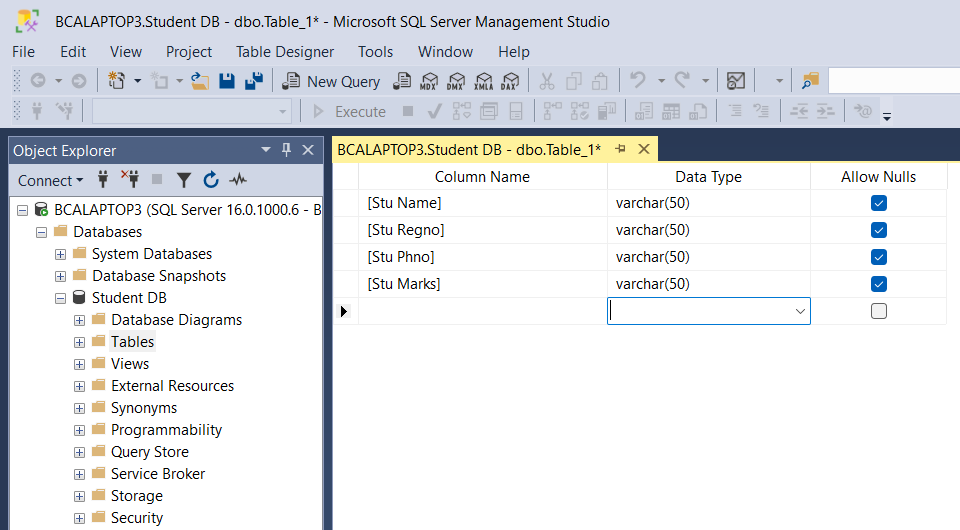
Select and click on + symbol and select Table and right click and select new 🡪 Table.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Create columns

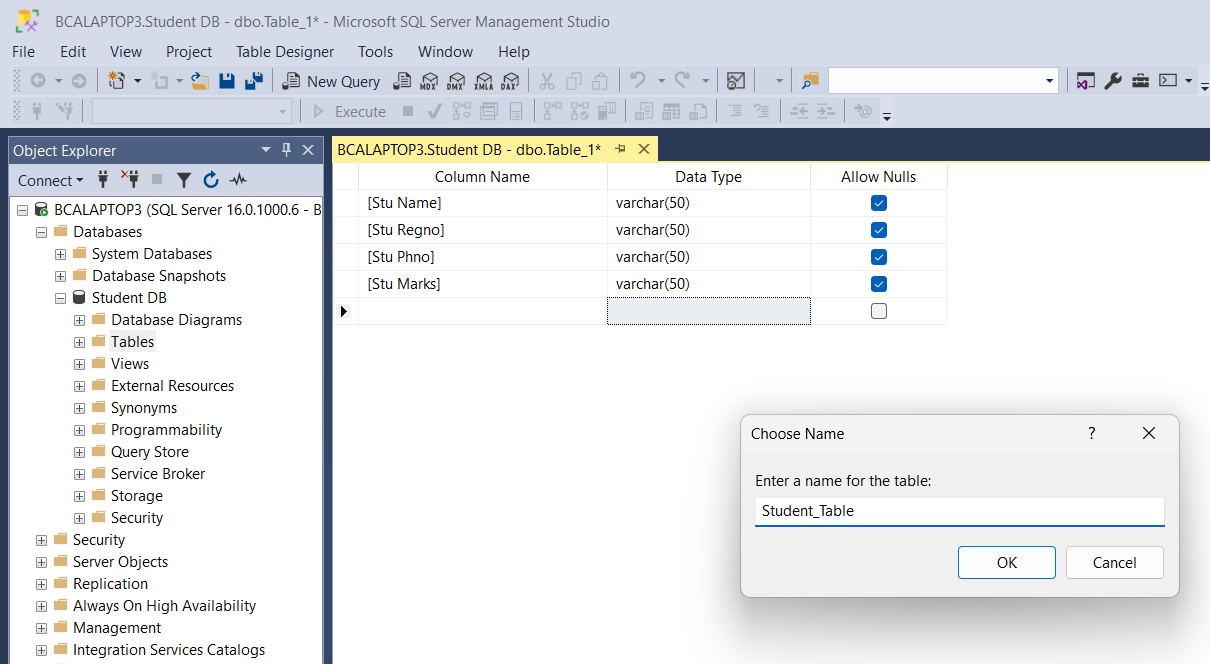


Right click and save table:

A screenshot of a computer

Description automatically generated

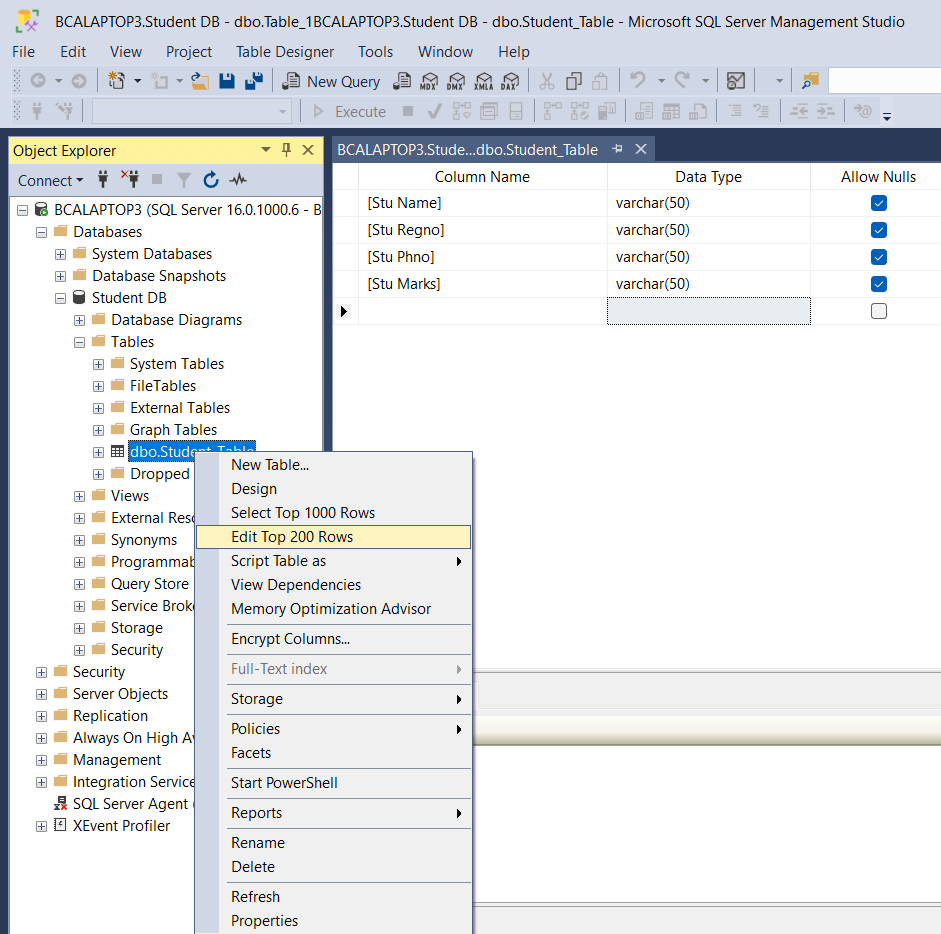
Enter table name and click on Ok.



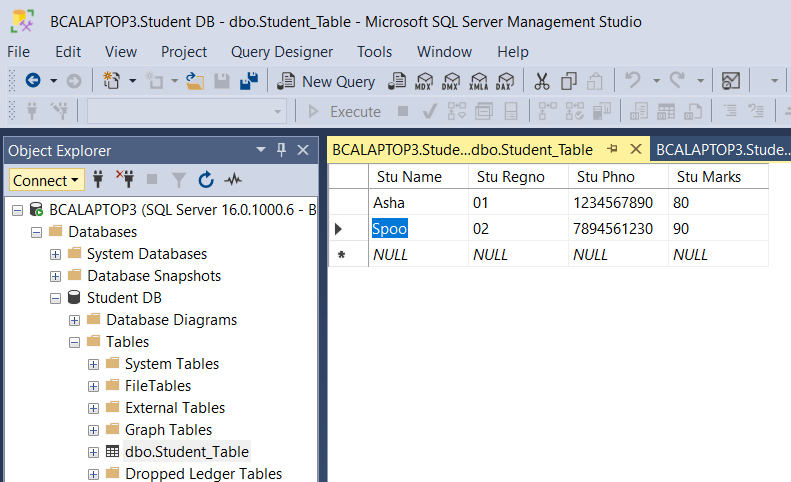
**Refresh the DB: Table will be visible.**

**Task 3:**

Right click on Table and select Edit Top 200 Rows and Insert values into the table.



**Insert Values.**



**Task 4:**

A screenshot of a computer

Description automatically generated**After creation of Table, insert values into the table – Go to Power BI Desktop**

Select SQL Server and Connect.

A screenshot of a computer

Description automatically generated

Give Server name which is in SSMS and click on OK

A screenshot of a computer

Description automatically generated

Select use my current credentials and click on Connect.

A screenshot of a computer

Description automatically generated

Select Student DB which was created using SSMS.

A screenshot of a computer

Description automatically generated

Select Student DB and select table and click on Load.

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

Once the Loding is completed click on Report view on top left.

A screenshot of a computer

Description automatically generated

The table and values will be visible.

A screenshot of a computer

Description automatically generated

3. Transforming data in Power BI desktop (Transform data can be done by using different transform techniques like merge columns, split columns, replace values, change datatypes, add or remove rows/columns, Unpivot columns, Applied steps etc.)

<https://hevodata.com/learn/power-bi-transform-data/>

<https://www.geeksforgeeks.org/query-editor-in-power-bi-for-data-transformation/>

**Task 1: Go to Power BI and select SQL server.**

**Select Student DB 🡪 and click on Transform Data.**

**A screenshot of a computer

Description automatically generated**

**New power query window will open.**

**A screenshot of a computer

Description automatically generated**

To perform above mentioned task, you should have multiple tables in the Student DB.

**Merging Columns in Power BI**

* **Step 1**: Choose the two columns you wish to merge and go to “**Add Column**”.
* **Step 2**: Next, choose the “**Merge Columns**” option.
* **Step 3**: The previous step prompts the Merge Columns page where you can specify the specific separator for the columns and click on “**OK**” to finish this Power BI transform data step.

**Using the Pivots in Power BI**

* **Step 1**: Select the column you wish to pivot.
* **Step 2**: On the Transform tab in the **Any column** group, choose the **Pivot column**.
* **Step 3**: Next, in the pivot dialog box, in the Value column list, you can choose **Value**.
* **Step 4**: By default, Power Query will try to execute a sum as the aggregation, but you can choose the Advanced option to sift through the other available aggregations before deciding on the one you like.

