Data Transformation with Power Query / Query Editor

Data Transformation

Data transformation is the process of converting data or information from one format to another, usually from the format of a source system into the required format of a new destination system.

Data Transformation – Why?

When existing Business model is hard to understand we use power query to Shape or Transform the data, and build a model that will be easily understandable for a Report User.

If existing Business model contains too many tables and many relationships between tables makes a reporting query very slow and not efficient. Here we use Power Query to Shape and Transform the data to build a star or snow flake schema by creating dimension tables and fact table, which is more comfortable for report development.

Transactional databases are not best option for reporting purpose because

- ✓ The model is hard to understand for a Report User.
- ✓ Too many tables and many relationships between tables makes a reporting query (that might use 20 of these tables at once) very slow and not efficient.
- ✓ Also we don't need all the transactional data to be loaded into Reporting Tools we just load whatever data we need for reports into our reporting tools.

Shape or Transform Data using Power Query

With Power BI Desktop or Query Editor or Power Query, you can connect different types of data sources, and then shape the data to meet your reporting needs.

In Power Query or Query Editor we will transform or shape the data using built-in GUI transformations in the ribbon or using M language code.

Benefits of Data Transformation

Data transformation ensures that data that enters your enterprise is usable and manageable.

It facilitates cost-efficient storage, ease of analysis for greater business intelligence, and operational efficiency.

On the flip side, storing data that has not been transformed wastes resources and creates the possibility of compliance risk because the data cannot be managed under the organization's data governance rules.

Overview of Power Query / Query Editor

- ✓ Power Query is a Data Extraction, Transformation and Loading Engine.
- ✓ The Engine comes with a Graphical Tool and a Formula Language (M Language).
- ✓ Power Query can connect to set of data sources and read data from them for data preparation.

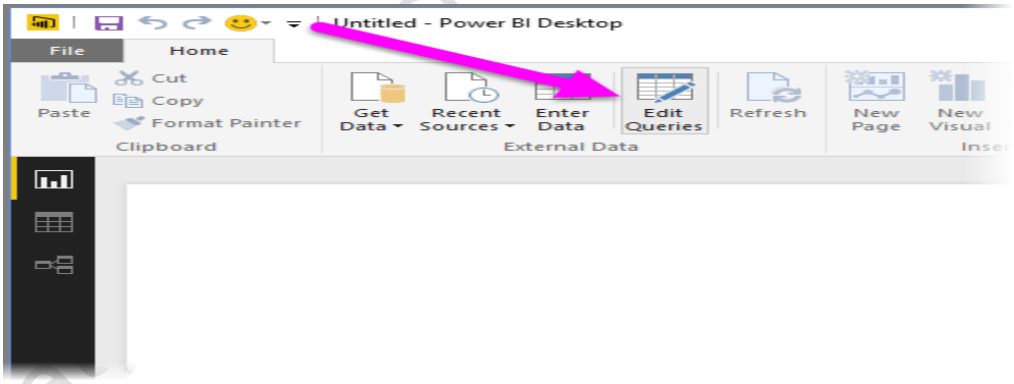
- ✓ Once connected to any data source, then Queries (one for each table, or entity) are listed and available for selection, viewing, and shaping.
- ✓ The Graphical Tool has list of Transformations that can be applied on a data set or Queries, and it also supports different data sources.
- ✓ Power Query graphical interface is so easy to work with that even business analyst or a power user can work with it, on the other hand Power Query M language is so powerful that can be used for complex real world challenges of data transformations.
- ✓ However, the Power Query formula language (M Language) is much more powerful than the GUI. Actually, there are some features in Power Query engine that not yet has been implemented through GUI, but they are available through M Language.
- ✓ Power Query can load the result set into Power Pivot for data modeling.
- ✓ M is the formula language behind the scenes of Power Query. Everything you do in the Query Editor will be translated to an M script. M contains full list of functions that you can use. So the powerful side of Power Query is actually M. M is a functional language and it has a simple structure.
- ✓ Every data preparation steps or applied steps on Queries will be recorded and displayed in Query Editor under Applied Steps Section.

Query Editor User Interface

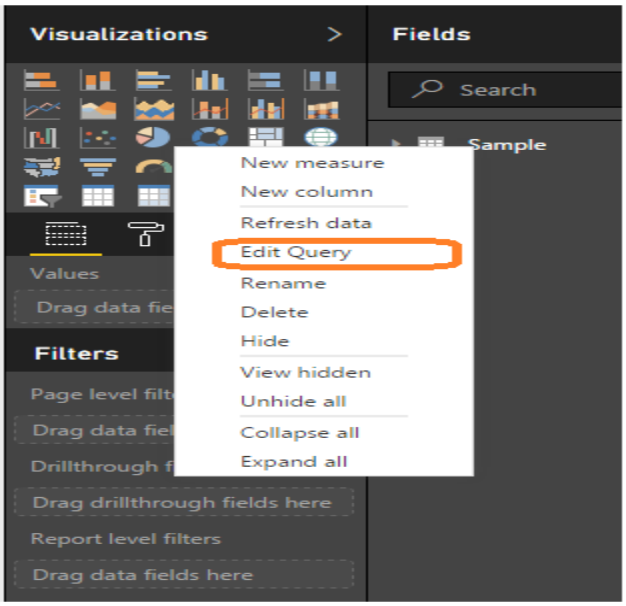
You can open Power Query Editor in three different ways

1. From Home Tab you can find Edit Queries.
2. In the Table Level Options you can find Edit Query.
3. While loading the table edit option that takes you to the Edit Queries.

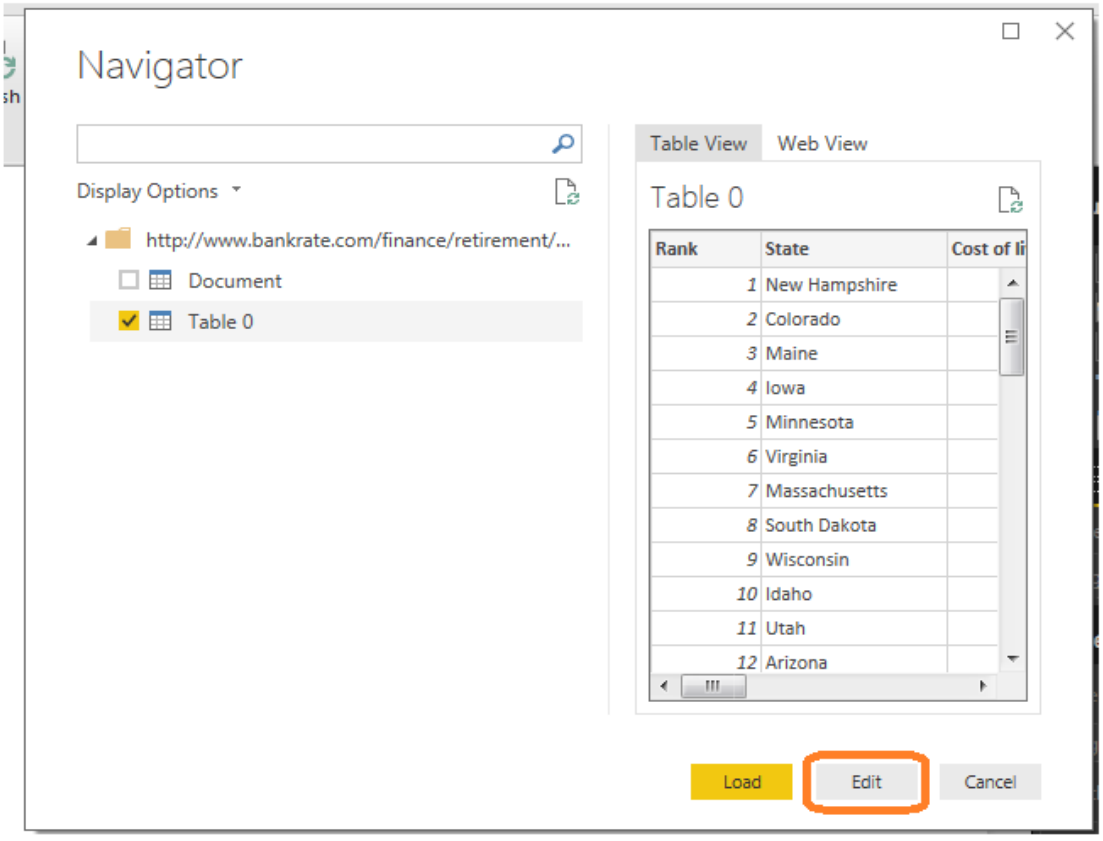
To get into Query Editor, select Edit Queries from the Home tab of Power BI Desktop.



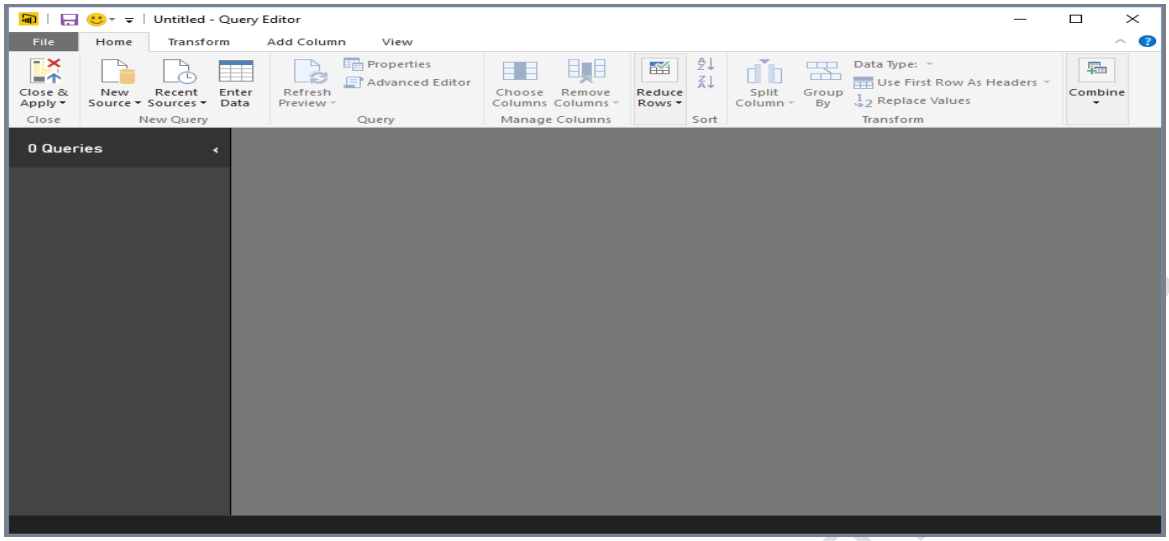
Other way to get or open Query Editor is, go to Table Level Options you can find Edit Query.



Third way is while loading the table "Edit" option that takes you to the Edit Queries.



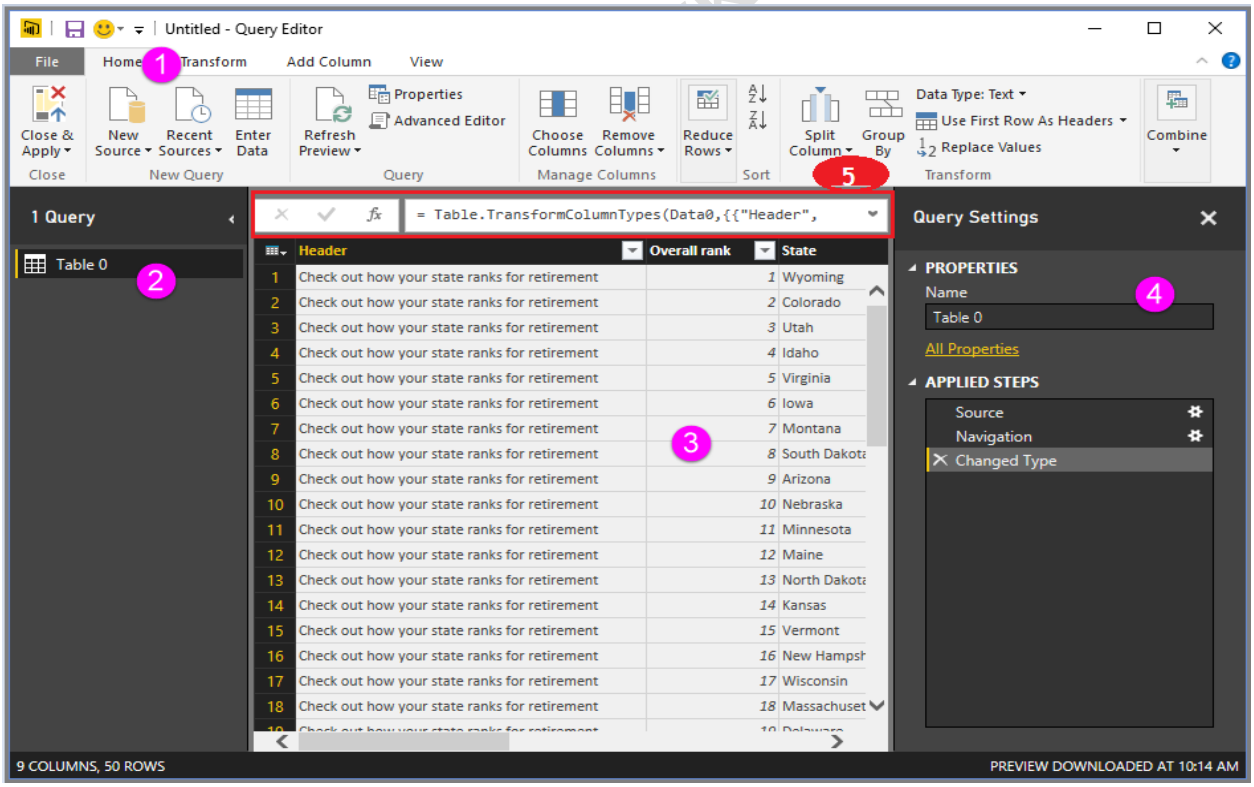
With no data connections, Query Editor appears as a blank pane, ready for data as shown below.



How to establish connection to the source?

Home Tab →New Source → Get Data window→Select the Source Type → Select the Source → Ok

Once Query Editor is loaded with data that is ready for you to shape, you see a handful of sections. Here's how Query Editor appears once a data connection is established.



1. In the ribbon, many buttons are now active to interact with the data in the query for data preparation.
2. In the left pane or queries pane, queries (one for each table, or entity) are listed and available for selection, viewing, and shaping.
3. In the center pane or Results Pane, data from the selected query is displayed and available for shaping.

4. The Query Settings window appears, listing the query's properties and applied steps.
5. The Formula bar is the place where you can see and edit the M code of the current transformation step.

We'll look at each of these four areas

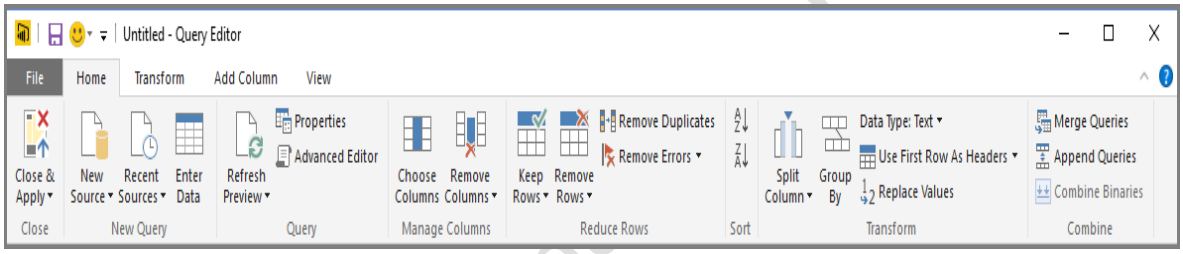
- ✓ The ribbon
- ✓ The queries pane
- ✓ The data view / Results Pane
- ✓ The Query Settings pane
- ✓ Formula Bar

The Query Ribbon

The ribbon in Query Editor consists of four tabs – Home, Transform, Add Column, and View.

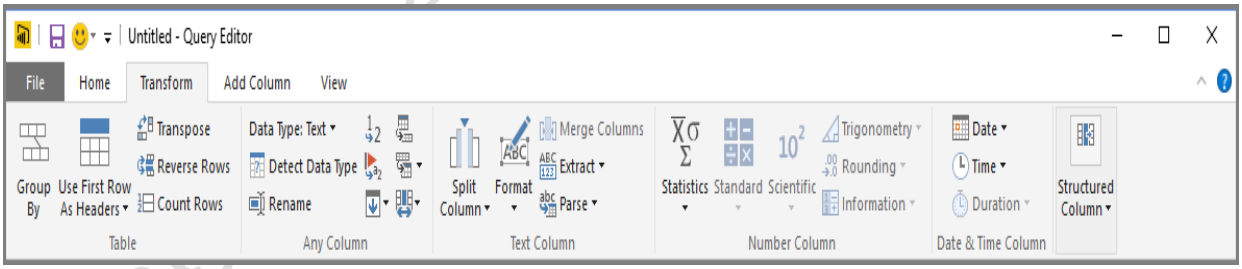
Home Tab

The Home tab contains the **common query tasks**, including the first step in any query, which is Get Data. The following image shows the Home ribbon.



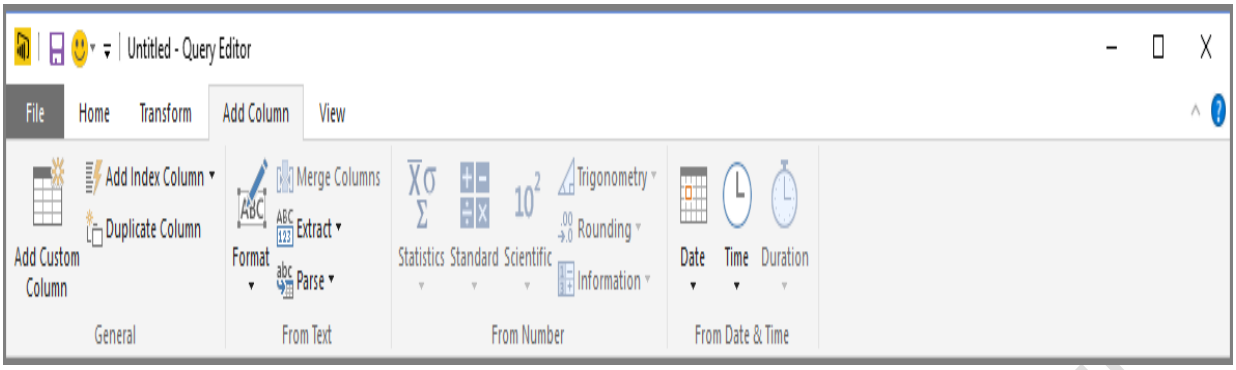
Transform Tab

The Transform tab provides access to common **data transformation tasks**, such as adding or removing columns, changing data types, splitting columns, and other data-driven tasks. The following image shows the Transform tab.



Add Column Tab

The Add Column tab provides additional tasks associated with **adding a column, formatting column data, and adding custom columns.** The following image shows the Add Column tab.



The Difference between the **Transform** and **Add Column** Tabs

The bulk of all transformations available in power query can be accessed through either the Transform tab or the Add Column tab.

You might think there is a lot of duplication between these two tabs. For example, both tabs contain a form Text section with a lot of the same commands. It's not really the case, there is a subtle difference!

When you use a command from the Add Column tab that is found in both tabs, it will create a new column with the transformed data and the original column will stay intact. Whereas using the equivalent command from the Transform tab will change the original column and no new column is created. This is a critical point to be aware of!

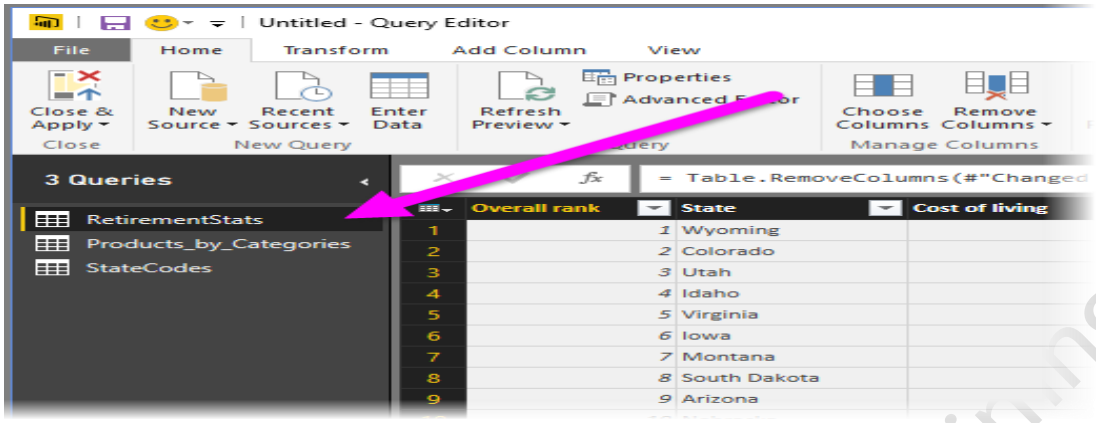
View Tab

The View tab on the ribbon is used to toggle whether certain panes or windows are displayed. It's also used to display the Advanced Editor. The following image shows the View tab.

It's useful to know that many of the tasks available from the ribbon are also available by right-clicking a column, or other data, in the center pane.

The Left Pane / Queries Pane

The left pane displays the number of active queries, as well as the name of the query. When you select a query from the left pane, its data is displayed in the center pane, where you can shape and transform the data to meet your needs. The following image shows the left pane with multiple queries.



