Contents

Create and optimize data models

To let create visualizations and reports from data, Power BI requires you to optimize data by creating proper data models out of it.

Below are common techniques applied for creating and optimizing data models.

Manage relationships

As per relational database theory, data tables need to relate to each other to form relationships within them, such that we can use this related data together to make sense out of it.

Before looking at an example of managing relations in Power BI, you must know about two important terms, cardinality, filtering direction and active relationship.

Cardinality

Many to One (*:1) - This is the most common, default type. This means the column in one table can have more than one instance of a value, and the other related table, often known as the Lookup table, has only one instance of a value.

One to One (1:1) - This means the column in one table has only one instance of a value, and the other related table also has only one instance of that particular value.

Cross filter direction

Both - This is the most common, default direction. This means for filtering purposes, both tables are treated as if they're a single table. This works well with a single table that has several lookup tables that surround it. An example is a Sales actuals table with a lookup table for department. This is often called a Star schema configuration (a central table with several lookup tables.) However, if you have two or more tables that also have lookup tables (with some in common) then you wouldn't want to use the Both setting. To continue the previous example, in this case, you also have a budget sales table that records target budget for each department. And, the department table is connected to both the sales and the budget table. Avoid the Both setting for this kind of configuration.

Single - This means that filtering choices in connected tables work on the table where values are being aggregated. If you import a Power Pivot in Excel 2013 or earlier data model, all relationships will have a single direction.

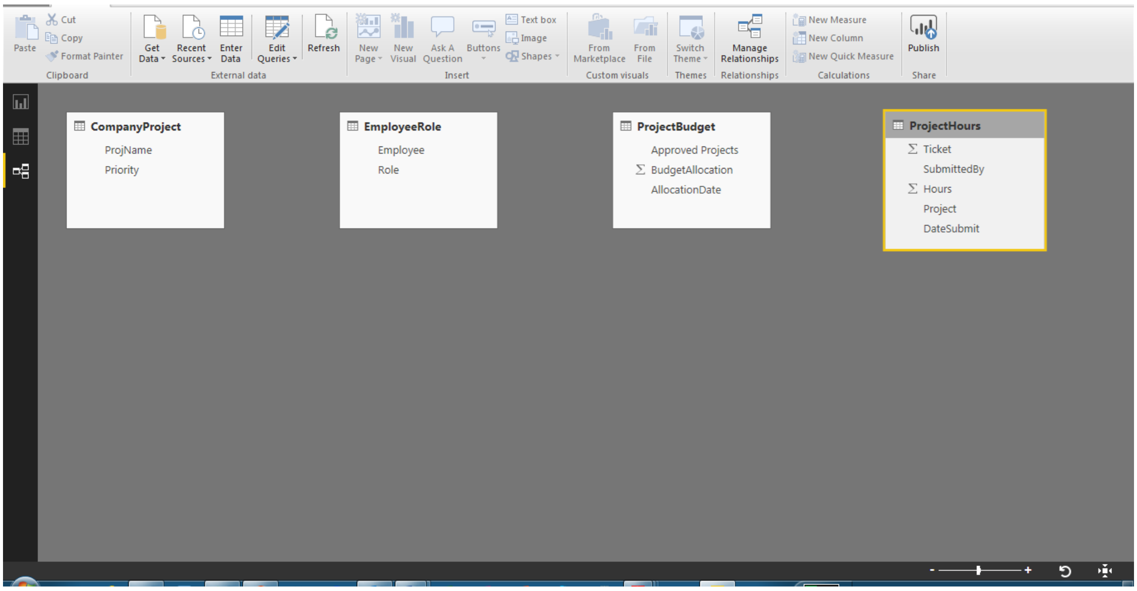
See the Understanding additional options section later in this article for more details about when to change cross filter direction.

Make relationship active

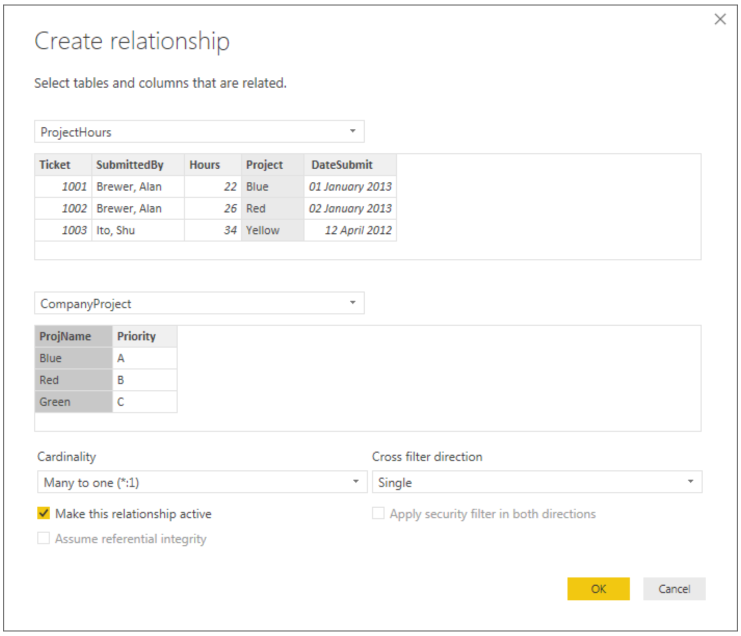
When checked, this means the relationship serves as the active, default relationship. In cases where there is more than one relationship between two tables, the active relationship provides a way for Power BI Desktop to automatically create visualizations that include both tables.

Let’s walk through an example of creating and managing relationships.

1. Connect and load data from the MR.xlsx file found in **…\PBI-Training\Day 2** folder.
2. Once you load data, switch to Relationships view, which should look like follows.



3. As you see, the tables are not related in this case, so follow the steps below to create the relationships.
4. Click Manage Relationships.
5. In Manage Relationships, click New. This opens the Create Relationship dialog, where we can select the tables, columns, and any additional settings we want for our relationship.
6. In the first table, select ProjectHours, then select the Project column. This is the many side of our relationship.
7. In the second table, select CompanyProject, then select the ProjName column. This is the one side of our relationship.
8. Go ahead and click OK in both the Create Relationship dialog and the Manage Relationships dialog.



Create relationship

Select tables and columns that are related.