**Unpivoting Data in Power BI**

Therefore, you might want to separate it into two columns where one column shows you the Sale made and another column depicts the Region.

* **Step 1**: Choose all four columns, right-click on any heading, and then choose “**Unpivot Columns**”. This will result in two new columns, called Value and Attribute as shown below. You can rename these two columns by following the steps mentioned in the Renaming Columns section.

**How to Change/Modify Data Types in Power BI?**

* **Step 1**: Select the column and “**Right Click**” on the column header.
* **Step 2**: This prompts a drop-down list from which you can select “**Change Type**”.
* **Step 3**: When you select it, a drop-down list appears with a list of different data types to choose from. You can choose the data type you like, say **Fixed Decimal Number**. Here are the results for the same:

**Rename Operation in Power BI**

The power query editor provides us with a variety of possible renaming. We can rename the data sources or tables, columns, and queries.

**Rename Data Source**

It can be achieved by right-clicking on the source and opting for the rename option. Here, we have renamed the data source from sheet 1 to Movie Data. On transformation, we get:

A screenshot of a computer

Description automatically generated

**Rename Column**

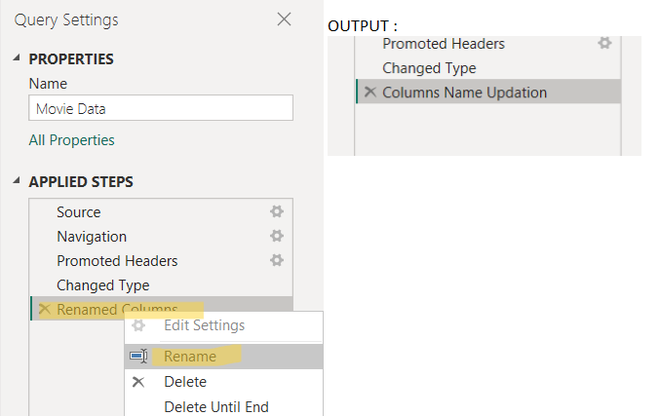
It’s also done by right clicking the column in which you wish to change the name and selecting the option of renaming and renaming as per the wish of the user. Here, we have renamed the column “TITLE” to “MOVIE NAME”. On renaming the column it appears as:

A screenshot of a computer

Description automatically generated

**Rename Query**

Under the query settings pane an “Applied Steps” section under which all the changes we made are stored as queries. Using the query editor, we can rename the queries also. Here, we have renamed the query “Renamed Columns” to “Columns Name Updation”. On renaming it appears as:



**Change Data Types of a Column in Power BI**

We can change or modify the data type of any column by right-clicking on the column which is followed by a drop-down under which we have to select the option change type which further provides us with varied suitable data types.

Here, we have modified the data type of column “RANK”. Earlier, it was of “Whole Number” type which has been modified to “Decimal Number” by us. This can be observed from the image below as well, where before implementing the operation the numbers mentioned prior to the RANK are 1,2,3 i.e. symbolizing whole numbers. Whereas after the operation is being done those (1,2,3) changed to 1.2 i.e. indication of decimal numbers.

This same operation can also be achieved from an option present in the home bar of the ribbon as “Data type:<type>”.

A screenshot of a computer

Description automatically generated

**Format Operation in Power BI**

The Format feature is provided in the “Transform” tab of the ribbon. It’s useful in formatting the text. We have further options for formatting the text like we can change the data to lowercase or UPPERCASE, also we can add prefixes or suffixes. Formatting has other options as well like trim and clean. Trim is used to remove leading and trailing whitespaces for each data entry of the specifically selected column. Clean is used to remove nonprintable characters of selected columns. Performing all these operations is very simple as it just involves clicking over the option. And so, we have performed only UPPERCASE operation here. Rest all can also be done in the same fashion.

A screenshot of a computer

Description automatically generated

**Removal Operation in Power BI**

**Remove Row Operation**

The removal of rows operation is supported by the “Reduce Rows” block. This feature is present in the “Home” bar. The removal of rows operation has several sub-operations like removing rows(removing the top, bottom, or alternate rows) and columns, removing duplicates, removing blank rows, and removing errors. Here is an illustration performing the removal of the bottom-most row as follows. Rest all removal operations can also be performed in the same fashion.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated*Removing the bottom row*

*“Reduce Rows” Block*

**Remove Column Operation in Power BI**

The removal of columns operation provides us with the feature to remove columns or multiple columns. Remove column is present in the “Home” bar.

The “Remove Columns” feature is supported by the “Manage Columns” block. On clicking Remove columns it provides whether you want to remove the selected column by clicking on “Remove Columns”.

We can also remove the rest of the columns other than the selected one by selecting the option “Remove Other Columns”.

Here, we will illustrate the “Remove Columns” by removing the column “RATING”. After the operation will be performed “RATING” will not exist anymore. The “Remove Other Columns” can be performed the way.

A screenshot of a computer

Description automatically generated

**Output:**

A screenshot of a computer

Description automatically generated

**Merge Operation in Power BI**

We can merge multiple columns. To select multiple columns hold down the Ctrl key, navigate to the columns you want to be selected using the left and right arrows, and then press the Space bar to actually select those columns. The “Add Column” bar supports the “Merge Columns” feature which is followed by a prompt of merge columns that asks for the name of the merged column and to set the separator. Here, we have merged the columns “GENRE” and “RATING”. We have used a custom separator ” 😉 ” and kept the name of the merged column simple as “Merged”. The illustration along with the output is as follows :

A screenshot of a computer

Description automatically generated

*Merging Columns GENRE and RATING*

**Output:**

A screenshot of a computer

Description automatically generated

**Replace Values Operation in Power BI**

Replace values operation replaces some specific value to our desired value. It’s present in the “Transform” bar as “Replace Values”. Here, we have replaced “null” to “geeksforgeeks” for column “GENRE”.

A screenshot of a computer

Description automatically generated

*Replacing values of column “GENRE”*

A screenshot of a computer

Description automatically generated

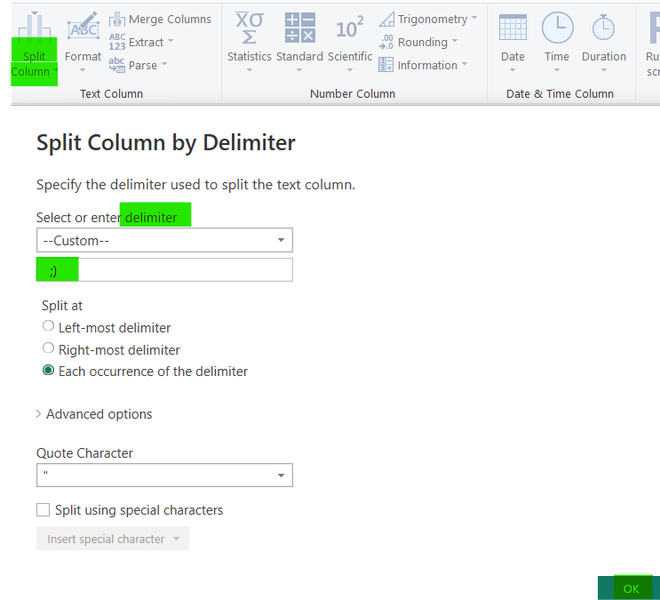
A screen shot of a computer

Description automatically generated**Output:**

*Successfully Replaced*

**Split Column Operation in Power BI**

It’s present in the “Transform” bar on the ribbon. In the transform bar lies an operation as “split column”. We can split either by using a delimiter, by providing no. of characters or positions, and so on. Here, we have split the column “Merged” using the customized delimiter” 😉 ” that splits the column into two columns “Merged.1” and “Merged.2”. The illustration is as follows:



*Splitting column “Merged”*

**Before**

A screen shot of a movie list

Description automatically generated

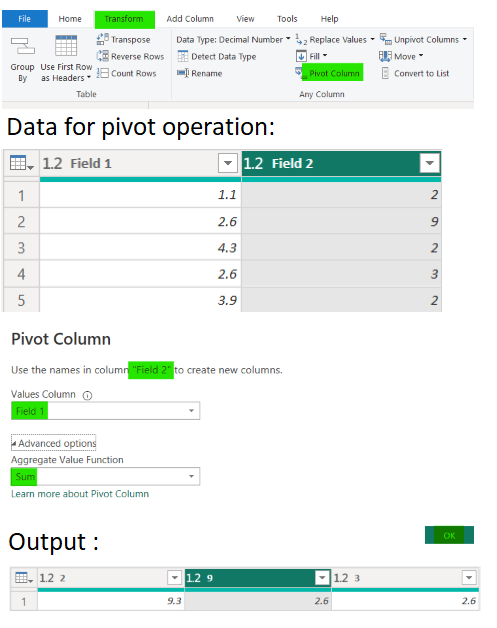
*Before Split Scenario*

A screenshot of a computer

Description automatically generated**After**

*After Split Scenario*

**Pivot Column Operation in Power BI**

The pivot operation basically turns rows into columns. By default, the query editor does sum as aggregation which can also be set as don’t aggregate or minimum or maximum or whatever as per the user’s wish from available options. It is present in the “Transform” bar.

