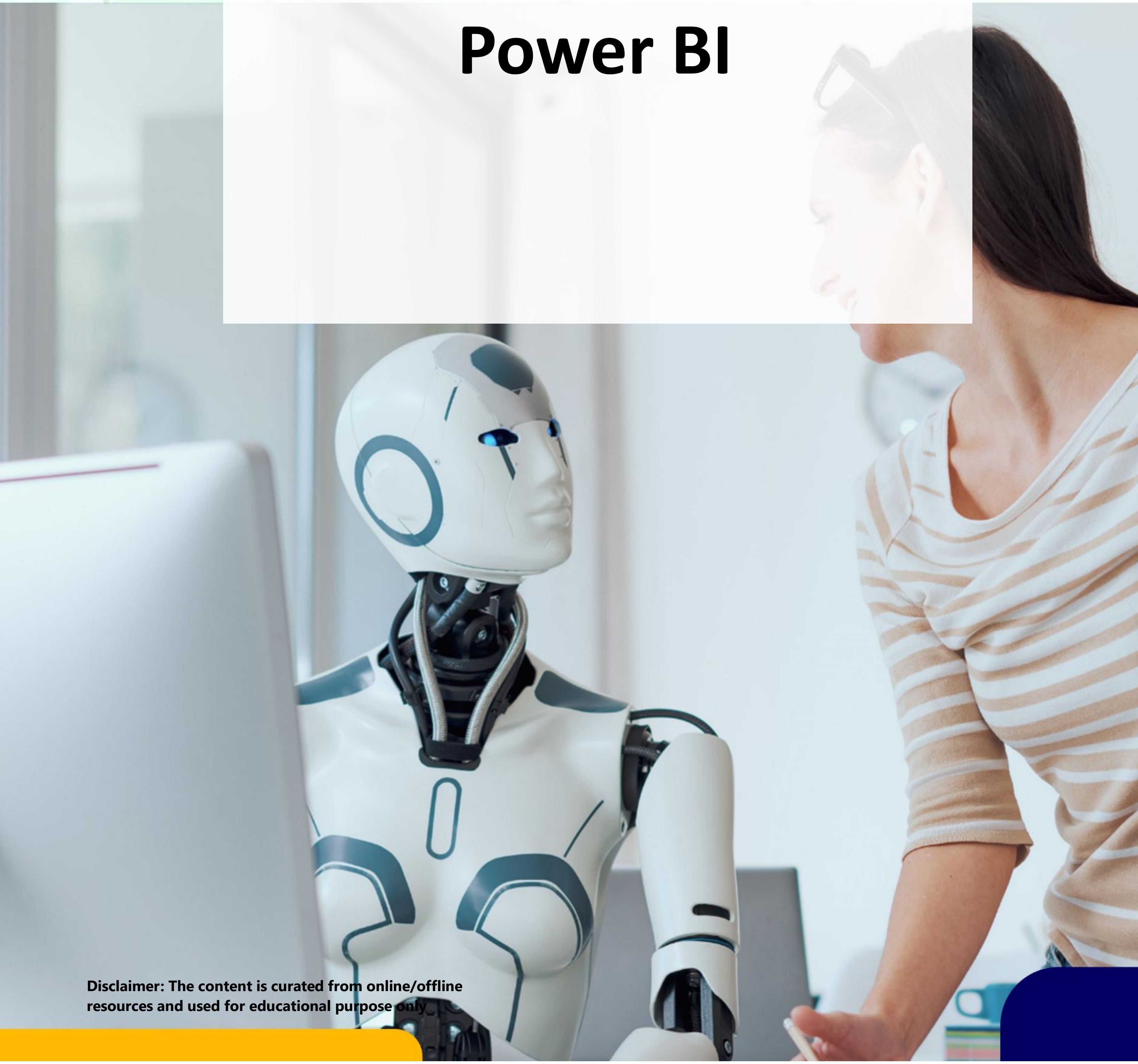
Microsoft

Proudly supported by SAP

edunet foundation

Lab Manual

Power BI



Disclaimer: The content is curated from online/offline resources and used for educational purpose only

Power BI installation

Lab 26:

Objective:

The objective of this lab is to utilize ChatGPT to generate Python code that creates an animation of an elastic ball dropping from a height and bouncing when it touches the ground using the Pygame library. Additionally, this lab aims to demonstrate how to execute the generated code to visualize the animation.

Equipment Required:

The following list provides the minimum requirements to run Power BI Desktop:

Important

- Power BI Desktop is no longer supported on Windows 7.
- Windows 8.1 or Windows Server 2012 R2 or later.
- .NET 4.7.2 or later.
- Microsoft Edge browser (Internet Explorer is no longer supported)
- Memory (RAM): At least 2 GB available, 4 GB or more recommended.
- Display: At least 1440x900 or 1600x900 (16:9) required. Lower resolutions such as 1024x768 or 1280x800 aren't supported because some controls (such as closing the startup screens) display beyond those resolutions.
- Windows display settings: If you set your display to change the size of text, apps, and other items to more than 100%, you won't see some dialogs that you must interact with to continue using Power BI Desktop. If you encounter this issue, check your display settings in Windows by going to Settings > System > Display, and use the slider to return display settings to 100%.
- CPU: 1 gigahertz (GHz) 64-bit (x64) processor or better recommended.
- WebView2: If WebView2 wasn't automatically installed with Power BI Desktop or if it was uninstalled, download and run the installer for WebView2.

Note

We recommend using a client version of Windows, such as Windows 10, instead of Windows Server. Power BI Desktop doesn't support Internet Explorer Enhanced Security Configuration because it will stop Power BI Desktop from signing in to the Power BI service.

Prerequisites:

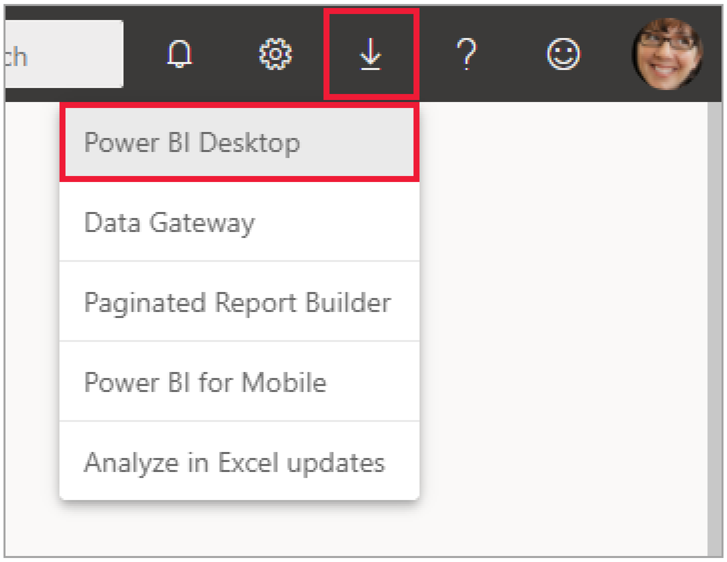
1. Internet skill to access the Power BI site
2. Computer OS skill to install the software

Problem Statement:

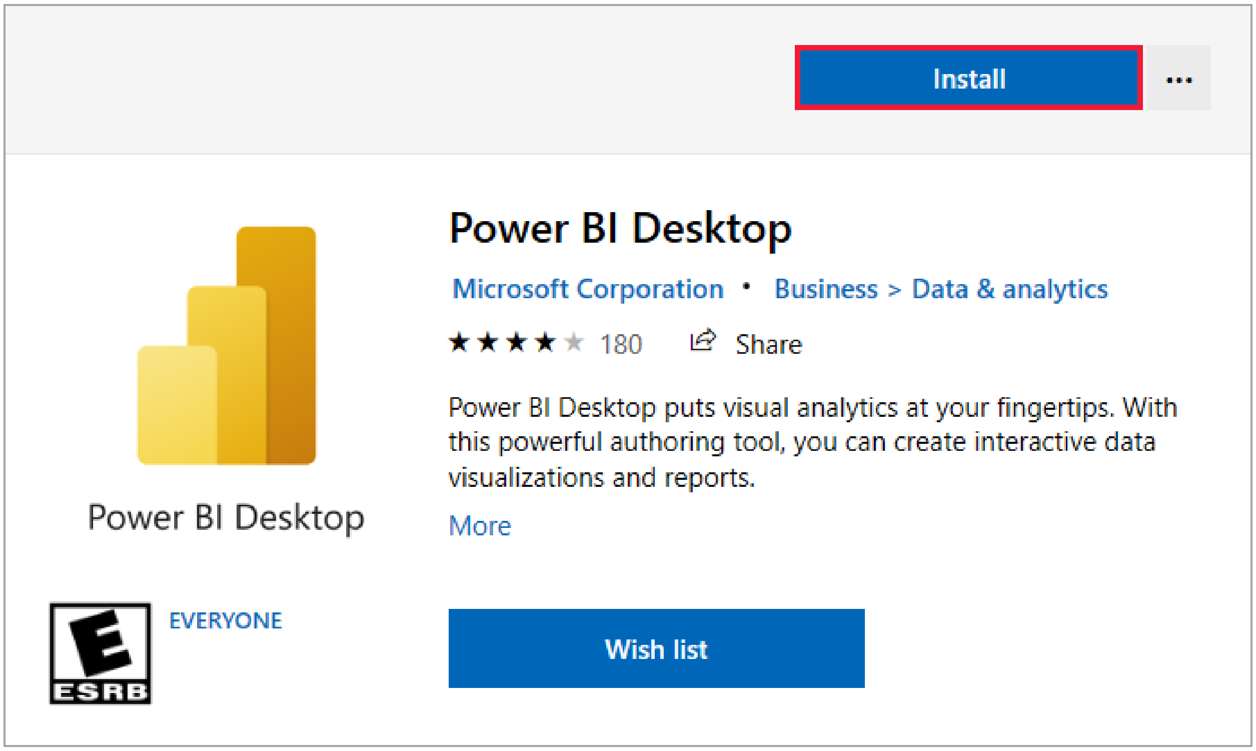
To Install the power BI for handling the data.

Install as an app from

1. Use one of the following options to open the **Power BI Desktop** page of the Microsoft Store:
   - ○ Open a browser and go directly to the Power BI Desktop page of the Microsoft Store.
   - ○ From the Power BI service, in the upper right corner, select the **Download** icon and then choose **Power BI Desktop**.



   - ○ Go to the Power BI Desktop product page, and then select **Download Free**.
2. After you've landed on the **Power BI Desktop** page of the Microsoft Store, select **Install**.



Power BI Desktop puts visual analytics at your fingertips. With this powerful authoring tool, you can create interactive data visualizations and reports.

There are a few advantages to getting Power BI Desktop from the Microsoft Store:

- **Automatic updates**: Windows downloads the latest version automatically in the background as soon as it's available, so your version is always up to date.
- **Smaller downloads**: Microsoft Store ensures only components that changed in each update are downloaded to your computer, resulting in smaller downloads for each update.
- **Admin privilege isn't required**: When you download the package directly and install it, you must be an administrator for the installation to complete successfully. If you get Power BI Desktop from the Microsoft Store, admin privilege isn't required.
- **IT roll-out enabled**: Through the Microsoft Store for Business, you can more easily deploy, or *roll out*, Power BI Desktop to everyone in your organization.
- **Language detection**: The Microsoft Store version includes all supported languages, and checks the language used on your computer each time it's launched. This language support also affects the localization of models created in Power BI Desktop. For example, built-in date hierarchies match the language that Power BI Desktop uses when the *.pbix* file is created.