ProjectHours

Ticket	SubmittedBy	Hours	Project	DateSubmit
1001	Brewer, Alan	22	Blue	01 January 2013
1002	Brewer, Alan	26	Red	02 January 2013
1003	Ito, Shu	34	Yellow	12 April 2012

CompanyProject

ProjName	Priority
Blue	A
Red	B
Green	C

Cardinality: Many to one (*:1)

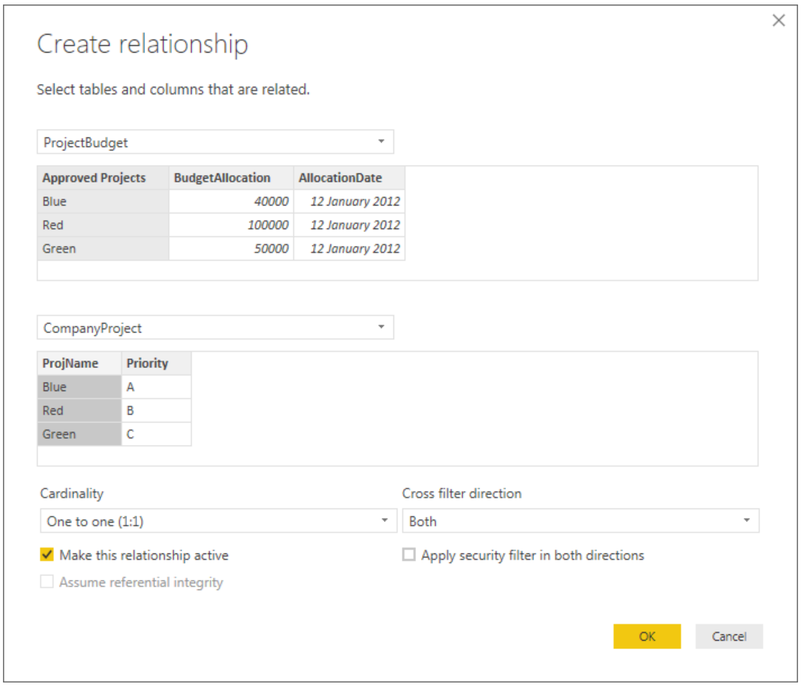
Cross filter direction: Single

☑ Make this relationship active

□ Apply security filter in both directions

□ Assume referential integrity

Similarly, create additional relationships among other tables to complete this example. (look at the figures below)



Create relationship

Select tables and columns that are related.

ProjectBudget

Approved Projects	BudgetAllocation	AllocationDate
Blue	*40000*	*12 January 2012*
Red	*100000*	*12 January 2012*
Green	*50000*	*12 January 2012*

CompanyProject

ProjName	Priority
Blue	A
Red	B
Green	C

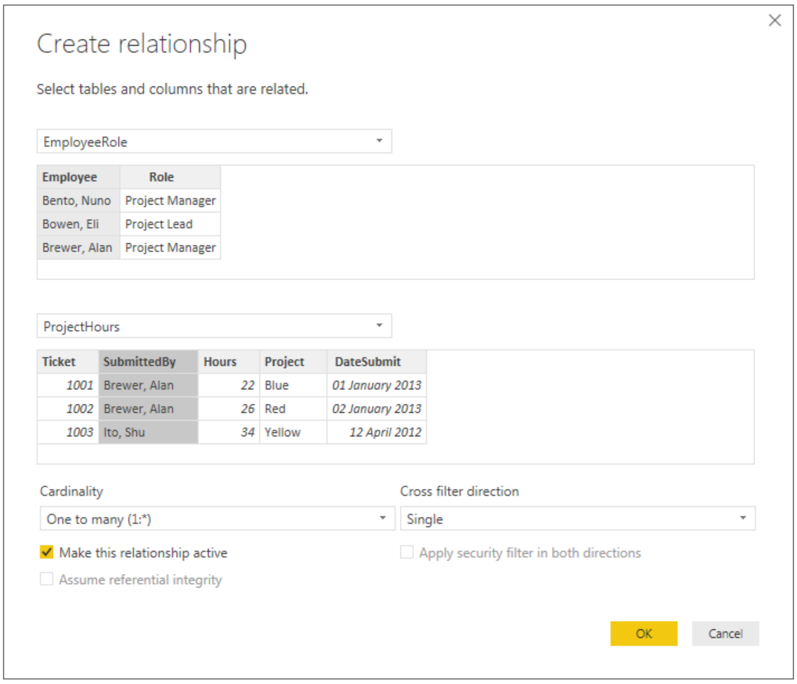
Cardinality: One to one (1:1)

Cross filter direction: Both

☑ Make this relationship active

□ Apply security filter in both directions

□ Assume referential integrity



Create relationship

Select tables and columns that are related.

EmployeeRole

Employee	Role
Bento, Nuno	Project Manager
Bowen, Eli	Project Lead
Brewer, Alan	Project Manager

ProjectHours

Ticket	SubmittedBy	Hours	Project	DateSubmit
1001	Brewer, Alan	*22*	Blue	*01 January 2013*
1002	Brewer, Alan	*26*	Red	*02 January 2013*
1003	Ito, Shu	*34*	Yellow	*12 April 2012*

Cardinality: One to many (1:*)

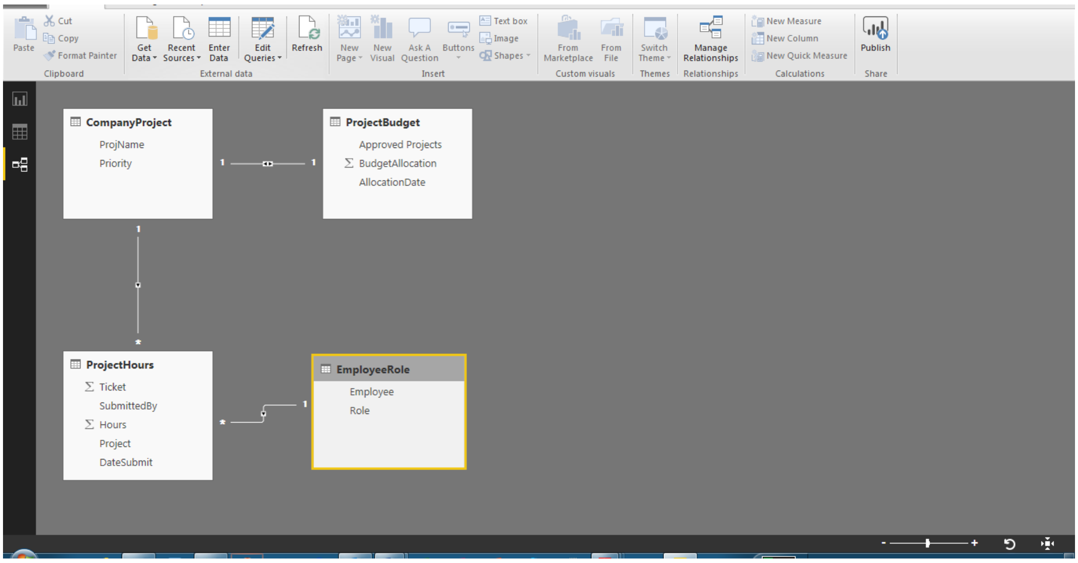
Cross filter direction: Single

☑ Make this relationship active

□ Apply security filter in both directions

□ Assume referential integrity

Finally, the relationship diagram should look like below.