*Pivot column “Field 2”*

**Unpivot Column Operation in Power BI**

A screenshot of a computer

Description automatically generatedUnpivot column operation as the name also suggests does the opposite of what pivot does. Unpivot basically unpacks similar values and gathers them under one label. When we did unpivot on the same data we took for the pivot operation it produces the output:

*Unpivot operation implemented on Field 1 and Field 2 both*

**Output:**

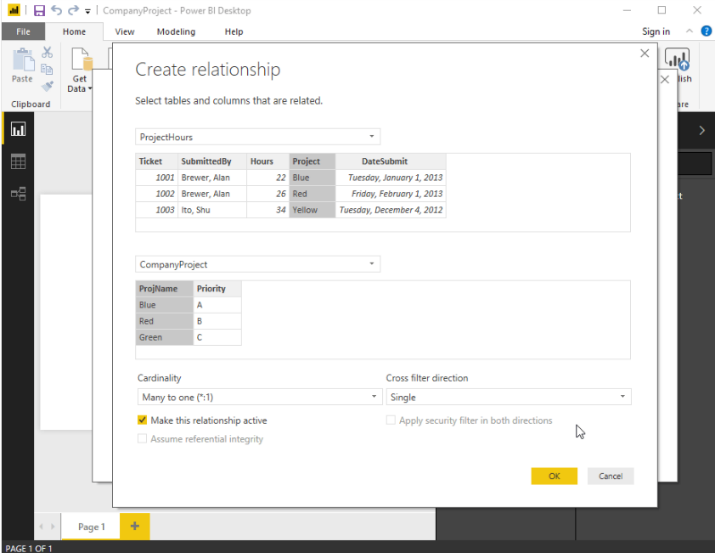
A screenshot of a computer

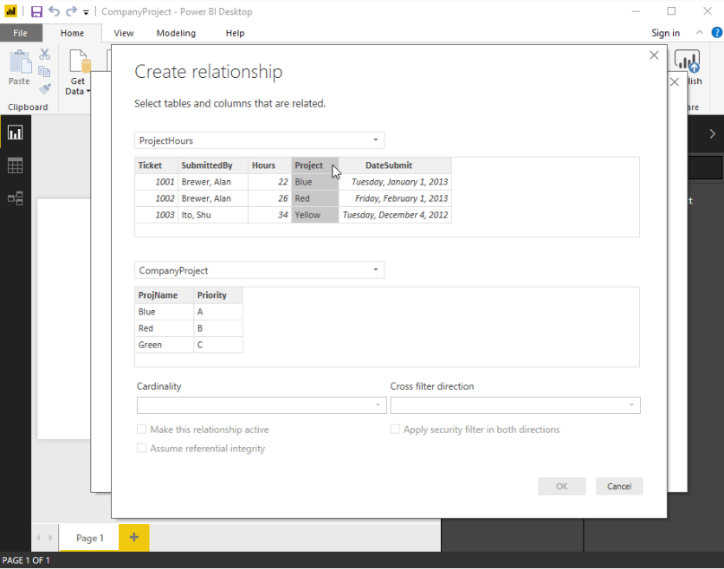
Description automatically generated

4. Developing Data Modelling in Power BI Desktop (creating the relationship among the table like One to One, one to many, many to one, many to many).

<https://learn.microsoft.com/en-us/power-bi/transform-model/desktop-create-and-manage-relationships>

**Create a relationship manually**

1. On the **Modelling** tab, select **Manage relationships** > **New**.
2. In the **Create relationship** dialog box, in the first table drop-down list, select a table. Select the column you want to use in the relationship.
3. In the second table drop-down list, select the other table you want in the relationship. Select the other column you want to use, and then select **OK**.



By default, Power BI Desktop automatically configures the options **Cardinality** (direction), **Cross filter direction**, and **Make this relationship active** for your new relationship. However, you can change these settings if necessary.

If none of the tables selected for the relationship has unique values, you'll see the following error: One of the columns must have unique values. At least one table in a relationship must have a distinct, unique list of key values, which is a common requirement for all relational database technologies.

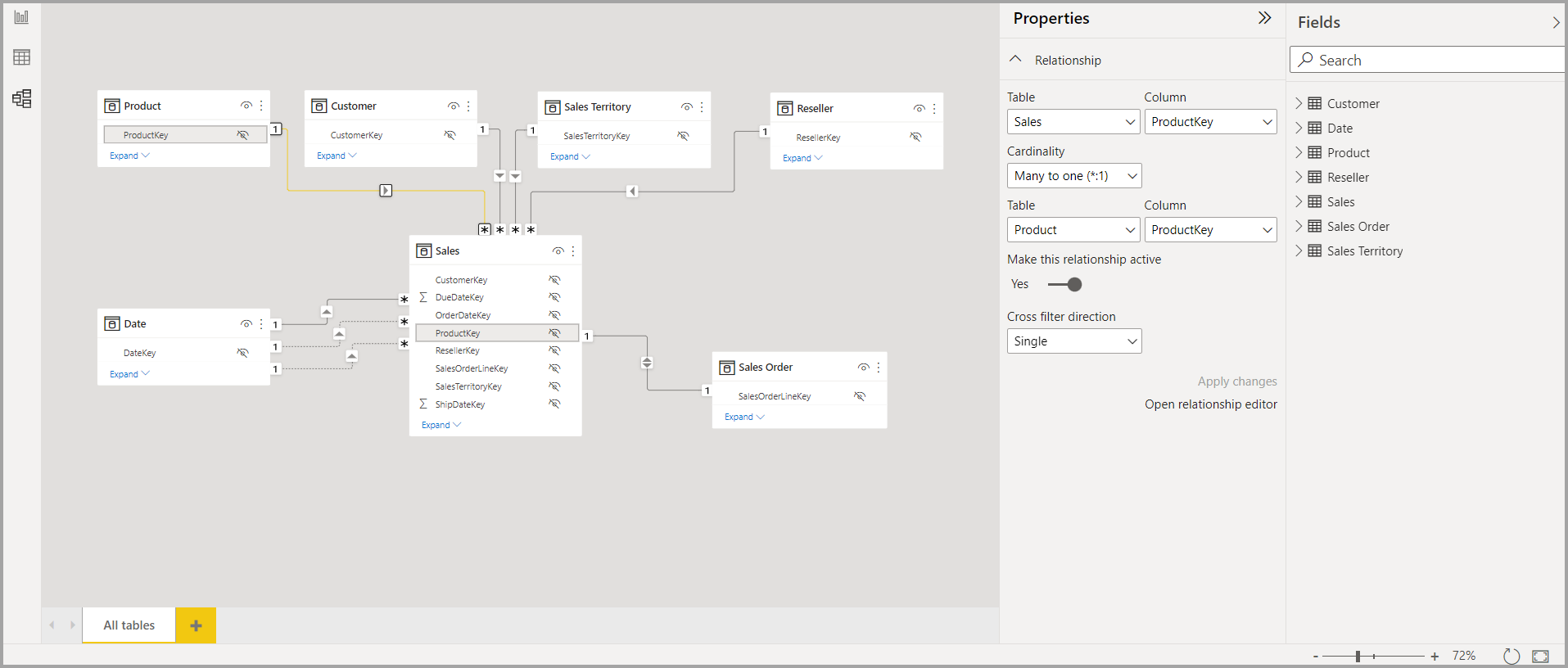
If you encounter that error, there are a couple ways to fix the issue:

* Use **Remove Duplicates** to create a column with unique values. The drawback to this approach is that you might lose information when duplicate rows are removed. Often a key (row) is duplicated for good reason.
* Add an intermediary table made of the list of distinct key values to the model, which will then be linked to both original columns in the relationship.
* Alternatively, in the **Model view** diagram layouts, you can drag and drop a column from one table to a column in another table to create a relationship.

**Edit a relationship**

There are two ways to edit a relationship in Power BI.

The first method to edit a relationship is using the **Editing relationships in the Properties** pane in **Model view**, where you can select any line between two tables to see the relationship options in the **Properties** pane. Be sure to expand the **Properties** pane to see the relationship options.



The other method of editing a relationship is using the **Relationship editor dialog**, which you can open many ways from within Power BI Desktop. The following list shows different ways you can open the **Relationship editor dialog**:

From **Report view** do any of the following:

* Select the **Modeling** ribbon > **Manage relationships**, then select the relationship and select **Edit**.
* Select a table in the **Fields** list then select the **Table tools** ribbon > **Manage relationships**, then select the relationship and then select **Edit**.

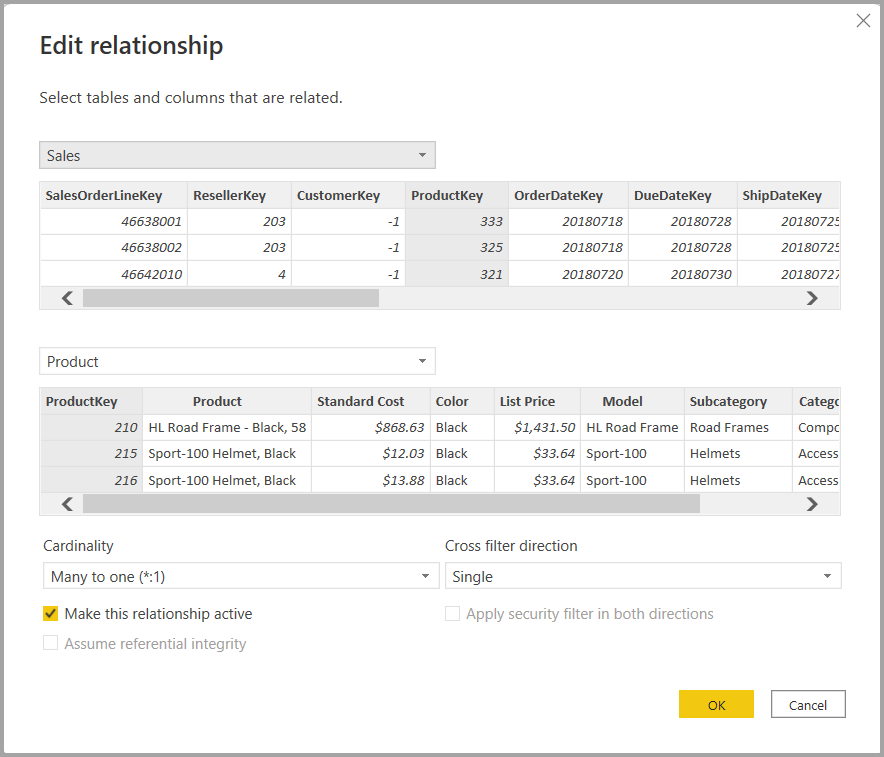
From the Data view, select the **Table tools** ribbon > **Manage relationships**, then select the relationship and then choose **Edit**.

From the **Model** view do any of the following:

* Select the **Home** ribbon > **Manage relationships**, then choose the relationship and then select **Edit**.
* Double-click any line between two tables.
* Right-click any line between two tables and then choose **Properties**.
* Select any line between two tables, then choose **Open relationship editor** in the **Properties** pane.

Finally, you can also edit a relationship from any view, right-click or select the ellipsis to get to the context menu of any table, then select **Manage relationships**, select the relationship and then select **Edit**

The following image shows a screenshot of the **Edit relationship** window.



5. Create Quick measures & Hierarchy in Power BI Desktop.

<https://learn.microsoft.com/en-us/power-bi/transform-model/desktop-tutorial-create-measures>

Many common calculations are available as *quick measures*, which write the DAX formulas for you based on your inputs in a window. These quick, powerful calculations are also great for learning DAX or seeding your own customized measures.

Create a quick measure using one of these methods:

* From a table in the **Fields** pane, right-click or select **More options** (**...**), and then choose **New quick measure** from the list.
* Under **Calculations** in the **Home** tab of the Power BI Desktop ribbon, select **New Quick Measure**.