The following considerations and limitations apply when you install Power BI Desktop from the Microsoft Store:

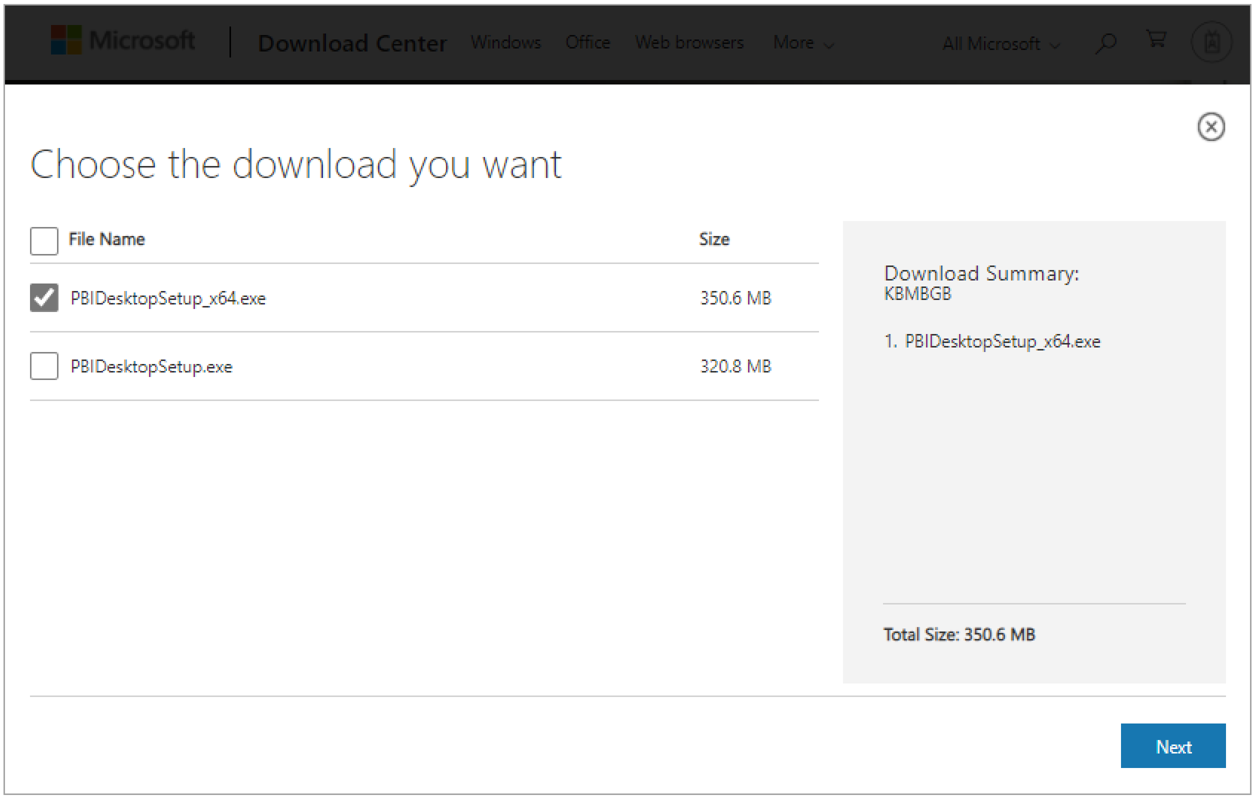
- If you use the SAP connector, you might need to move your SAP driver files to the *Windows\System32* folder.
- Installing Power BI Desktop from the Microsoft Store doesn't copy user settings from the *.exe* version. You might have to reconnect to your recent data sources and reenter your credentials.

Note

The Power BI Report Server version of Power BI Desktop is a separate and different installation from the versions discussed in this article. For information about the Report Server version of Power BI Desktop, see **Create a Power BI report for Power BI Report Server**.

Download Power BI Desktop directly

To download the Power BI Desktop executable from the Download Center, select **Download** from the Download Center page. Then specify the 32-bit or 64-bit installation file to download.



Choose the download you want

□ File Name	Size
☑ PBIDesktopSetup_x64.exe	350.6 MB
□ PBIDesktopSetup.exe	320.8 MB

Download Summary:
KBMBGB

1. PBIDesktopSetup_x64.exe

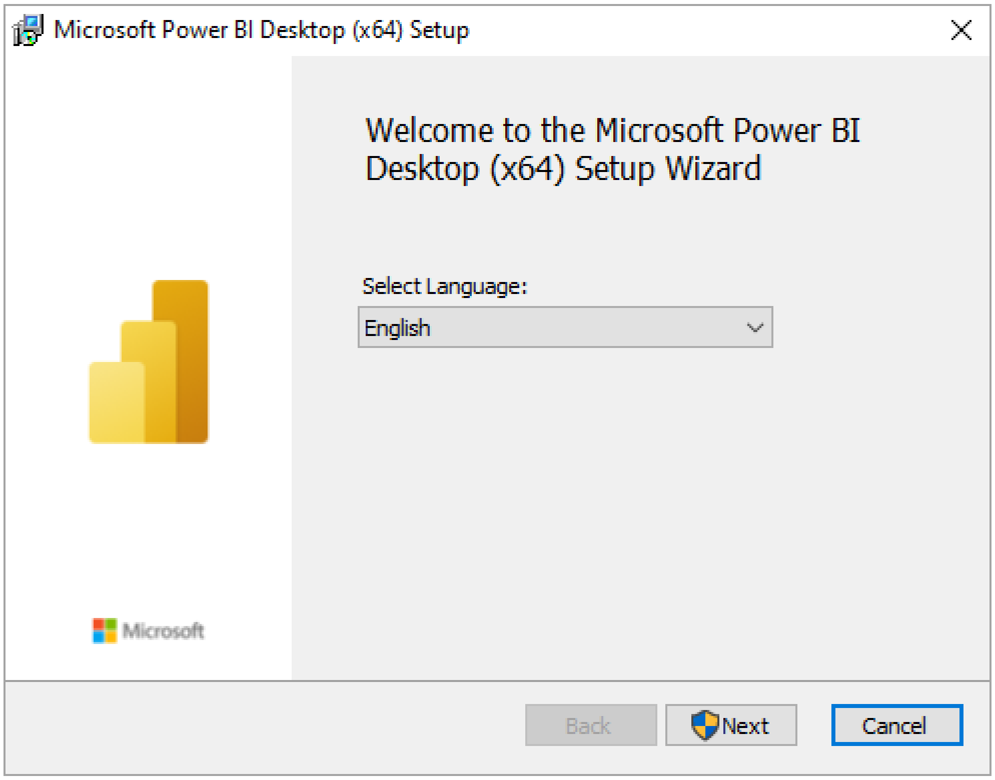
Total Size: 350.6 MB

Install Power BI Desktop after download

You're prompted to run the installation file after you finish downloading it.

Power BI Desktop ships as a single *.exe* installation package that contains all supported languages, with separate *.exe* files for the 32-bit and 64-bit versions. The *.msi* packages are no longer available. You need the executable for installation. This approach makes distribution, updates, and installation easier and more convenient, especially for administrators. You can also use command-line parameters to customize the installation process, as described in Using command-line options during installation.

After you launch the installation package, Power BI Desktop installs as an application and runs on your desktop.



Microsoft Power BI Desktop (x64) Setup

Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard

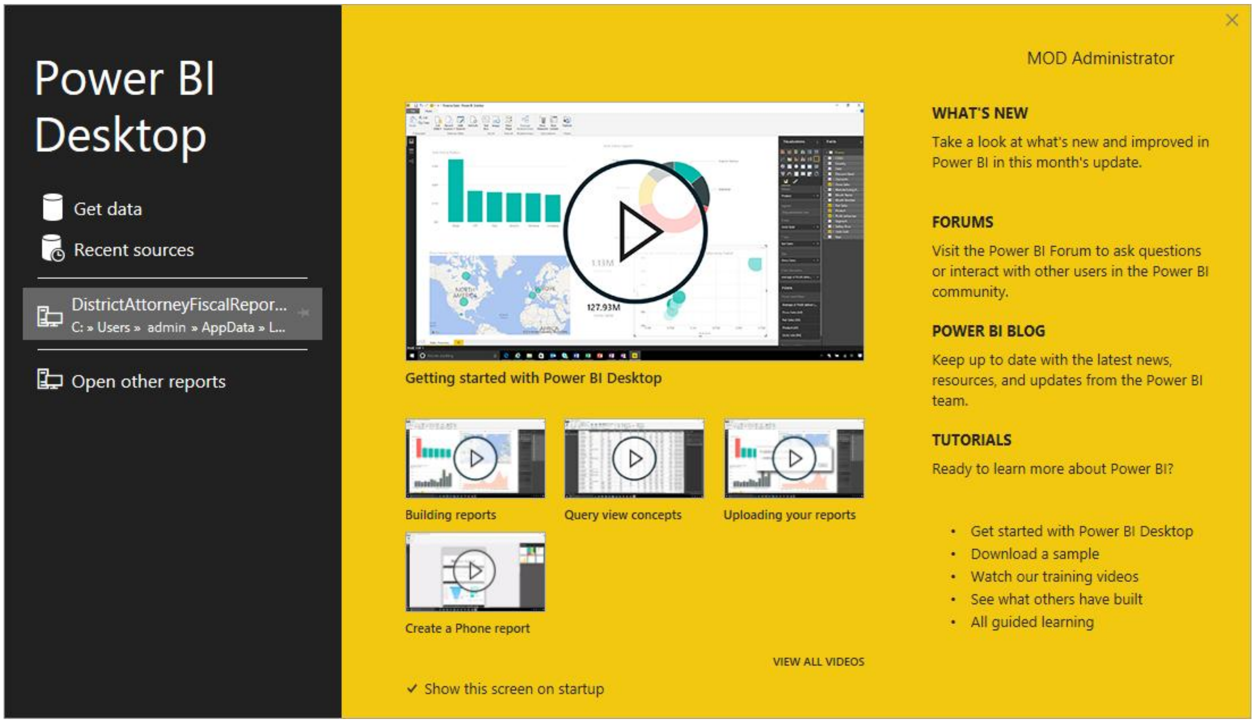
Select Language: English

Note

Installing the deprecated *msi* version and the Microsoft Store version of Power BI Desktop on the same computer. Sometimes referred to as a *side-by-side* installation isn't supported. Manually uninstall Power BI Desktop before you download it from the Microsoft Store.

Use Power BI Desktop

When you launch Power BI Desktop, a welcome screen appears.



WHAT'S NEW

Take a look at what's new and improved in Power BI in this month's update.

FORUMS

Visit the Power BI Forum to ask questions or interact with other users in the Power BI community.

POWER BI BLOG

Keep up to date with the latest news, resources, and updates from the Power BI team.

TUTORIALS

Ready to learn more about Power BI?

- Get started with Power BI Desktop
- Download a sample
- Watch our training videos
- See what others have built
- All guided learning

When you launch Power BI Desktop for the first time, if the installation isn't an upgrade, you're prompted to fill out a form or sign in to the Power BI service before you can continue.

After that, you can begin creating data models or reports, and share them with others on the Power BI service. Check out the Next steps section for links to guides to help you get started using Power BI Desktop.

Power Query for Data Transformation in Real time Data

Objective:

- To Consolidate Data: Combine sales data from multiple sources into a single, cohesive dataset.
- To Clean Data: Remove duplicates, handle missing values, and correct errors in the data.
- To Transform Data: Reshape the data to fit the analysis requirements.

Equipment Required:

The following list provides the minimum requirements to run Power BI Desktop:

Important

- Power BI Desktop is no longer supported on Windows 7.
- Windows 8.1 or Windows Server 2012 R2 or later.
- .NET 4.7.2 or later.
- Microsoft Edge browser (Internet Explorer is no longer supported)
- Memory (RAM): At least 2 GB available, 4 GB or more recommended.
- Display: At least 1440x900 or 1600x900 (16:9) required. Lower resolutions such as 1024x768 or 1280x800 aren't supported because some controls (such as closing the startup screens) display beyond those resolutions.
- Windows display settings: If you set your display to change the size of text, apps, and other items to more than 100%, you won't see some dialogs that you must interact with to continue using Power BI Desktop. If you encounter this issue, check your display settings in Windows by going to Settings > System > Display, and use the slider to return display settings to 100%.
- CPU: 1 gigahertz (GHz) 64-bit (x64) processor or better recommended.
- WebView2: If WebView2 wasn't automatically installed with Power BI Desktop or if it was uninstalled, download and run the installer for WebView2.

Note

We recommend using a client version of Windows, such as Windows 10, instead of Windows Server. Power BI Desktop doesn't support Internet Explorer Enhanced Security Configuration because it will stop Power BI Desktop from signing in to the Power BI service.

Prerequisites:

3. Internet skill to access the Power BI site
4. Computer OS skill to install the software

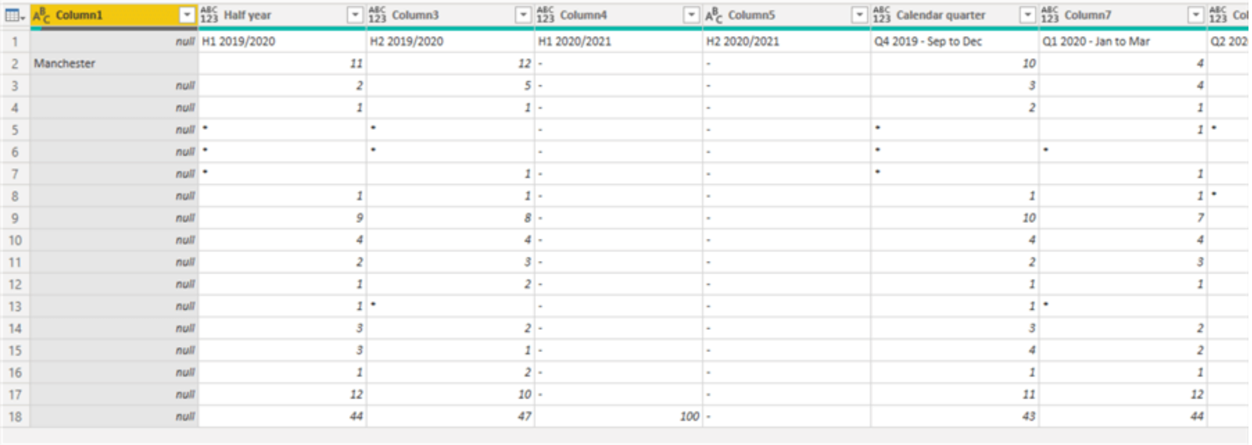
Problem Statement:

Sometimes we get data from data sources (SQL Server, Excel, CSV file, OData, etc.) in formats which are not in a format that can be easily transformed using simple methods in Microsoft Power BI. This might be due to the complexity and structure of the dataset. Thus, it becomes a hectic task to write simple DAX calculations from the datasets or even use it in any form in Power BI development.