The Center (Data) Pane / Results Pane

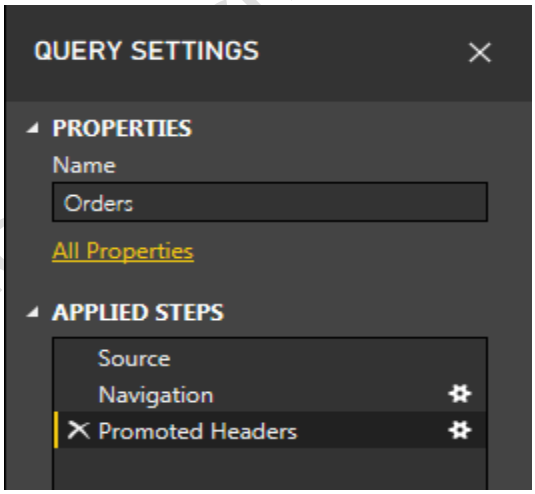
In the center pane, or Data pane, data from the selected query is displayed. This is where much of the work of the Query view is accomplished.

Notice that many of these right-click menu items are the same as buttons in the ribbon tabs.

When you select a right-click menu item (or a ribbon button), Query applies the step to the data, and saves it as part of the query itself. The steps are recorded in the Query Settings pane in sequential order, as described in the next section.

The Query Settings Pane

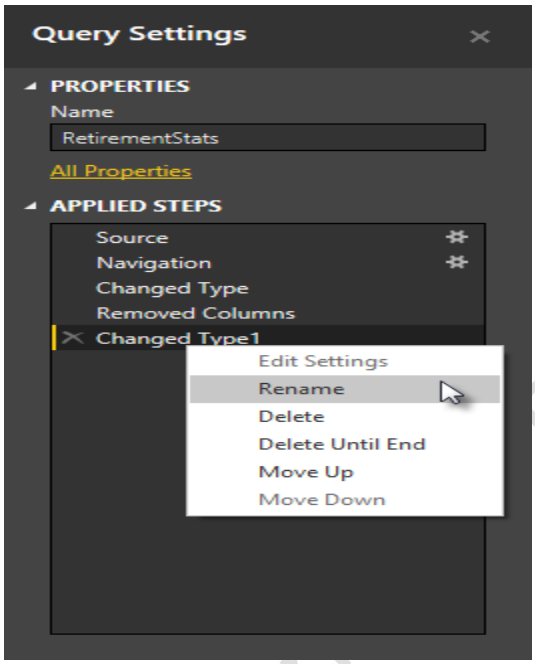
The Query Settings pane is where all steps associated with a query are displayed. For example, in the following image, the Applied Steps section of the Query Settings pane reflects the fact that we just changed the type of the Overall score column.



As additional shaping steps are applied to the query, they are captured in the **Applied Steps** section.

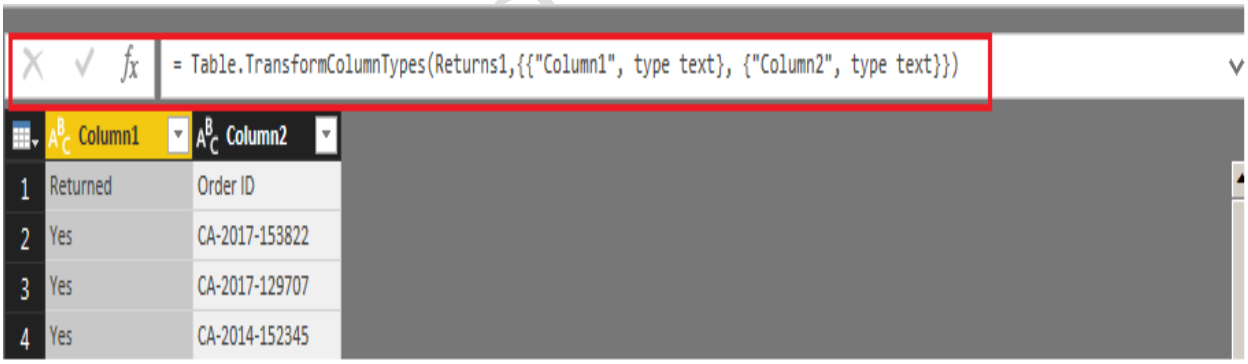
It's important to know that the underlying data is not changed rather Query Editor adjusts and shapes its view of the data, and any interaction with the underlying data occurs based on Query Editor's shaped and modified view of that data.

In the Query Settings pane, you can rename steps, delete steps, or reorder the steps as you see fit. To do so, right-click the step in the Applied Steps section and choose from the menu that appears. All query steps are carried out in the order they appear in the Applied Steps pane.



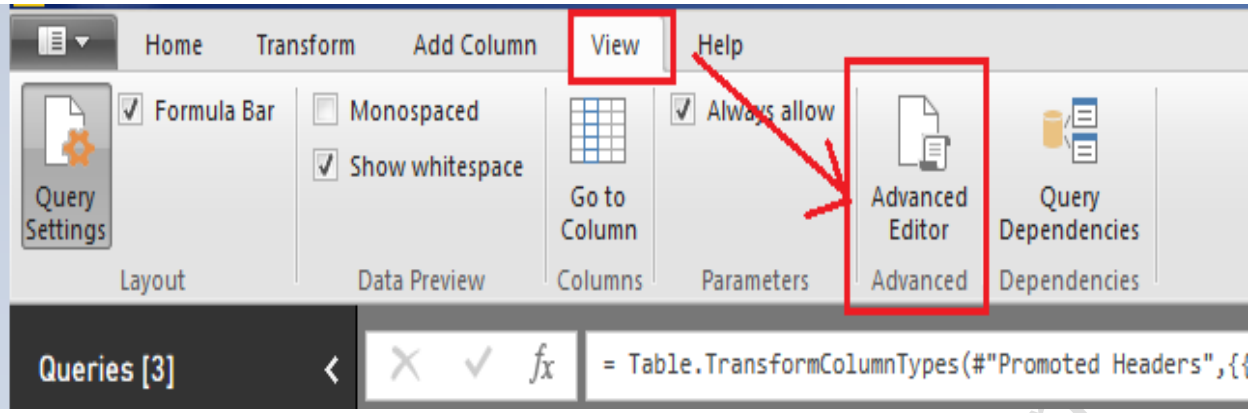
Formula Bar

This is where you can see and edit the M code of the current transformation step. Each transformation you make on your data is recorded and appears as a step in the applied steps area.

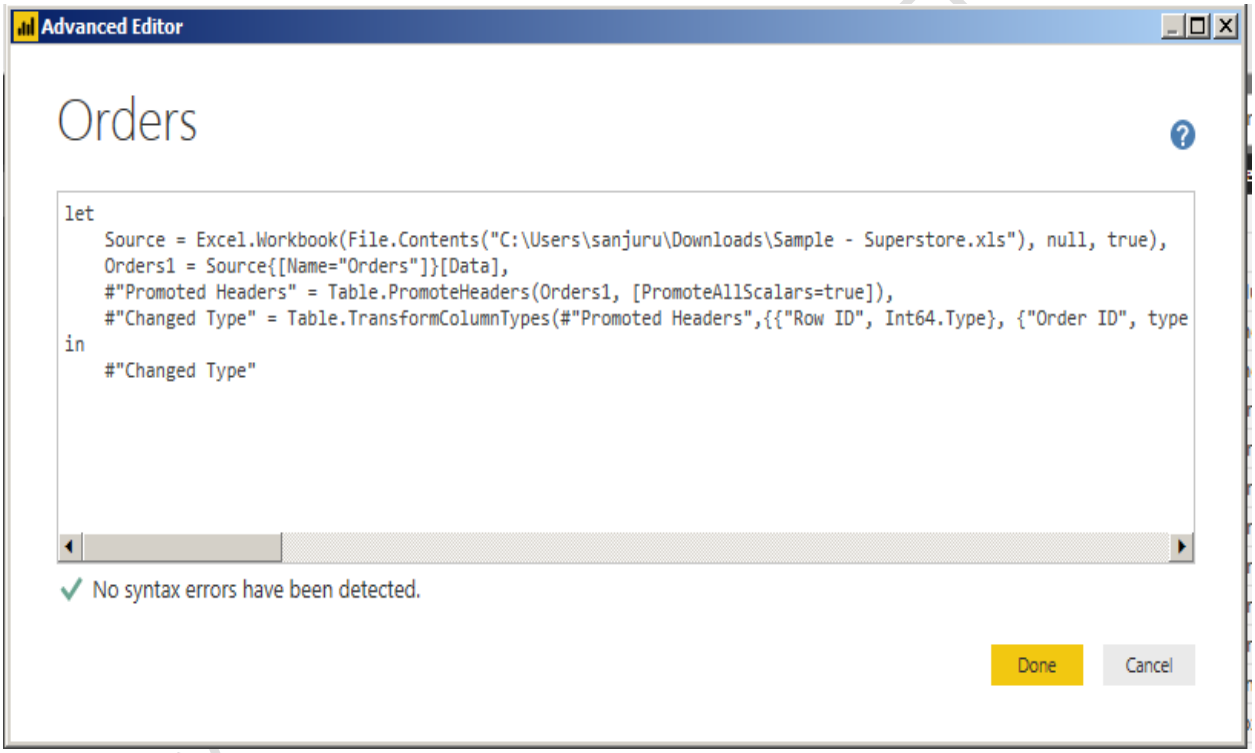


The Advanced Editor

If you want to see the code that Query Editor is creating with each step, or want to create your own shaping code, you can use the Advanced Editor. To launch the advanced editor, select View from the ribbon, then select Advanced Editor.

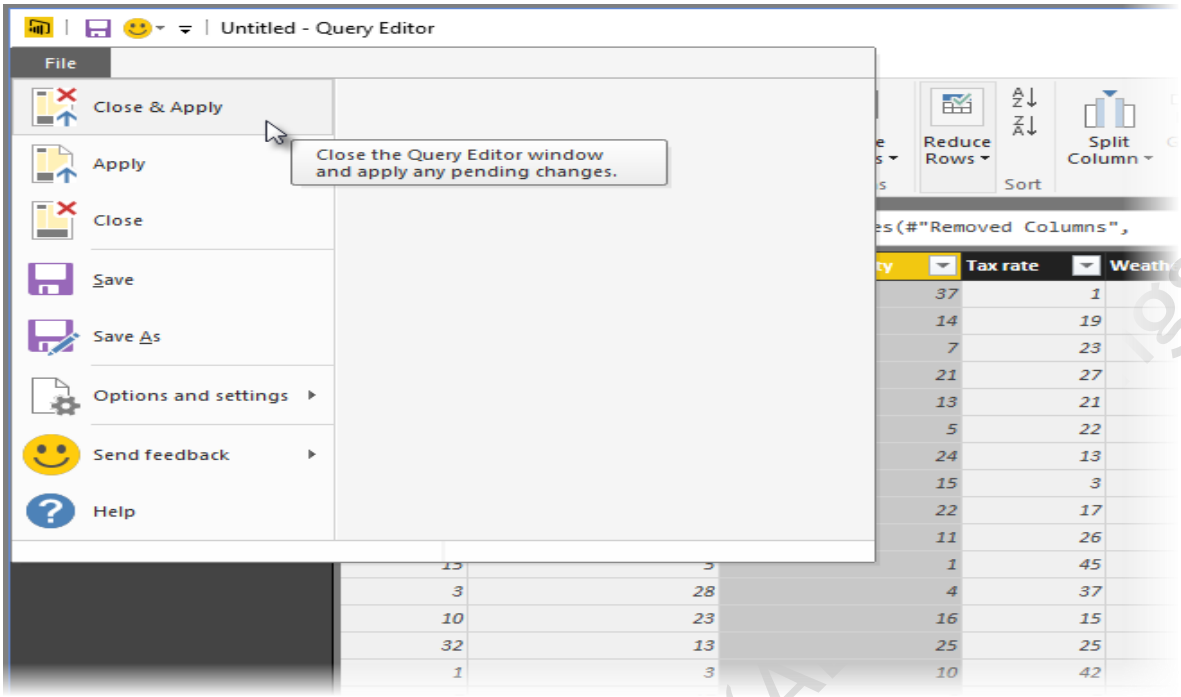


A window appears, showing the existing Query code. You can directly edit the code in the Advanced Editor window. To close the window, select the done or Cancel button.



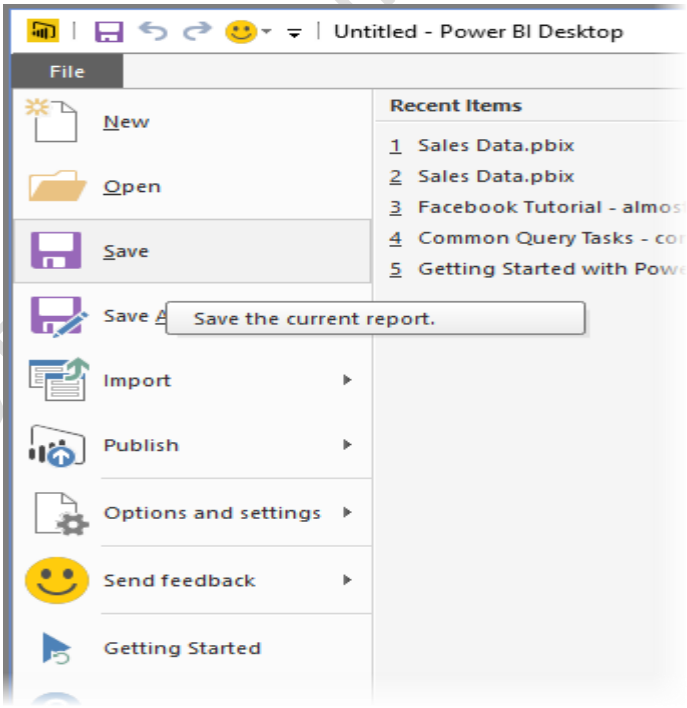
Saving Your Work

When your query is where you want it, you can have Query Editor apply the changes to the data model into Power BI Desktop, and close Query Editor. To do that, select Close & Apply from Query Editor's File menu as shown below.

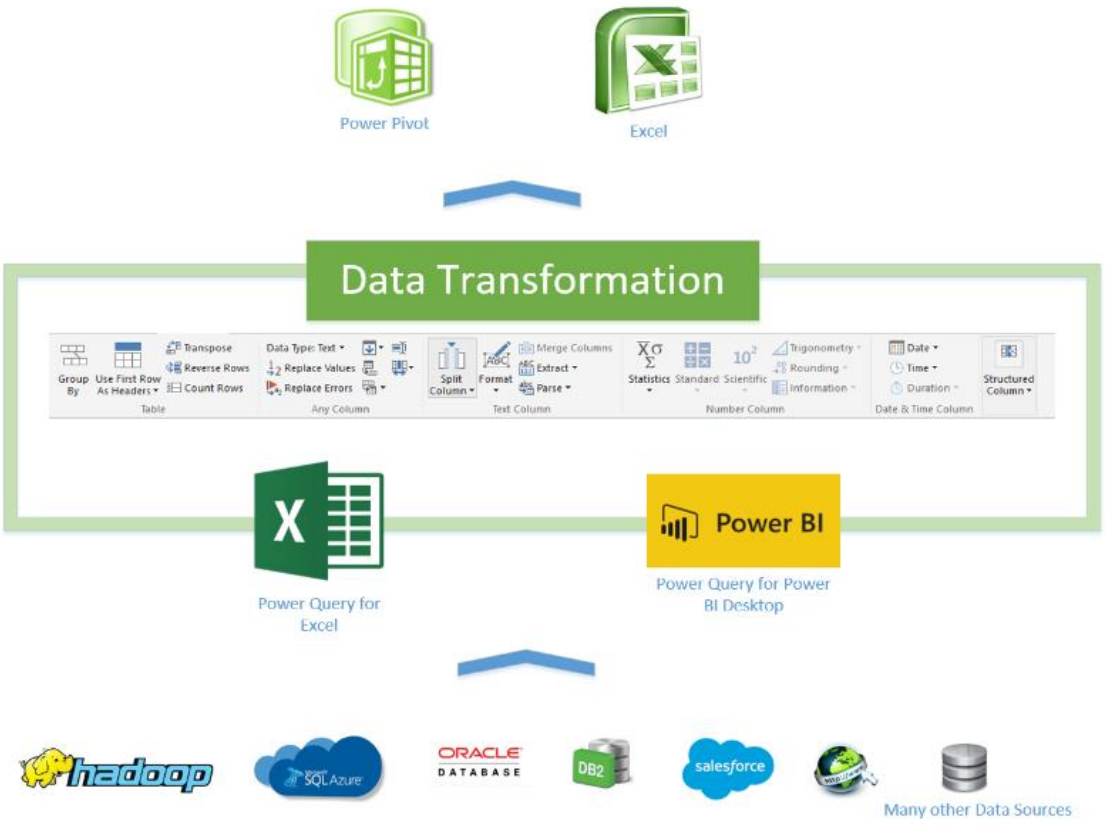


Once you have your query where you want it, or if you just want to make sure your work is saved, Power BI Desktop can save your work in the form of **". pbix"** file.

To save your work, select File > Save (or File > Save As), as shown in the following image.



In below diagram you can see a high-level diagram of Power Query conceptually



Data Type

Data Type represents the type of information stored in the memory location or Column. Each column should have only one Data Type.

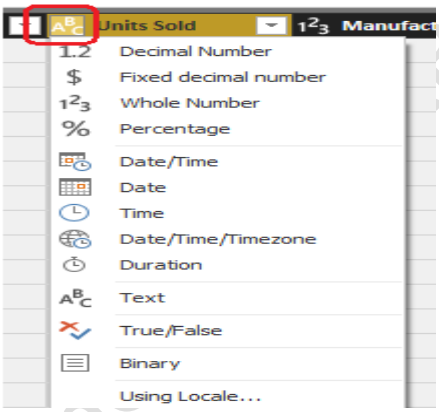
Change Data Type of a Column in Power BI

When you import or load a table from any data source, Power BI will automatically detect the data type of a column. However, there may be some situations where Power BI might get them wrong. For example, it may consider amounts, values, or even dates as the text. Now we will see how to Change Data Types of a Column in Power BI with example. Changing data type of the column is important as DAX functions have special data type requirements and also filtering options will change based on data type of the column.

In Query Editor or Power Query you can change the Data Type of a column in different ways.

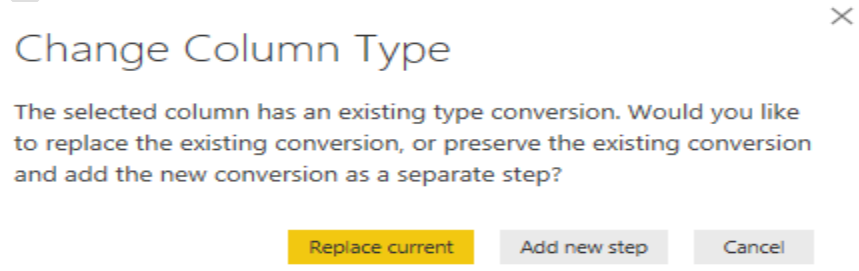
Approach 1

In the below image for Units Sold Column, Power BI identified it as string column. But actually we have Decimal Numbers as that column values.



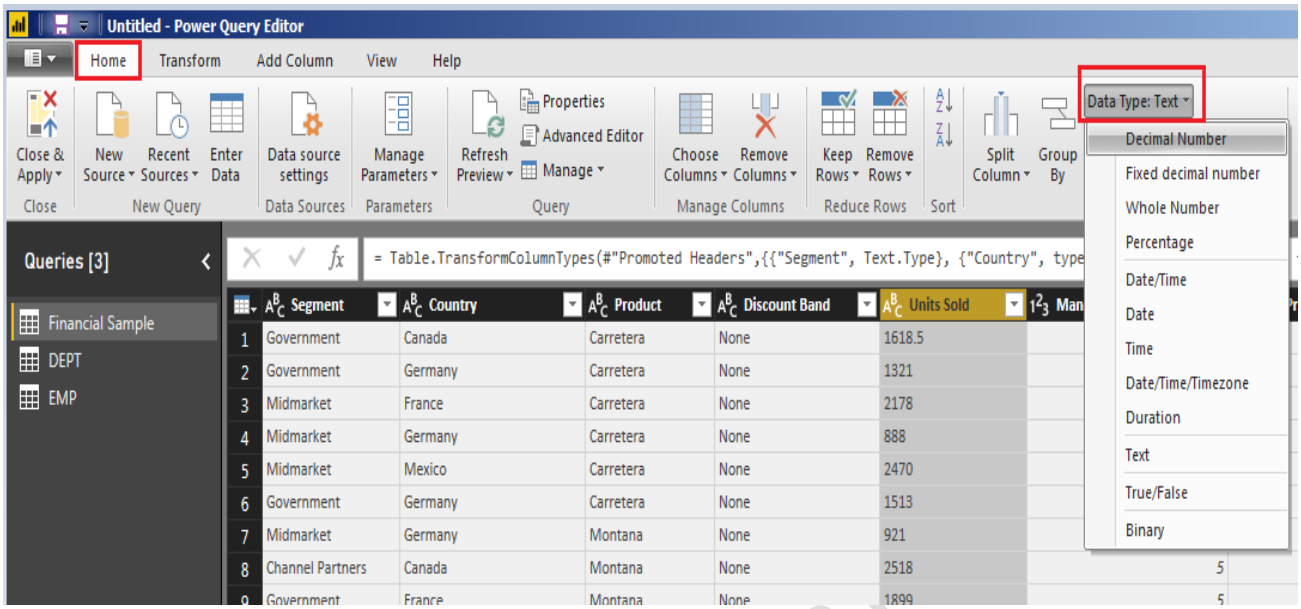
So to change the data type, select the Column for which you want to change the data type. Next, click on the left corner of the column header which is marked in Red Box. Then select the data type which is appropriate, here Decimal Number.