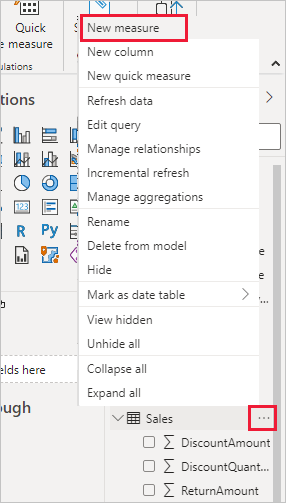
**Create a measure**

Suppose you want to analyze your net sales by subtracting discounts and returns from total sales amounts. For the context that exists in your visualization, you need a measure that subtracts the sum of DiscountAmount and ReturnAmount from the sum of SalesAmount. There's no field for Net Sales in the **Fields** list, but you have the building blocks to create your own measure to calculate net sales.

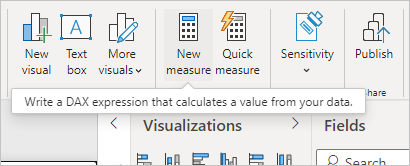
To create a measure, follow these steps:

1. In the **Fields** pane, right-click the **Sales** table, or hover over the table and select **More options** (**...**).
2. From the menu that appears, choose **New measure**.

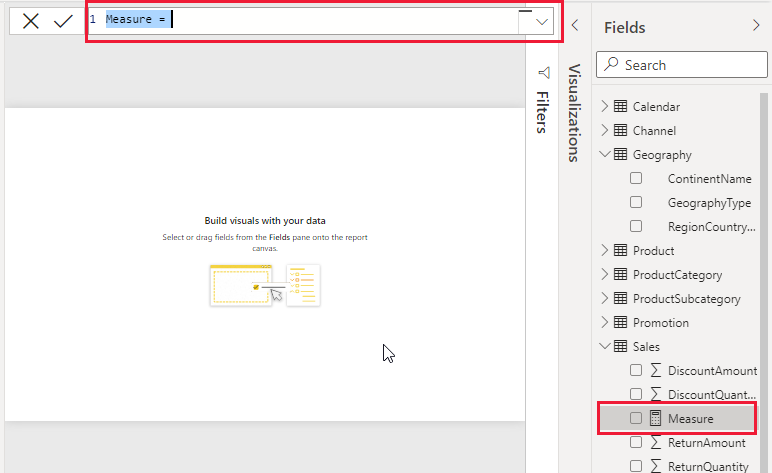
This action saves your new measure in the **Sales** table, where it's easy to find.



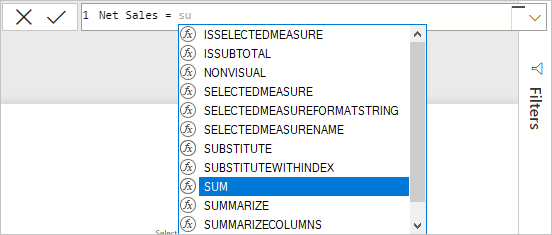
You can also create a new measure by selecting **New Measure** in the **Calculations** group on the **Home** tab of the Power BI Desktop ribbon.



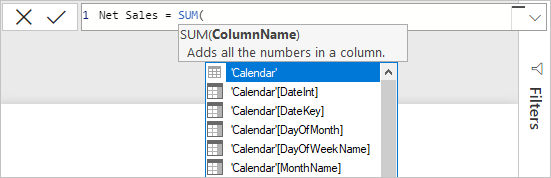
The formula bar appears along the top of the report canvas, where you can rename your measure and enter a DAX formula.



1. By default, each new measure is named *Measure*. If you don’t rename it, new measures are named *Measure 2*, *Measure 3*, and so on. Because we want this measure to be more identifiable, highlight *Measure* in the formula bar, and then change it to *Net Sales*.
2. Begin entering your formula. After the equals sign, start to type *Sum*. As you type, a drop-down suggestion list appears, showing all the DAX functions, beginning with the letters you type. Scroll down, if necessary, to select **SUM** from the list, and then press **Enter**.

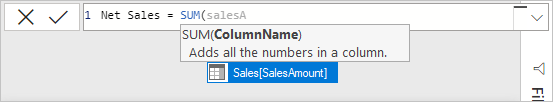


An opening parenthesis appears, along with a drop-down suggestion list of the available columns you can pass to the SUM function.



1. Expressions always appear between opening and closing parentheses. For this example, your expression contains a single argument to pass to the SUM function: the **SalesAmount** column. Begin typing *SalesAmount* until **Sales(SalesAmount)** is the only value left in the list.

The column name preceded by the table name is called the fully qualified name of the column. Fully qualified column names make your formulas easier to read.



1. Select **Sales[SalesAmount]** from the list, and then enter a closing parenthesis.

7. Subtract the other two columns inside the formula:

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

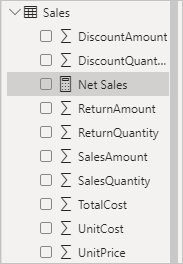
b. Enter another SUM function, and start typing *DiscountAmount* until you can choose the **Sales[DiscountAmount]** column as the argument. Add a closing parenthesis.

c. Type a space, a minus operator, a space, another SUM function with **Sales[ReturnAmount]** as the argument, and then a closing parenthesis.

Screenshot of the complete formula.

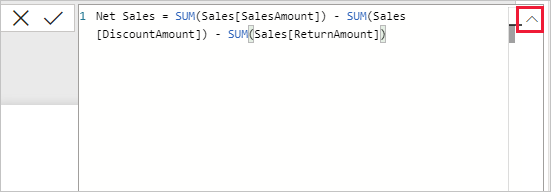
8. Press **Enter** or select **Commit** (checkmark icon) in the formula bar to complete and validate the formula.

The validated **Net Sales** measure is now ready to use in the **Sales** table in the **Fields** pane.

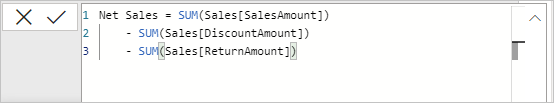


9. If you run out of room for entering a formula or want it on separate lines, select the down arrow on the right side of the formula bar to provide more space.

The down arrow turns into an up arrow and a large box appears.



10. Separate parts of your formula by pressing **Alt** + **Enter** for separate lines, or pressing **Tab** to add tab spacing.

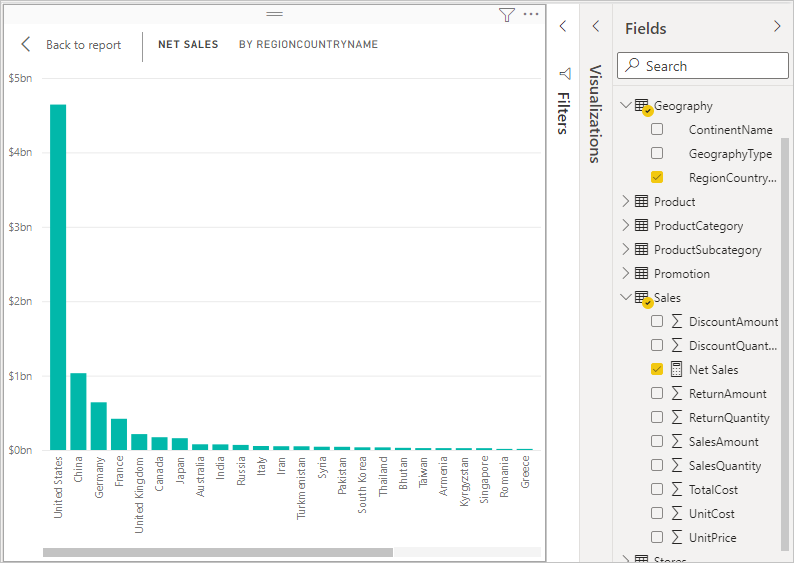


**Use your measure in the report**

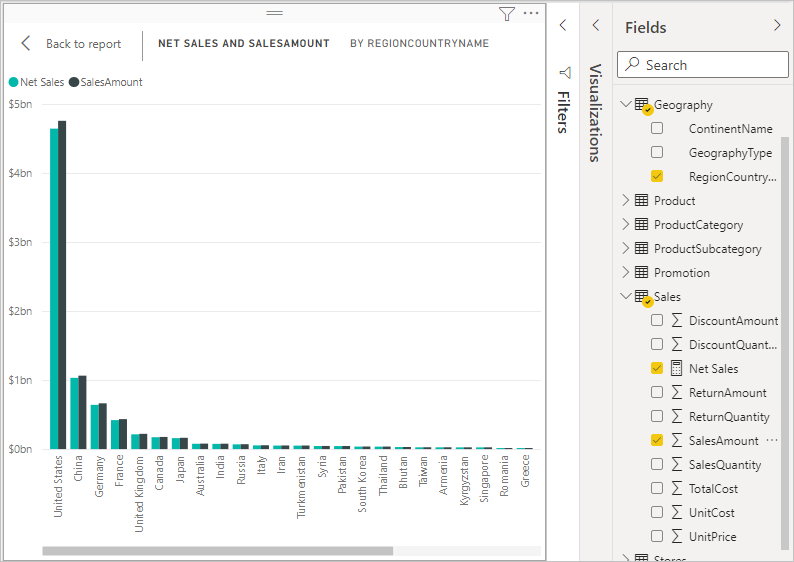
Add your new **Net Sales** measure to the report canvas, and calculate net sales for whatever other fields you add to the report.

To look at net sales by country/region:

1. Select the **Net Sales** measure from the **Sales** table, or drag it onto the report canvas.
2. Select the **RegionCountryName** field from the **Geography** table, or drag it onto the **Net Sales** chart.



1. To see the difference between net sales and total sales by country/region, select the **SalesAmount** field or drag it onto the chart.



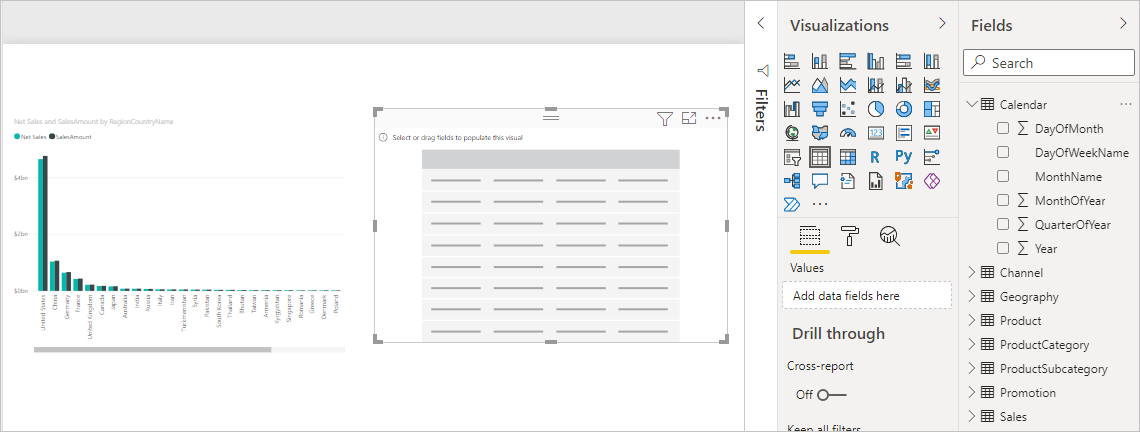
The chart now uses two measures: **SalesAmount**, which Power BI summed automatically, and the **Net Sales** measure, which you manually created. Each measure was calculated in the context of another field, **RegionCountryName**.