Solution:
To demonstrate this process, we are going to transform the dataset as shown in the worksheet below from the current format into an easy to read format which would then make writing calculations and performing data analysis on it easier.

STEP 1: Import the datasets into Power BI

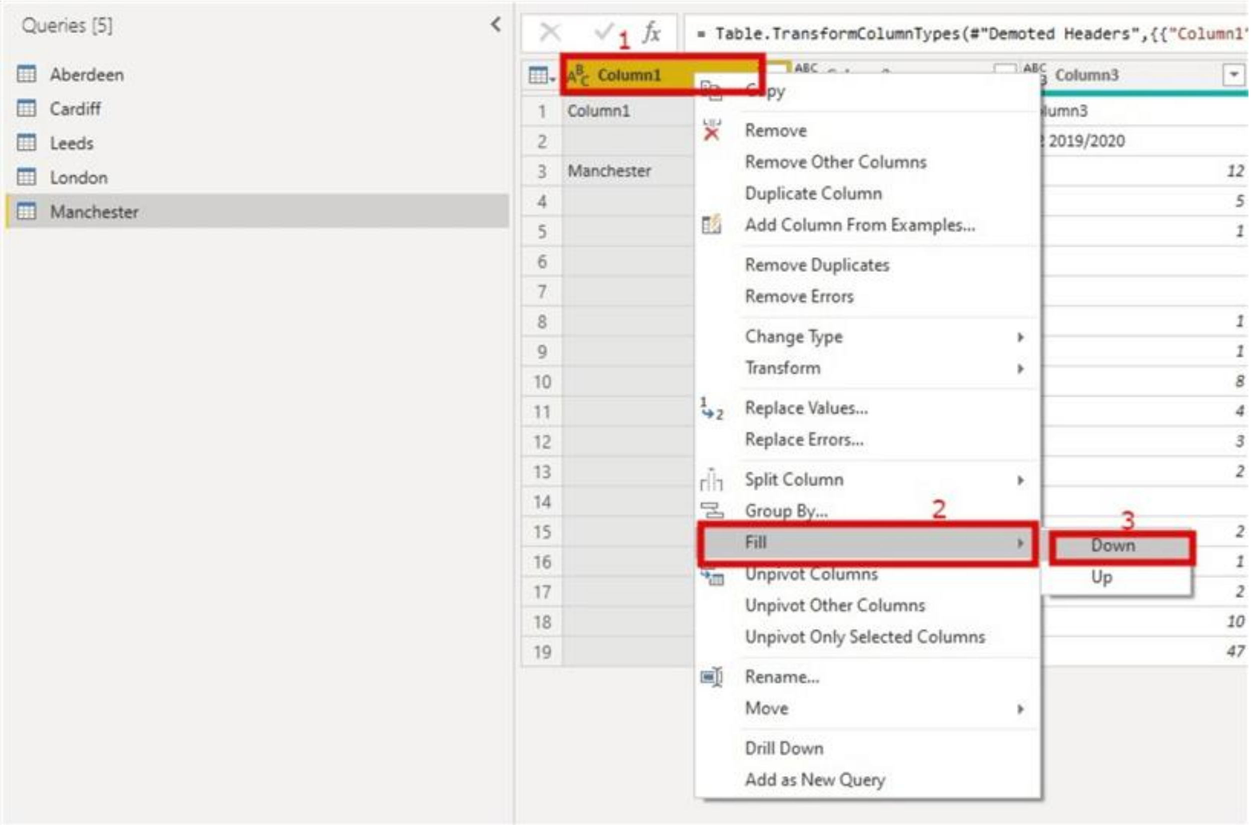
Once data importing is complete, you should be able to see the dataset like the screenshot below.



	Column1	Half year	Column3	Column4	Column5	Calendar quarter	Column7	Col
1	null	H1 2019/2020	H2 2019/2020	H1 2020/2021	H2 2020/2021	Q4 2019 - Sep to Dec	Q1 2020 - Jan to Mar	Q2 202
2	Manchester	11	12	-	-	10	4	
3	null	2	5	-	-	3	4	
4	null	1	1	-	-	2	1	
5	null	•	•	-	-	•	1	•
6	null	•	•	-	-	•	•	
7	null	•	1	-	-	•	1	
8	null	1	1	-	-	1	1	•
9	null	9	8	-	-	10	7	
10	null	4	4	-	-	4	4	
11	null	2	3	-	-	2	3	
12	null	1	2	-	-	1	1	
13	null	1	•	-	-	1	•	
14	null	3	2	-	-	3	2	
15	null	3	1	-	-	4	2	
16	null	1	2	-	-	1	1	
17	null	12	10	-	-	11	12	
18	null	44	47	100	-	43	44	

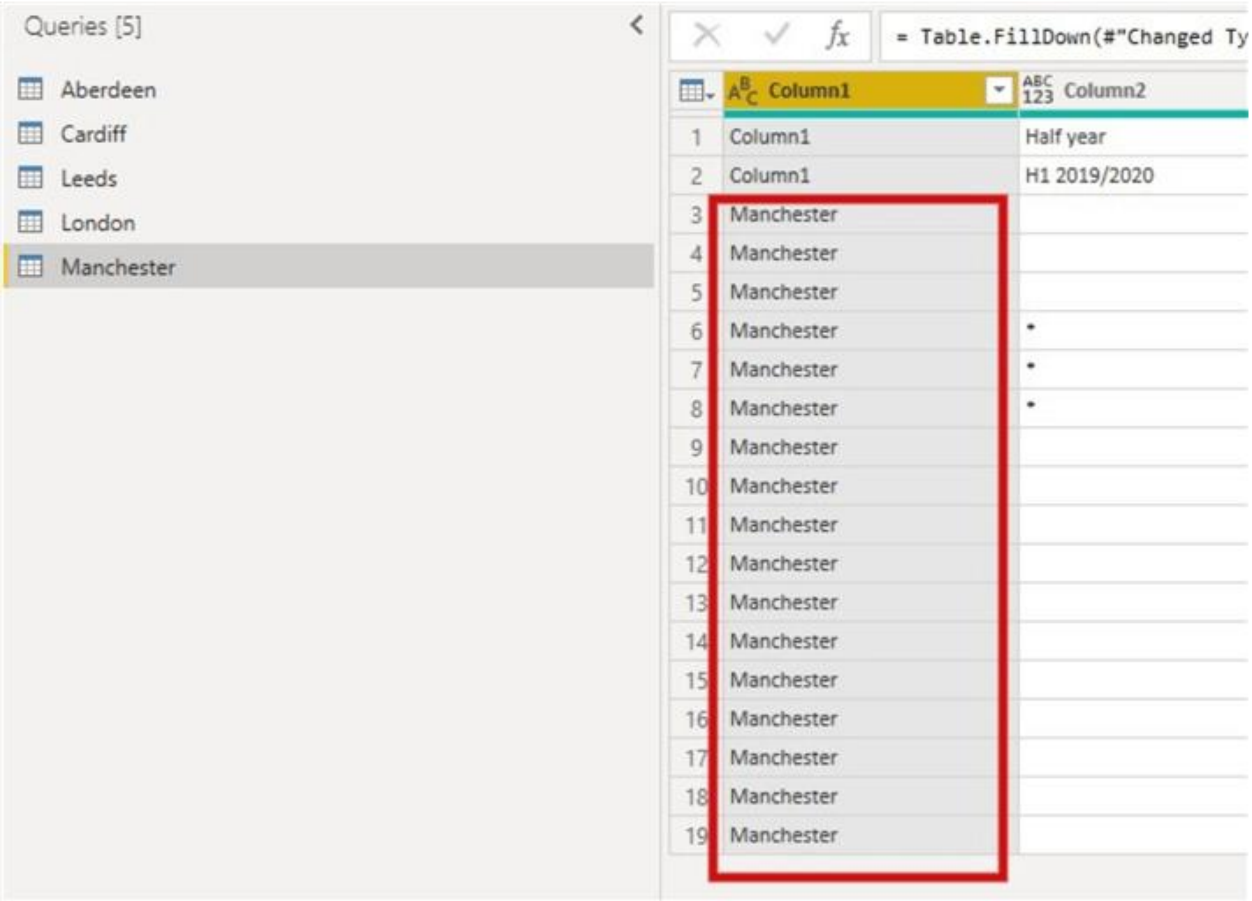
STEP 2: Fill Down the City names

As can be seen in the diagram above, the city name has a lot of "nulls" which would make the table a bit messy later, so we need to do a fill down to exclude the nulls. To do this, we need to right click on the "Column1" and then click on "Fill" and then "Down" as shown in the diagram below.



```
= Table.TransformColumnTypes(#"Demoted Headers",{{"Column1'
```

This would ensure each row with "nulls" under the city name is populated with the city name.



Queries [5]

- Aberdeen
- Cardiff
- Leeds
- London
- Manchester

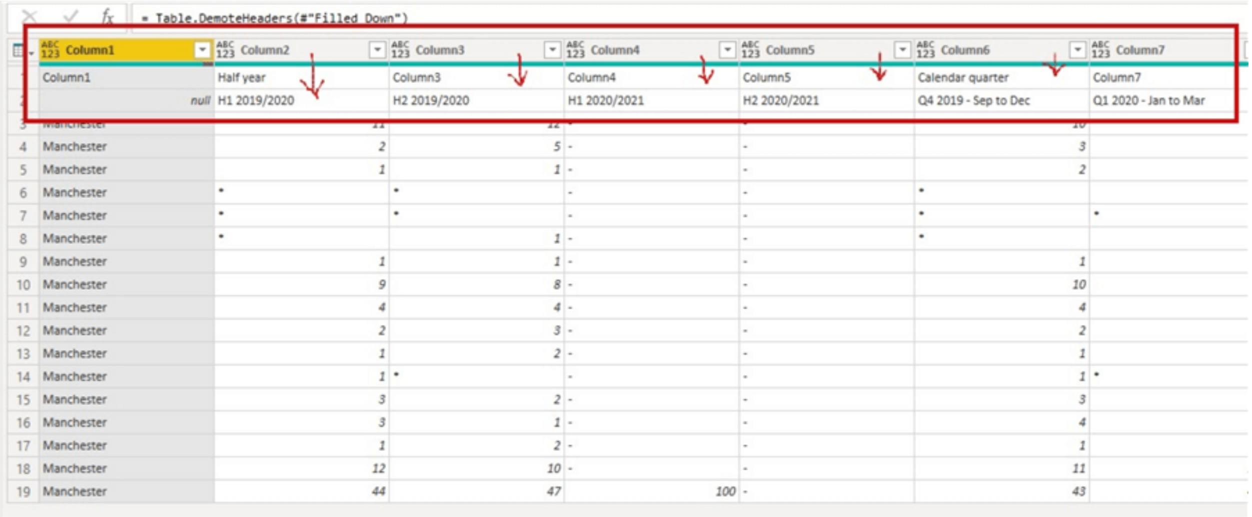
```
= Table.FillDown(#"Changed Ty
```

	Column1	Column2
1	Column1	Half year
2	Column1	H1 2019/2020
3	Manchester	
4	Manchester	
5	Manchester	
6	Manchester	•
7	Manchester	•
8	Manchester	•
9	Manchester	
10	Manchester	
11	Manchester	
12	Manchester	
13	Manchester	
14	Manchester	
15	Manchester	
16	Manchester	
17	Manchester	
18	Manchester	
19	Manchester	

STEP 3: Move headers into first rows

As you can see in the dataset, we have two rows that contain possible headers, we need to ensure this is in a less complex format by moving the "Half Year" and "Calendar Quarter" into columns. We also need to ensure the first-row values are in Columns too.