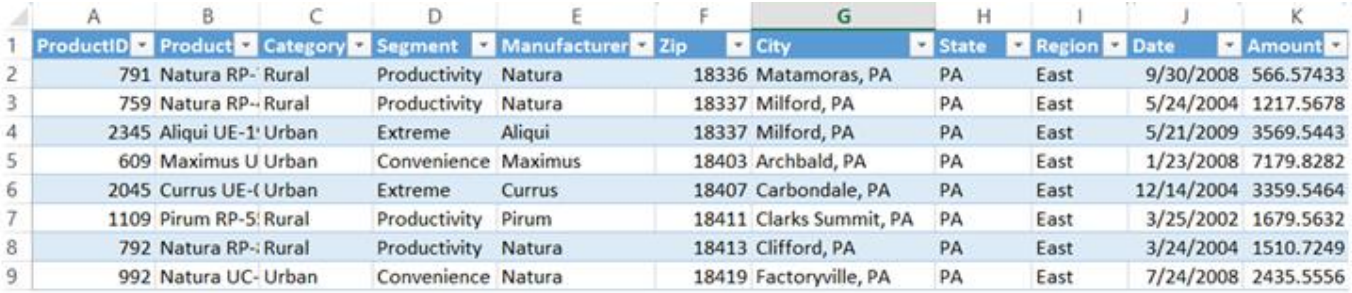


Optimize models for reporting

In addition to creating relationships among tables, there are additional techniques to optimize the model for reporting. Let's walk through some of these below.

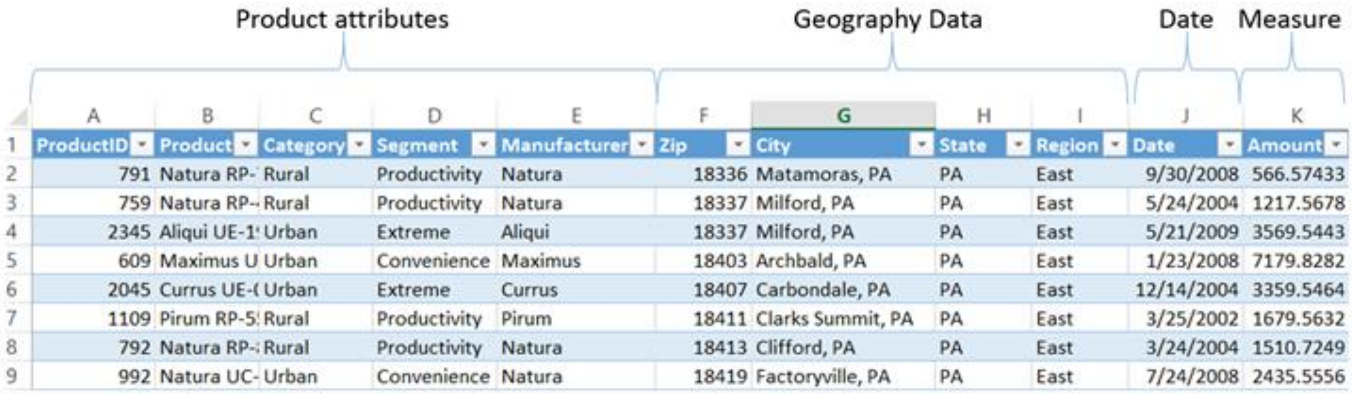
Create a Star schema

Consider following data…



	ProductID	Product	Category	Segment	Manufacturer	Zip	City	State	Region	Date	Amount
2	791	Natura RP-	Rural	Productivity	Natura	18336	Matamoras, PA	PA	East	9/30/2008	566.57433
3	759	Natura RP-	Rural	Productivity	Natura	18337	Milford, PA	PA	East	5/24/2004	1217.5678
4	2345	Aliqui UE-1	Urban	Extreme	Aliqui	18337	Milford, PA	PA	East	5/21/2009	3569.5443
5	609	Maximus U	Urban	Convenience	Maximus	18403	Archbald, PA	PA	East	1/23/2008	7179.8282
6	2045	Currus UE-(	Urban	Extreme	Currus	18407	Carbondale, PA	PA	East	12/14/2004	3359.5464
7	1109	Pirum RP-5	Rural	Productivity	Pirum	18411	Clarks Summit, PA	PA	East	3/25/2002	1679.5632
8	792	Natura RP-	Rural	Productivity	Natura	18413	Clifford, PA	PA	East	3/24/2004	1510.7249
9	992	Natura UC-	Urban	Convenience	Natura	18419	Factoryville, PA	PA	East	7/24/2008	2435.5556

This data represents sales transactions and if we observe closely, we can make out that there are dimensions and facts among these as next figure shows.

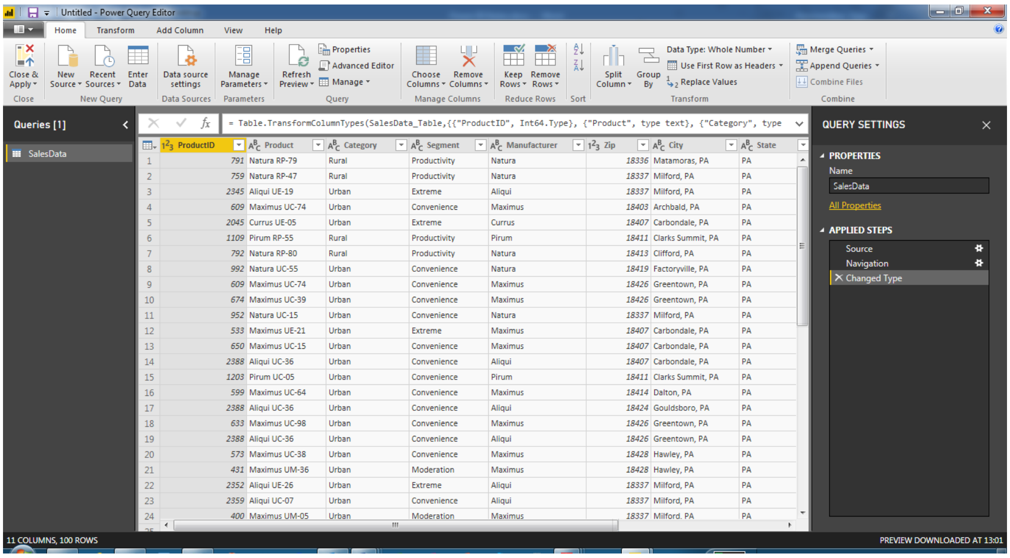

Product attributes
Geography Data
Date
Measure

	ProductID	Product	Category	Segment	Manufacturer	Zip	City	State	Region	Date	Amount
2	791	Natura RP-	Rural	Productivity	Natura	18336	Matamoras, PA	PA	East	9/30/2008	566.57433
3	759	Natura RP-	Rural	Productivity	Natura	18337	Milford, PA	PA	East	5/24/2004	1217.5678
4	2345	Aliqui UE-1	Urban	Extreme	Aliqui	18337	Milford, PA	PA	East	5/21/2009	3569.5443
5	609	Maximus U	Urban	Convenience	Maximus	18403	Archbald, PA	PA	East	1/23/2008	7179.8282
6	2045	Currus UE-(	Urban	Extreme	Currus	18407	Carbondale, PA	PA	East	12/14/2004	3359.5464
7	1109	Pirum RP-5	Rural	Productivity	Pirum	18411	Clarks Summit, PA	PA	East	3/25/2002	1679.5632
8	792	Natura RP-	Rural	Productivity	Natura	18413	Clifford, PA	PA	East	3/24/2004	1510.7249
9	992	Natura UC-	Urban	Convenience	Natura	18419	Factoryville, PA	PA	East	7/24/2008	2435.5556

The first five attributes represent the Product dimension, the next four represent Geography dimension, the single Date attribute represents Date/Time dimension and the last remaining single attribute is the Sales measure or fact.