**Use your measure with a slicer**

Add a slicer to further filter net sales and sales amounts by calendar year:

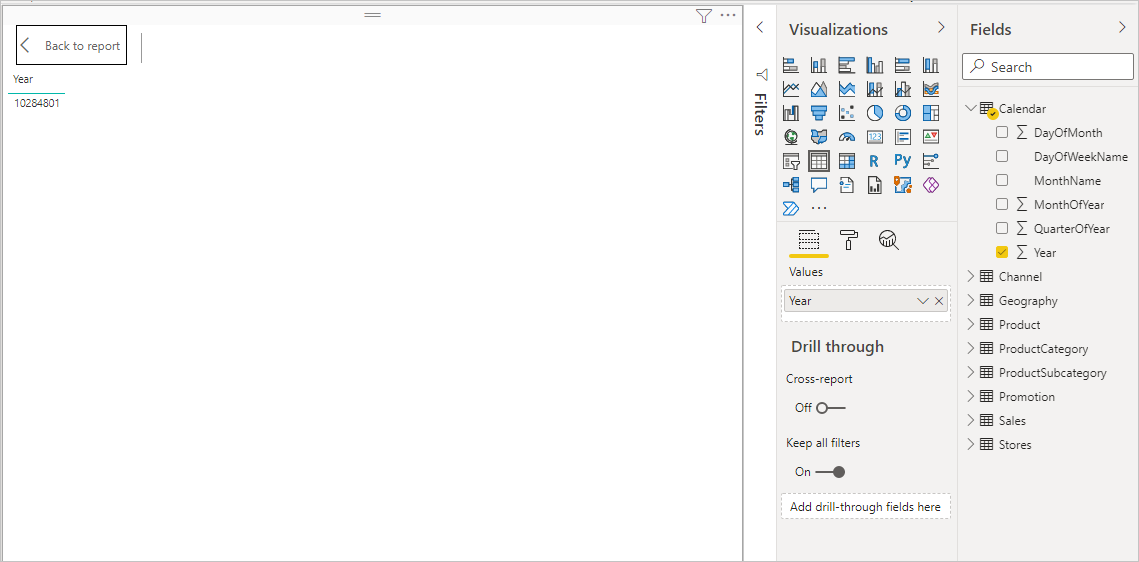
1. Select a blank area next to the chart. In the **Visualizations** pane, select the **Table** visualization.

This action creates a blank table visualization on the report canvas.

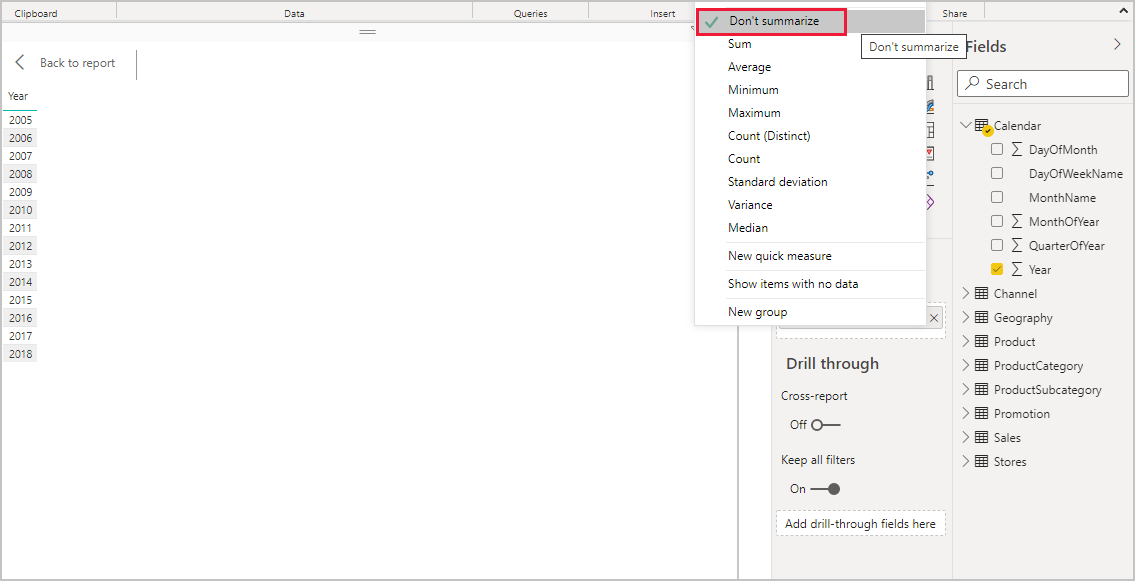


1. Drag the **Year** field from the **Calendar** table onto the new blank table visualization.

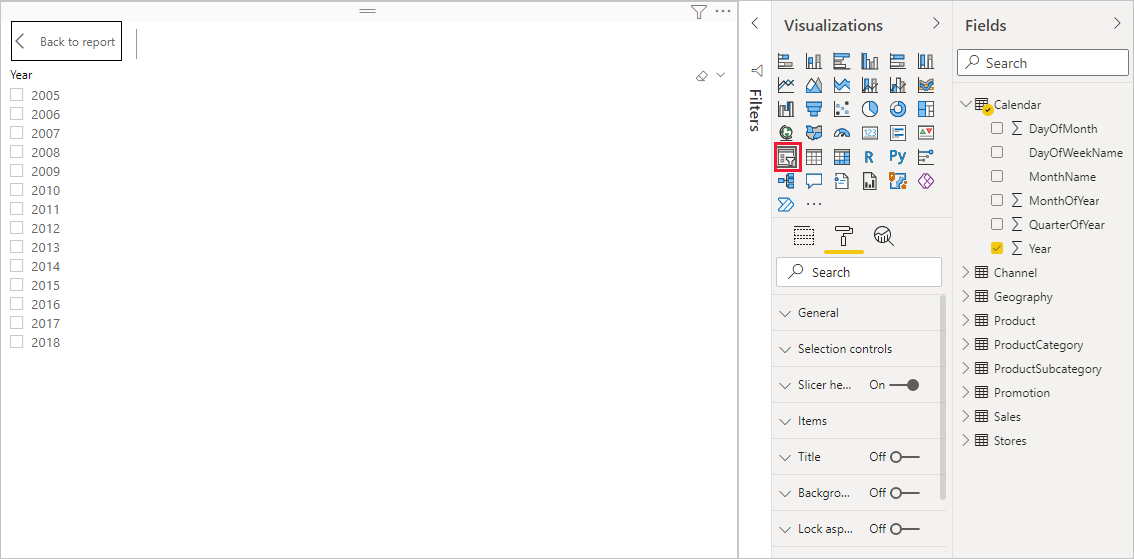
Because **Year** is a numeric field, Power BI Desktop sums up its values. This summation doesn’t work well as an aggregation; we'll address that in the next step.



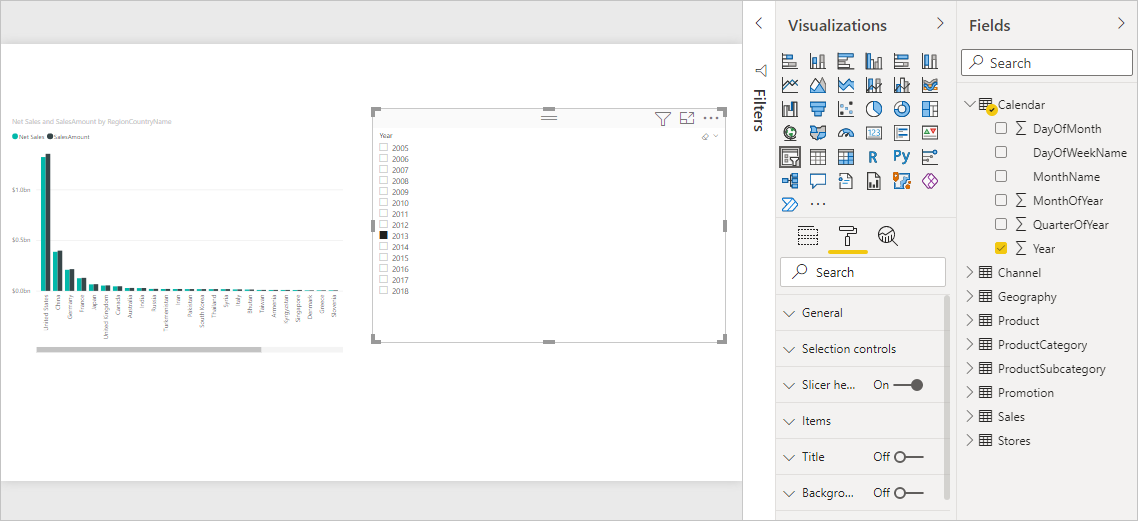
1. In the **Values** box in the **Visualizations** pane, select the down arrow next to **Year**, and then choose **Don't summarize** from the list. The table now lists individual years.



1. Select the **Slicer** icon in the **Visualizations** pane to convert the table to a slicer. If the visualization displays a slider instead of a list, choose **List** from the down arrow in the slider.



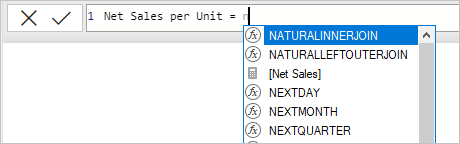
1. Select any value in the **Year** slicer to filter the **Net Sales and Sales Amount by RegionCountryName** chart accordingly. The **Net Sales** and **SalesAmount** measures recalculate and display results in the context of the selected **Year** field.



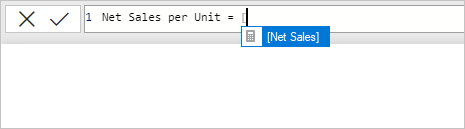
**Use your measure in another measure**

Suppose you want to find out which products have the highest net sales amount per unit sold. You'll need a measure that divides net sales by the quantity of units sold. Create a new measure that divides the result of your **Net Sales** measure by the sum of **Sales[SalesQuantity]**.