To do this, we first need to move the headers into first rows as seen in the diagram below. Within the "Home" ribbon click on the drop down on "Use First Rows as Headers" and select "Use Headers as First Row"

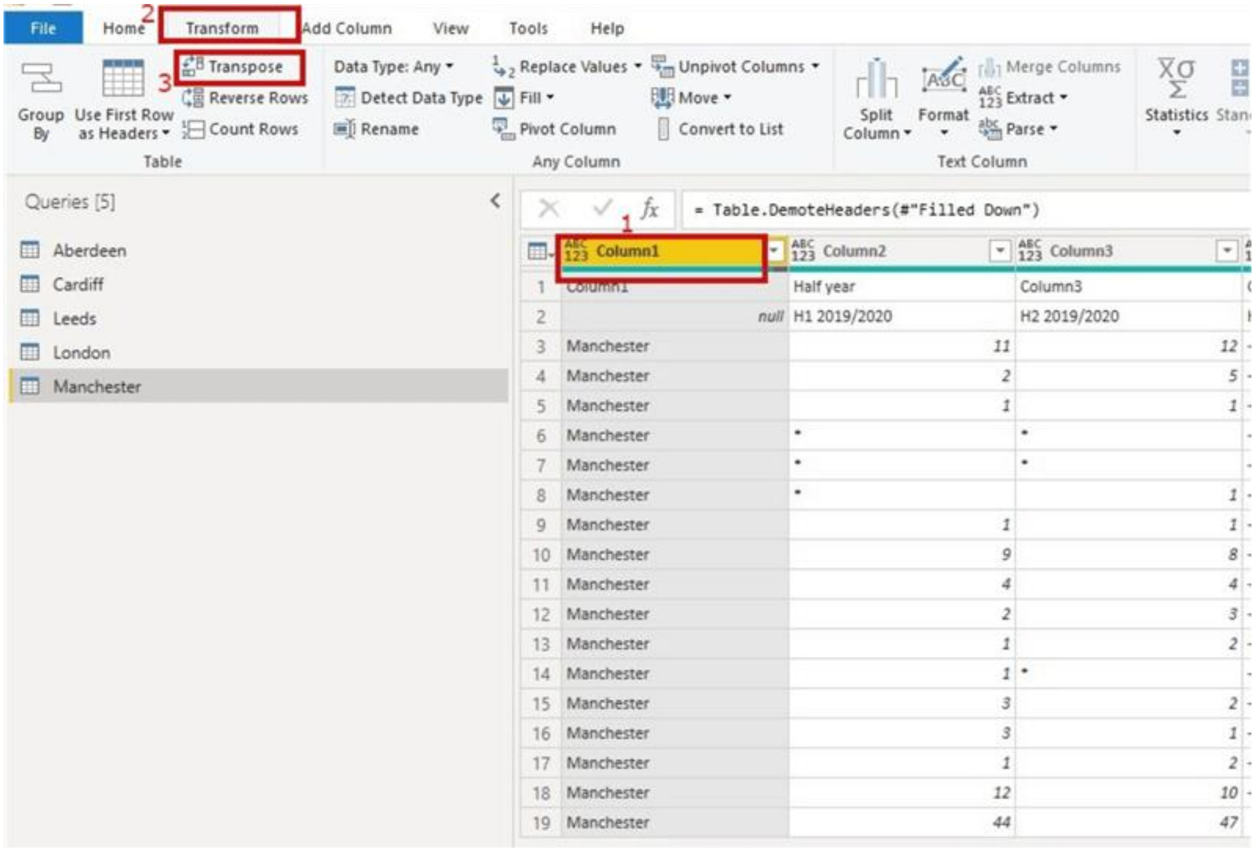


```
= Table.DemoteHeaders(#"Filled Down")
```

	Column1	Column2	Column3	Column4	Column5	Column6	Column7
1	Column1	Half year	Column3	Column4	Column5	Calendar quarter	Column7
2	null	H1 2019/2020	H2 2019/2020	H1 2020/2021	H2 2020/2021	Q4 2019 - Sep to Dec	Q1 2020 - Jan to Mar
3	Manchester	11	12	-	-	10	
4	Manchester	2	5	-	-	3	
5	Manchester	1	1	-	-	2	
6	Manchester	•	•	-	-	•	
7	Manchester	•	•	-	-	•	•
8	Manchester	•	1	-	-	•	
9	Manchester	1	1	-	-	1	
10	Manchester	9	8	-	-	10	
11	Manchester	4	4	-	-	4	
12	Manchester	2	3	-	-	2	
13	Manchester	1	2	-	-	1	
14	Manchester	1	•	-	-	1	•
15	Manchester	3	2	-	-	3	
16	Manchester	3	1	-	-	4	
17	Manchester	1	2	-	-	1	
18	Manchester	12	10	-	-	11	
19	Manchester	44	47	100	-	43	

STEP 4: Transpose the table

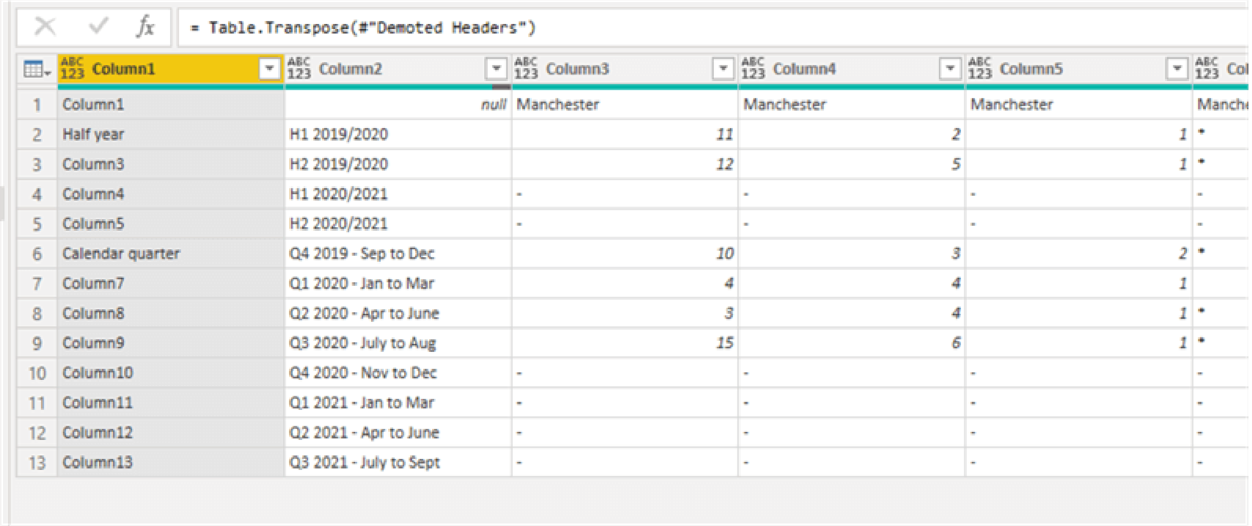
To do this click on "Column1", then click on the "Transform" tab, then click on "Transpose" as seen in the diagram below.



```
= Table.DemoteHeaders(#"Filled Down")
```

	Column1	Column2	Column3
1	Column1	Half year	Column3
2	null	H1 2019/2020	H2 2019/2020
3	Manchester	11	12
4	Manchester	2	5
5	Manchester	1	1
6	Manchester	*	*
7	Manchester	*	*
8	Manchester	*	1
9	Manchester	1	1
10	Manchester	9	8
11	Manchester	4	4
12	Manchester	2	3
13	Manchester	1	2
14	Manchester	1	*
15	Manchester	3	2
16	Manchester	3	1
17	Manchester	1	2
18	Manchester	12	10
19	Manchester	44	47

You should now be able to see the table part as seen in the snapshot below.



```
= Table.Transpose(#"Demoted Headers")
```

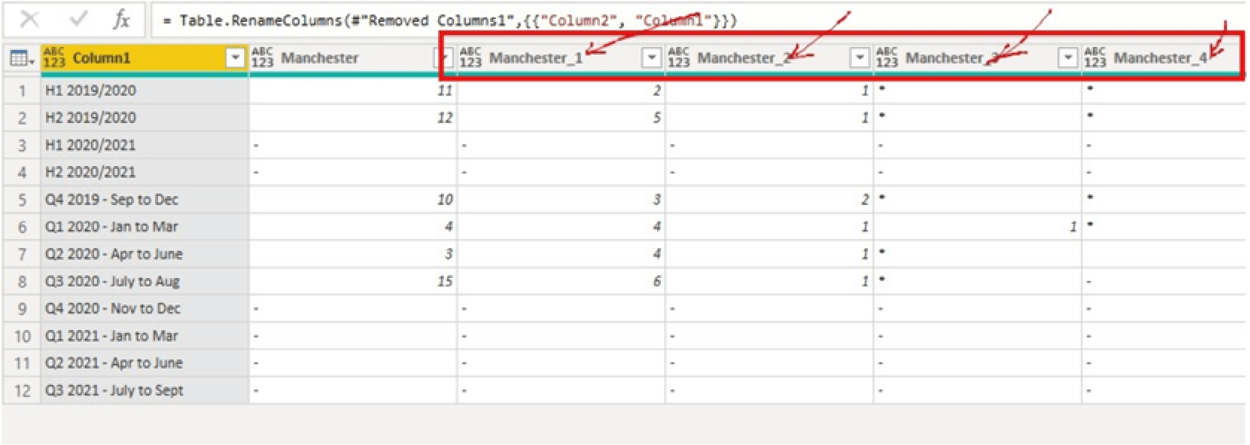
	Column1	Column2	Column3	Column4	Column5	Col
1	Column1	null	Manchester	Manchester	Manchester	Manche
2	Half year	H1 2019/2020	11	2	1	*
3	Column3	H2 2019/2020	12	5	1	*
4	Column4	H1 2020/2021	-	-	-	-
5	Column5	H2 2020/2021	-	-	-	-
6	Calendar quarter	Q4 2019 - Sep to Dec	10	3	2	*
7	Column7	Q1 2020 - Jan to Mar	4	4	1	
8	Column8	Q2 2020 - Apr to June	3	4	1	*
9	Column9	Q3 2020 - July to Aug	15	6	1	*
10	Column10	Q4 2020 - Nov to Dec	-	-	-	-
11	Column11	Q1 2021 - Jan to Mar	-	-	-	-
12	Column12	Q2 2021 - Apr to June	-	-	-	-
13	Column13	Q3 2021 - July to Sept	-	-	-	-

And since we may not need "Column1" after all, we remove columns at this point.

STEP 5: Promote first rows as headers

Next, we need to promote the first rows as headers as seen in the diagram below.

To do this, we need go to the "Home" tab ribbon and select "Use First rows as Headers".



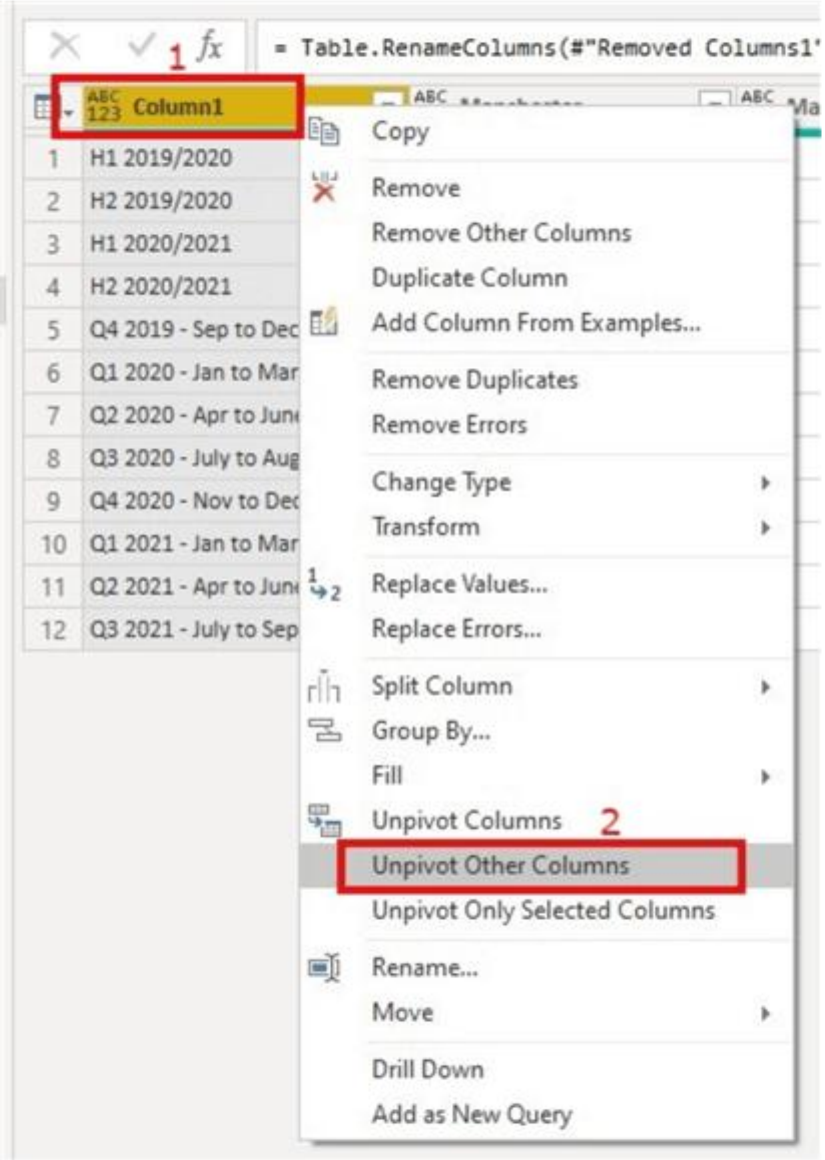
```
= Table.RenameColumns(#"Removed Columns1",{{"Column2", "Column1"}})
```

	Column1	Manchester	Manchester_1	Manchester_2	Manchester_3	Manchester_4
1	H1 2019/2020	11	2	1	*	*
2	H2 2019/2020	12	5	1	*	*
3	H1 2020/2021	-	-	-	-	-
4	H2 2020/2021	-	-	-	-	-
5	Q4 2019 - Sep to Dec	10	3	2	*	*
6	Q1 2020 - Jan to Mar	4	4	1	1	*
7	Q2 2020 - Apr to June	3	4	1	*	
8	Q3 2020 - July to Aug	15	6	1	*	-
9	Q4 2020 - Nov to Dec	-	-	-	-	-
10	Q1 2021 - Jan to Mar	-	-	-	-	-
11	Q2 2021 - Apr to June	-	-	-	-	-
12	Q3 2021 - July to Sept	-	-	-	-	-

As you can see, there is some of the column headers city names having an attached underscore and a number. We will deal with this later.