Changing data type of a column will open the following pop up window. You can Choose “Replace Current” to update current step or also you can choose “Add new step” to add a new transformation step to the Query.



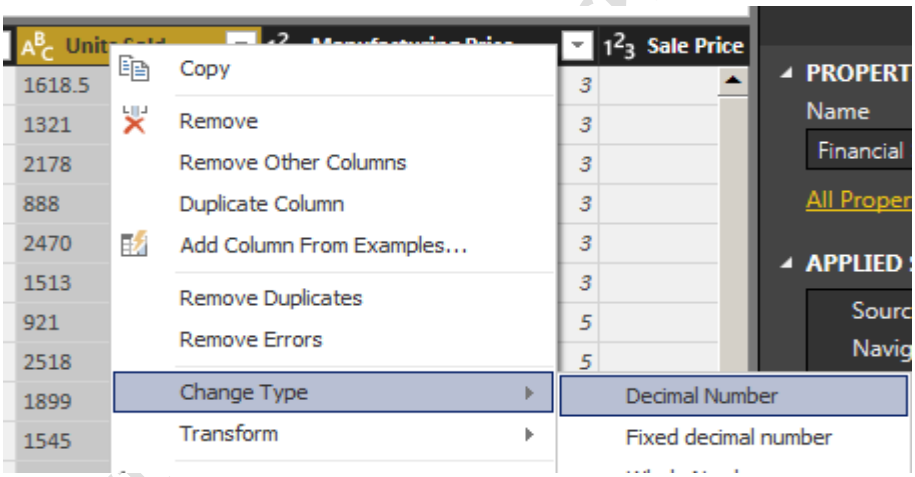
Approach 2

Select the Column name that you want to alter the data type, and click on the Data Type button under the Home tab in Power Query Ribbon.



Approach 3

Select the Column that you want to change the data type and right-click on it will open the context menu. Select the Change Type and then select the data type from the list. For now, we are selecting the Decimal Number.



Filtering select Rows in Power Query / Filters in Power Query

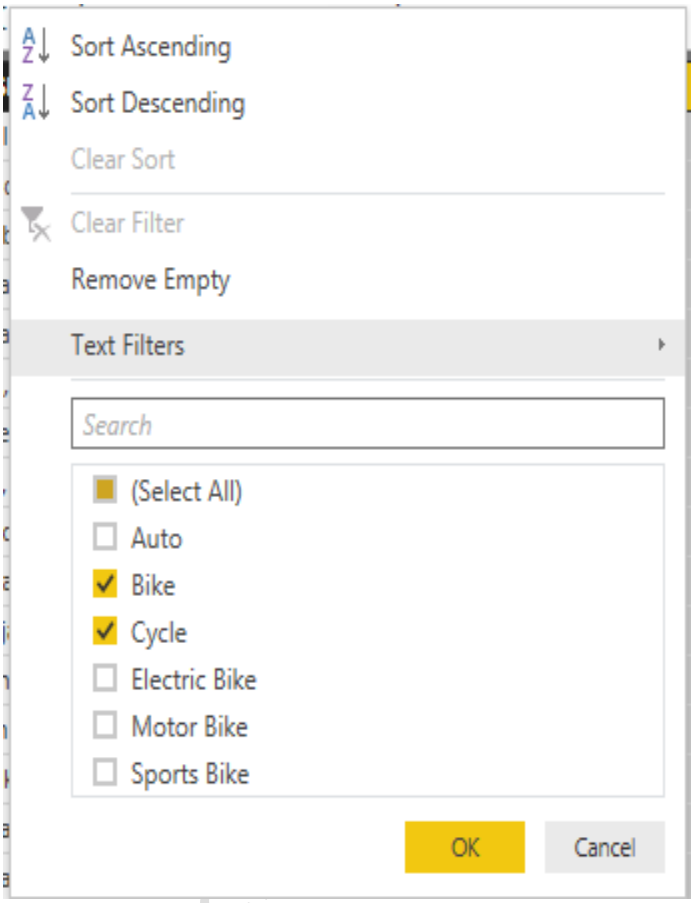
Data Type of column has impact on filtering options available. Filter options changes with respect to data types. Before going to filter rows check the data types of the columns.

"Text Filtering Options Are Case Sensitive".

Filter a column using an Auto Filter / Basic Filtering

- ✓ Select the column that you need to filter.
- ✓ Click the down arrow (▾).
- ✓ Uncheck the Select All box to deselect all Column Values.
- ✓ Select the column values you want to include in your table.

✓ Click OK.



"Search Bar is Case Insensitive".

Note:

Be careful if you are filtering the rows using Search Bar. Always look at the M code return by Power Query and cross check it is filtering as expected.

Basic Filtering is good only if you want to do equity filtering for values that exists in the current data set, however it won't work correctly if you want to check ranges, or contains or things that is not an exact equity filter. Advanced Filtering is the correct way of filtering in Power Query, and there are advanced filters for all types of data types; Numbers, Text, Date…

When you filter a column, only the top 1,000 distinct values in the column will load into the filter list. If there are 1,000 or more values in the column in Query Editor that you are filtering, a message will appears indicating that the list of values in the filter list may be incomplete, and the Load more link is shown. Click the Load more link to load another 1,000 distinct values.

If exactly 1,000 distinct values are found again, the list is displayed with a message stating that the list could still be incomplete.

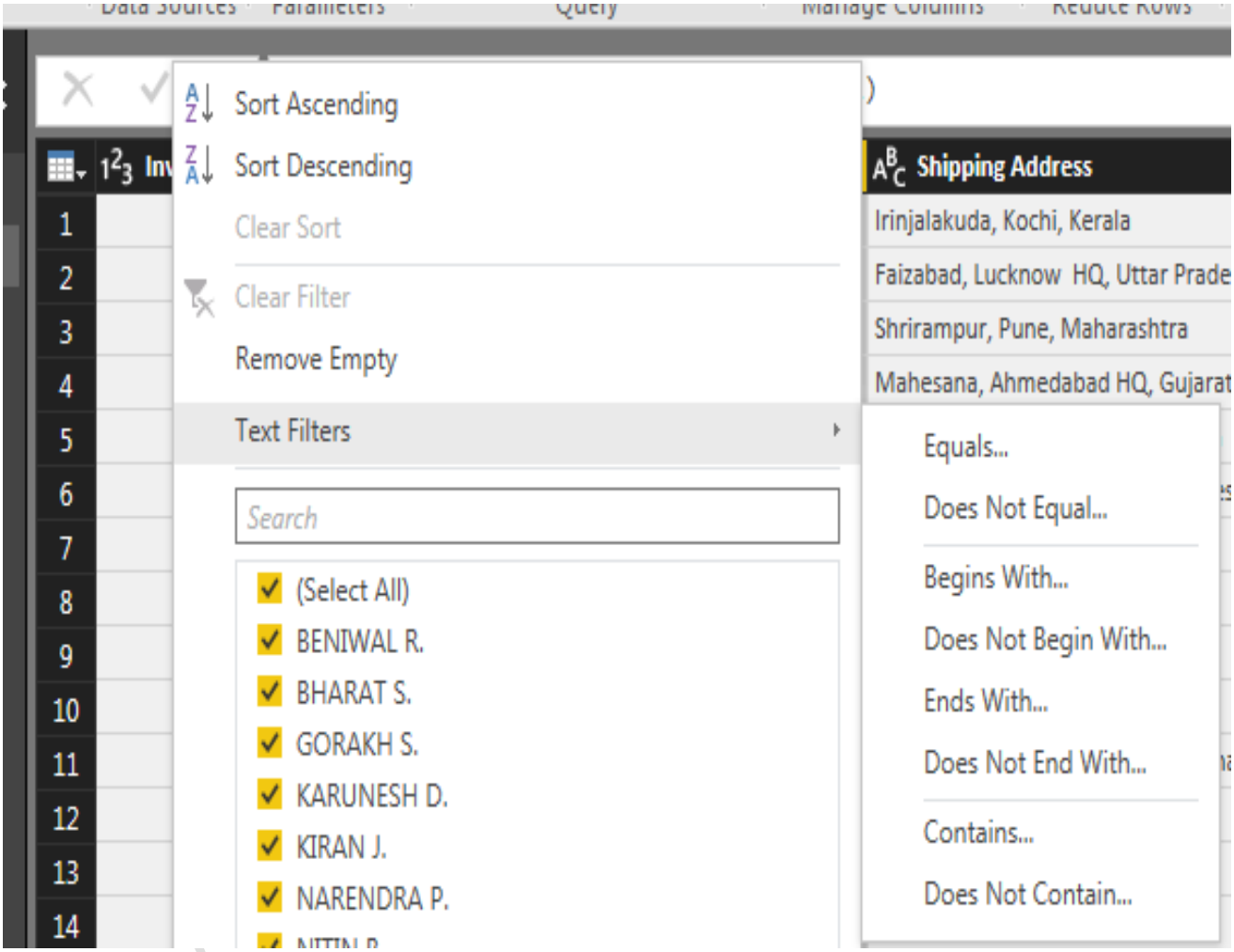
If less than 1,000 distinct values are found, the full list of values is shown.

Filter a Column using Text Filters

In addition to the "Auto Filters" or Basic Filtering, you can filter a Text values using the Text Filters context menu.

Click the down arrow (▾) of the column containing a Text values you want to filter on.

Click Text Filters and select the filter option required from Context Menu.

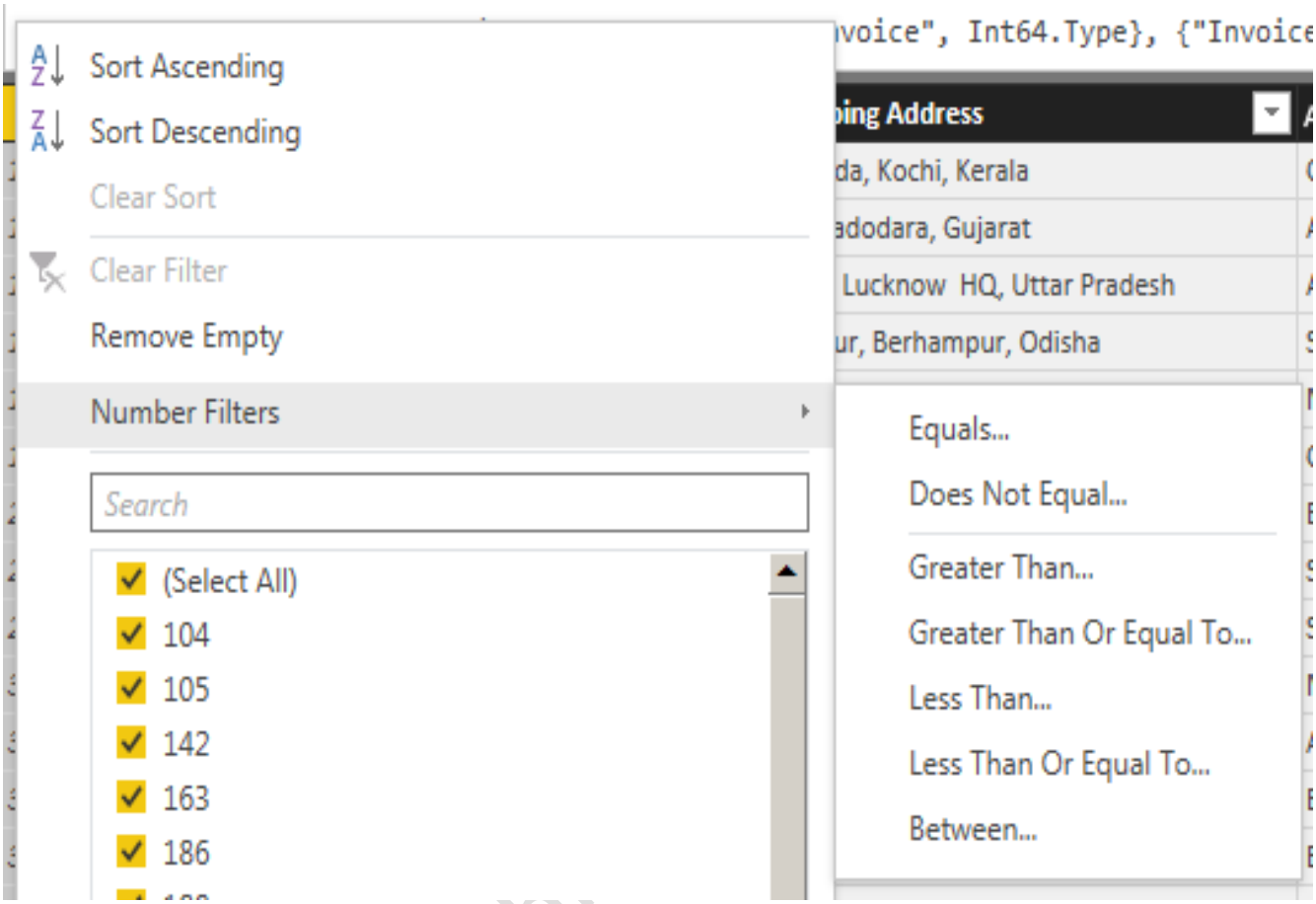


Filter a Column using Number Filters

In addition to the "Auto Filters", you can filter Number values using the Number Filters Context Menu.

To filter a column using Number Filters, Click the down arrow (▾) of the column containing a Number values you want to filter on.

Click Number Filters, and select the filter option required from Context Menu.

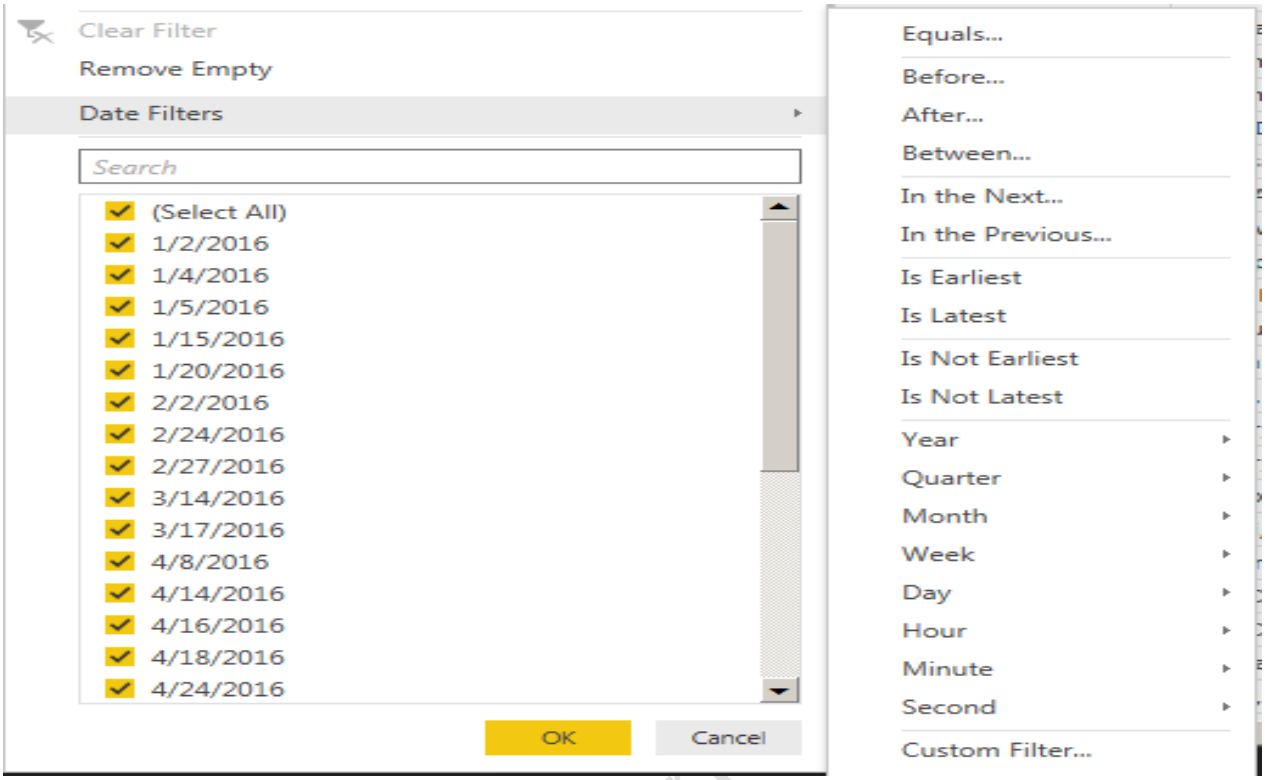


Filter a Column using Date Filters

In addition to the "Auto Filters", you can filter Date values using the Date Filters Context Menu.

To filter a column values using Date Filters, Click the down arrow (▾) of the column containing Date values you want to filter on.

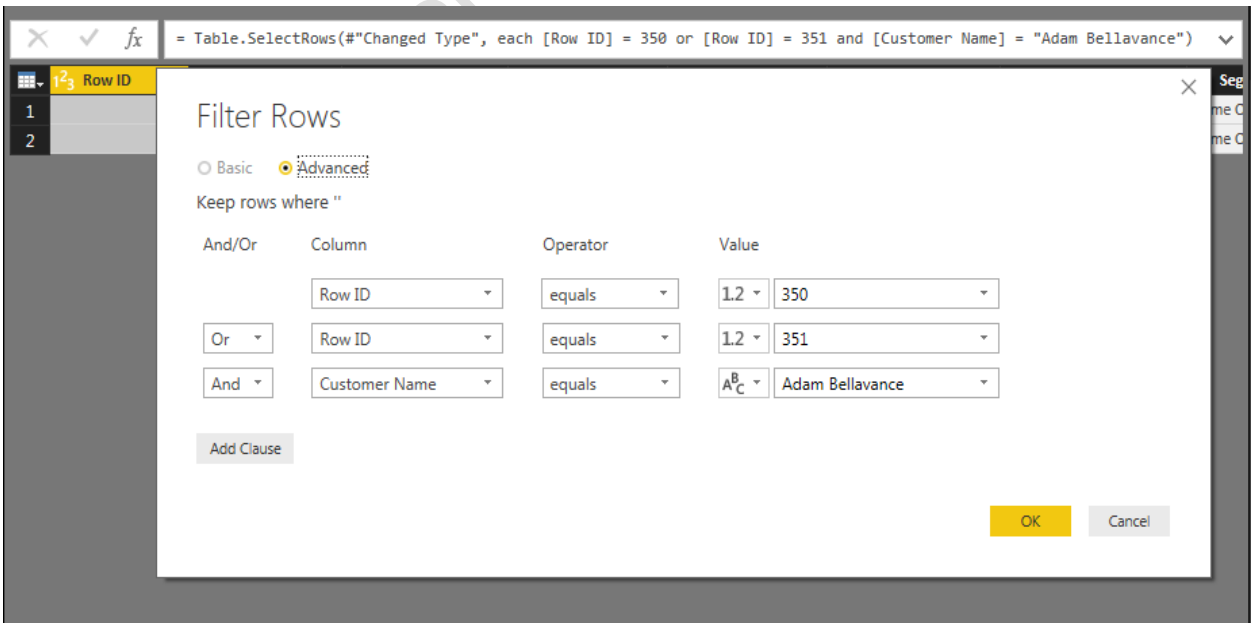
Click Date Filters, and select the filter option required from Context Menu.



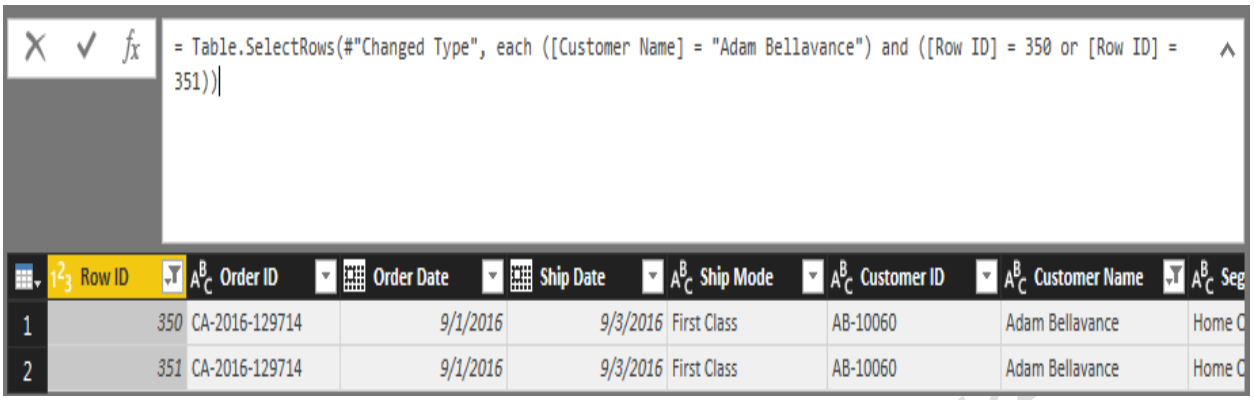
Filter Multiple Columns

To filter multiple columns, select an additional column, and repeat one of the column filter steps. **AND** Operation will be performed between the columns if you apply filters on multiple columns individually.