1. In the **Fields** pane, create a new measure named **Net Sales per Unit** in the **Sales** table.
2. In the formula bar, begin typing *Net Sales*. The suggestion list shows what you can add. Select **[Net Sales]**.



1. You can also reference measures by just typing an opening bracket (**[**). The suggestion list shows only measures to add to your formula.



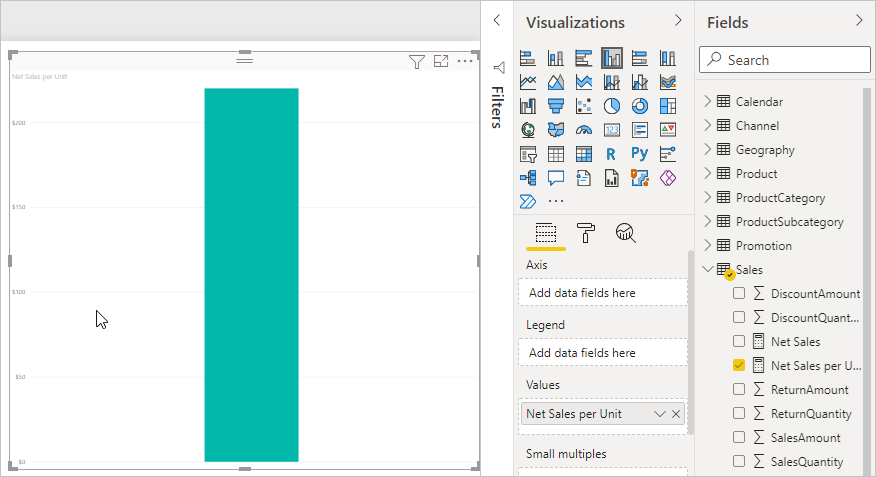
1. Enter a space, a divide operator (/), another space, a SUM function, and then type *Quantity*. The suggestion list shows all the columns with *Quantity* in the name. Select **Sales[SalesQuantity]**, type the closing parenthesis, and press **ENTER** or choose **Commit** (checkmark icon) to validate your formula.

The resulting formula should appear as:

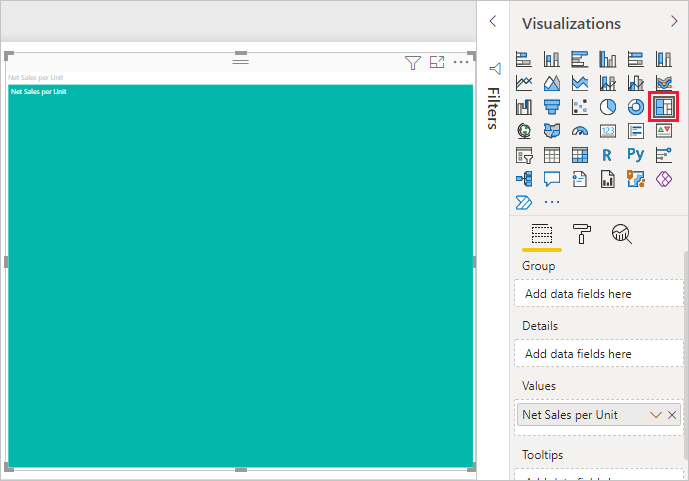
Net Sales per Unit = [Net Sales] / SUM(Sales[SalesQuantity])

1. Select the **Net Sales per Unit** measure from the **Sales** table, or drag it onto a blank area in the report canvas.

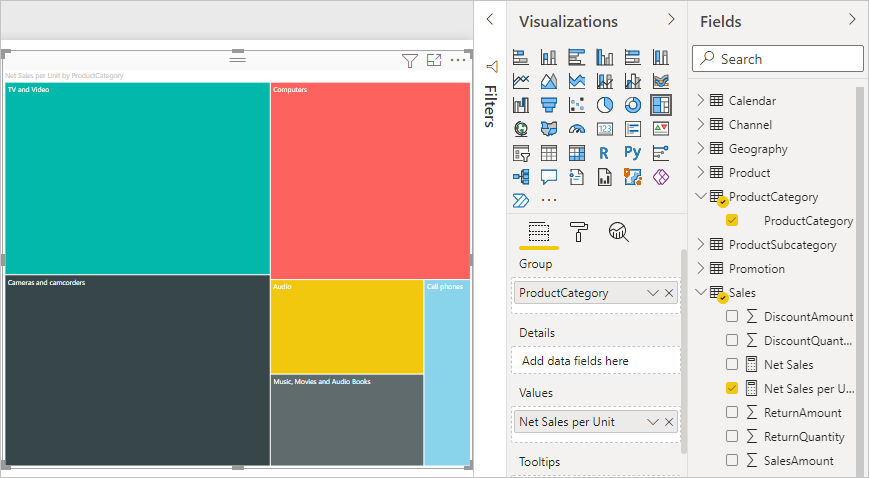
The chart shows the net sales amount per unit over all products sold. This chart isn't informative; we'll address it in the next step.



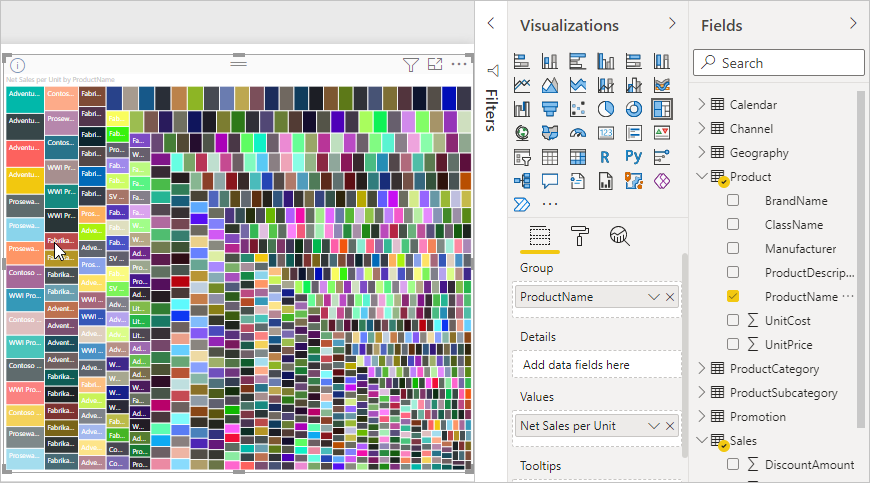
1. For a different look, change the chart visualization type to **Treemap**.



1. Select the **Product Category** field, or drag it onto the treemap or the **Group** field of the **Visualizations** pane. Now you have some good info!



1. Try removing the **ProductCategory** field and dragging the **ProductName** field onto the chart instead.



1. Create calculated tables, calculated columns, and simple measures using Data Analysis Expressions (DAX).

<https://microsoftlearning.github.io/PL-300-Microsoft-Power-BI-Data-Analyst/Instructions/Labs/04-create-dax-calculations-in-power-bi-desktop.html#:~:text=On%20the%20Table%20Tools%20contextual,already%20exist%20in%20the%20table>.

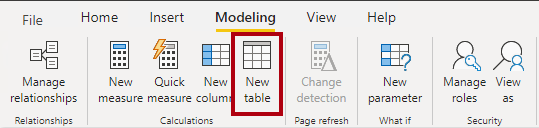
* Create calculated tables
* Create calculated columns
* Create measures

**Create the Salesperson calculated table**

In this task, you’ll create the **Salesperson** calculated table (direct relationship to **Sales**).

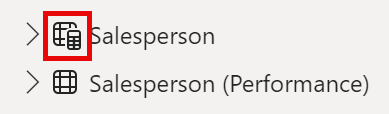
A calculated table is created by first entering the table name, followed by the equals symbol (=), followed by a DAX formula that returns a table. The table name can’t already exist in the data model.

The formula bar supports entering a valid DAX formula. It includes features like auto-complete, Intellisense and color-coding, enabling you to quickly and accurately enter the formula.

1. In Power BI Desktop, in Report view, on the **Modeling** ribbon, from inside the **Calculations** group, select **New Table**.
2. In the formula bar (which opens directly beneath the ribbon when creating or editing calculations), type **Salesperson =**, press **Shift+Enter**, type **‘Salesperson (Performance)’**, and then press **Enter**.

Picture 4

1. In the **Data** pane, notice that the table icon has an additional calculator in front of it (denoting a calculated table).



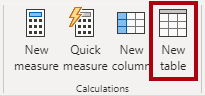
1. Switch to Model view, and notice that the **Salesperson** table is available (you may need to reset view to find table).
2. Create a relationship from the **Salesperson | EmployeeKey** column to the **Sales | EmployeeKey** column.
3. Right-click the inactive relationship between the **Salesperson (Performance)** and **Sales** tables, and then select **Delete**. When prompted to confirm the deletion, select **Yes**.
4. In the **Salesperson** table, multi-select the following columns, and then hide them (set the **Is Hidden** property to **Yes**):
   1. EmployeeID
   2. EmployeeKey
   3. UPN
5. In the model diagram, select the **Salesperson** table.
6. In the **Properties** pane, in the **Description** box, enter: **Salesperson related to Sales**
7. For the **Salesperson (Performance)** table, set the description to: **Salesperson related to region(s)**

The data model now provides two alternatives when analyzing salespeople. The **Salesperson** table allows analyzing sales made by a salesperson, while the **Salesperson (Performance)** table allows analyzing sales made in the sales region(s) assigned to the salesperson.

**Create the Date table**

In this task, you’ll create the Date table.

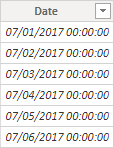
1. Switch to Table view. On the Home ribbon tab, from inside the Calculations group, select New Table.



1. In the formula bar, enter the following DAX:

**Date = CALENDARAUTO(6)**

1. Notice the column of date values which are formatted using US regional settings (that is, mm/dd/yyyy).



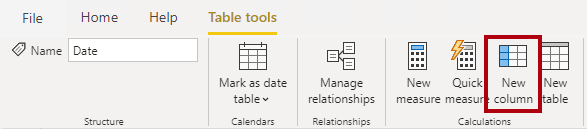
1. At the bottom-left corner, in the status bar, notice the table statistics, confirming that 1826 rows of data have been generated, which represents five full years’ data.

Picture 9

**Create calculated columns**

In this task, you’ll add more columns to enable filtering and grouping by different time periods. You’ll also create a calculated column to control the sort order of other columns.