STEP 6: Unpivot columns

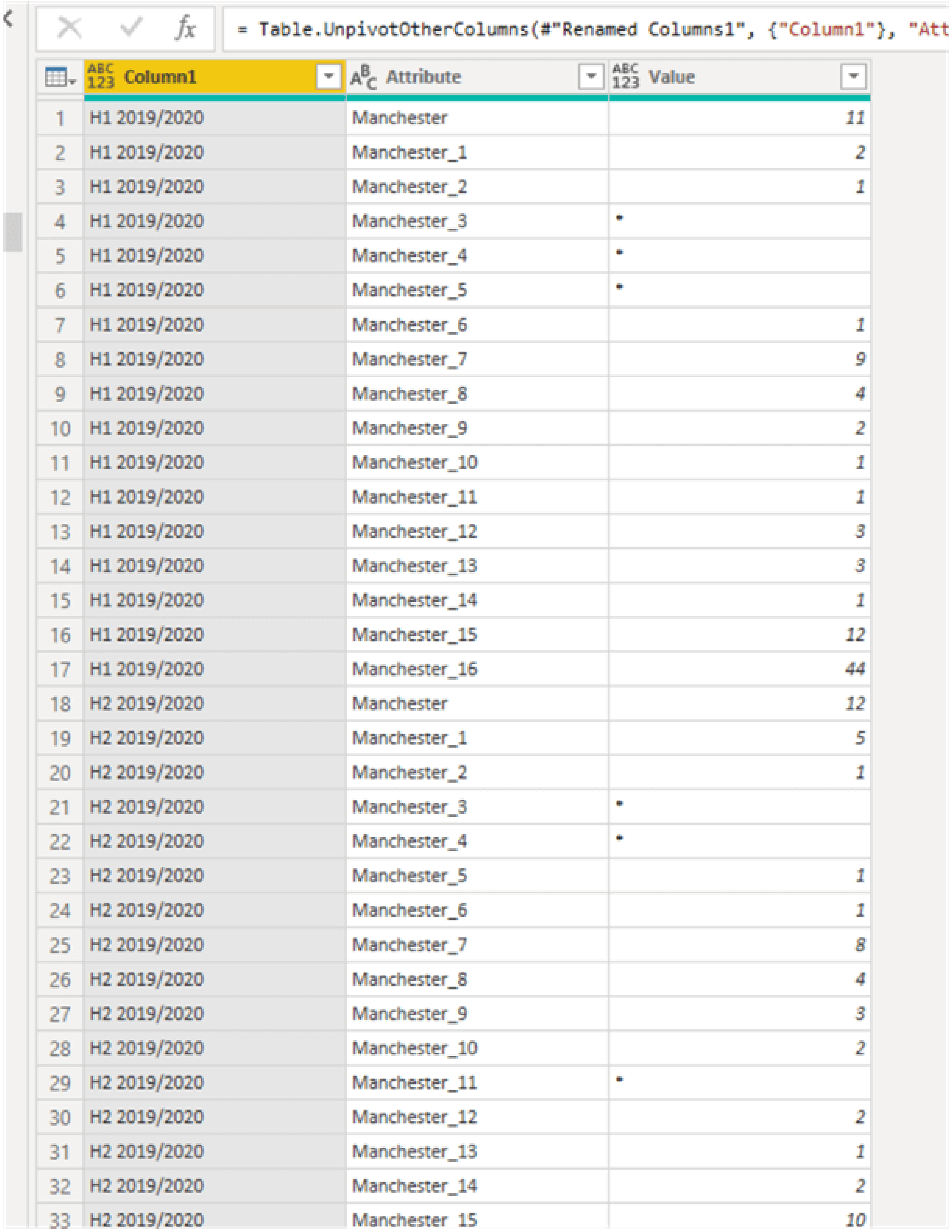
To do this, we need to right click on "Column1" and then select the "Unpivot Other Columns" as seen in the diagram below. Note that this column doesn’t contain numerical values.



```
= Table.RenameColumns(#"Removed Columns1'
```

	Column1
1	H1 2019/2020
2	H2 2019/2020
3	H1 2020/2021
4	H2 2020/2021
5	Q4 2019 - Sep to Dec
6	Q1 2020 - Jan to Mar
7	Q2 2020 - Apr to June
8	Q3 2020 - July to Aug
9	Q4 2020 - Nov to Dec
10	Q1 2021 - Jan to Mar
11	Q2 2021 - Apr to June
12	Q3 2021 - July to Sep

This should transform the table into a more simplified three-column table as seen in the diagram below.



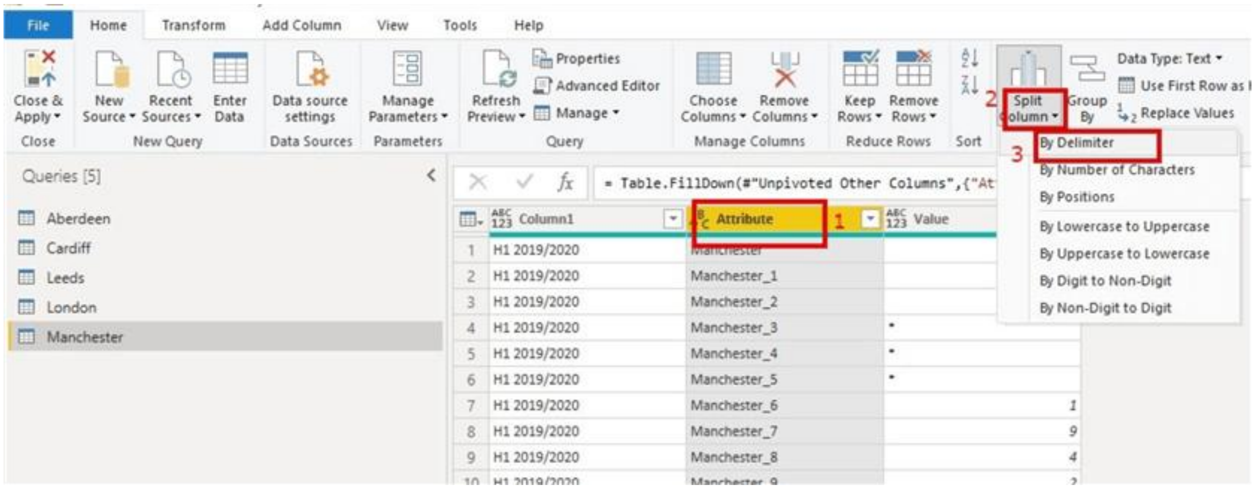
```
= Table.UnpivotOtherColumns(#"Renamed Columns1", {"Column1"}, "Att
```

	Column1	Attribute	Value
1	H1 2019/2020	Manchester	11
2	H1 2019/2020	Manchester_1	2
3	H1 2019/2020	Manchester_2	1
4	H1 2019/2020	Manchester_3	•
5	H1 2019/2020	Manchester_4	•
6	H1 2019/2020	Manchester_5	•
7	H1 2019/2020	Manchester_6	1
8	H1 2019/2020	Manchester_7	9
9	H1 2019/2020	Manchester_8	4
10	H1 2019/2020	Manchester_9	2
11	H1 2019/2020	Manchester_10	1
12	H1 2019/2020	Manchester_11	1
13	H1 2019/2020	Manchester_12	3
14	H1 2019/2020	Manchester_13	3
15	H1 2019/2020	Manchester_14	1
16	H1 2019/2020	Manchester_15	12
17	H1 2019/2020	Manchester_16	44
18	H2 2019/2020	Manchester	12
19	H2 2019/2020	Manchester_1	5
20	H2 2019/2020	Manchester_2	1
21	H2 2019/2020	Manchester_3	•
22	H2 2019/2020	Manchester_4	•
23	H2 2019/2020	Manchester_5	1
24	H2 2019/2020	Manchester_6	1
25	H2 2019/2020	Manchester_7	8
26	H2 2019/2020	Manchester_8	4
27	H2 2019/2020	Manchester_9	3
28	H2 2019/2020	Manchester_10	2
29	H2 2019/2020	Manchester_11	•
30	H2 2019/2020	Manchester_12	2
31	H2 2019/2020	Manchester_13	1
32	H2 2019/2020	Manchester_14	2
33	H2 2019/2020	Manchester_15	10

STEP 7: Cleanse the City Name Column

The column holding city names is currently having some values concatenated with underscore and a number as seen in the "Attribute" column above. We need to eliminate the underscore and number from the City Name, just leaving "Manchester" alone in this case.

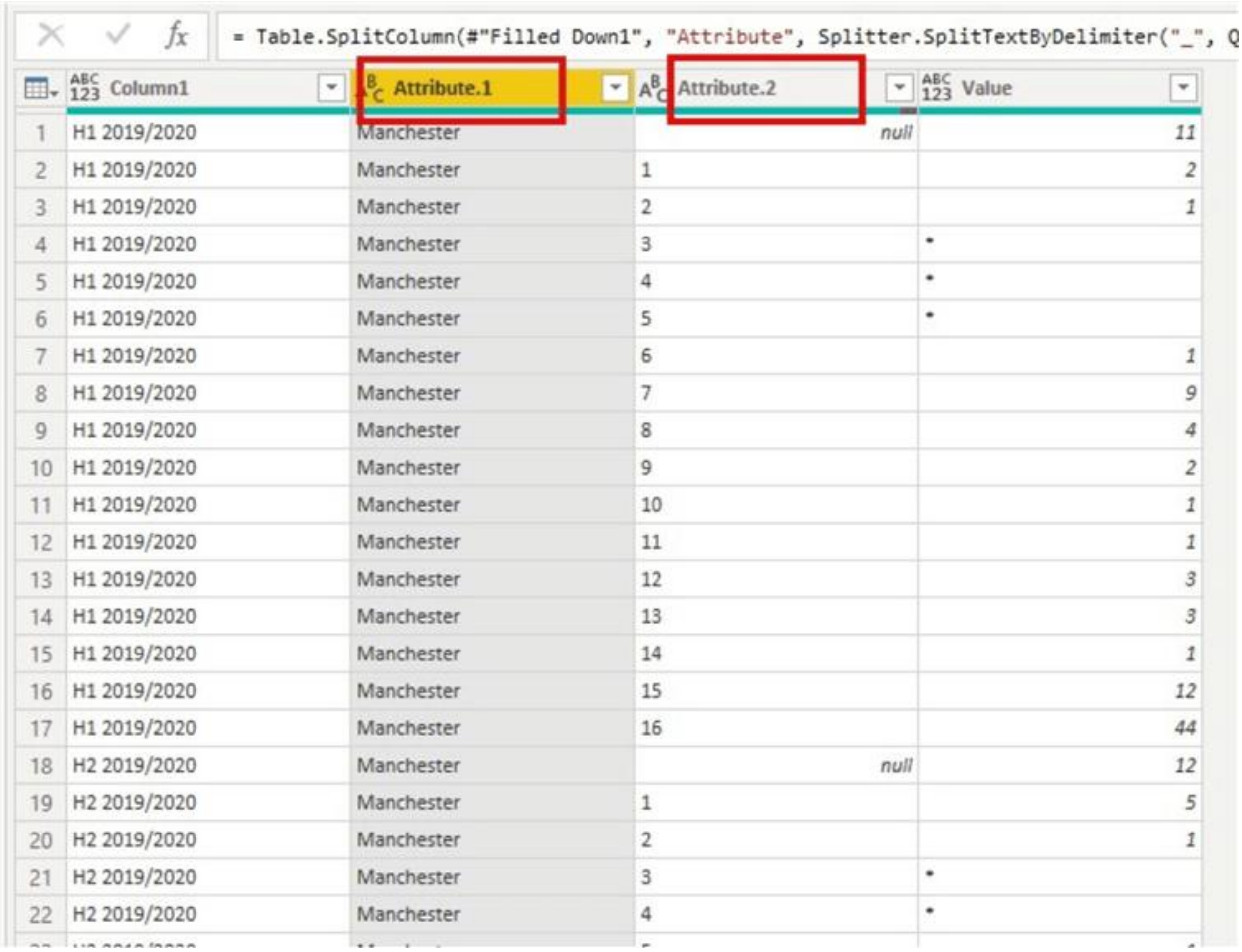
To do this, we need to click on the "Attribute" column, then on the "Home" tab click on the dropdown on "Split Column", then select "By Delimiter" as shown in the diagram below.



```
= Table.FillDown(#"Unpivoted Other Columns",{"At
```

	Column1	Attribute	Value
1	H1 2019/2020	Manchester	
2	H1 2019/2020	Manchester_1	
3	H1 2019/2020	Manchester_2	
4	H1 2019/2020	Manchester_3	•
5	H1 2019/2020	Manchester_4	•
6	H1 2019/2020	Manchester_5	•
7	H1 2019/2020	Manchester_6	1
8	H1 2019/2020	Manchester_7	9
9	H1 2019/2020	Manchester_8	4
10	H1 2019/2020	Manchester_9	2

Once this is done, "Attribute" column would split into two columns "Attribute.1" and "Attribute.2" as shown in the diagram below, we then need to delete "Attribute.2" as we don’t need it.