In order to use this data for any reporting/visualization purpose, we need to convert this single table into a Star schema, so follow along…

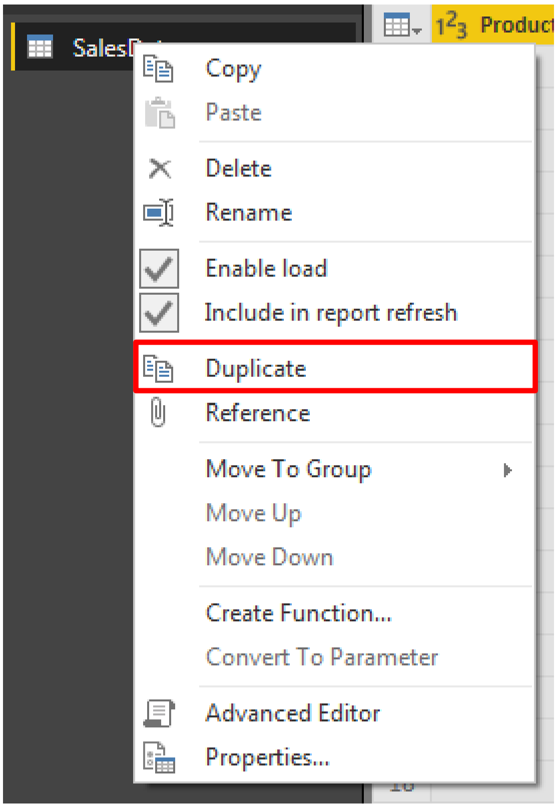
1. Connect and load the SalesData.xlsx from **…\PBI-Training\Day 2** folder.
2. Click on the Edit Queries button on Home tab of the ribbon, to open the Power Query editor.
3. The opened window should be like this.



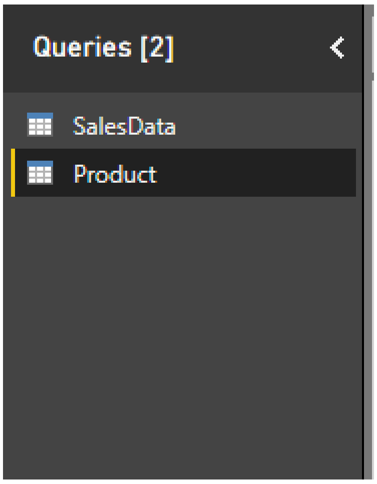
```
= Table.TransformColumnTypes(SalesData_Table,{{"ProductID", Int64.Type}, {"Product", type text}, {"Category", type
```

	ProductID	Product	Category	Segment	Manufacturer	Zip	City	State
1	791	Natura RP-79	Rural	Productivity	Natura	18336	Matamoras, PA	PA
2	759	Natura RP-47	Rural	Productivity	Natura	18337	Milford, PA	PA
3	2345	Aliqui UE-19	Urban	Extreme	Aliqui	18337	Milford, PA	PA
4	609	Maximus UC-74	Urban	Convenience	Maximus	18403	Archbald, PA	PA
5	2045	Currus UE-05	Urban	Extreme	Currus	18407	Carbondale, PA	PA
6	1109	Pirum RP-55	Rural	Productivity	Pirum	18411	Clarks Summit, PA	PA
7	792	Natura RP-80	Rural	Productivity	Natura	18413	Clifford, PA	PA
8	992	Natura UC-55	Urban	Convenience	Natura	18419	Factoryville, PA	PA
9	609	Maximus UC-74	Urban	Convenience	Maximus	18426	Greentown, PA	PA
10	674	Maximus UC-39	Urban	Convenience	Maximus	18426	Greentown, PA	PA
11	952	Natura UC-15	Urban	Convenience	Natura	18337	Milford, PA	PA
12	533	Maximus UE-21	Urban	Extreme	Maximus	18407	Carbondale, PA	PA
13	650	Maximus UC-15	Urban	Convenience	Maximus	18407	Carbondale, PA	PA
14	2388	Aliqui UC-36	Urban	Convenience	Aliqui	18407	Carbondale, PA	PA
15	1203	Pirum UC-05	Urban	Convenience	Pirum	18411	Clarks Summit, PA	PA
16	599	Maximus UC-64	Urban	Convenience	Maximus	18414	Dalton, PA	PA
17	2388	Aliqui UC-36	Urban	Convenience	Aliqui	18424	Gouldsboro, PA	PA
18	633	Maximus UC-98	Urban	Convenience	Maximus	18426	Greentown, PA	PA
19	2388	Aliqui UC-36	Urban	Convenience	Aliqui	18426	Greentown, PA	PA
20	573	Maximus UC-38	Urban	Convenience	Maximus	18428	Hawley, PA	PA
21	431	Maximus UM-36	Urban	Moderation	Maximus	18428	Hawley, PA	PA
22	2352	Aliqui UE-26	Urban	Extreme	Aliqui	18337	Milford, PA	PA
23	2359	Aliqui UC-07	Urban	Convenience	Aliqui	18337	Milford, PA	PA
24	400	Maximus UM-05	Urban	Moderation	Maximus	18337	Milford, PA	PA

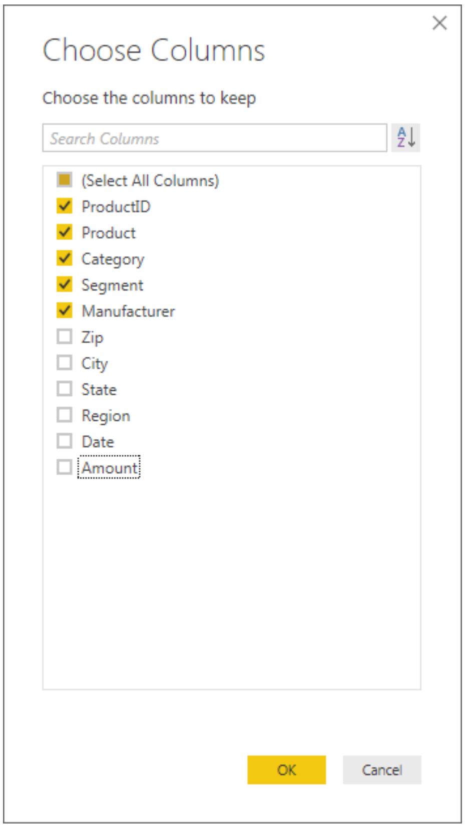
4. Now, right-click on the SalesData query name on the left-hand side of the screen and select Duplicate option to create a duplicate data table.