Other way is by Using **Advanced** option in Filter Rows you can apply filters on multiple columns at a time. Here you can select **And / Or** operation between columns.



In below image we applied filters on Row ID and Customer Name Column and you can see M Language Code.



	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Seg
1	350	CA-2016-129714	9/1/2016	9/3/2016	First Class	AB-10060	Adam Bellavance	Home C
2	351	CA-2016-129714	9/1/2016	9/3/2016	First Class	AB-10060	Adam Bellavance	Home C

Inbuilt Column Transformations

- ✓ Remove Columns / Remove Other Columns
- ✓ Name / Rename a Column
- ✓ Reorder Columns or Sort Columns
- ✓ Add Column / Custom Column
- ✓ Split Columns
- ✓ Merge Columns
- ✓ Pivot, Unpivot Columns
- ✓ Transpose Columns

Remove Columns / Remove Other Columns

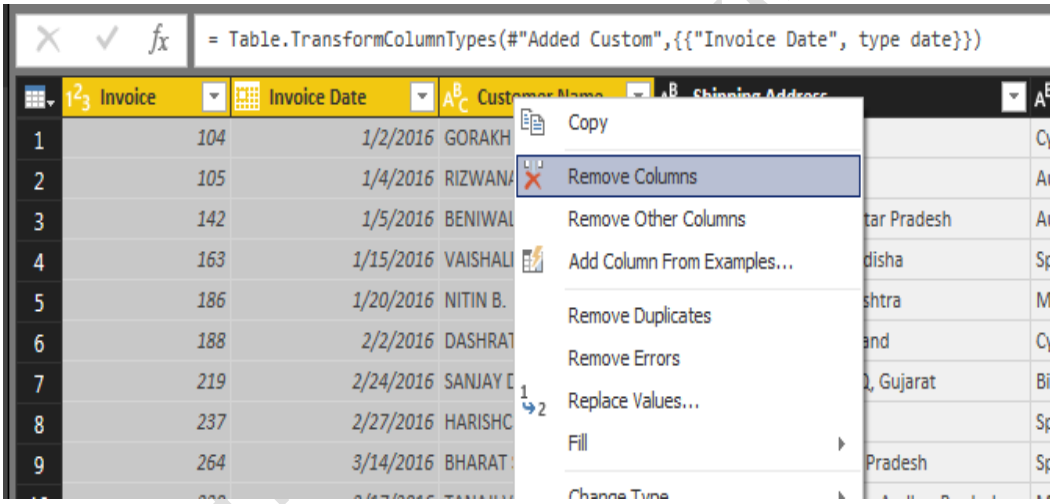
If you want to remove unwanted columns which are not necessary in your data model for the data source in your query, you can use Remove Columns / Remove Other Columns option as shown in the below image.

If you want remove selected columns

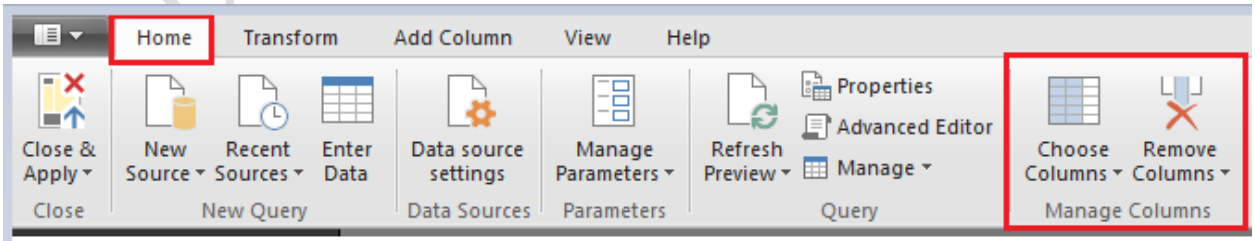
Select the columns you want to remove → Right Click → Remove Columns

If you want to remove all other columns other than selected

Select the columns you want to keep → Right Click → Remove Other Columns



You can also remove columns from **Manage Columns** Section in **Home** Tab in **Query Editor** Ribbon.



Name or Rename a Column

When you want give a meaningful name for a column that needs in Report, you can do it by renaming a column. To rename a column

Right click a column → Rename → Enter a meaningful Name

(OR)

Double click the column →that will allow you to edit Name → Enter a meaningful Name

Reorder Columns or Sort Columns in Power BI

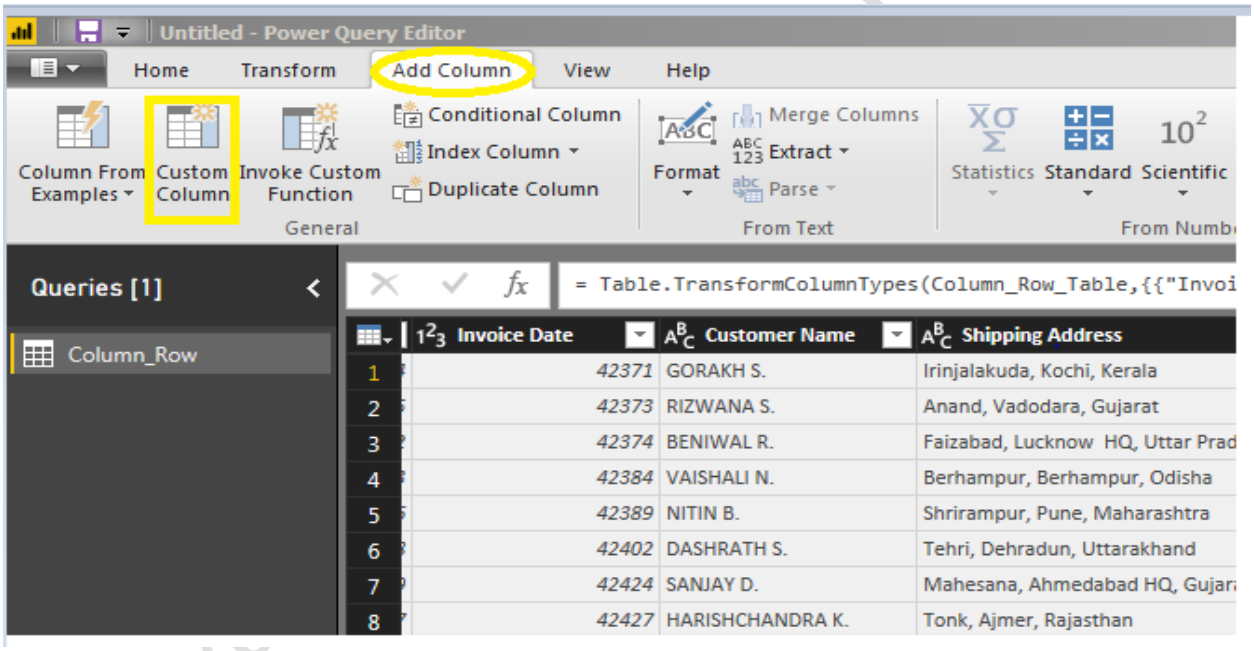
One way is drag the required column and Drop at the position you want to place.

Second way, right-click on the column name that you want to move will open the context menu. Select the **Move** and then select **Right**, **Left**, **To End**, or **To Beginning** options.

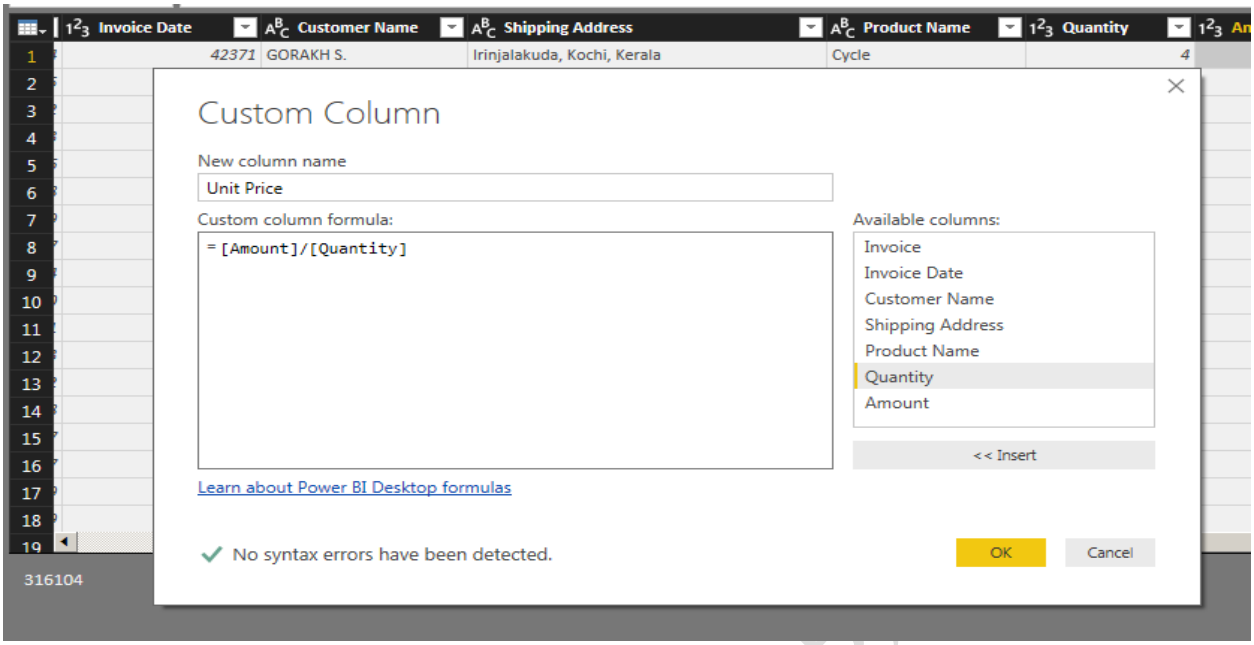
Add Column / Add Custom Column

In Query Editor you can create custom formulas that operate on multiple columns in your table, and then place the results of such formulas into a new (custom) column. Query Editor makes it easy to create custom columns.

In Query Editor, select Custom Column from the Add Column tab on the ribbon.



Once we select custom column a Custom Column window opens as below where we can provide New column name and Custom column formula.

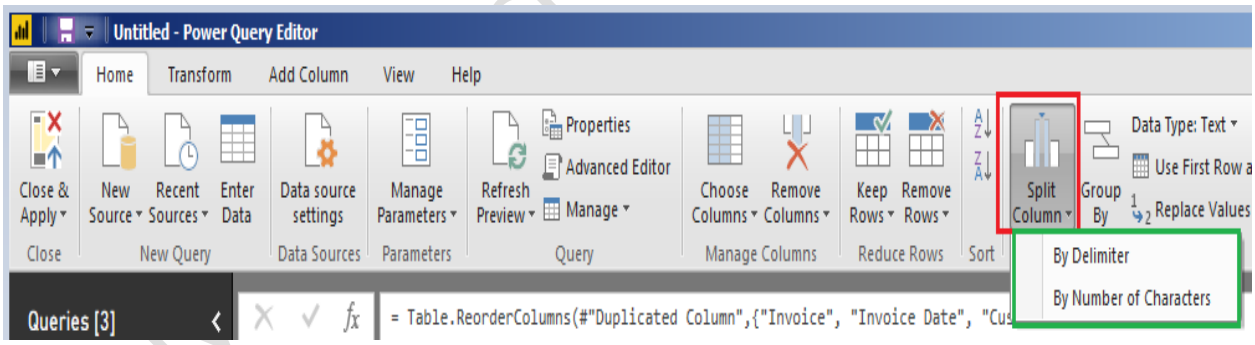


Split Columns

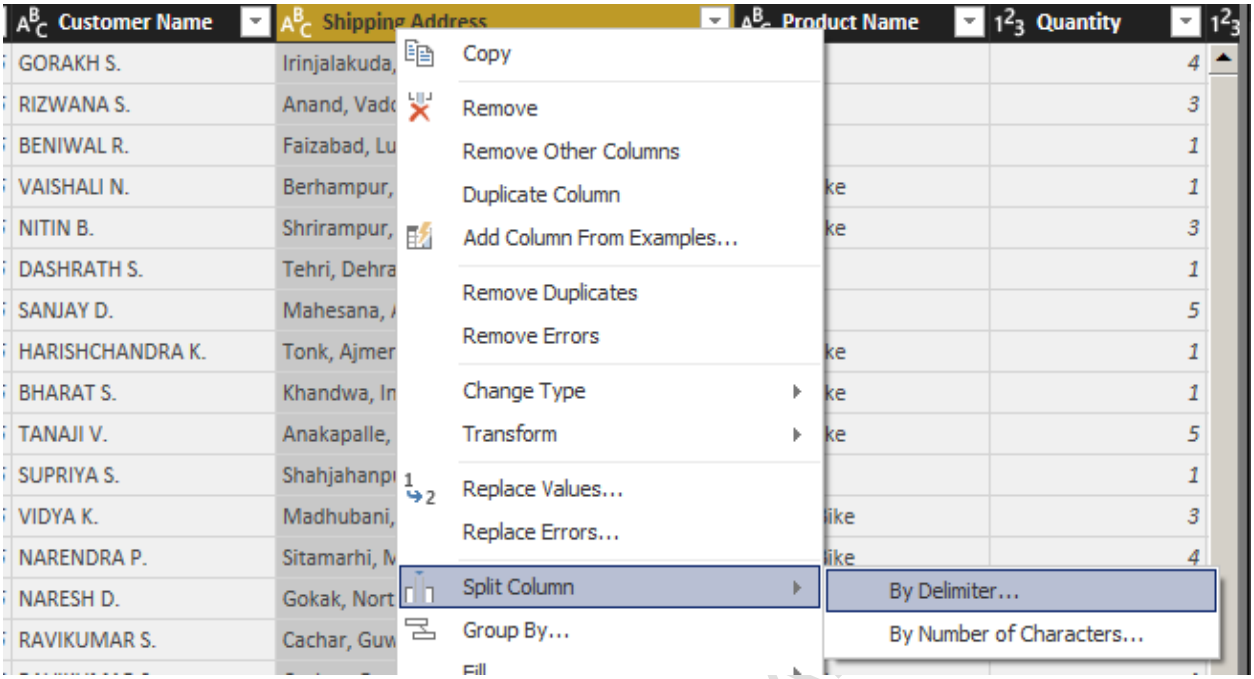
Sometimes you will get merged columns (one column with too much information). In that situation, you can use Power BI Split Columns option to split that column into multiple columns.

We have two options here to split columns as shown in below image.

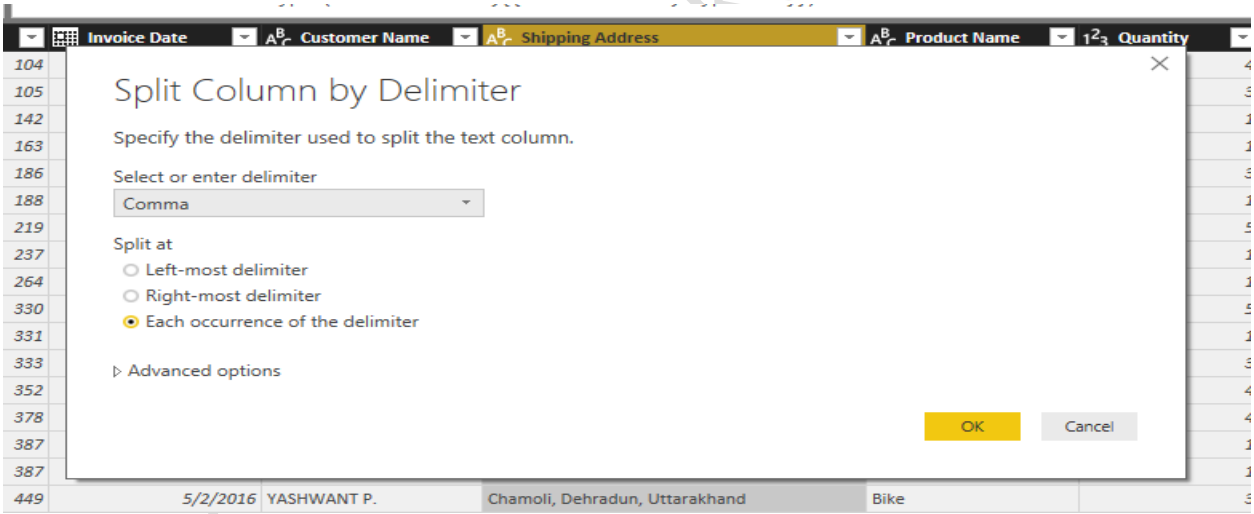
- By Delimiter
- By Number of Characters



In order to split the columns in a table, right-click on the column that you want to split will open the context menu. Select the Split Columns and then select "**By Delimiter"** option.



Selecting the "By Delimiter" option will open the following window.



Select or enter delimiter

Please select the delimiter that you want to use as the split character from the drop down list. If it is not there in the list, then select Custom option in the drop down and specify that custom character.

Left most delimiter

This option will split the left most string before first delimiter.

Right most delimiter

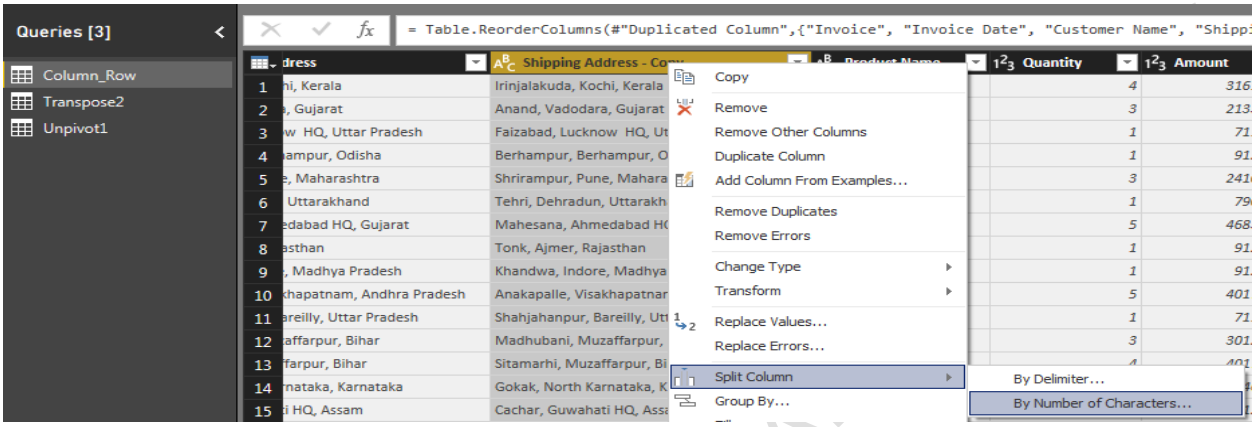
This option will split right most string after the last delimiter.

Each Occurrence of the delimiter

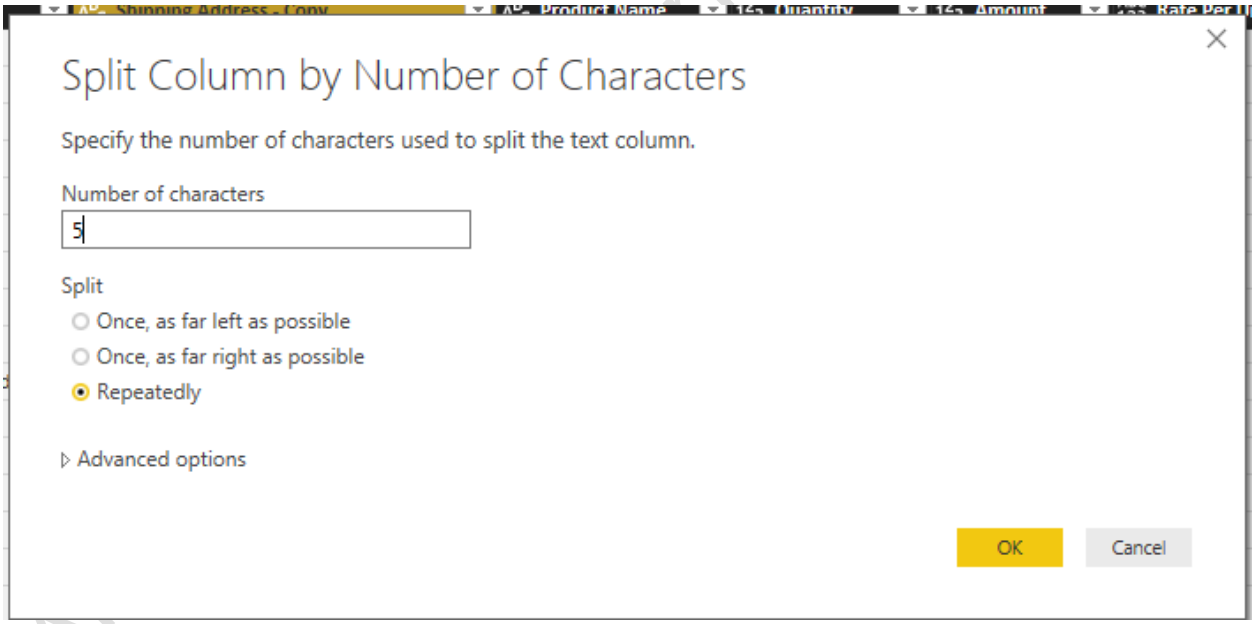
Text will split at each occurrence of a delimiter.

Split Columns by Number of Characters

Right-click on the column that you want to split will open the context menu. Please select the Split Columns and then select “By Number of Characters” option.