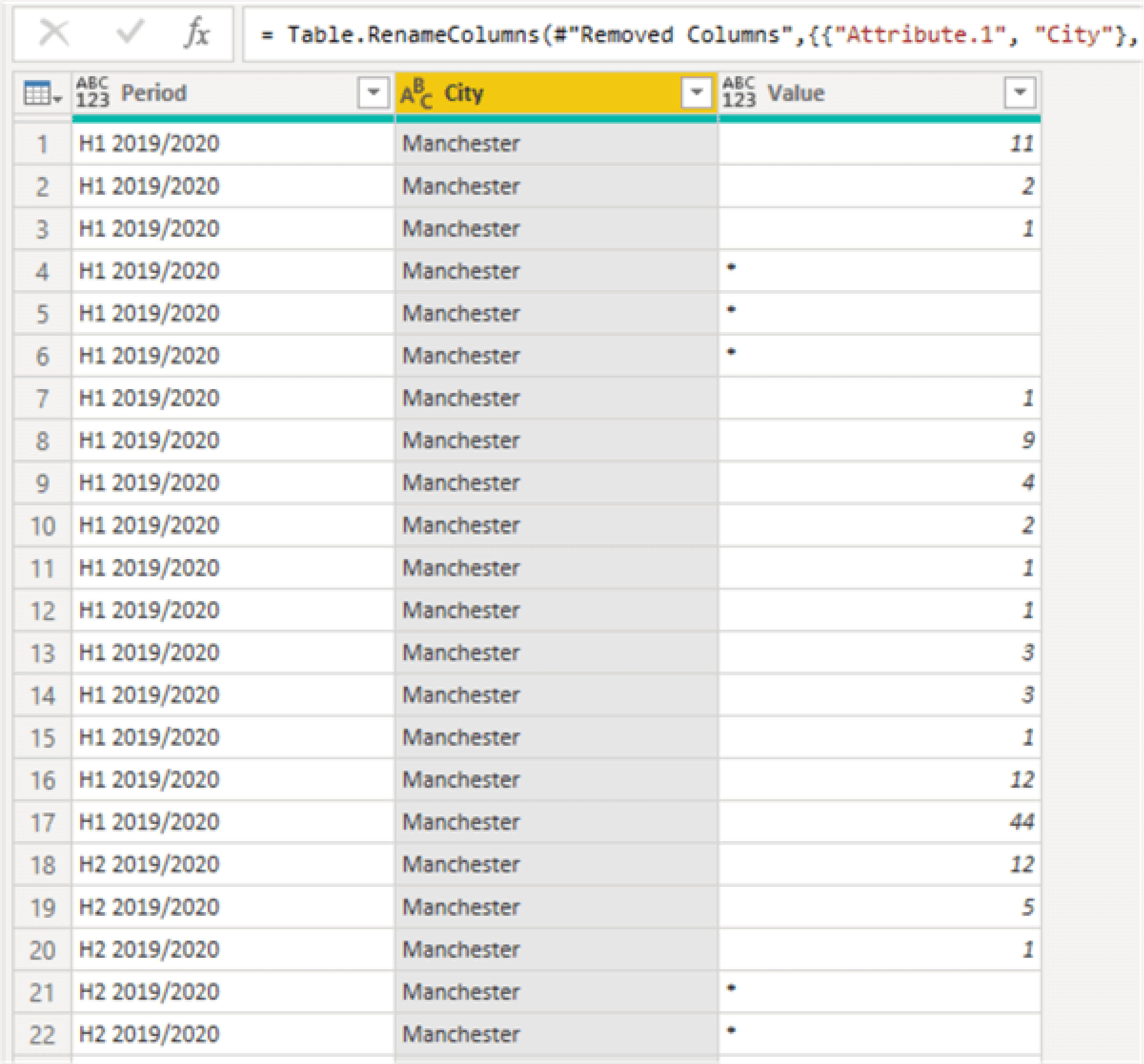


```
= Table.SplitColumn(#"Filled Down1", "Attribute", Splitter.SplitTextByDelimiter("_", Q
```

	Column1	Attribute.1	Attribute.2	Value
1	H1 2019/2020	Manchester	null	11
2	H1 2019/2020	Manchester	1	2
3	H1 2019/2020	Manchester	2	1
4	H1 2019/2020	Manchester	3	•
5	H1 2019/2020	Manchester	4	•
6	H1 2019/2020	Manchester	5	•
7	H1 2019/2020	Manchester	6	1
8	H1 2019/2020	Manchester	7	9
9	H1 2019/2020	Manchester	8	4
10	H1 2019/2020	Manchester	9	2
11	H1 2019/2020	Manchester	10	1
12	H1 2019/2020	Manchester	11	1
13	H1 2019/2020	Manchester	12	3
14	H1 2019/2020	Manchester	13	3
15	H1 2019/2020	Manchester	14	1
16	H1 2019/2020	Manchester	15	12
17	H1 2019/2020	Manchester	16	44
18	H2 2019/2020	Manchester	null	12
19	H2 2019/2020	Manchester	1	5
20	H2 2019/2020	Manchester	2	1
21	H2 2019/2020	Manchester	3	•
22	H2 2019/2020	Manchester	4	•

STEP 8: Rename columns appropriately

As you would observe the table has column names like "Column1" and "Attribute" which are not appropriate column names. So, we would rename "Column1" as "Period" and "Attribute.1" as "City" as shown in the diagram below.



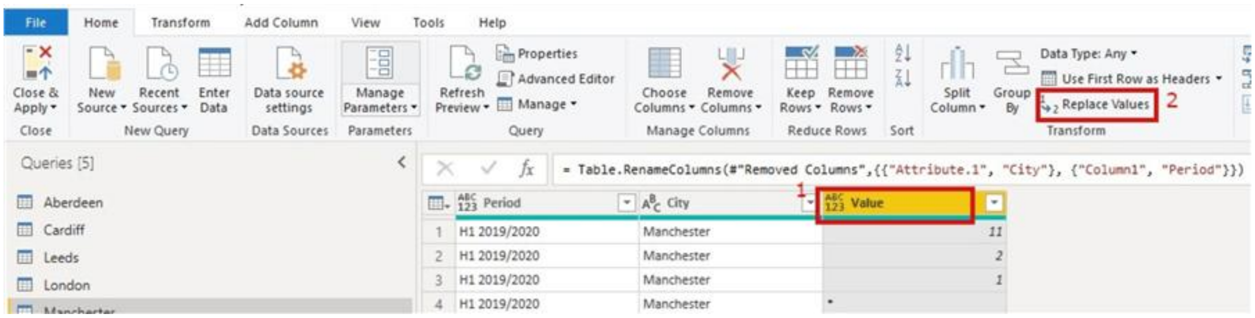
```
= Table.RenameColumns(#"Removed Columns",{{"Attribute.1", "City"},
```

	Period	City	Value
1	H1 2019/2020	Manchester	11
2	H1 2019/2020	Manchester	2
3	H1 2019/2020	Manchester	1
4	H1 2019/2020	Manchester	*
5	H1 2019/2020	Manchester	*
6	H1 2019/2020	Manchester	*
7	H1 2019/2020	Manchester	1
8	H1 2019/2020	Manchester	9
9	H1 2019/2020	Manchester	4
10	H1 2019/2020	Manchester	2
11	H1 2019/2020	Manchester	1
12	H1 2019/2020	Manchester	1
13	H1 2019/2020	Manchester	3
14	H1 2019/2020	Manchester	3
15	H1 2019/2020	Manchester	1
16	H1 2019/2020	Manchester	12
17	H1 2019/2020	Manchester	44
18	H2 2019/2020	Manchester	12
19	H2 2019/2020	Manchester	5
20	H2 2019/2020	Manchester	1
21	H2 2019/2020	Manchester	*
22	H2 2019/2020	Manchester	*

STEP 9: Replace any special character values within the "Values" column with "0"

As can be seen in the "Values" column, we have some special characters like "*" and "- "both of which will throw errors on the cell when we try to correct the datatype of the "Values" column. To ensure this is corrected, we need to replace these special character values with a numerical value such as zero ("0").

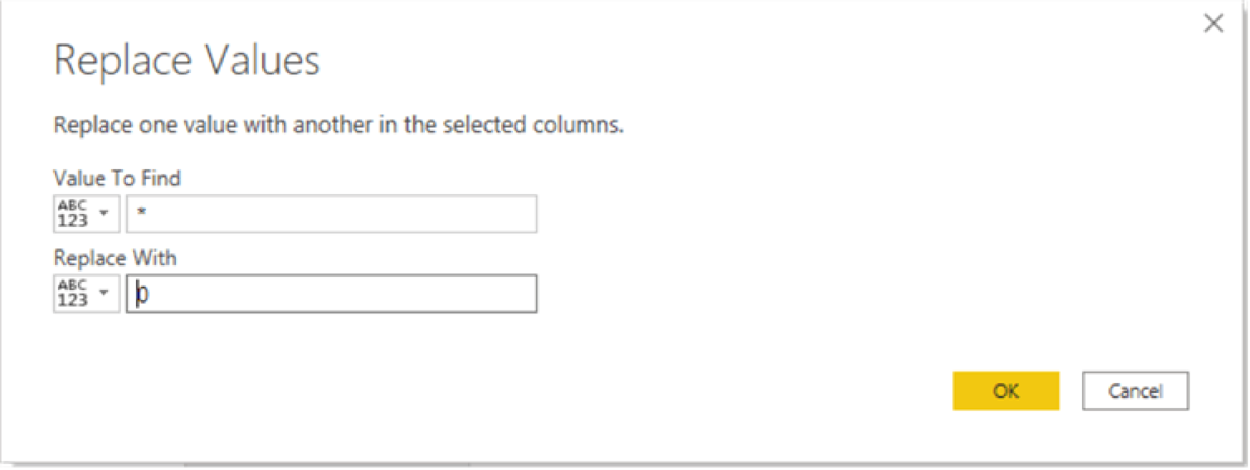
To do this, we click on the "Values" column, then within the "Home" tab and the transform section select "Replace Values" as seen in the diagram below.



```
= Table.RenameColumns(#"Removed Columns",{{"Attribute.1", "City"}, {"Column1", "Period"}})
```

	Period	City	Value
1	H1 2019/2020	Manchester	11
2	H1 2019/2020	Manchester	2
3	H1 2019/2020	Manchester	1
4	H1 2019/2020	Manchester	*

The below dialog box will open, then just enter the values as seen in the diagram below.

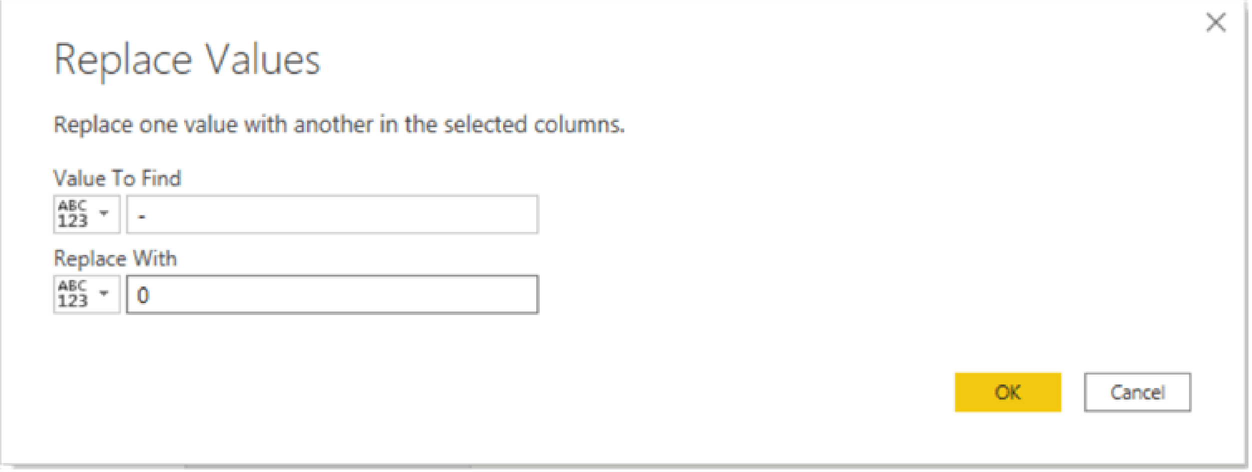

Replace Values

Replace one value with another in the selected columns.

Value To Find: *

Replace With: 0

Repeat same actions for replacing "- "with "0" as seen in the diagram below.