5. Double-click the duplicated SalesData (2) query name and rename it to Product.



6. Click on the Choose Columns option on the ribbon and uncheck all columns unrelated to Product. Check the figure.



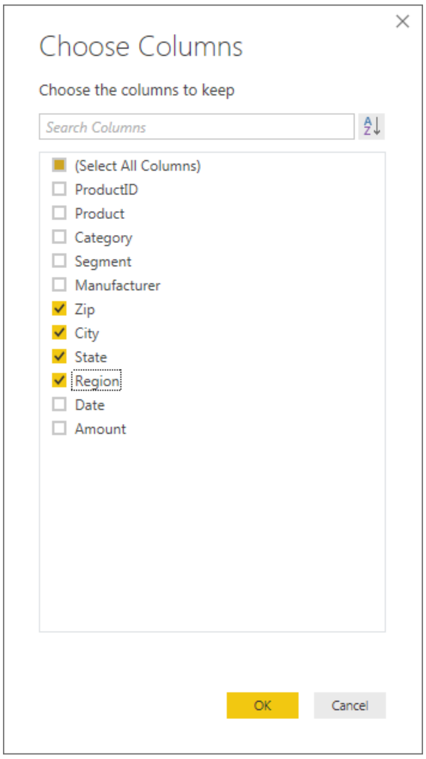
Choose Columns

Choose the columns to keep

- ▣ (Select All Columns)
- ☑ ProductID
- ☑ Product
- ☑ Category
- ☑ Segment
- ☑ Manufacturer
- ☐ Zip
- ☐ City
- ☐ State
- ☐ Region
- ☐ Date
- ☐ Amount

7. Next step is to remove duplicate records by right clicking on Product ID and then selecting Remove Duplicate option from the menu.
8. This gives us the Product dimension table and now we need to **repeat steps 4 through 7 above** to create the Geography and Date dimensions *. Refer below figures to complete.

* Duplicate SalesData and rename to Geography, then choose relevant columns as follows, remove duplicates from Zip column

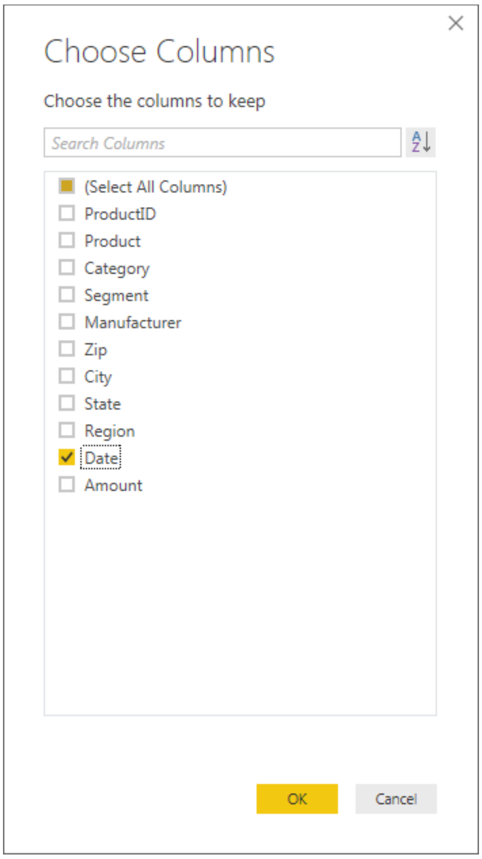


Choose Columns

Choose the columns to keep

- ■ (Select All Columns)
- □ ProductID
- □ Product
- □ Category
- □ Segment
- □ Manufacturer
- ☑ Zip
- ☑ City
- ☑ State
- ☑ Region
- □ Date
- □ Amount

* Duplicate SalesData and rename to Date, then choose relevant columns as follows, remove duplicates from Date column

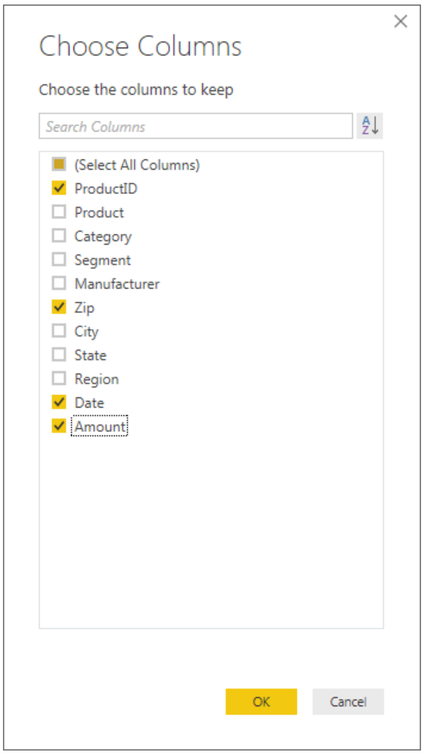


Choose Columns

Choose the columns to keep

- ■ (Select All Columns)
- □ ProductID
- □ Product
- □ Category
- □ Segment
- □ Manufacturer
- □ Zip
- □ City
- □ State
- □ Region
- ☑ Date
- □ Amount

Finally, duplicate SalesData one more time, rename it to SalesFact and choose following columns.