Selecting the “By Number of Characters” option will open the Split Column by Number of Characters window.



Number of Characters

Please specify the number of characters used to split the column. Let us give as 5.

Once, as far left as possible

This option will split the given string into two strings first string with first 5 Characters and second string with remaining characters.

Once, as far right as possible

This option will split the given string into two strings first string with all the characters except last 5 Characters and second string will be last 5 characters.

Repeatedly

Text will split for every 5 characters.

Merge Columns

With Power Query, you can merge two or more columns in your query. You can merge columns to replace them with a merged column, or create a new merged column alongside the columns that are merged.

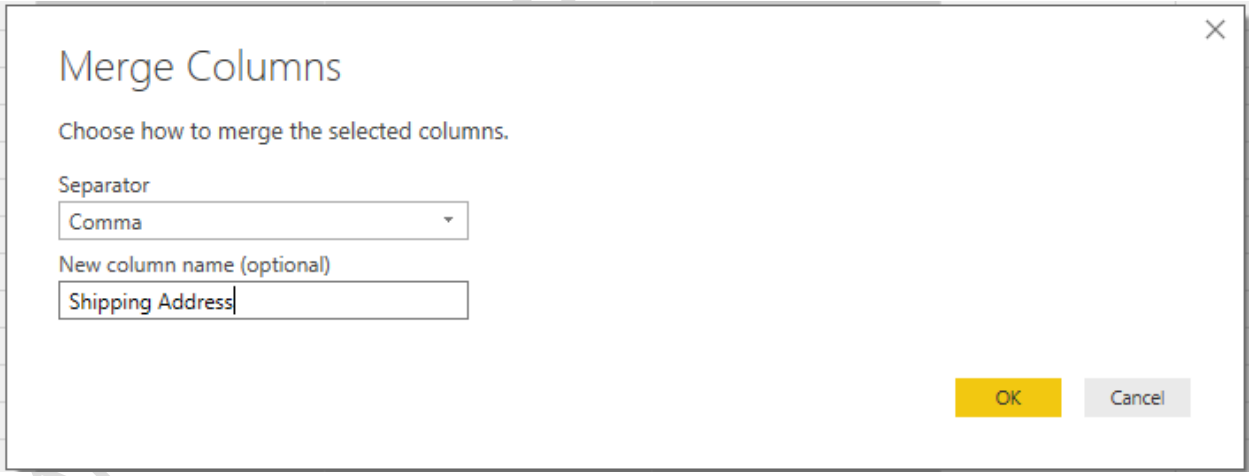
Merge columns to replace existing columns

Select two or more columns that you need to merge. Press the CTRL key, and then click on the column headers to select each of the columns that you'll include in the merge.

NOTE: The order in which you select the columns sets the order of the values in the merged column.

Right-click the columns and click Merge Columns.

In the Merge Columns popup window, specify the separator that is in use between each of the column values. You can select from predefined separator values, or specify a custom separator value. Give a meaningful Name in "**New column name"** section.



Click OK to create a merge column that replaces the columns selected for the merge operation.

Merge Columns to create a new column

Perform all the above steps in **Add Column** Tab in **Query Ribbon** to create a new column for all the merged columns.

PIVOT and UNPIVOT with Power BI

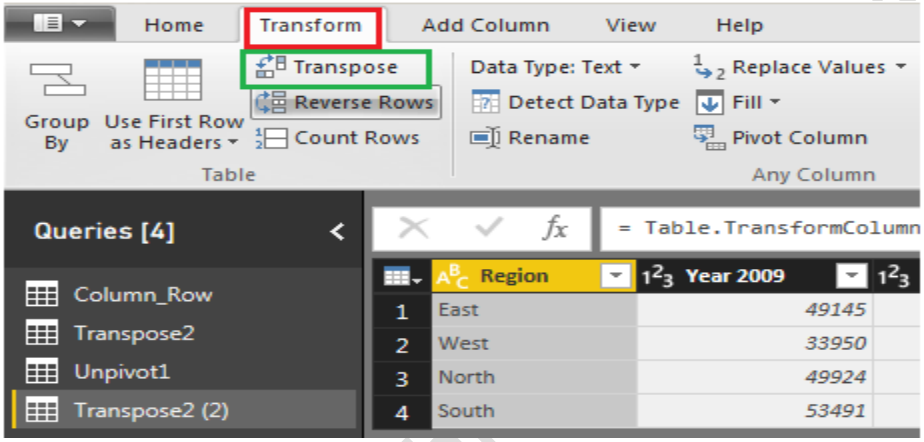
Turning columns to rows, or rows to columns is easy with Power Query and Power BI.

UNPIVOT → Converting Columns to Rows

PIVOT → Converting Rows to Columns

Transpose

This is used to reverse the rows and column of a table. Once you select the column and click on transpose option the rows becomes columns and column becomes rows. For using it click on Transpose option of Transform tab as below.



Region values are in Rows and Year Values are in columns in the below Query and we will use this for Transpose.

	Region	Year 2009	Year 2010	Year 2011	Year 2012	Year 2013	Year 2014	Year 2015	Year 2016
1	East	49145	15805	63544	14588	27252	20240	28094	38449
2	West	33950	43735	66850	23125	10444	41878	41825	27325
3	North	49924	76718	96314	99194	94416	31575	75907	20659
4	South	53491	20462	99796	57194	29032	34539	62883	32993

Once after Transpose the result will looks like below. Year values in rows and Region values in columns.

	Year	East	West	North	South
1	Year 2009	49145	33950	49924	53491
2	Year 2010	15805	43735	76718	20462
3	Year 2011	63544	66850	96314	99796
4	Year 2012	14588	23125	99194	57194
5	Year 2013	27252	10444	94416	29032
6	Year 2014	20240	41878	31575	34539
7	Year 2015	28094	41825	75907	62883
8	Year 2016	38449	27325	20659	32993

In built Row Transformations

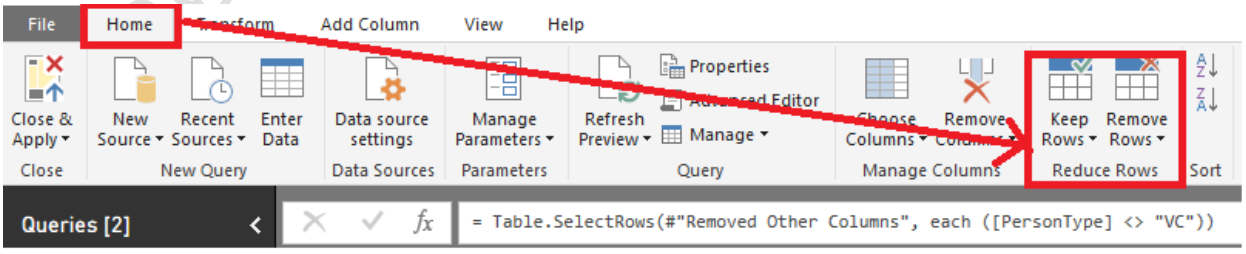
- ✓ Header Row or Use First Row as Headers
- ✓ Keep Top Rows
- ✓ Keep Bottom Rows
- ✓ Keep Range of Rows
- ✓ Keep Duplicates
- ✓ Keep Errors
- ✓ Remove Top Rows
- ✓ Remove Bottom Rows
- ✓ Remove Alternative Rows
- ✓ Remove Duplicates
- ✓ Remove Blank Rows
- ✓ Remove Errors
- ✓ Group Rows

Use First Row as Header

When power bi is not able to identify headers automatically, you can manually do that using "**Use First Row as Header"**.

Reduce Rows

In case you want to filter the data you are importing, you have two options: either by keeping the specific rows or removing rows. Both options can be found by clicking **Home → Reduce Rows**.



Under **Keep Rows**, you have the following options

Keep Top Rows → where you specify the number of top rows to keep.

Keep Bottom Rows → for which you pick the number of bottom rows to keep.
Keep Range of Rows → which skips a specified number of top rows and then keeps the chosen number of rows.

In addition to the first three options, which work on whole tables, you have **Keep Duplicates** and **Keep Errors**, both of which can work on either the whole table or the selected columns only. For example, if you select the whole table and choose Keep Duplicates, you will only see the rows that are complete duplicates of each other. However, if you choose only one column and click Keep Duplicates, you will get the rows where the values in the selected column are duplicates, regardless of other columns' values.

Under **Remove Rows**, you have six options

Remove Top Rows → Removes a specified number of top rows. Works on the whole table only.
Remove Bottom Rows → Removes a specified number of bottom rows. Works on the whole table only.
Remove Alternate Rows → Removes rows following a user-supplied pattern: it starts with a specified row, then alternates between removing the selected number of rows and keeping the chosen number of rows. Works on the whole table only.
Remove Duplicates → Removes rows that are duplicates of other rows. Works on either the whole table or the selected columns only.
Remove Blank Rows → Removes rows that completely consist of either empty strings or nulls; if you need to remove blank values from one column, you can click on the arrow to the right of a column's name and click Remove Empty. Remove Blank Rows Works on the whole table only.
Remove Errors → Removes rows that contain errors. Works on either the whole table or the selected columns only.

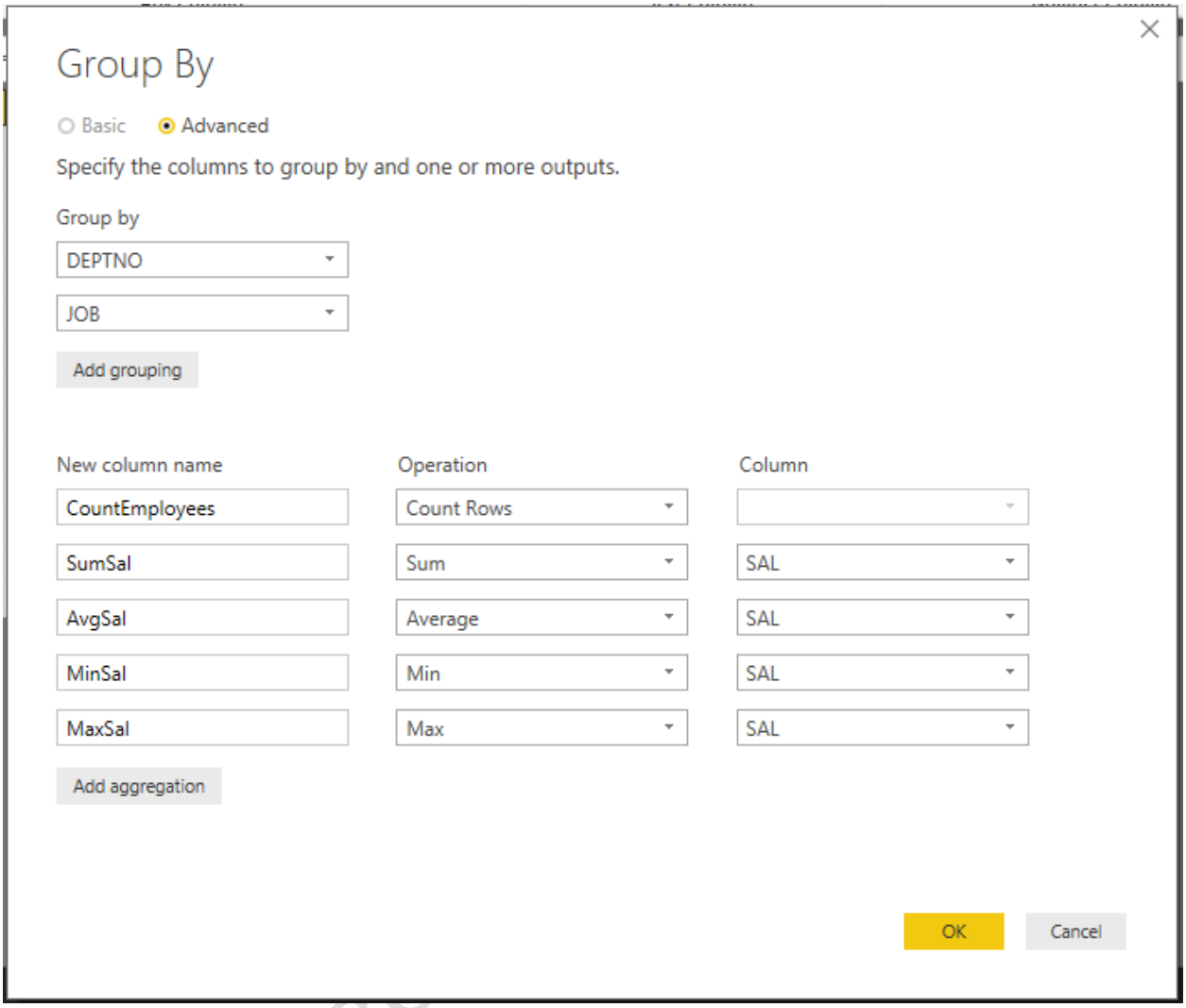
In case of Remove Duplicates and Remove Errors, there is a difference between applying these options to all selected columns or the whole table. In the first case, if you have new columns added to your query, the functions will not work on the new columns, because selecting all columns keeps their names in the code. To remove duplicates or errors from the whole table, select the table icon above row numbers and choose either Remove Duplicates or Remove Errors.

Group rows / Group By

In Query Editor, you can group the values in multiple rows into a single value. This can be useful when summarizing the number of products offered, the total sales, or the count of students or total salary paid for each department.

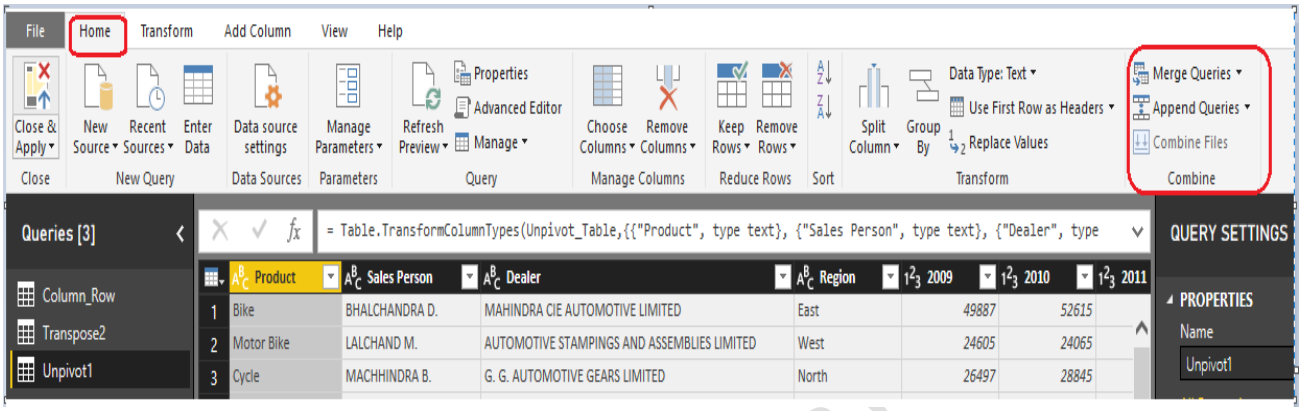
Let's summarize our EMP table by Deptno and Job wise TotSal, MinSal, MaxSal, CountOfEmp.

To do that, you can select Deptno and Job first, then click **Home → Transform →** Group By (**Or) Transform → Table → Group By**. The **Group By** window then opens; you'll see a radio button to switch between Basic and Advanced settings. Specify one or more columns to group by and how to aggregate data. To group by more than one column, switch to Advanced settings, or you could have pre-selected multiple columns before clicking **Group By**.



Combine Queries

Combining more than one queries either horizontal or vertical we call it as Combine Queries. Result of a combine operation on one or more queries will be only one query. You can find **Append Queries** or **Merge Queries** in the Combine Queries section of the Query Editor in Power BI. Append Queries will combine the queries data vertically. Merge Queries will combine the queries data horizontally.



Append Queries

Append Queries means results of two (or more) queries will be combined vertically into one query in the below way

Rows will be appended after each other. (For example appending a query with 50 rows with another query with 100 rows, will return a result set of 150 rows)

Each Query should contain same number of columns and same datatype for columns for better outputs. (For example col1, col2… col10 in first query, after appending with same columns in the second query will result into one query with single set of col1, col2… col10)

Append 2 files individually with different queries

Consider two sample data sets, one for each month sales,

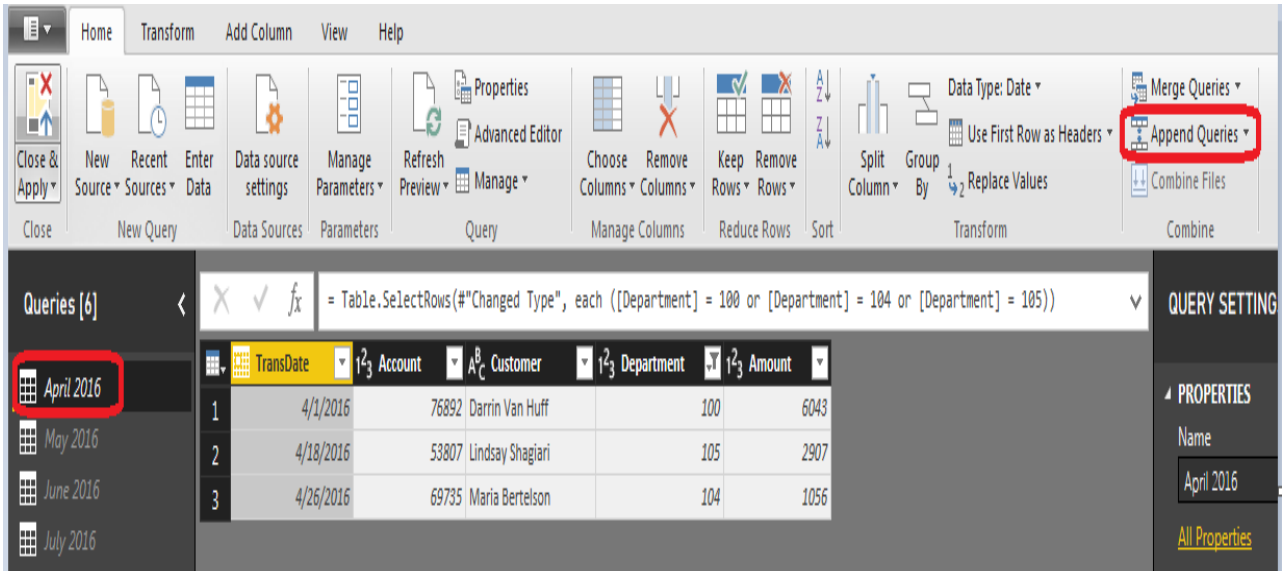
Sales for April Month

	TransDate	Account	Customer	Department	Amount
1	4/1/2016	76892	Darrin Van Huff	100	6043
2	4/18/2016	53807	Lindsay Shagiari	105	2907
3	4/26/2016	69735	Maria Bertelson	104	1056

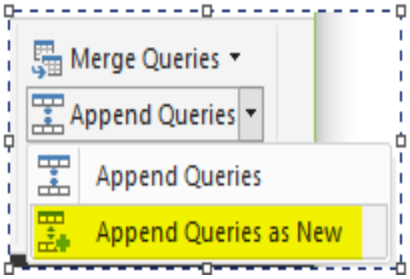
Sales for May Month

	TransDate	Account	Customer	Department	Amount
1	5/1/2016	80596	Craig Carreira	107	3484
2	5/16/2016	72039	Ross Baird	115	3794
3	5/17/2016	55021	Troy Blackwell	101	901
4	5/29/2016	72216	Linda Cazamias	107	9437

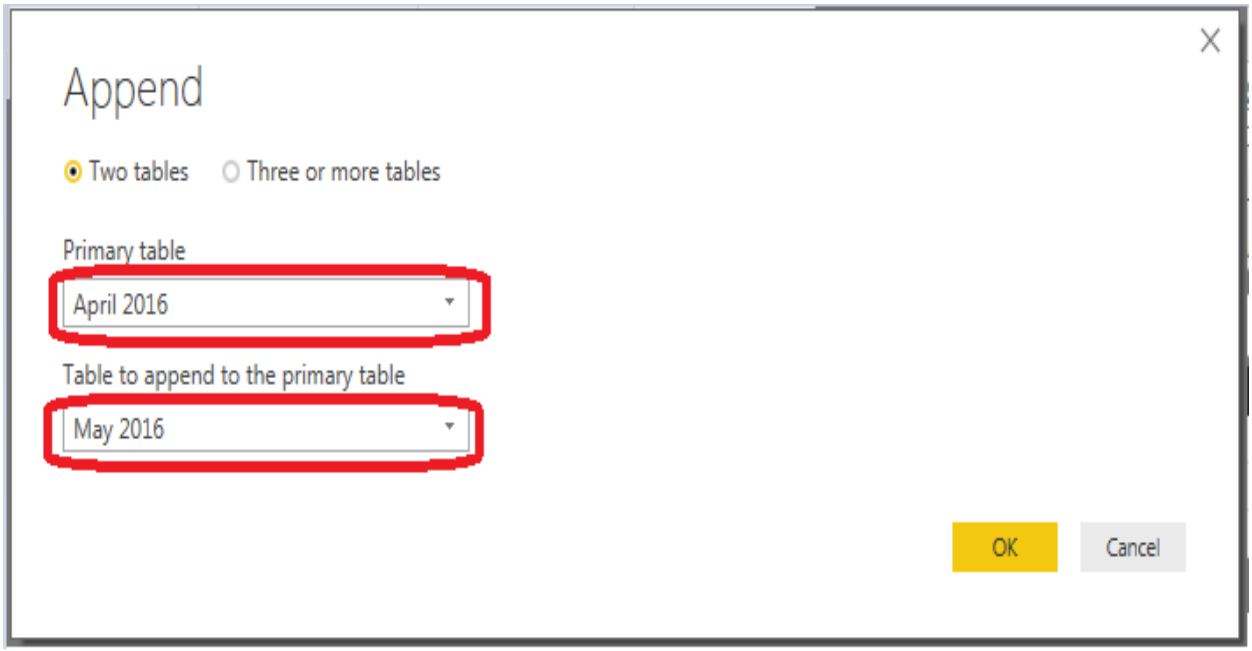
To append these queries, click on one of them and select Append Queries from the **Combine** section of **Home** tab in Query Editor.