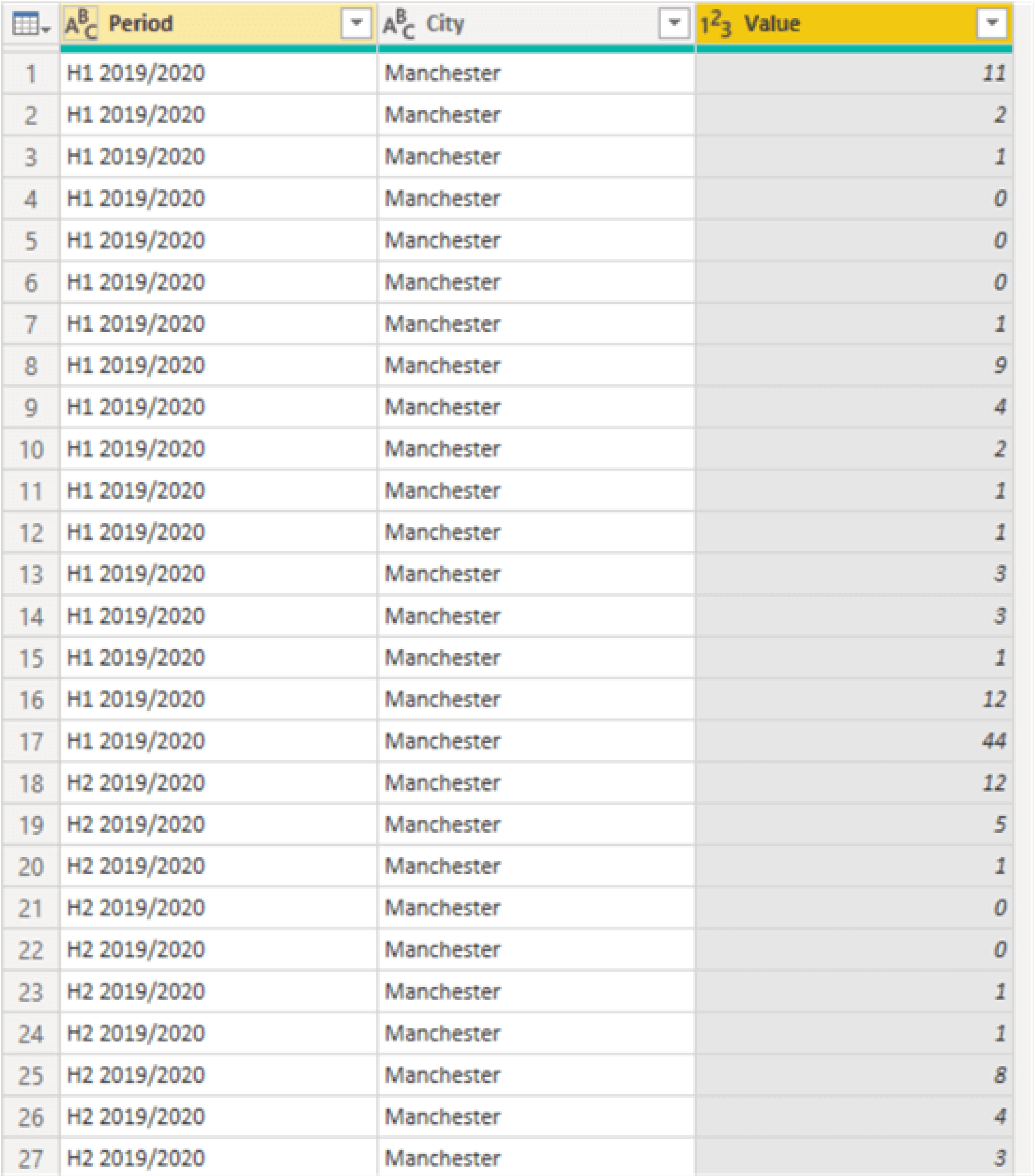

Replace Values

Replace one value with another in the selected columns.

Value To Find: -

Replace With: 0

Then after these we can now change the datatypes of the columns appropriately too. The product of the first transformation is as seen below.



	Period	City	Value
1	H1 2019/2020	Manchester	11
2	H1 2019/2020	Manchester	2
3	H1 2019/2020	Manchester	1
4	H1 2019/2020	Manchester	0
5	H1 2019/2020	Manchester	0
6	H1 2019/2020	Manchester	0
7	H1 2019/2020	Manchester	1
8	H1 2019/2020	Manchester	9
9	H1 2019/2020	Manchester	4
10	H1 2019/2020	Manchester	2
11	H1 2019/2020	Manchester	1
12	H1 2019/2020	Manchester	1
13	H1 2019/2020	Manchester	3
14	H1 2019/2020	Manchester	3
15	H1 2019/2020	Manchester	1
16	H1 2019/2020	Manchester	12
17	H1 2019/2020	Manchester	44
18	H2 2019/2020	Manchester	12
19	H2 2019/2020	Manchester	5
20	H2 2019/2020	Manchester	1
21	H2 2019/2020	Manchester	0
22	H2 2019/2020	Manchester	0
23	H2 2019/2020	Manchester	1
24	H2 2019/2020	Manchester	1
25	H2 2019/2020	Manchester	8
26	H2 2019/2020	Manchester	4
27	H2 2019/2020	Manchester	3

Data Modeling in Power BI and DAX for Calculated Columns and Measures

Objective:

To import data, visit the 'Home' menu from the top, then click on 'Get Data' and select the method you want to use. It will take some time to process and show results on your screen. If you feel that your data is properly loaded, click on the 'Load' button. Else, click 'Transform Data' to make some changes.

Here we will import from an excel sheet that contains tables. These tables represent the relationship with data inserted on rows & column. Power BI automatically recognize the data & its relation in a table on a single import.

Equipment Required:

The following list provides the minimum requirements to run Power BI Desktop:

Important

- Power BI Desktop is no longer supported on Windows 7.
- Windows 8.1 or Windows Server 2012 R2 or later.
- .NET 4.7.2 or later.
- Microsoft Edge browser (Internet Explorer is no longer supported)
- Memory (RAM): At least 2 GB available, 4 GB or more recommended.
- Display: At least 1440x900 or 1600x900 (16:9) required. Lower resolutions such as 1024x768 or 1280x800 aren't supported because some controls (such as closing the startup screens) display beyond those resolutions.
- Windows display settings: If you set your display to change the size of text, apps, and other items to more than 100%, you won't see some dialogs that you must interact with to continue using Power BI Desktop. If you encounter this issue, check your display settings in Windows by going to Settings > System > Display, and use the slider to return display settings to 100%.
- CPU: 1 gigahertz (GHz) 64-bit (x64) processor or better recommended.
- WebView2: If WebView2 wasn't automatically installed with Power BI Desktop or if it was uninstalled, download and run the installer for WebView2.

Note

We recommend using a client version of Windows, such as Windows 10, instead of Windows Server. Power BI Desktop doesn't support Internet Explorer Enhanced Security Configuration because it will stop Power BI Desktop from signing in to the Power BI service.

Prerequisites:

5. Internet skill to access the Power BI site
6. Computer OS skill to install the software

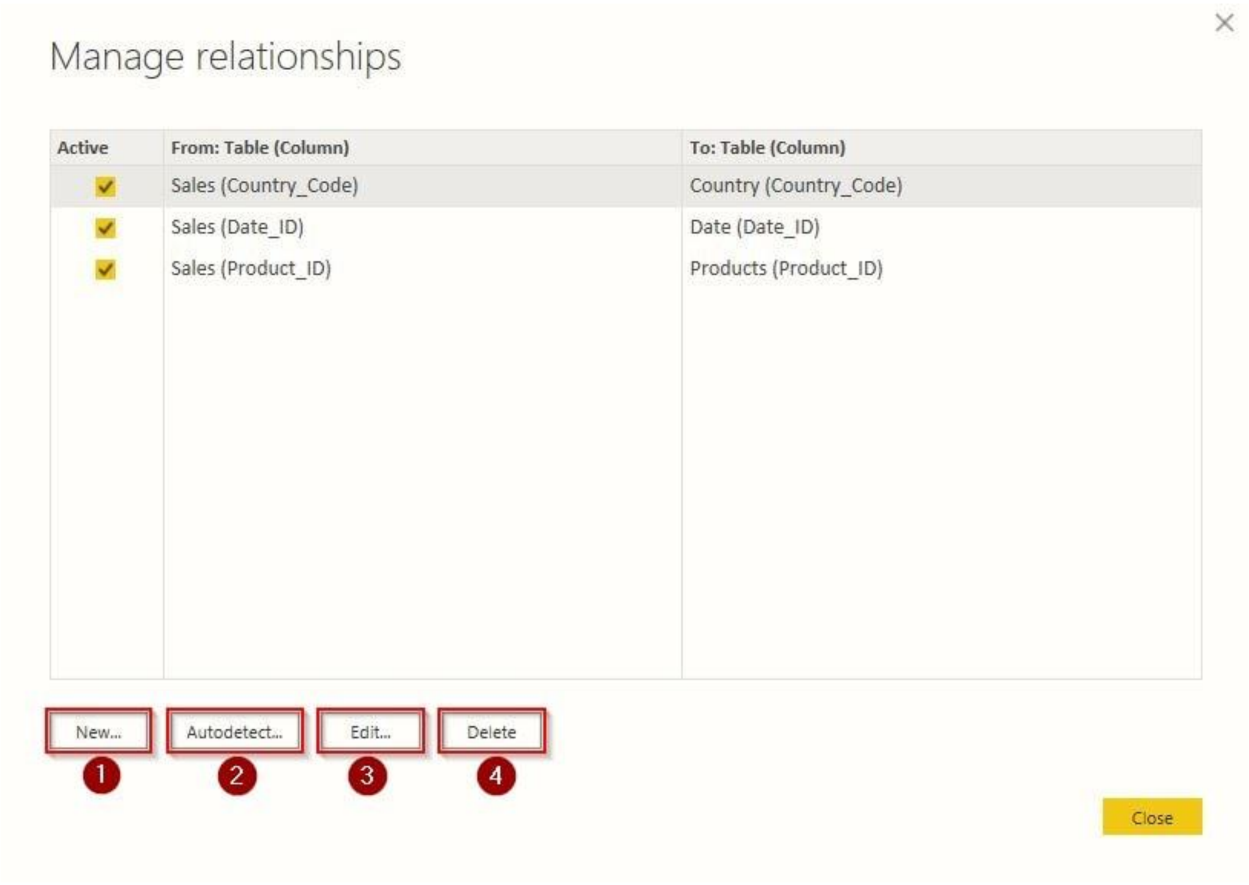
Problem Statement:

We will be using simple electronics shop sale records as our data to create Data Models. This data includes the Shop Sales with parameters as customer, country, product, cost, month and year.

Solution:
To automatically detect all the possible relations between different sets of data. Sometimes we need to create a relation between the data manually. Visit the 'Models' tab from the left side, as highlighted in the above image. Here you will see some automated relation created by Power BI. All the lines visible here in the 'Models' tab depicts

the cardinality and direction of the relation from one table to another. You can create & modify this default relationship created by Power BI using the Manage Relationship tool

Create And Manage Relationship

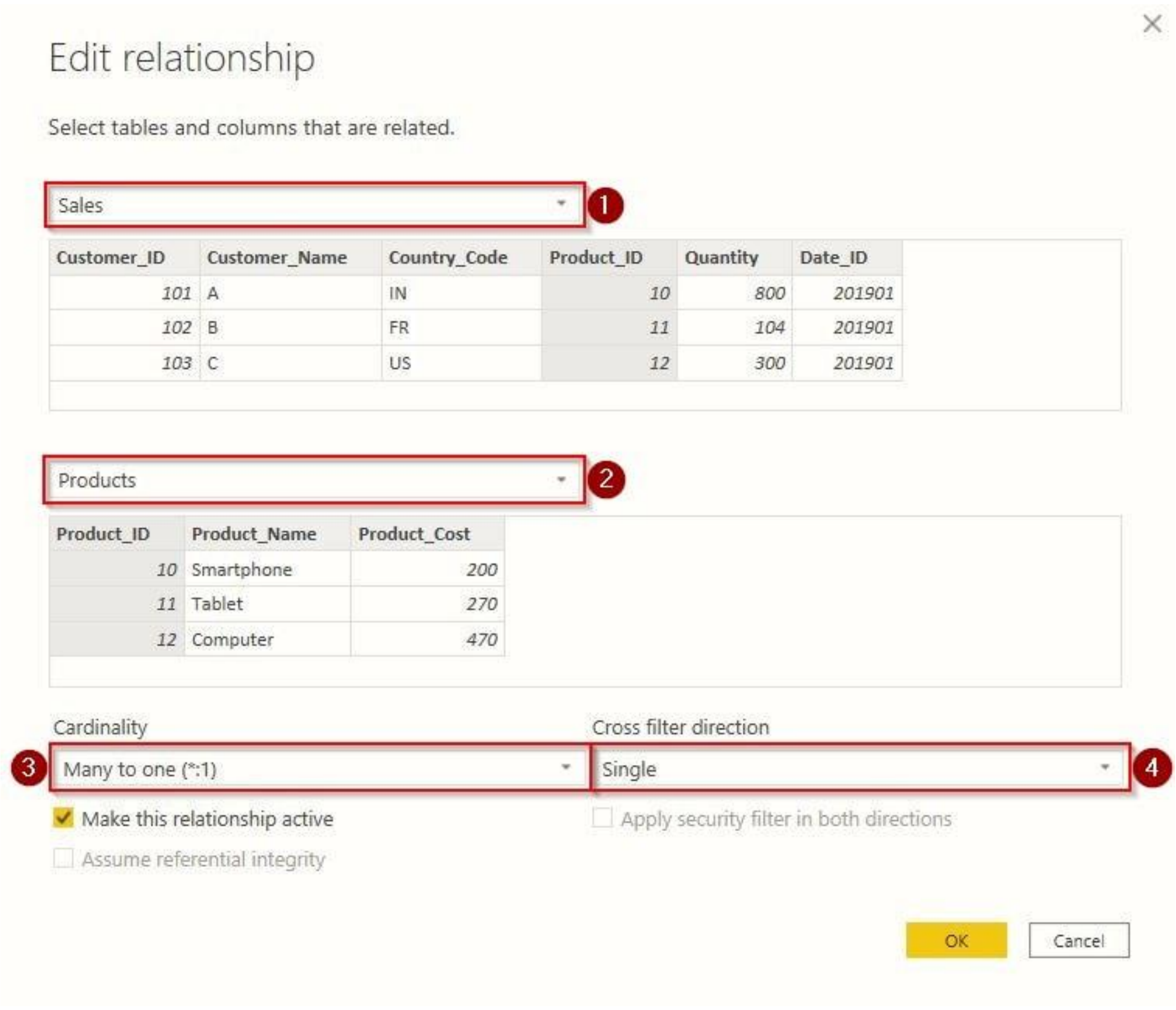


Manage relationships

Active	From: Table (Column)	To: Table (Column)
☑	Sales (Country_Code)	Country (Country_Code)
☑	Sales (Date_ID)	Date (Date_ID)
☑	Sales (Product_ID)	Products (Product_ID)

After clicking on ‘*Manage Relationship,*‘ a similar screen will appear, as shown in the above image. You can see all the active relations here from one table to another. All the things you can do with these relations are explained below:

1. **New** – This option will help you create a new relationship between tables.
2. **Autodetect** – Using this option, Power BI automatically detects the relationship between data present in tables.
3. **Edit** – This option will help you to edit your data relationship.
4. **Delete** – It deletes the selected relationship between the tables.