Choose Columns

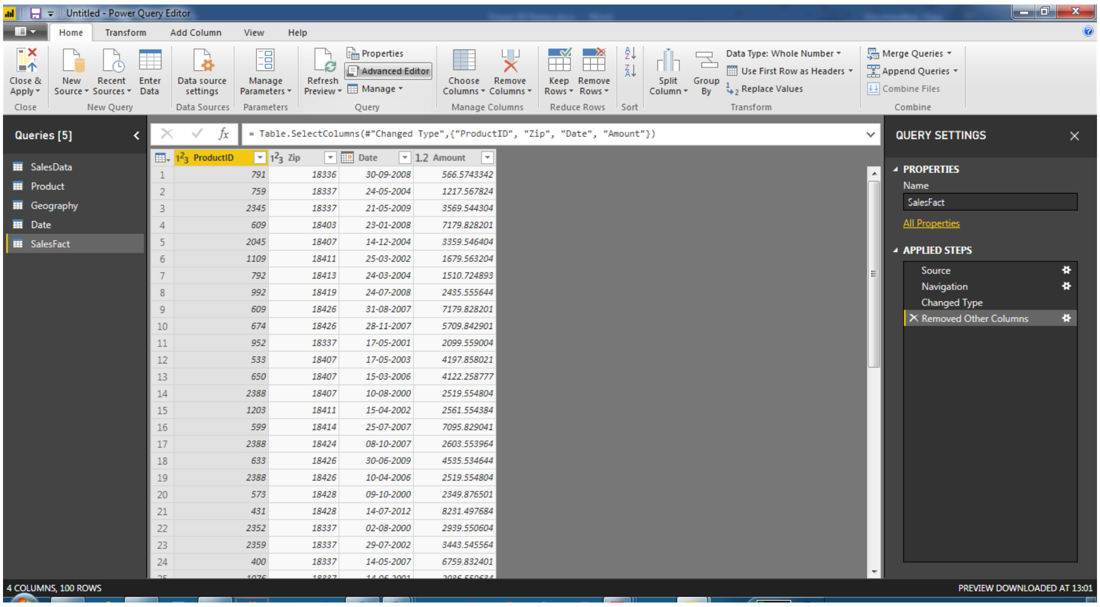
Choose the columns to keep

Search Columns

- ■ (Select All Columns)
- ☑ ProductID
- □ Product
- □ Category
- □ Segment
- □ Manufacturer
- ☑ Zip
- □ City
- □ State
- □ Region
- ☑ Date
- ☑ Amount

Important: As this is Fact table, do not remove duplicates.

The final screen should match below showing all queries as on left side. **As we won't need the original SalesData query anymore, right-click on it and unselect Enable Load option from the menu**. Once done, click the Close & Apply button to return data to Power BI.



Queries [5]

- SalesData
- Product
- Geography
- Date
- SalesFact

```
= Table.SelectColumns(#"Changed Type",{"ProductID", "Zip", "Date", "Amount"})
```

	ProductID	Zip	Date	Amount
1	791	18336	30-09-2008	566.5743342
2	759	18337	24-05-2004	1217.567824
3	2345	18337	21-05-2009	3569.544304
4	609	18403	23-01-2008	7179.828201
5	2045	18407	14-12-2004	3359.546404
6	1109	18411	25-03-2002	1679.563204
7	792	18413	24-03-2004	1510.724893
8	992	18419	24-07-2008	2435.555644
9	609	18426	31-08-2007	7179.828201
10	674	18426	28-11-2007	5709.842901
11	952	18337	17-05-2001	2099.559004
12	533	18407	17-05-2003	4197.858021
13	650	18407	15-03-2006	4122.258777
14	2388	18407	10-08-2000	2519.554804
15	1203	18411	15-04-2002	2561.554384
16	599	18414	25-07-2007	7095.829041
17	2388	18424	08-10-2007	2603.553964
18	633	18426	30-06-2009	4535.534644
19	2388	18426	10-04-2006	2519.554804
20	573	18428	09-10-2000	2349.876501
21	431	18428	14-07-2012	8231.497684
22	2352	18337	02-08-2000	2939.550604
23	2359	18337	29-07-2002	3443.545564
24	400	18337	14-05-2007	6759.832401

QUERY SETTINGS

PROPERTIES

Name: SalesFact

All Properties

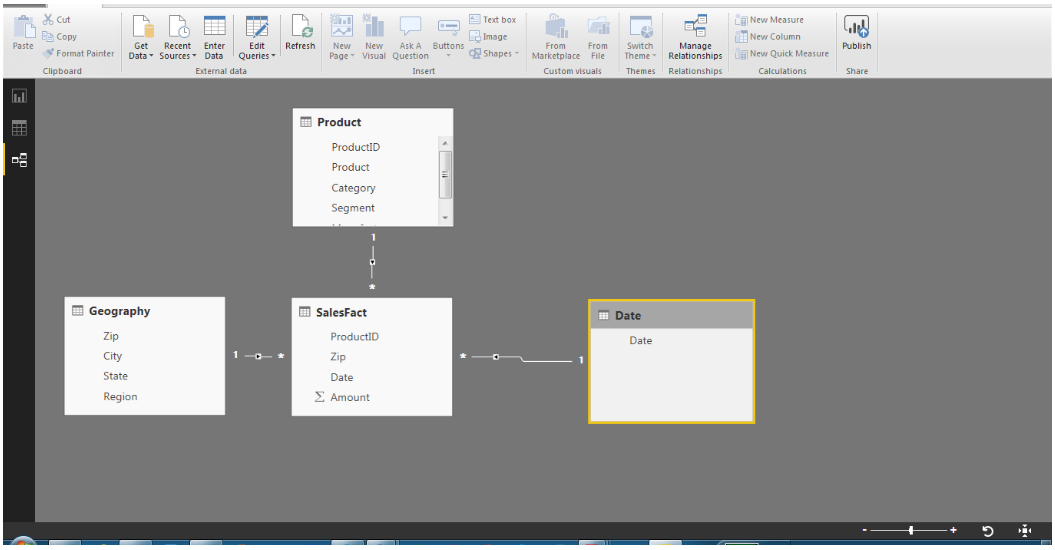
APPLIED STEPS

- Source
- Navigation
- Changed Type
- Removed Other Columns

4 COLUMNS, 100 ROWS

PREVIEW DOWNLOADED AT 13:01

As a last step to complete this example, switch to Relationships window and connect the tables as per diagram below. (Power BI should have connected tables automatically except the Date table which you can do manually).



Sort columns by other columns

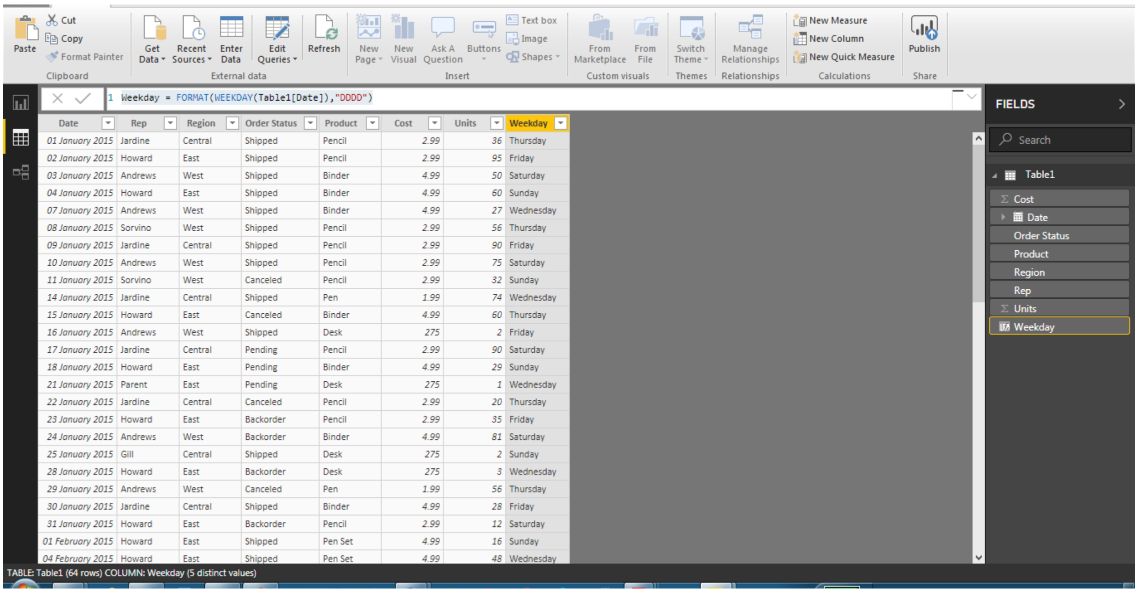
Sometimes data, especially date related like weekday names / month names can create an issue while sorting as it is done alphabetically instead of chronologically, or you may have a specific custom sort order for certain data. In such cases you need to tell Power BI, how you want the sorting of one column based on another.