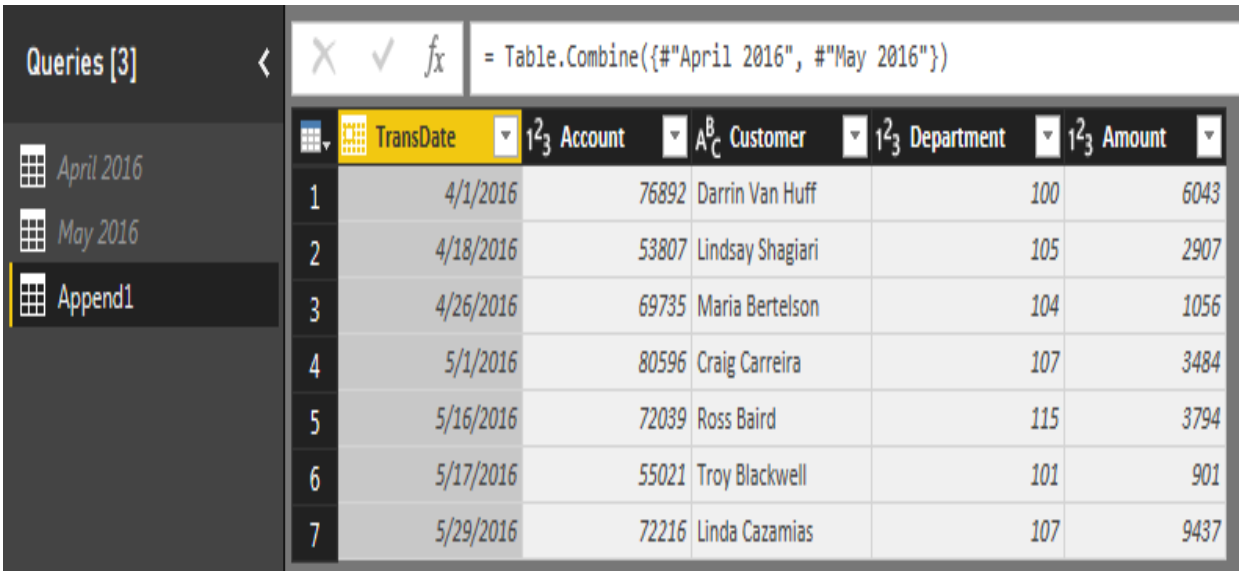
If you want to keep the existing query result as it is, and create a new query with the appended result choose Append Queries as New, otherwise just select Append Queries. In this example I'll do Append Queries as New, because I want to keep existing queries intact.



You can choose what is the primary table (normally this is the query that you have selected before clicking on Append Queries), and the table to append. In the append query result you will see primary table values first and others next.



For this example, I have only two tables, so I'll continue with above configuration. Append Queries simply append rows after each other, and because column names are exactly similar in both queries, the result set will have same columns.



	TransDate	Account	Customer	Department	Amount
1	4/1/2016	76892	Darrin Van Huff	100	6043
2	4/18/2016	53807	Lindsay Shagiari	105	2907
3	4/26/2016	69735	Maria Bertelson	104	1056
4	5/1/2016	80596	Craig Carreira	107	3484
5	5/16/2016	72039	Ross Baird	115	3794
6	5/17/2016	55021	Troy Blackwell	101	901
7	5/29/2016	72216	Linda Cazamias	107	9437

Result of Append as simple as below. Append is similar to UNION ALL in T-SQL.

	TransDate	Account	Department	Amount
1	4/1/2016	76892	100	6043
2	4/18/2016	53807	105	2907
3	4/26/2016	69735	104	1056

	TransDate	Account	Department	Amount
1	4/1/2016	76892	100	6043
2	4/18/2016	53807	105	2907
3	4/26/2016	69735	104	1056
4	5/1/2016	80596	107	3484
5	5/16/2016	72039	115	3794
6	5/17/2016	55021	101	901
7	5/29/2016	72216	107	9437

	TransDate	Account	Department	Amount
1	5/1/2016	80596	107	3484
2	5/16/2016	72039	115	3794
3	5/17/2016	55021	101	901
4	5/29/2016	72216	107	9437

What if files contain duplicates?

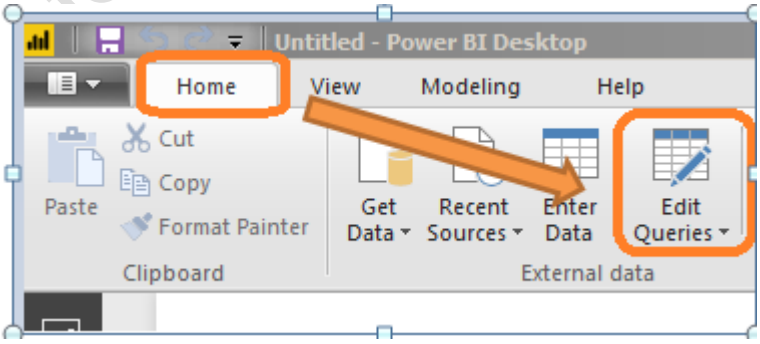
Append Queries will NOT remove duplicates. You have to use Group By or Remove Duplicate Rows to get rid of duplicates.

What if the Columns in the query are not matched exactly?

Append requires columns to be exactly similar to work in best condition. If columns in source queries are different, append still works, but will create one column in the output per each new column, if one of the sources doesn't have that column the cell value of that column for those rows will be null.

Append 3 or more files individually with different queries

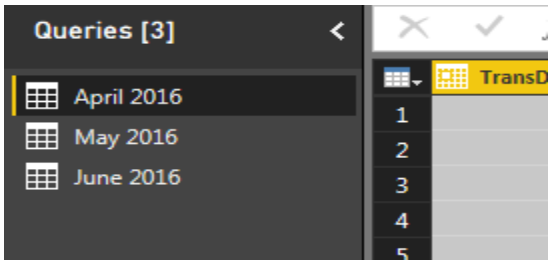
Open Edit Queries as shown below.



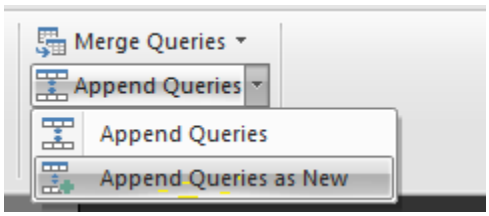
Query Editor Will opens → From Home Tab → selects **New Source** → Select the type of source for example **Text/CSV** as shown below → Browse for the file location → Select the file → Open.

Apply all steps above for each file to get multiple files into Query Editor for Appending.

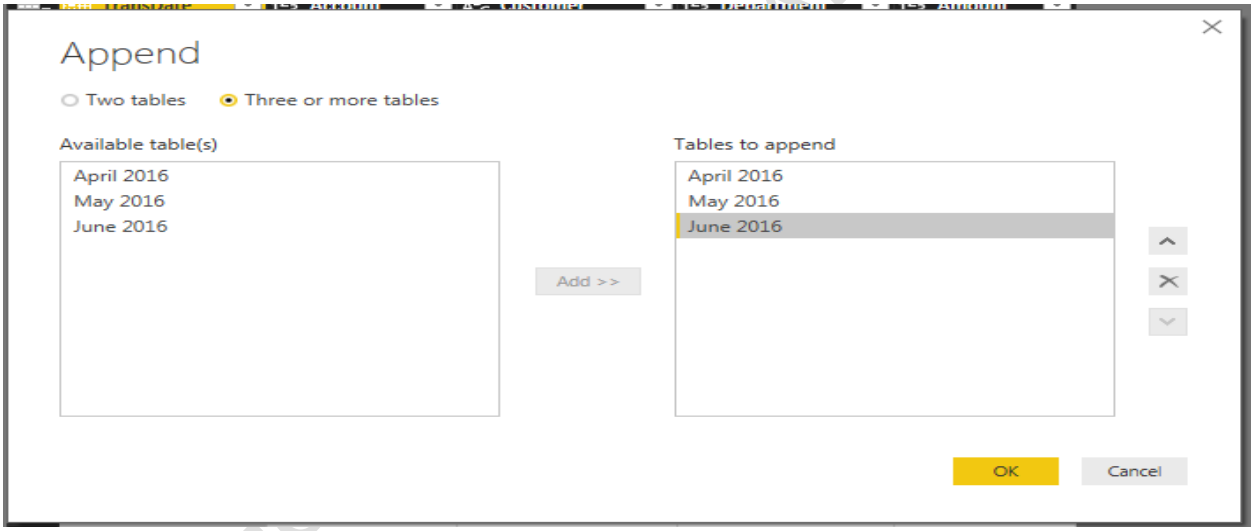
Now all the three files are there in Query Editor as shown below. Now we will see how to append them.



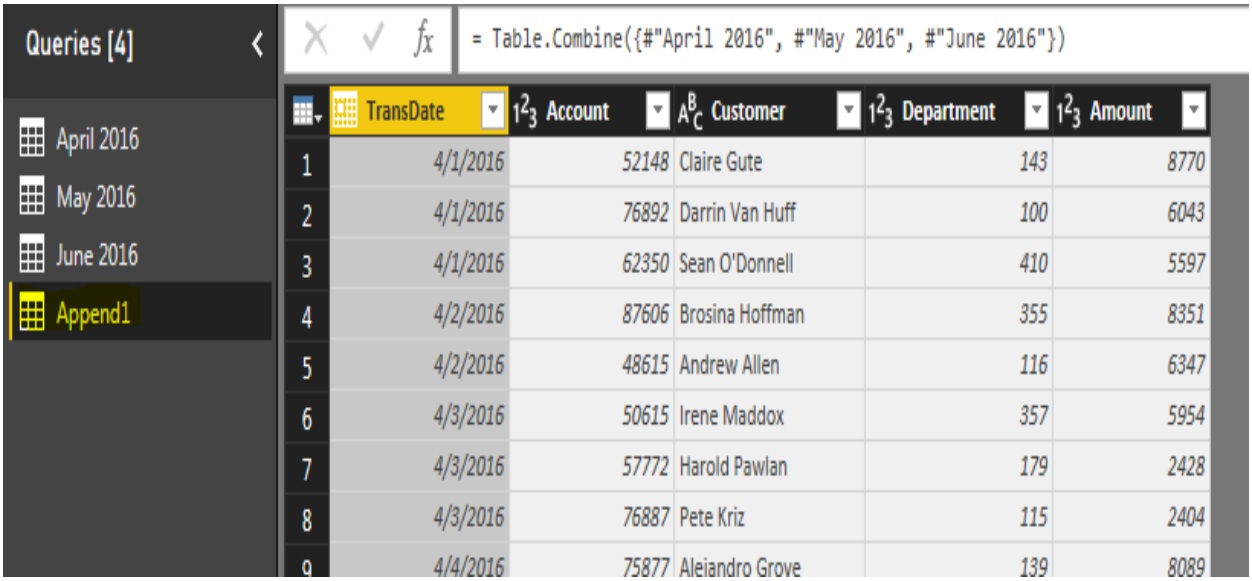
Select Append Queries → Under select **Append Queries as New**



Select three or more tables → Add tables from **Available tables** section to **Tables to append** section



Change the order using up and down button as shown in above image, the way you are expecting the data in append query result→ Ok. Now you will see an Append Query with the data from all the three files we selected in the same order as we placed in **Tables to append section** as shown below.

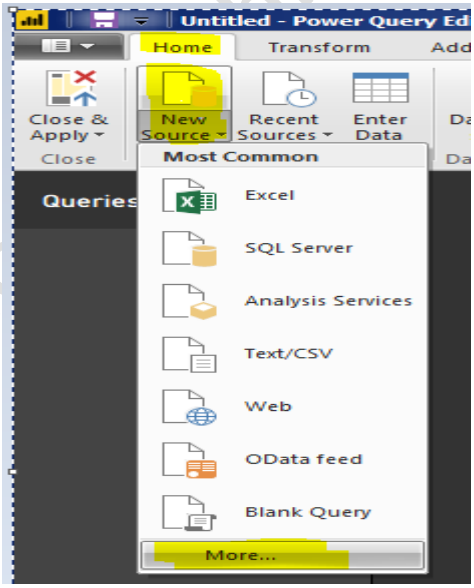


	TransDate	Account	Customer	Department	Amount
1	4/1/2016	52148	Claire Gute	143	8770
2	4/1/2016	76892	Darrin Van Huff	100	6043
3	4/1/2016	62350	Sean O'Donnell	410	5597
4	4/2/2016	87606	Brosina Hoffman	355	8351
5	4/2/2016	48615	Andrew Allen	116	6347
6	4/3/2016	50615	Irene Maddox	357	5954
7	4/3/2016	57772	Harold Pawlan	179	2428
8	4/3/2016	76887	Pete Kriz	115	2404
9	4/4/2016	75877	Alejandro Grove	139	8089

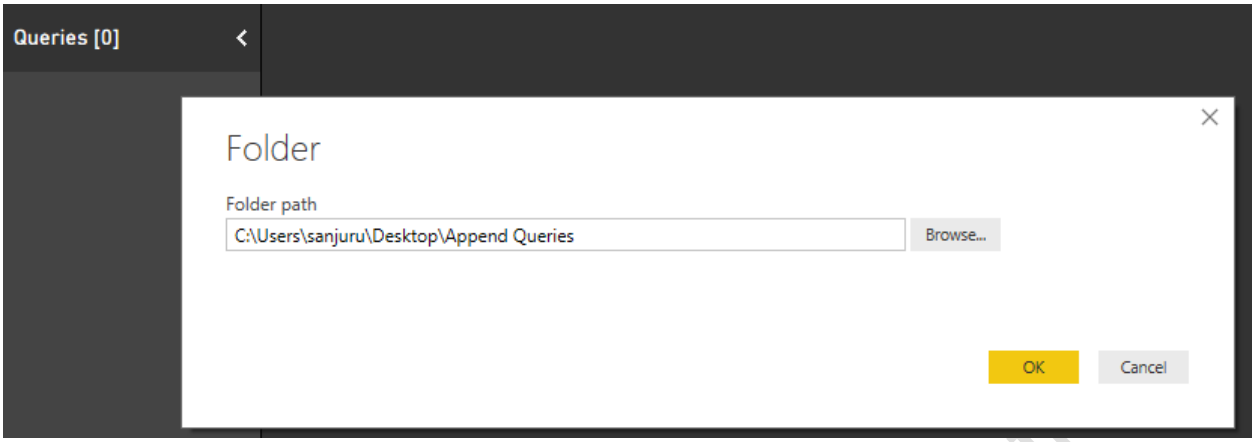
Appending multiple files of same type from a folder using single query

In the above example we append April, May and June files. Tomorrow let us assume we get a file for July Month. In the above process you need to import it manually and need to append all the four files. It's a manual and lengthy process. Let us see how we can automate it by a single refresh and how to import and append with single query.

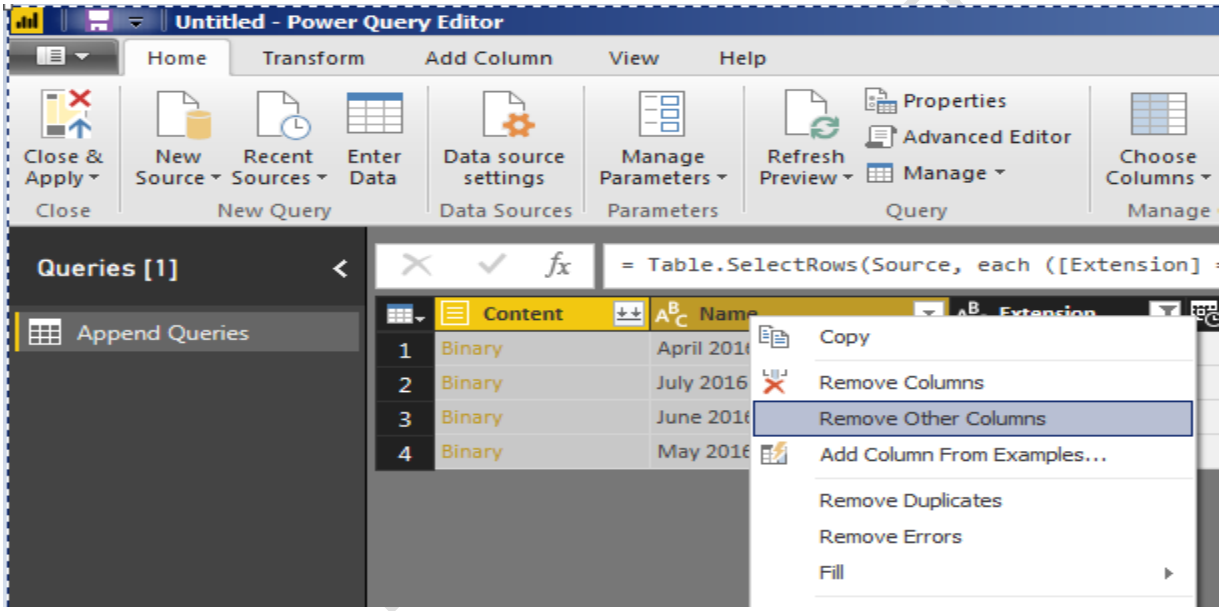
Open Query Editor → Home → New Source → More...



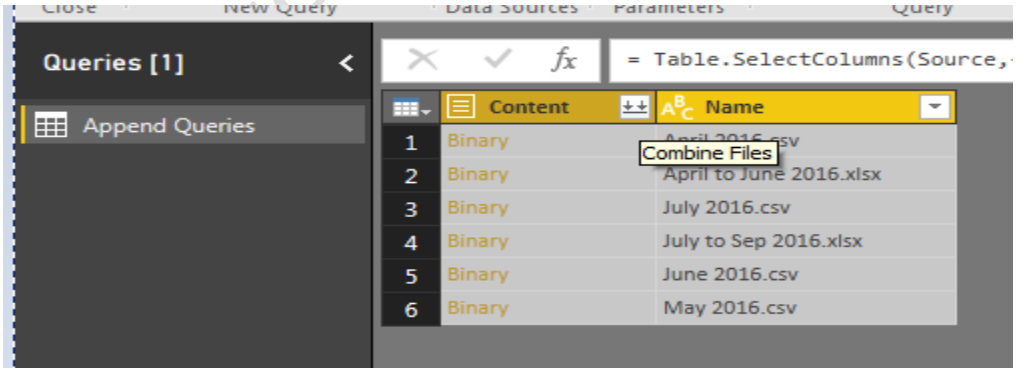
Select **Folder** → Connect → Enter the folder Path where files are located