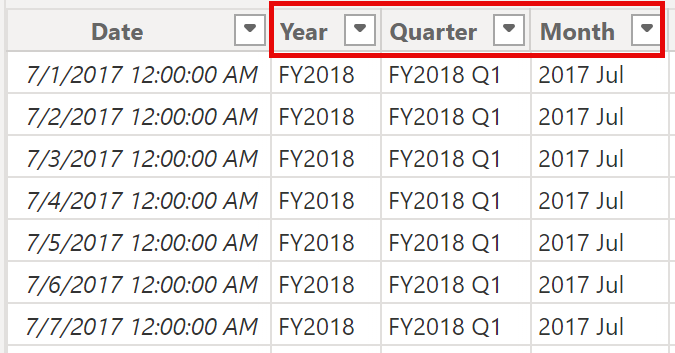
* 1. On the **Table Tools** contextual ribbon, from inside the **Calculations** group, select **New Column**.



* 1. In the formula bar, type the following (or copy from the snippets file), and then press **Enter.**

**Year ="FY" & YEAR('Date'[Date]) + IF(MONTH('Date'[Date]) > 6, 1)**

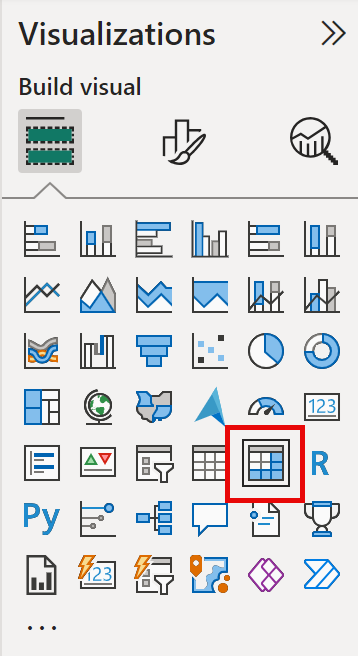
* 1. Use the snippets file definitions to create the following two calculated columns for the **Date** table:
* Quarter
* Month
  1. Verify the new columns have been added.



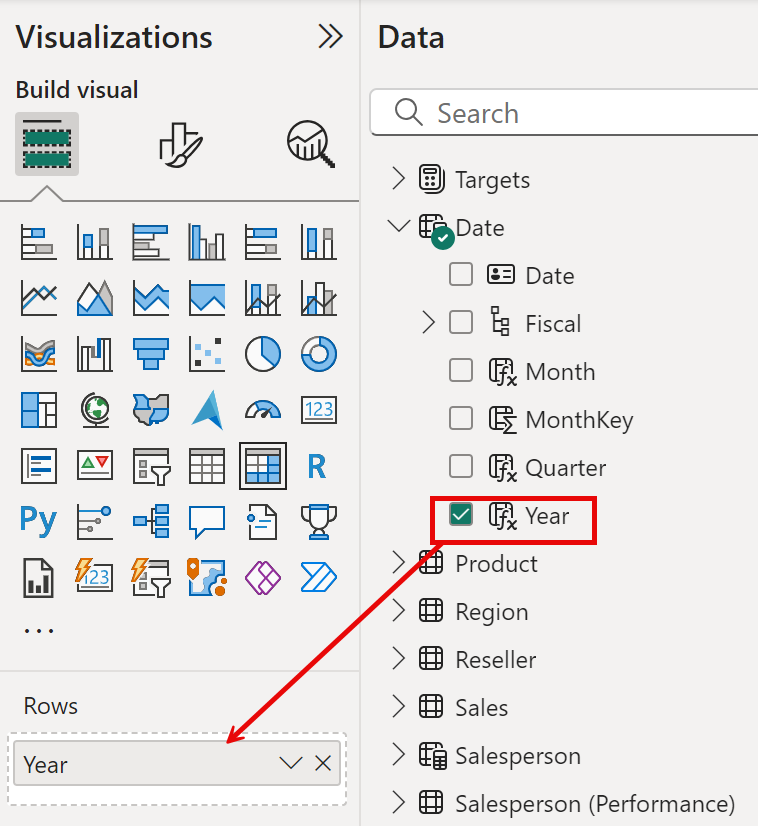
* 1. To validate the calculations, switch to Report view.
  2. To create a new report page, select the plus icon next to Page 1.



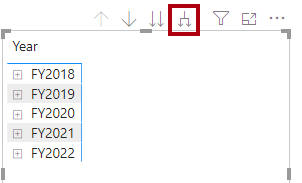
* 1. To add a matrix visual to the new report page, in the **Visualizations** pane, select the matrix visual type.



* 1. In the **Data** pane, from inside the **Date** table, drag the **Year** field into the **Rows** well/area.



* 1. Drag the **Month** field into the **Rows** well/area, directly beneath the **Year** field.
  2. At the top-right of the matrix visual (or bottom, depending on the location of the visual), select the forked-double arrow icon (which will expand all years down one level).



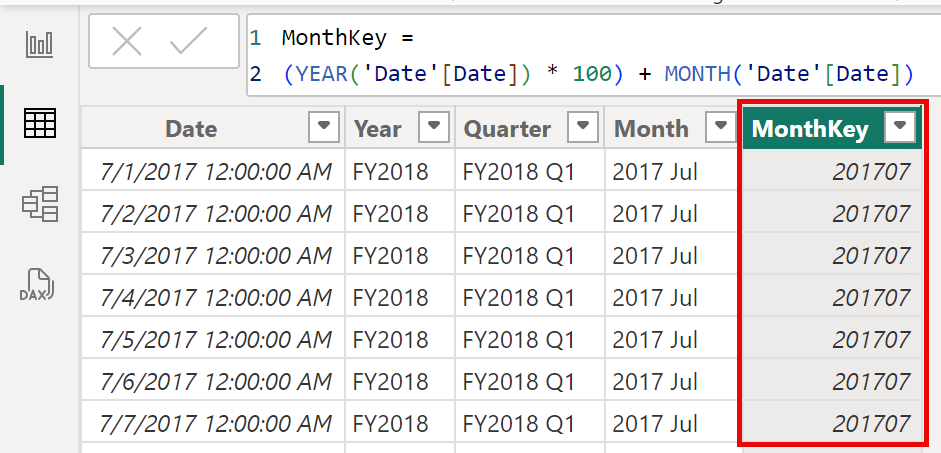
* 1. Notice that the years expand to months, and that the months are sorted alphabetically rather than chronologically.



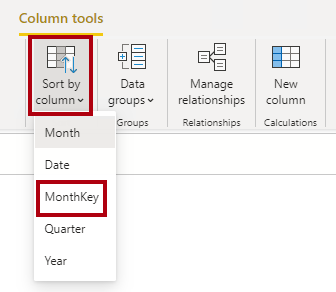
* 1. To customize the **Month** field sort order, switch to Table view.
  2. Add the **MonthKey** column to the **Date** table.

**MonthKey =(YEAR('Date'[Date]) \* 100) + MONTH('Date'[Date])**

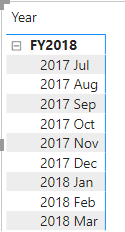
* 1. In Table view, verify that the new column contains numeric values (for example, 201707 for July 2017, etc.).



* 1. Switch back to Report view. In the **Data** pane and select **Month.**
  2. On the **Column Tools** contextual ribbon, from inside the **Sort** group, select **Sort by Column**, and then select **MonthKey**.



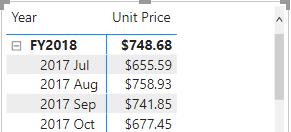
* 1. In the matrix visual, notice that the months are now chronologically sorted.



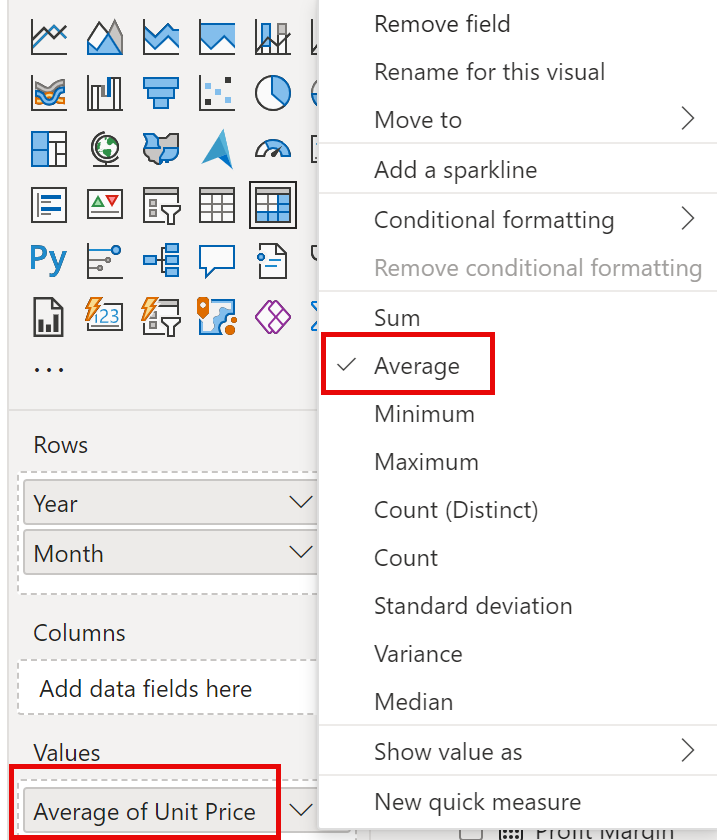
**Create simple measures**

In this task, you’ll create simple measures. Simple measures aggregate values in a single column or count rows of a table.

1. In Report view, on **Page 2**, in the **Data** pane, drag the **Sales | Unit Price** field into the matrix visual.



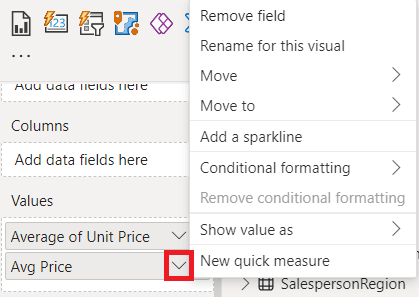
1. In the visual fields pane (located beneath the **Visualizations** pane), in the **Values** field well/area, notice that **Unit Price** is listed as the **Average of Unit Price**. Select the down-arrow for **Unit Price**, and then notice the available menu options.