Let’s work through an example.

1. Connect and load the Orders1.xlsx file from **…\PBI-Training\Day 2** folder.
2. Switch to Data view and add a new column as follows:
   a. Click on New Column button on Home tab of ribbon.
   b. In the formula bar, replace the expression Column = with the following:

      Weekday = FORMAT(WEEKDAY(Table1[Date]),"DDDD")

   c. This will add a new column with week day names as Sunday, Monday etc. to table. Check figure.

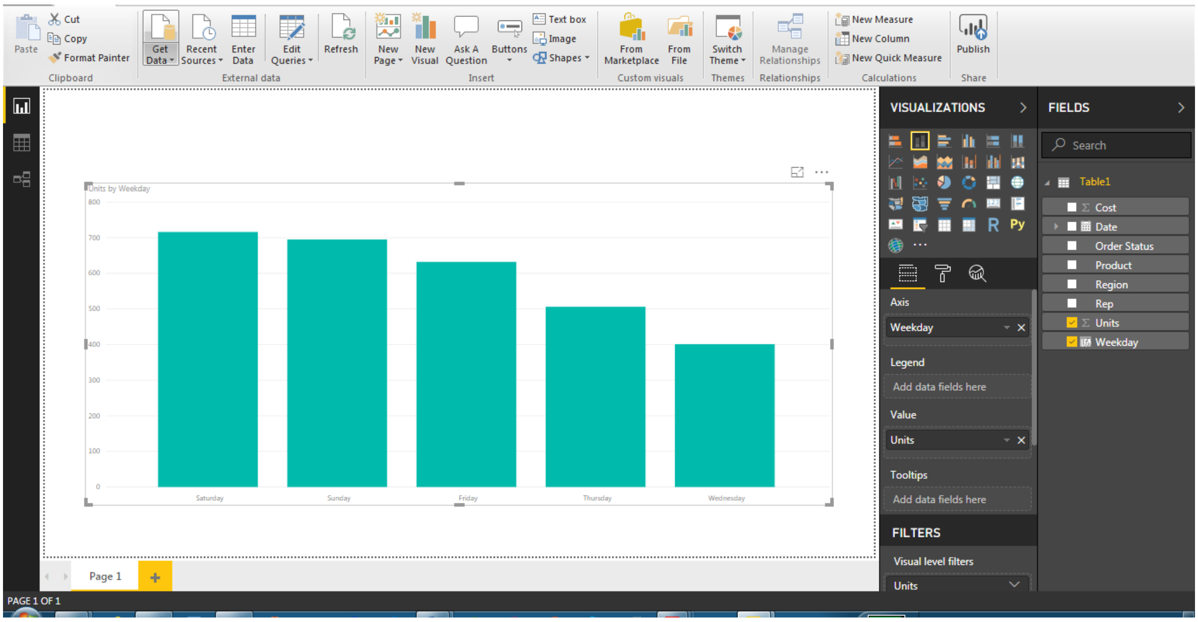


```
1 Weekday = FORMAT(WEEKDAY(Table1[Date]),"DDDD")
```

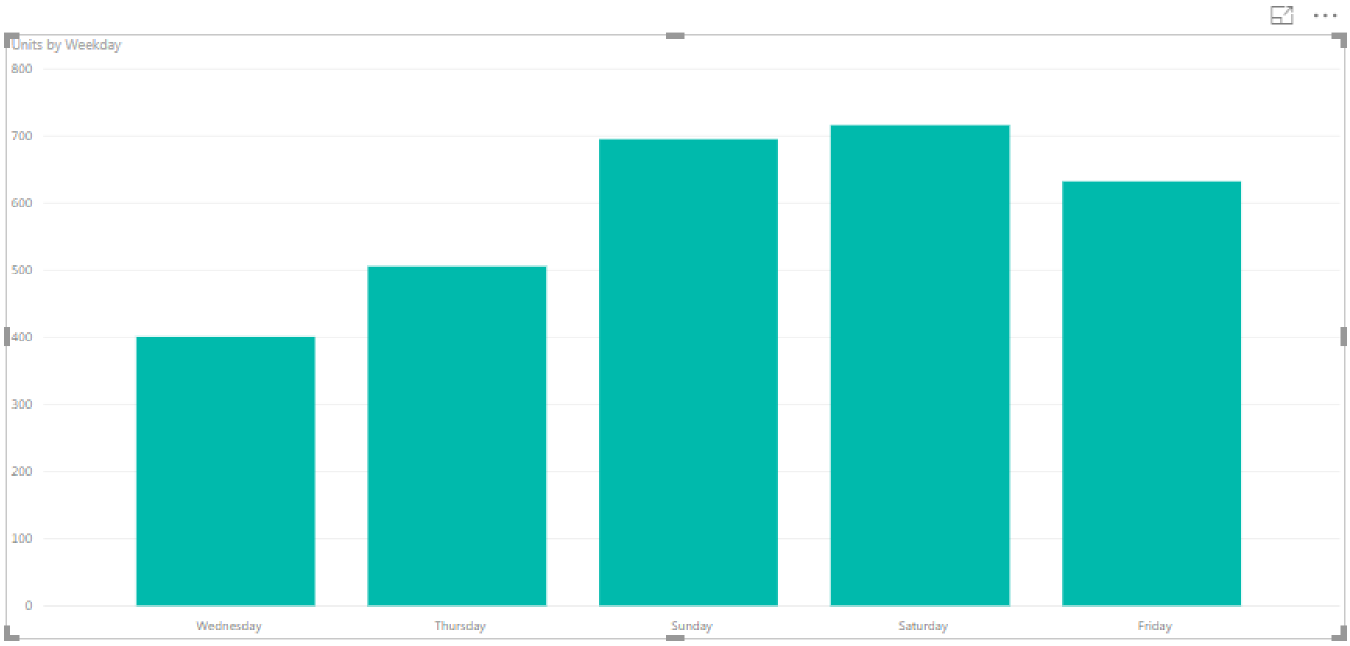
Date	Rep	Region	Order Status	Product	Cost	Units	Weekday
01 January 2015	Jardine	Central	Shipped	Pencil	2.99	36	Thursday
02 January 2015	Howard	East	Shipped	Pencil	2.99	95	Friday
03 January 2015	Andrews	West	Shipped	Binder	4.99	50	Saturday
04 January 2015	Howard	East	Shipped	Binder	4.99	60	Sunday
07 January 2015	Andrews	West	Shipped	Binder	4.99	27	Wednesday
08 January 2015	Sorvino	West	Shipped	Pencil	2.99	56	Thursday
09 January 2015	Jardine	Central	Shipped	Pencil	2.99	90	Friday
10 January 2015	Andrews	West	Shipped	Pencil	2.99	75	Saturday
11 January 2015	Sorvino	West	Canceled	Pencil	2.99	32	Sunday
14 January 2015	Jardine	Central	Shipped	Pen	1.99	74	Wednesday
15 January 2015	Howard	East	Canceled	Binder	4.99	60	Thursday
16 January 2015	Andrews	West	Shipped	Desk	275	2	Friday
17 January 2015	Jardine	Central	Pending	Pencil	2.99	90	Saturday
18 January 2015	Howard	East	Pending	Binder	4.99	29	Sunday
21 January 2015	Parent	East	Pending	Desk	275	1	Wednesday
22 January 2015	Jardine	Central	Canceled	Pencil	2.99	20	Thursday
23 January 2015	Howard	East	Backorder	Pencil	2.99	35	Friday
24 January 2015	Andrews	West	Backorder	Binder	4.99	81	Saturday
25 January 2015	Gill	Central	Shipped	Desk	275	2	Sunday
28 January 2015	Howard	East	Backorder	Desk	275	3	Wednesday
29 January 2015	Andrews	West	Canceled	Pen	1.99	56	Thursday
30 January 2015	Jardine	Central	Shipped	Binder	4.99	28	Friday
31 January 2015	Howard	East	Backorder	Pencil	2.99	12	Saturday
01 February 2015	Howard	East	Shipped	Pen Set	4.99	16	Sunday
04 February 2015	Howard	East	Shipped	Pen Set	4.99	48	Wednesday