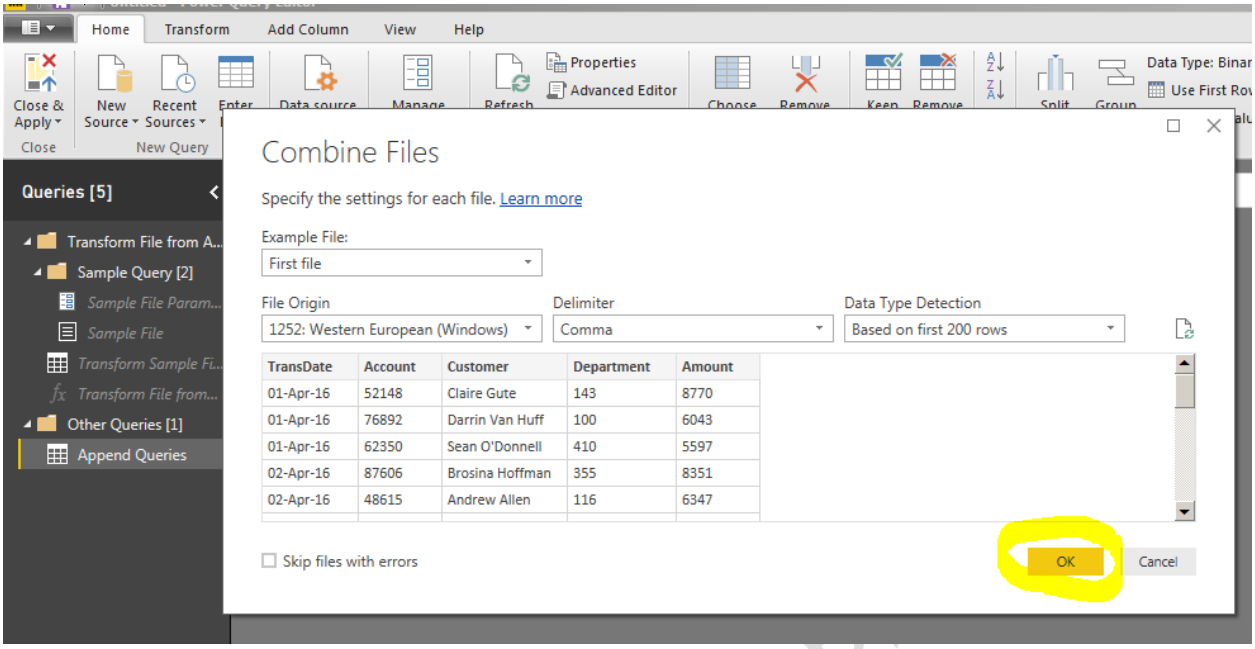


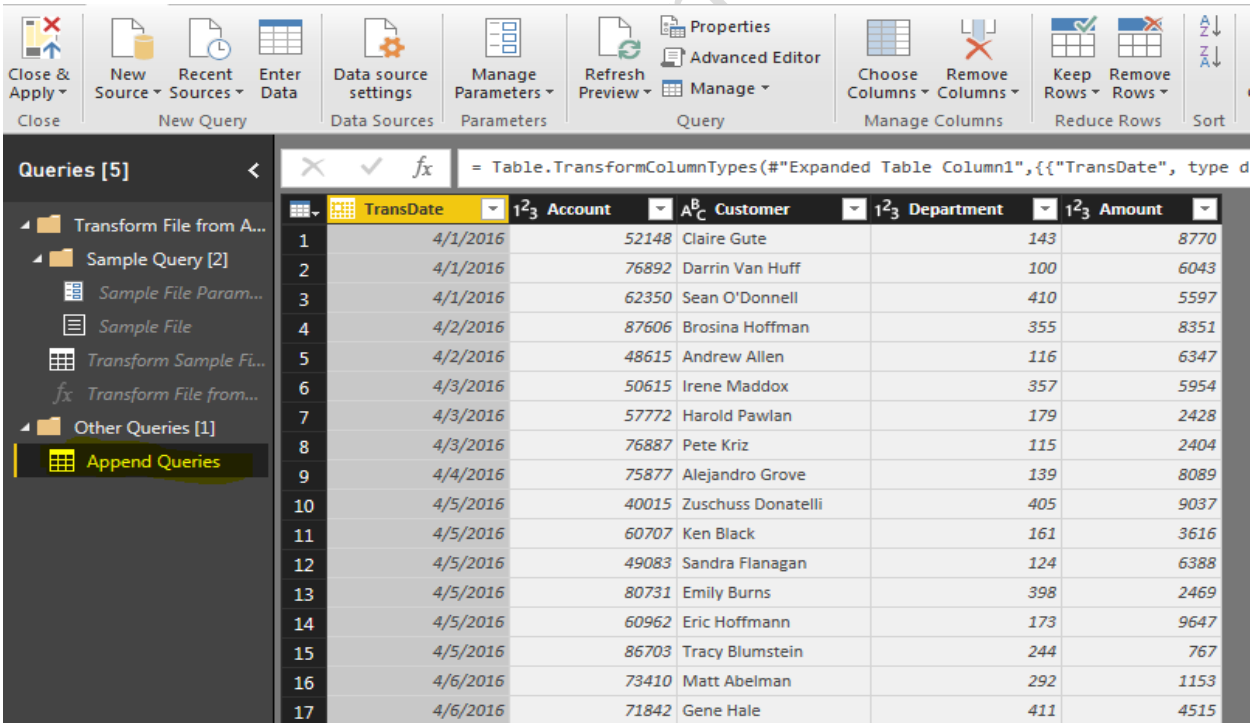
Select Ok → Edit → Perform Transformation Steps as shown below.



Select two down arrows to combine files as shown in below image







Tomorrow when you get a new file, place it in the folder and simply refresh in Query Editor. You will find updated data in Append Queries section.

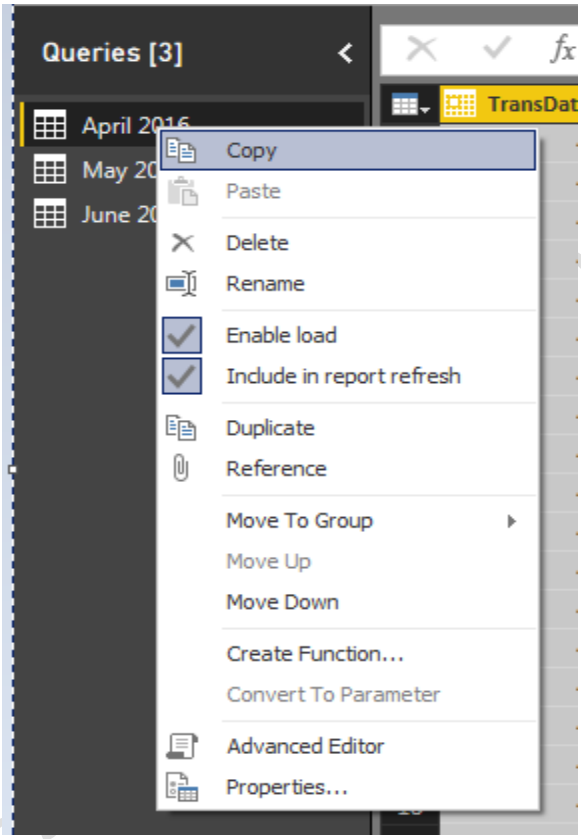
Task

Append the data from multiple excel sheets in an excel workbook and multiple workbooks or files in folder with a single query.

Excel.Workbook([Content])

Right-click options of a Query

When you right-click on the name of a query (in the left-most pane of Power Query's window) you will see list of options a show below. We will discuss each of them now.



Copy

Copy option will copy the query and you can paste the query where ever you needed.

Paste

Paste is used to paste the copied query under the queries section.

Delete

Delete is used to delete a particular Query.

Rename

Rename will help you to rename a query.

Enable Load

Always remember that Query Editor is your ETL (Extract, Transform, and Load) engine. It will apply all transformations before loading the data into the model, but once you finished the transformation all queries will be loaded into the model and they take memory. By default, all queries are enabled to Load into the model. Simply change that for queries that are not required in the model by disabling the option “Enable Load”.

"Enable Load" means query results are available for report builder. Otherwise you may use it in your other queries (for example to merge data), but it is not shown in the report builder.

Include in report refresh

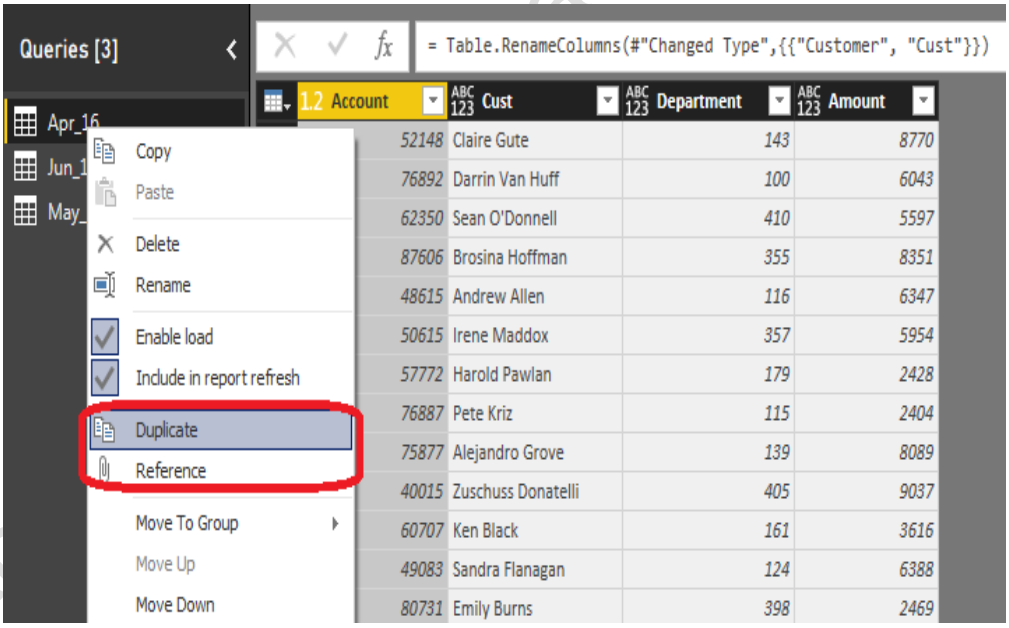
"Include in Report Refresh" means query is automatically refreshed when you press "Refresh" button on the ribbon. Disable this option for the static tables to improve the performance of report refresh.

Duplicate Query

Duplicate will duplicate the code of the query. Duplicate is generally used when you would like to create a similar query and you do not want to type the same code. Once you created a duplicate query from source query then duplicate query is independent of source query. You can make changes to this duplicate query.

Reference Query

Reference Query means you would like to use that Query results in some other queries where your original query remains as a base Query. Once you created a reference query from source query then reference query is dependent of source query. When you make changes to source query those will reflect in reference query. You can make additional changes to this reference query.



Merge Queries

Merge Queries are used when we need to add one or more columns to a Query from another Query. Merge is similar to JOINS in T-SQL or Other Databases.

Now we will see how to merge queries with an example. Take the EMP and DEPT Queries as shown below.

EMP

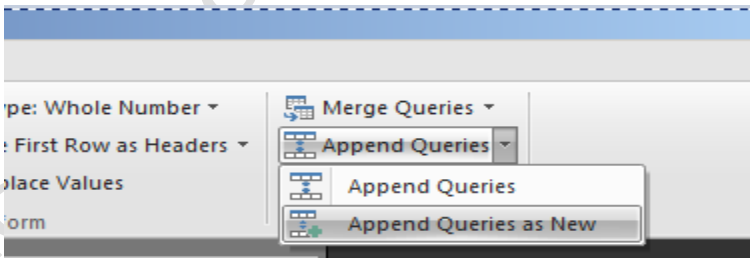
EMPNO	ENAME	JOB	MGR	SAL	DEPTNO
7369	SMITH	CLERK	1001	800	20
7499	ALLEN	SALESMAN	7566	1600	30
7521	WARD	SALESMAN	7566	1250	30
7566	JONES	MANAGER	7839	2975	20
7788	SCOTT	ANALYST	7839	3000	20
7839	KING	PRESIDENT		5000	10
7934	MILLER	CLERK	1001	1300	10
1001	SUNIL	TRAINER	7839	10000	50

DEPT

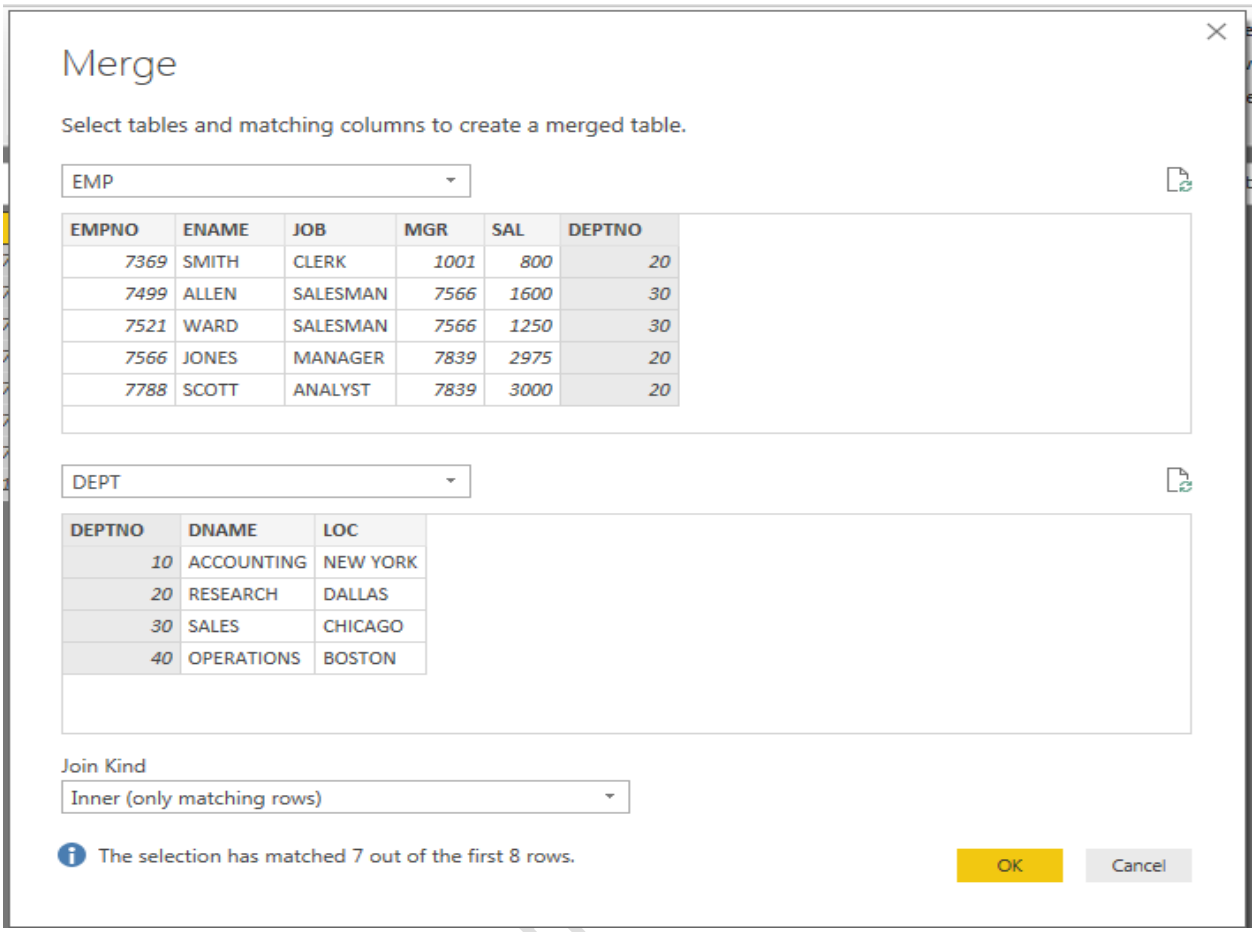
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

If you want to see employee details along with the department details in the same query then you need to merge EMP, DEPT Queries. Now we will see how to merge both of them.

Get both the Queries (EMP, DEPT) into Power Query. Select EMP Query first, and then select Merge Queries (Merge Queries as New) from the Home tab on the ribbon as shown below.



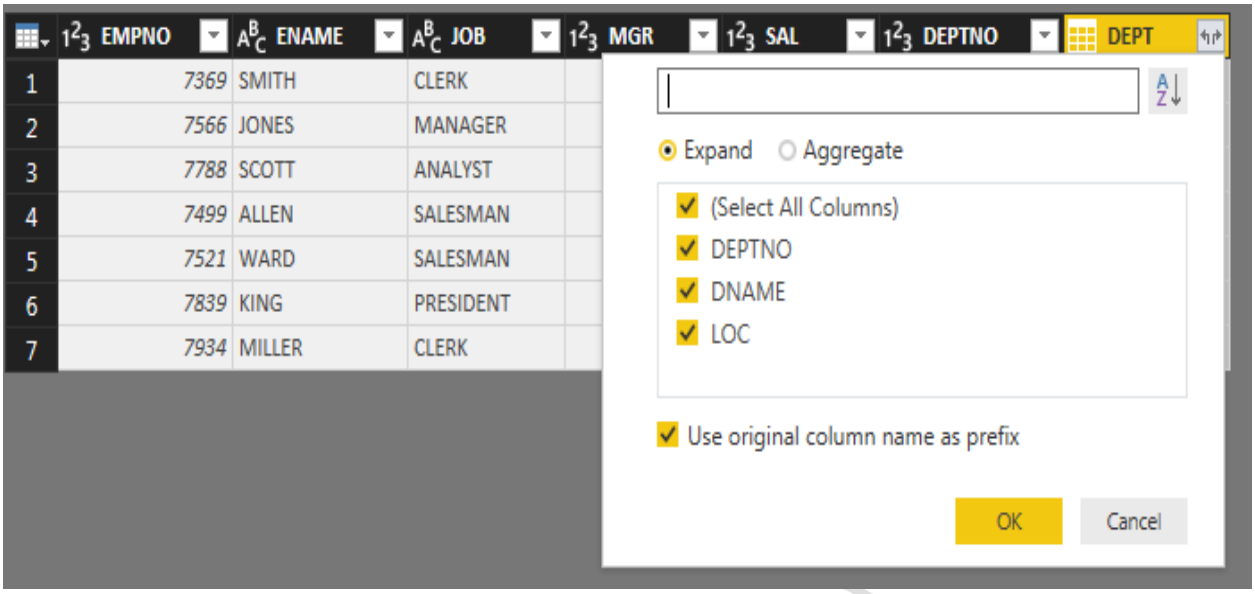
Next the Merge window appears prompting us to select the tables, matching columns and **Join Kind** to create a merged table as shown below. We will discuss more about Join Kind later.



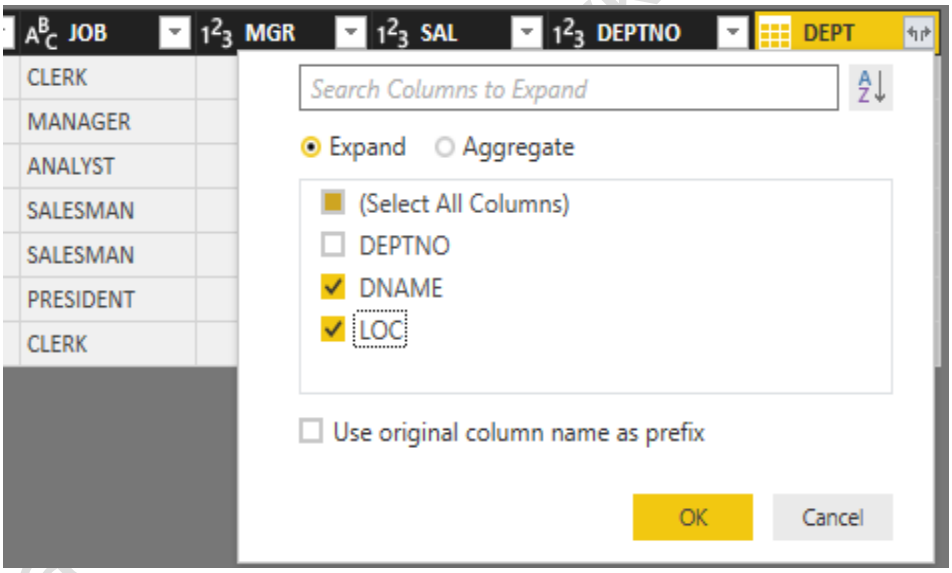
Once you select tables, matching columns and Join Kind Then Ok button enabled. Once you press **OK**, a NewColumn is created at the end of the query as shown below. The NewColumn is the contents of the table (query) that was merged with the existing query. All columns from the merged query are condensed into the NewColumn.

	EMPNO	ENAME	JOB	MGR	SAL	DEPTNO	DEPT
1	7369	SMITH	CLERK	1001	800	20	Table
2	7566	JONES	MANAGER	7839	2975	20	Table
3	7788	SCOTT	ANALYST	7839	3000	20	Table
4	7499	ALLEN	SALESMAN	7566	1600	30	Table
5	7521	WARD	SALESMAN	7566	1250	30	Table
6	7839	KING	PRESIDENT	null	5000	10	Table
7	7934	MILLER	CLERK	1001	1300	10	Table

To Expand the merged table, and select which columns to include, select the expand icon ( ). The **Expand** window appears as shown below.



In this case, we only want the DNAME and LOC columns, so we select only that columns and then select OK. We clear the checkbox from Use original column name as prefix because we don't need or want that, if we leave that selected, the merged column would be named **DEPT.DNAME, DEPT.LOC** (the original column name, or NewColumn, then a dot, then the name of the column being brought into the query).



Select OK to see the merged query output as shown below.

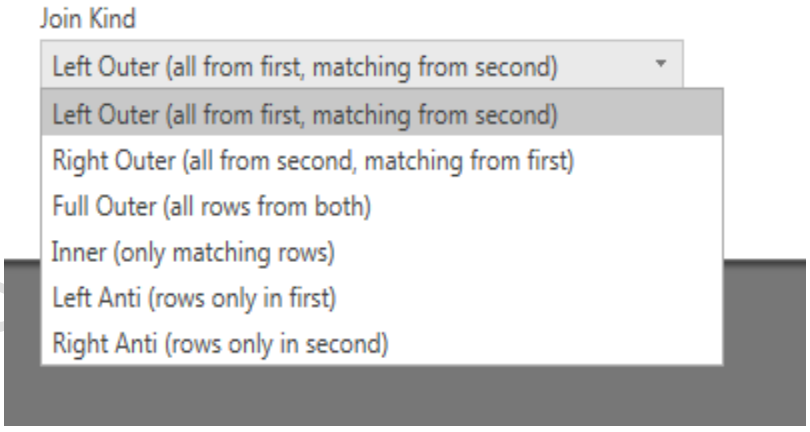
	EMPNO	ENAME	JOB	MGR	SAL	DEPTNO	DNAME	LOC
1	7369	SMITH	CLERK	1001	800	20	RESEARCH	DALLAS
2	7566	JONES	MANAGER	7839	2975	20	RESEARCH	DALLAS
3	7788	SCOTT	ANALYST	7839	3000	20	RESEARCH	DALLAS
4	7499	ALLEN	SALESMAN	7566	1600	30	SALES	CHICAGO
5	7521	WARD	SALESMAN	7566	1250	30	SALES	CHICAGO
6	7839	KING	PRESIDENT	null	5000	10	ACCOUNTING	NEW YORK
7	7934	MILLER	CLERK	1001	1300	10	ACCOUNTING	NEW YORK

You can easily use multiple columns in join condition for merging two data sets. Just select them in an order with holding Ctrl key of the keyboard.