1. To create a measure, in the **Data** pane, right-click the **Sales** table, and then select **New Measure**.
2. In the formula bar, add the following measure definition:

**Avg Price = AVERAGE(Sales[Unit Price])**

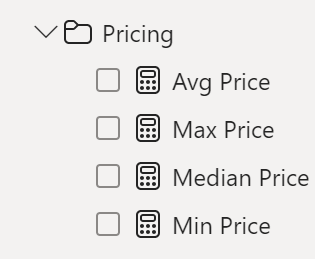
1. Add the **Avg Price** measure to the matrix visual, and notice that it produces the same result as the **Unit Price** column (but with different formatting).
2. In the **Values** well, open the context menu for the **Avg Price** field, and notice that it isn’t possible to change the aggregation technique.



1. Use the snippets file definitions to create the following five measures for the **Sales** table:

* Median Price
* Min Price
* Max Price
* Orders
* Order Lines

1. Switch to Model view, and then multi-select the four price measures: **Avg Price**, **Max Price**, **Median Price**, and **Min Price**.
2. For the multi-selection of measures, configure the following requirements:  
   * Set the format to two decimal places
   * Assign to a display folder named **Pricing**



1. Hide the **Unit Price** column.
2. Multi-select the **Order Lines** and **Orders** measures, and then configure the following requirements:

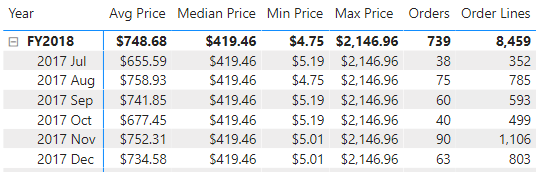
* Set the format use the thousands separator
* Assign to a display folder named **Counts**



1. In Report view, in the **Values** well/area of the matrix visual, for the **Unit Price** field, select **X** to remove it.
2. Increase the size of the matrix visual to fill the page width and height.
3. Add the following five measures to the matrix visual:

* Median Price
* Min Price
* Max Price
* Orders
* Order Lines

1. Verify that the results look sensible and are correctly formatted.



1. Create measures with DAX expressions involving filter context manipulation by using the CALCULATE () function & Time Intelligence functions.

<https://microsoftlearning.github.io/PL-300-Microsoft-Power-BI-Data-Analyst/Instructions/Labs/05-create-dax-calculations-in-power-bi-desktop-advanced.html#manipulate-filter-context>

**Syntax**

**CALCULATE(<expression>[, <filter1> [, <filter2> [, …]]])**

**Parameters:**

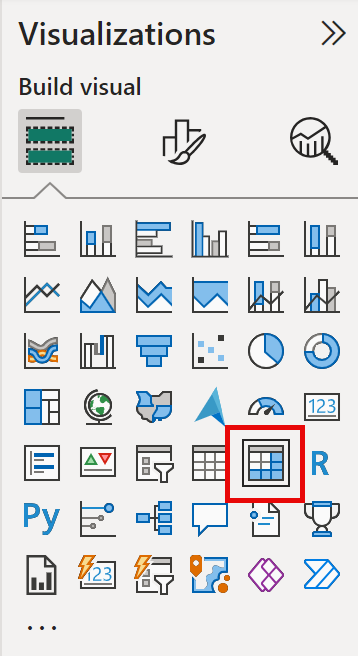
| Term | Definition |
| --- | --- |
| expression | The expression to be evaluated. |
| filter1, filter2,… | (Optional) Boolean expressions or table expressions that defines filters, or  filter modifier functions. |

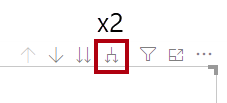
* **Use the CALCULATE() function to manipulate filter context**
* **Use Time Intelligence functions**

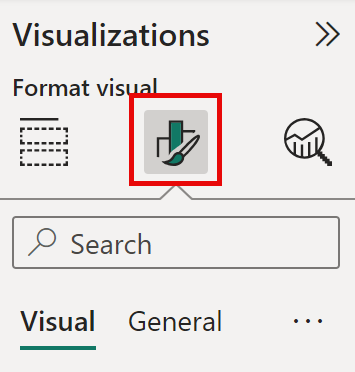
**Create a matrix visual**

In this task, you’ll create a matrix visual to support testing your new measures.

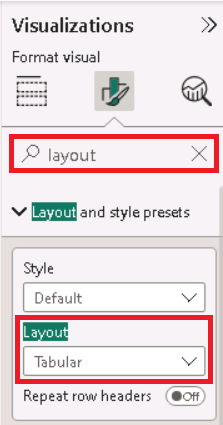
1. In Power BI Desktop, Report view, create a new report page.
2. On Page 3, add a matrix visual.



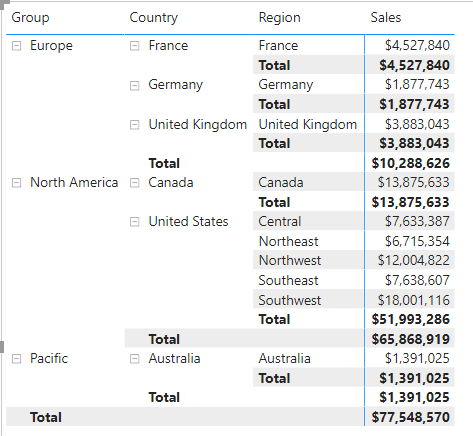
1. Resize the matrix visual to fill the entire page.
2. To configure the matrix visual fields, from the **Data** pane, drag the **Region | Regions** hierarchy, and drop it inside the visual.
3. Add also the **Sales | Sales** field to the Values well.
4. To expand the entire hierarchy, at the top-right of the matrix visual, select the forked-double arrow icon twice.
5. To format the visual, in the **Visualizations** pane, select the **Format** pane.



1. In the **Search** box, enter **Layout**.
2. Set the **Layout** property to **Tabular**.



1. Verify that the matrix visual now has 4 column headers.



**Manipulate filter context**

In this task, you’ll create several measures with DAX expressions that use the CALCULATE() function to manipulate filter context.

1. Add a measure to the **Sales** table, based on the following expression:

**Sales All Region =CALCULATE(SUM(Sales[Sales]), REMOVEFILTERS(Region))**

1. Add the **Sales All Region** measure to the matrix visual.



1. Notice that the **Sales All Region** measure computes the total of all region sales for each region, country (subtotal) and group (subtotal).

*The new measure is yet to deliver a useful result. When the sales for a group, country, or region is divided by this value it will produce a useful ratio known as “percent of grand total”.*

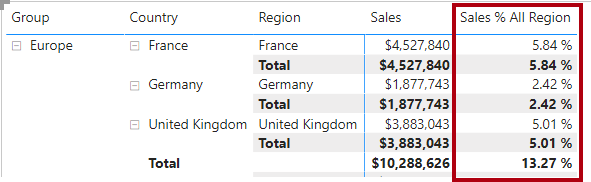
1. In the **Data** pane, ensure that the **Sales All Region** measure is selected (when selected, it will have a dark gray background), and then in the formula bar, replace the measure name and formula with the following formula:

Tip: To replace the existing formula, first copy the snippet. Then, select inside the formula bar and press **Ctrl+A** to select all text. Then, press **Ctrl+V** to paste the snippet to overwrite the selected text. Then press **Enter**.

**Formula:**

**Sales%AllRegion=DIVIDE(SUM(Sales[Sales]),CALCULATE(SUM(Sales[Sales]), REMOVEFILTERS(Region)))**

1. In the matrix visual, notice that the measure has been renamed and that a different value now appears for each group, country, and region.
2. Format the **Sales % All Region** measure as a percentage with two decimal places.
3. In the matrix visual, review the **Sales % All Region** measure values.



1. Add another measure to the **Sales** table, based on the following expression, and format as a percentage:

**Formula:**

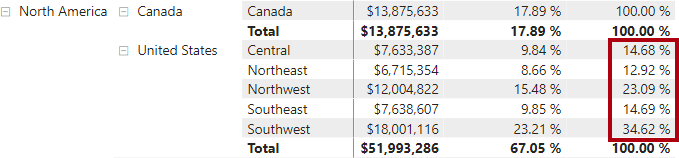
**Sales % Country = DIVIDE( SUM(Sales[Sales]), CALCULATE( SUM(Sales[Sales]),**

**REMOVEFILTERS(Region[Region]) ) )**

1. Notice that the **Sales % Country** measure formula differs slightly from the **Sales % All Region** measure formula.

*The difference is that the denominator modifies the filter context by removing filters on the****Region****column of the****Region****table, not all columns of the****Region****table. It means that any filters applied to the group or country columns are preserved. It will achieve a result that represents the sales as a percentage of country.*

1. Add the **Sales % Country** measure to the matrix visual.
2. Notice that only the United States’ regions produce a value that isn’t 100%.



1. To improve the readability of this measure in visual, overwrite the **Sales % Country** measure with this improved formula.

**Formula:**

**Sales % Country =**

**IF(**

**ISINSCOPE(Region[Region]),**

**DIVIDE(**

**SUM(Sales[Sales]),**

**CALCULATE(**

**SUM(Sales[Sales]),**

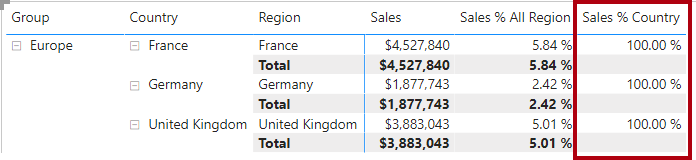
**REMOVEFILTERS(Region[Region])**

**)**

**)**

**)**

1. Notice that the **Sales % Country** measure now only returns a value when a region is in scope.



1. Add another measure to the **Sales** table, based on the following expression, and format as a percentage:

**Sales % Group =**

**DIVIDE(**

**SUM(Sales[Sales]),**

**CALCULATE(**

**SUM(Sales[Sales]),**

**REMOVEFILTERS(**

**Region[Region],**

**Region[Country]**

**)**

**)**

**)**

1. Add the **Sales % Group** measure to the matrix visual.
2. To improve the readability of this measure in visual, overwrite the **Sales % Group** measure with this improved formula.

**Sales % Group =**

**IF(**

**ISINSCOPE(Region[Region])**

**|| ISINSCOPE(Region[Country]),**

**DIVIDE(**

**SUM(Sales[Sales]),**

**CALCULATE(**

**SUM(Sales[Sales]),**

**REMOVEFILTERS(**

**Region[Region],**

**Region[Country]**

**)**

**)**

**)**

**)**

1. Notice that the **Sales % Group** measure now only returns a value when a region or country is in scope.
2. In Model view, place the three new measures into a display folder named **Ratios**.



1. Save the Power BI Desktop file.

8. Design a Report in Power BI using Sales Data.

9. Create the Sales Monitoring dashboard using charts & tiles.

10. Publish Data in Power BI Workspace.

**Part B: Mini project**

1.   Mini project – Collect Datasets & Create your own Dashboard.