Edit relationship

Select tables and columns that are related.

Sales

Customer_ID	Customer_Name	Country_Code	Product_ID	Quantity	Date_ID
101	A	IN	10	800	201901
102	B	FR	11	104	201901
103	C	US	12	300	201901

Products

Product_ID	Product_Name	Product_Cost
10	Smartphone	200
11	Tablet	270
12	Computer	470

Cardinality: Many to one (*:1)

Cross filter direction: Single

☑ Make this relationship active

□ Apply security filter in both directions

□ Assume referential integrity

Now after clicking on ‘*New*‘ or ‘*Edit*‘, a similar screen will appear in front of you, as shown above. Here we will explain to you the purpose of all the options one by one.

1. The first drop-down menu will allow you to select your table from which you want to create a relation.
2. The selection in the second drop-down menu will create a relation from the first table to the second.
3. You will select the Cardinality relation here, but you can not force Power BI to select a specific cardinality that doesn’t exist.
4. Here you can select the direction of the relationship as ‘Single’ or from ‘Both’.

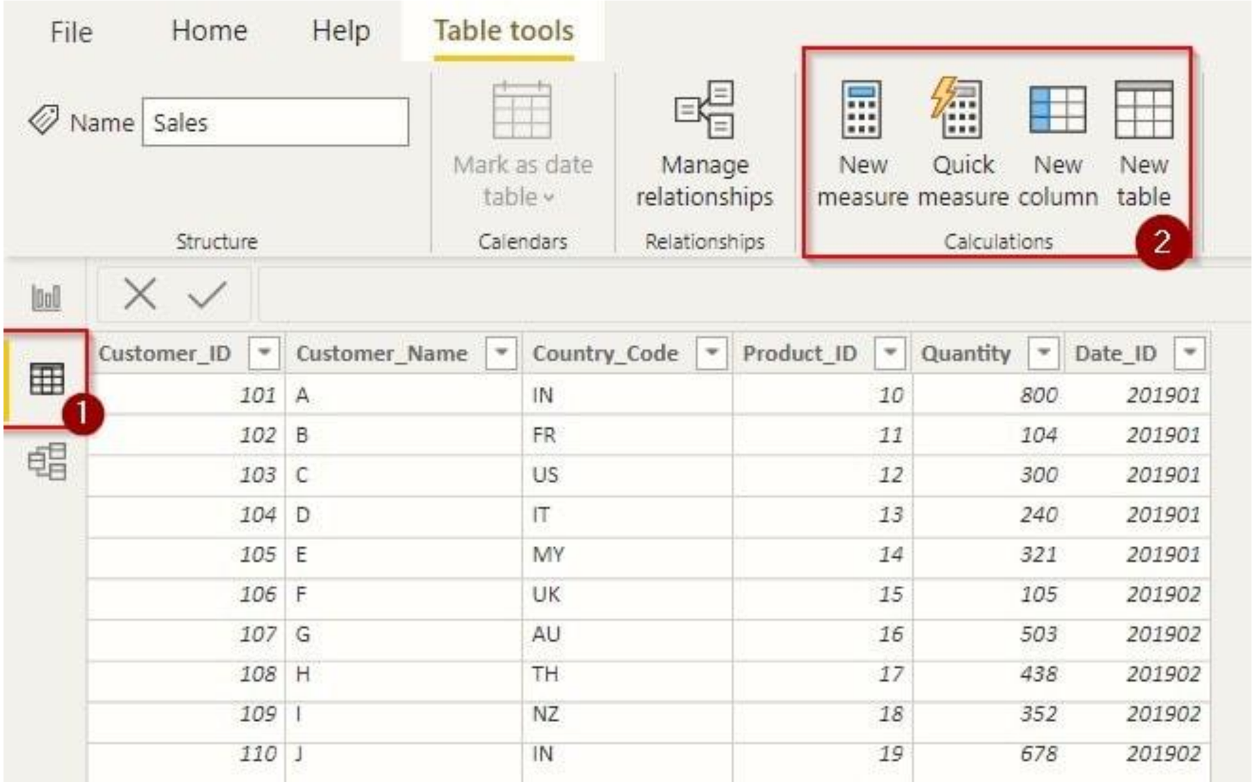
Calculate And Measure Data

If you are familiar with Excel, you may have worked on the DAX (Data Analysis Expression) formula. If not, then no worries. I will explain in short and simple words. Like programming, DAX is a set of instructions used to calculate data from the tables. These expressions include commands for Addition, Multiplication, Average, Percentage and others with various filters.

For Example :

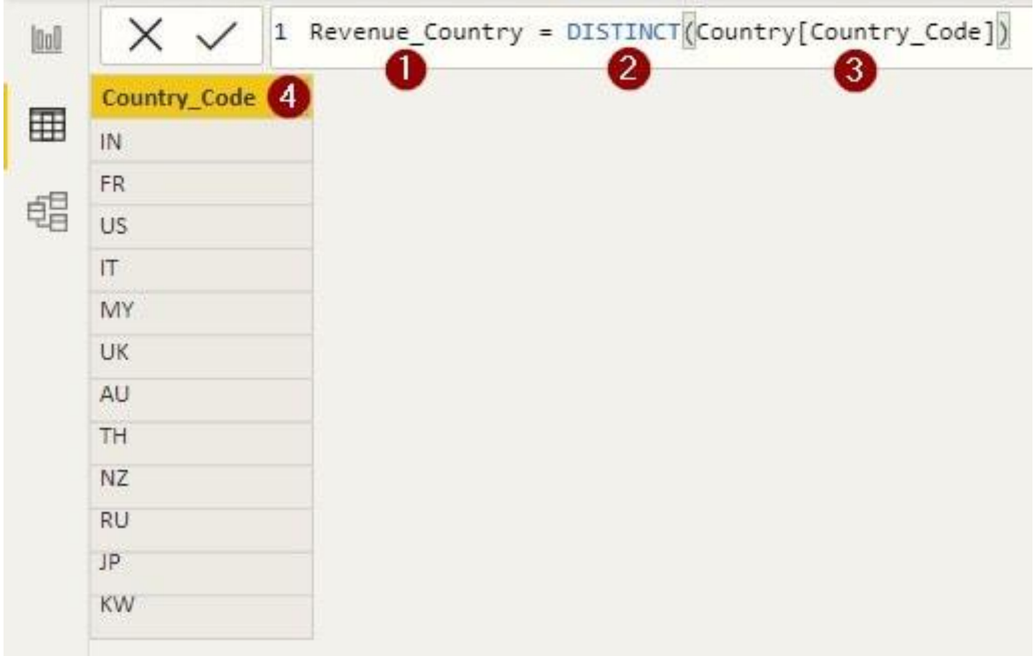
1. *Table 1 = DISTINCT(Table 2[Column_1])*, this expression will fill all the unique values in Table 1 from Column_1 present in Table 2.
2. *Column1 = RIGHT(Table1[Column_Name],3)*, this expression will get last 3 characters from Column_Name to fill in Column1.

Let's create some calculations with our shop data present in Power BI. Visit the '*Data*' tab from the left menu as highlighted in the image below. Here you will see some tools to calculate your data. We will be using these on Power BI.



Customer_ID	Customer_Name	Country_Code	Product_ID	Quantity	Date_ID
101	A	IN	10	800	201901
102	B	FR	11	104	201901
103	C	US	12	300	201901
104	D	IT	13	240	201901
105	E	MY	14	321	201901
106	F	UK	15	105	201902
107	G	AU	16	503	201902
108	H	TH	17	438	201902
109	I	NZ	18	352	201902
110	J	IN	19	678	201902

Create Table



```
Revenue_Country = DISTINCT(Country[Country_Code])
```

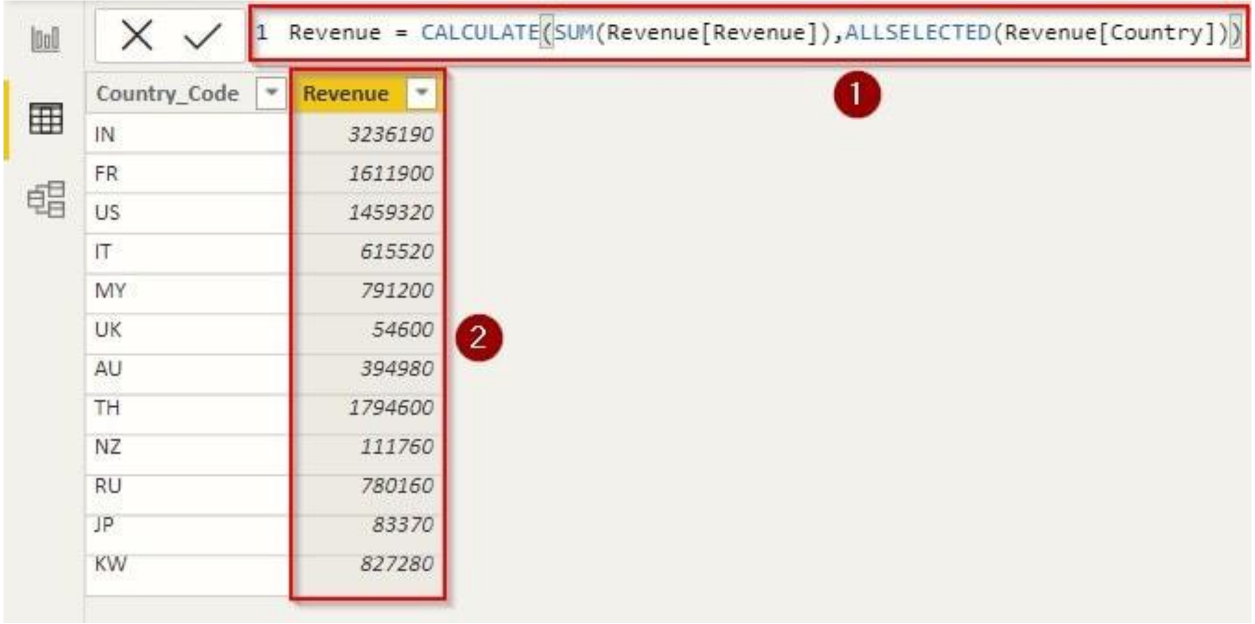
Country_Code
IN
FR
US
IT
MY
UK
AU
TH
NZ
RU
JP
KW

After clicking on '*New Table*', we need to enter the DAX expression shown in the above image.

1. The first part of the expression defines the name of the table.
2. The second is the filter; the ‘DISTINCT’ function will select only the unique values from the column.
3. We need to pass the parameters inside the ‘DISTINCT’ function, and these parameters are the location from which we will extract our data. So we have passed the table and column name where our country codes are present. When your expression is complete, click ‘Enter’.
4. After applying the expression, we will get our new column with the default name and results. To rename a column, you can double click on it.

Create Column

Click on ‘*New Column*‘ from the top menu to create a calculated column.



```
1 Revenue = CALCULATE(SUM(Revenue[Revenue]),ALLSELECTED(Revenue[Country]))
```

Country_Code	Revenue
IN	3236190
FR	1611900
US	1459320
IT	615520
MY	791200
UK	54600
AU	394980
TH	1794600
NZ	111760
RU	780160
JP	83370
KW	827280

1. The DAX expression will calculate all the revenue from the table ‘*Revenue*‘ with filter as ‘*Country*‘. Without this expression, we might have spent hours calculating the individual revenue generated from the country.
2. This is the result we get from the expression.

Although Power BI suggests you write an expression, it might be difficult to remember all the expressions. If this is the case, you can use the Quick Measure tool. You will only need to fill in the parameters and function for calculation. This tool will then generate the expression automatically, depending on your selection. These measurement tools are also helpful when you want quick calculations for your reports.