Types of Joins / Join Kinds / Merge Type

There are 6 types of joins you can perform for merging the queries by default as shown below. Each of these joins gives you different results in merge query output. Default Join Kind is Left Outer. Let's see what their difference is.

1. Left Outer (all from first, matching from second)
2. Right Outer (all from second, matching from first)
3. Full Outer (all rows from both)
4. Inner (only matching rows)
5. Left Anti (rows only in first)
6. Right Anti (rows only in second)



We use below EMP and DEPT Tables to illustrate these Join Types

EMP

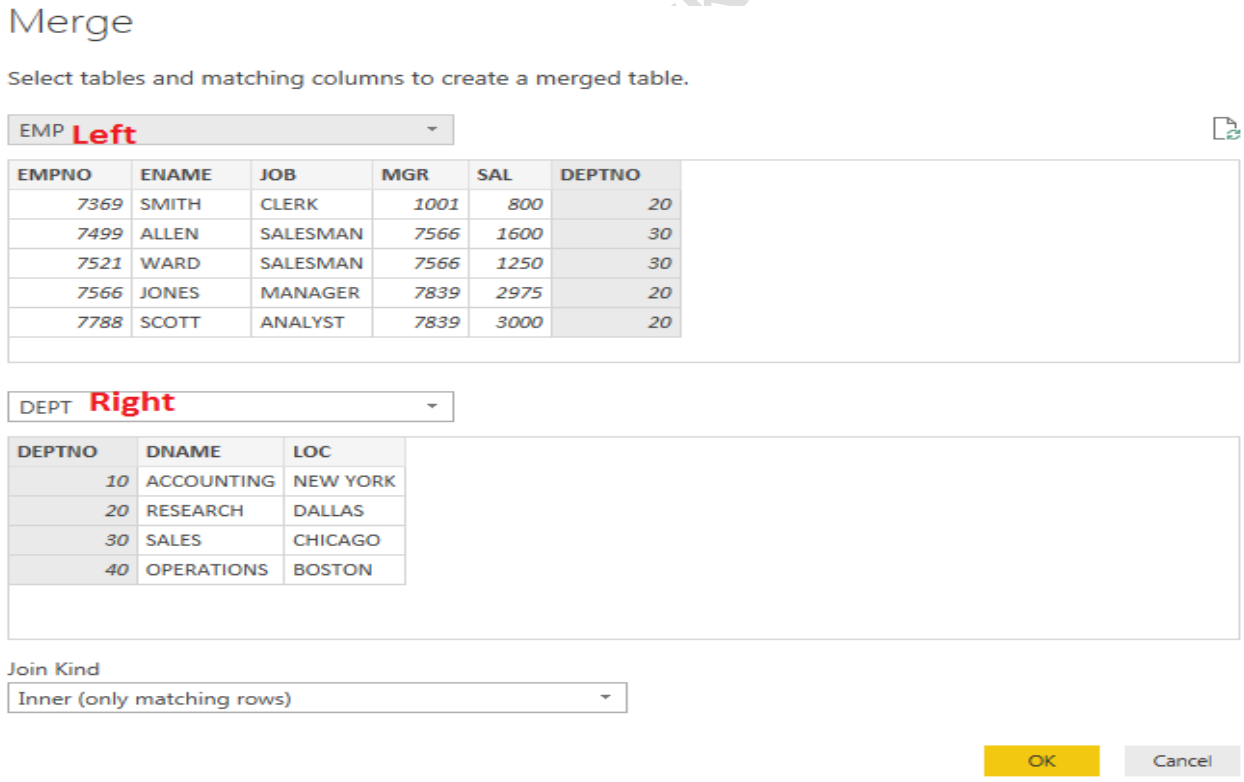
EMPNO	ENAME	JOB	MGR	SAL	DEPTNO
7369	SMITH	CLERK	1001	800	20
7499	ALLEN	SALESMAN	7566	1600	30
7521	WARD	SALESMAN	7566	1250	30
7566	JONES	MANAGER	7839	2975	20
7788	SCOTT	ANALYST	7839	3000	20
7839	KING	PRESIDENT		5000	10
7934	MILLER	CLERK	1001	1300	10
1001	SUNIL	TRAINER	7839	10000	50

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Left and Right

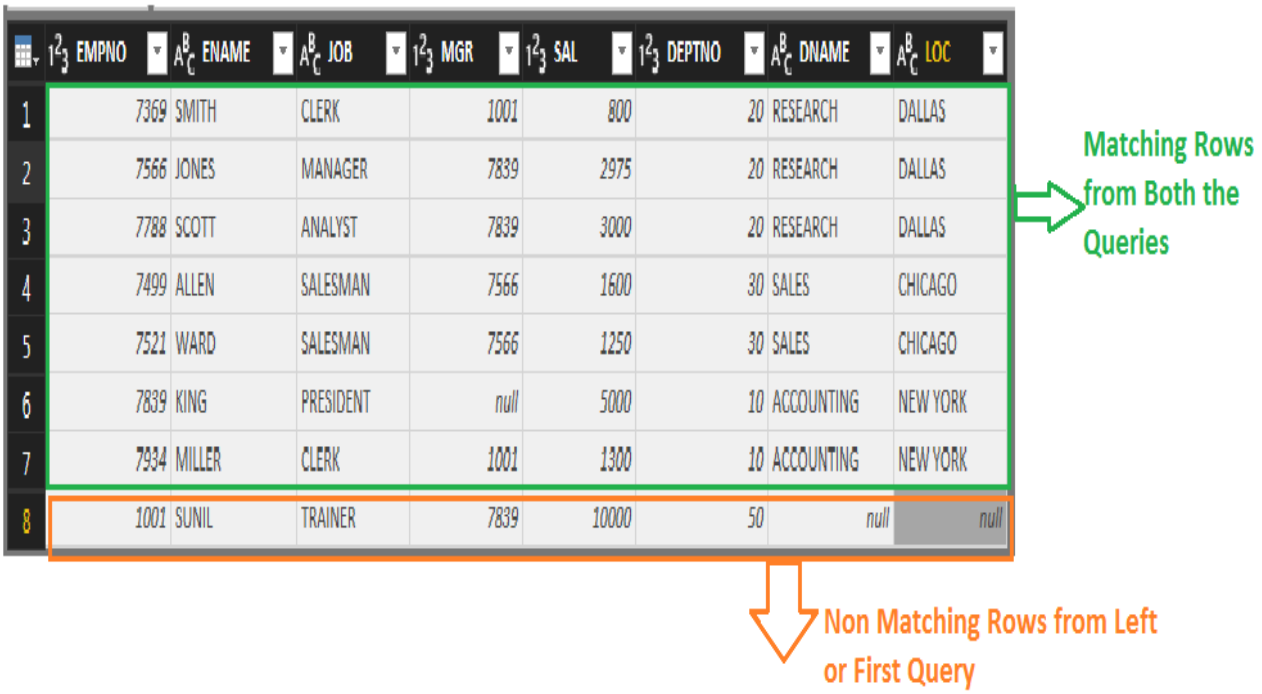
To start, you need to know the concept of Left and Right tables (or queries). When you merge two data sets with each other, the first query is considered as LEFT and the second as RIGHT.



In the above example EMP is Left and First and DEPT is Right and Second. Understanding this is important, because most of Join Kinds works with the concept of left or right or both.

Left Outer (All from first, matching from second)

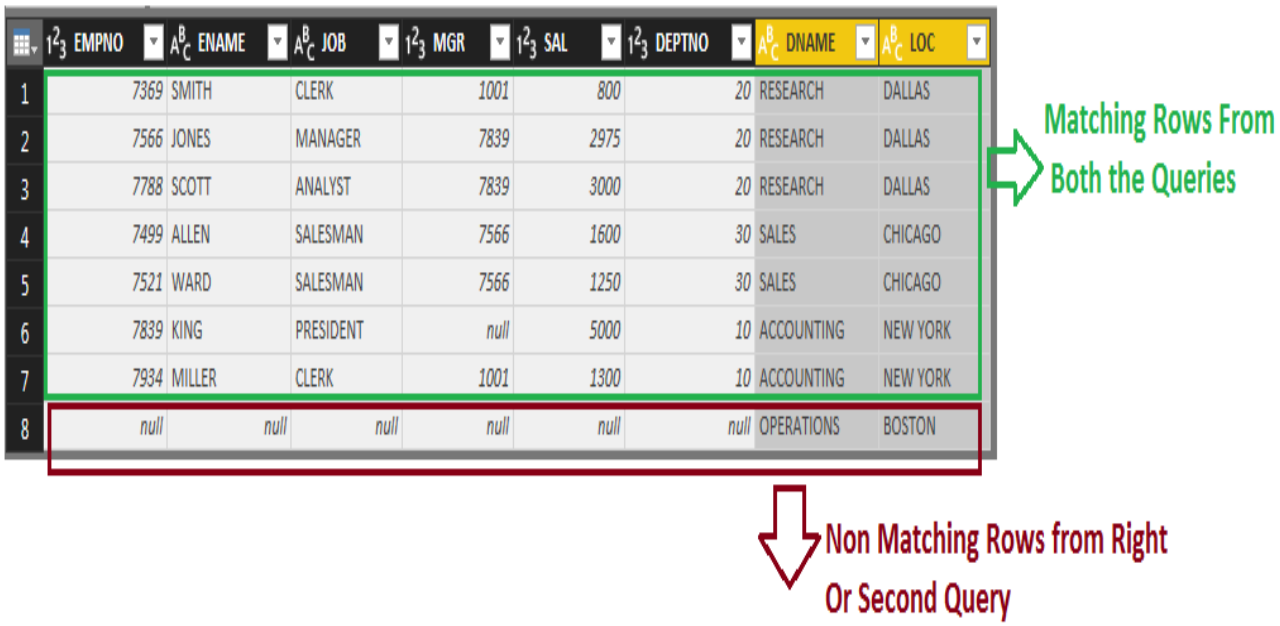
The first type of Join/Merge is Left Outer. All records from this query (LEFT or FIRST) will be showed in the result set plus their matching rows in the right (or second Query). This type of join is the default type. If you don't specify the Join Kind, it will be always Left Outer. The result set of EMP and DEPT with Join Kind Left Outer is shown below. We can see that all the Rows from EMP table are shown in the result. We don't have the DEPTNO=50 in DEPT table still it is showing in the merged query result as we select **Left Outer** Join Kind.



	EMPNO	ENAME	JOB	MGR	SAL	DEPTNO	DNAME	LOC
1	7369	SMITH	CLERK	1001	800	20	RESEARCH	DALLAS
2	7566	JONES	MANAGER	7839	2975	20	RESEARCH	DALLAS
3	7788	SCOTT	ANALYST	7839	3000	20	RESEARCH	DALLAS
4	7499	ALLEN	SALESMAN	7566	1600	30	SALES	CHICAGO
5	7521	WARD	SALESMAN	7566	1250	30	SALES	CHICAGO
6	7839	KING	PRESIDENT	null	5000	10	ACCOUNTING	NEW YORK
7	7934	MILLER	CLERK	1001	1300	10	ACCOUNTING	NEW YORK
8	1001	SUNIL	TRAINER	7839	10000	50	null	null

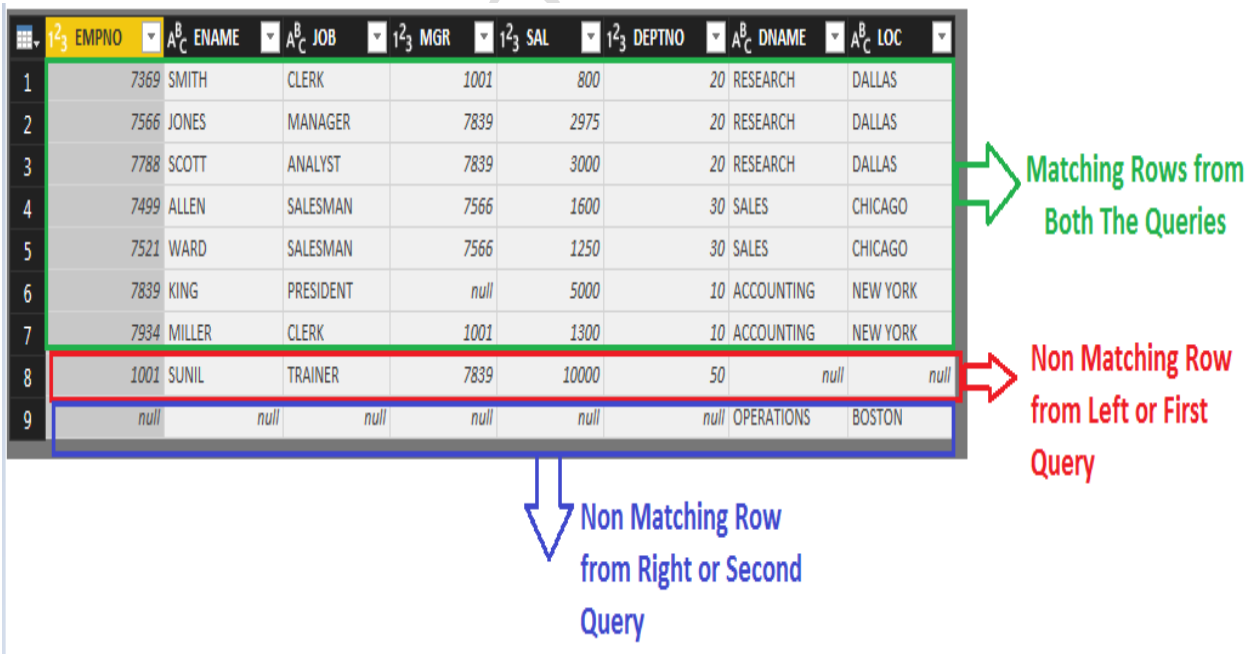
Right Outer (all rows from second, matching from first)

With this type of Join, you get all rows from the RIGHT (or second) Query, with their matching rows from left (or first Query). We can see that all the Rows from DEPT Query are shown in the result along with matching rows from Both the Queries.



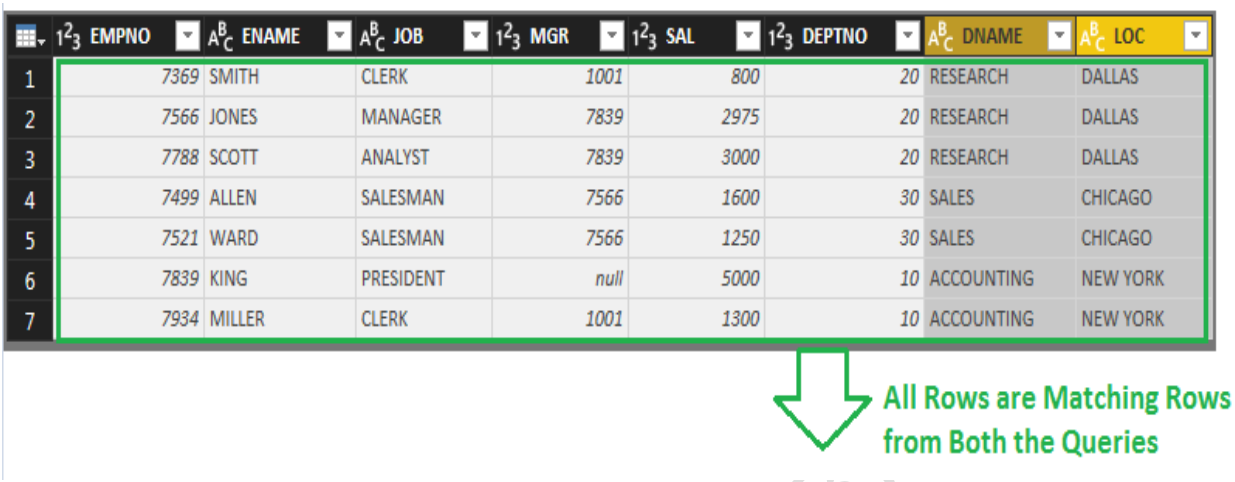
Full Outer (all rows from both)

This Join Kind will return all rows from both Queries (matching and non-matching). You will have all non-matching rows from first Query, and all non-matching rows from the second Query, and all matching rows from both the Queries. In the below image you can see Matching Rows from both the Queries in Green Box, Non matching rows from First or Left Query in Red Box and Non matching rows from Second or Right Query in Blue Box.



Inner (only matching rows)

This Join Kind will only return matching rows as shown below. You will not have any record with null values (because these records generate as a result of not matching).

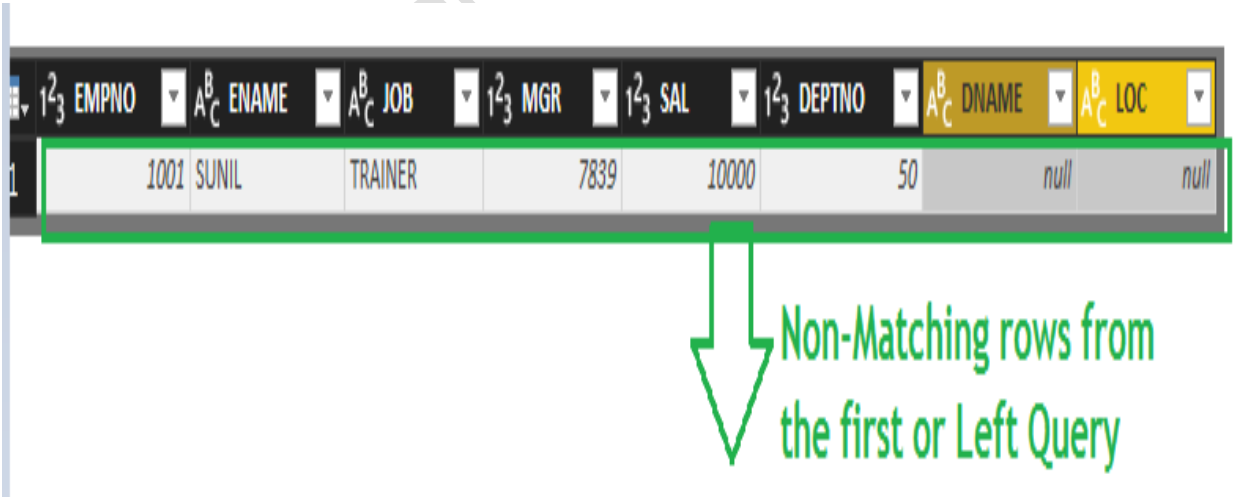


	EMPNO	ENAME	JOB	MGR	SAL	DEPTNO	DNAME	LOC
1	7369	SMITH	CLERK	1001	800	20	RESEARCH	DALLAS
2	7566	JONES	MANAGER	7839	2975	20	RESEARCH	DALLAS
3	7788	SCOTT	ANALYST	7839	3000	20	RESEARCH	DALLAS
4	7499	ALLEN	SALESMAN	7566	1600	30	SALES	CHICAGO
5	7521	WARD	SALESMAN	7566	1250	30	SALES	CHICAGO
6	7839	KING	PRESIDENT	null	5000	10	ACCOUNTING	NEW YORK
7	7934	MILLER	CLERK	1001	1300	10	ACCOUNTING	NEW YORK

Let Anti (rows only in first)

If you are only interested in rows from the LEFT (first) Query, then this is the option to select. This means rows that are in the First or Left Query and DO NOT match with the Second or Right Query. So this Join Kind returns only non-matching rows from the first or Left Query. With Anti options you always get null for the second data set, because these rows don't exist there. Anti-options are good for finding rows that exists in one Query but not in the other one.

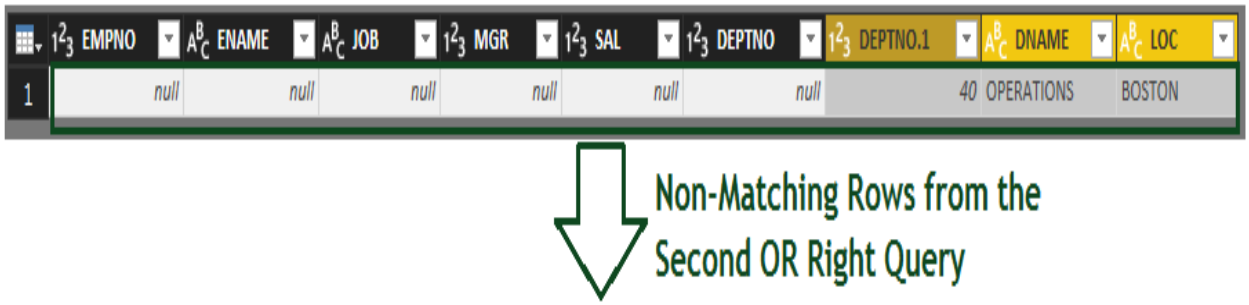
This Join Kind will find the only one row that exists in the EMP Query and does not match with any of rows in the DEPT Query.



	EMPNO	ENAME	JOB	MGR	SAL	DEPTNO	DNAME	LOC
1	1001	SUNIL	TRAINER	7839	10000	50	null	null

Right Anti (rows only in second)

This Join Kind will give you only non-matching rows, this time from the Second (Right) Query. You can find out what rows in the Right or Second Query are not matching with the Left or First Query.



Cartesian Join or Cross Join

By Default, in Power BI we have 6 types of joins and in that we don't have on more type of join that is Cross Join or Cartesian Join. To get the cross-join output we have a work around that we need to add a custom column for both the queries and populate that column with a unique value.