TABLE: Table1 (64 rows) COLUMN: Weekday (5 distinct values)

d. Now switch to Report view and create a simple column chart using the Units and Weekday fields as shown below.



As you can see the data is shown in random order of days with Thursday before Wednesday for example. If you now click the three small dots (…) at top right corner of the visual, select the Sort by option and choose Weekday it will sort and display as follows.



What we need is weekday should start on Sunday and follow it on till Saturday…

To achieve this, go back to Data view, and add a numeric Weekdaynum column to the table.

a. Click on New Column button
b. In the formula bar type the following expression

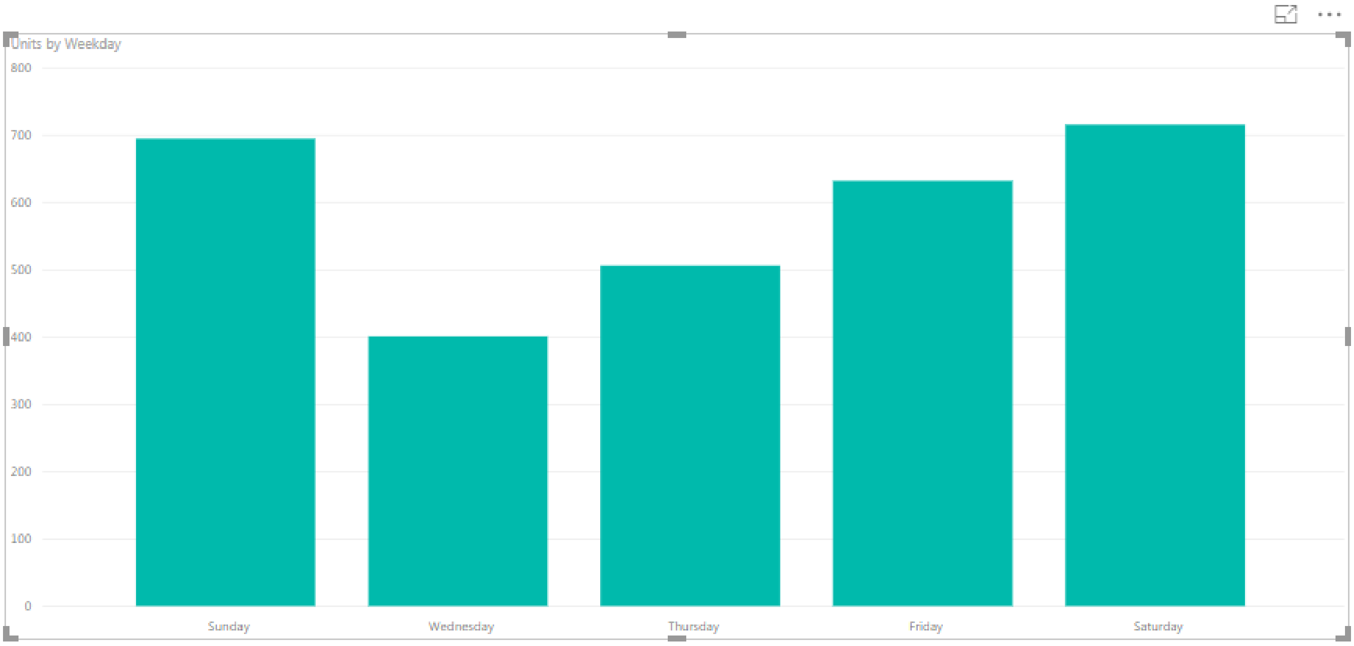
Weekdaynum = WEEKDAY(Table1[Date])

c. Right click on this new column and choose Hide in report view as we want to use this column only internally to sort the Weekday column

d. Now select the Weekday column and then switch to Modeling tab on ribbon to click the Sort by Column button

e. From the list displayed, click on Weekdaynum option.

f. Switch to report view and again select Sort Ascending option from the dot menu (…) and you should see correct sort order of weekdays as follows.

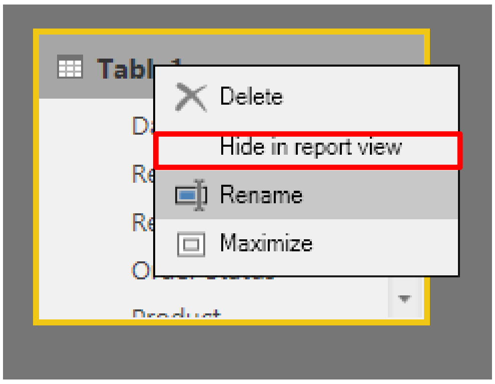


Hide fields and tables

As end users consume the data model, you may not want them to use all columns (fields) in the table, especially key fields. You can ensure this by hiding such columns and even entire tables. Let's try an example.

1. Continuing the last example, switch to Relationships window and right-click on the Table1 header of the box where you will see Hide in report view option. You won't do this in current example as it is the only table, but good to know.
2. Similarly, you can right-click on any column and find the same option to hide it in report view.

Note: In the previous demo of Sort column, you had already hidden the Weekdaynum column in the Data view window, which is another way of hiding.

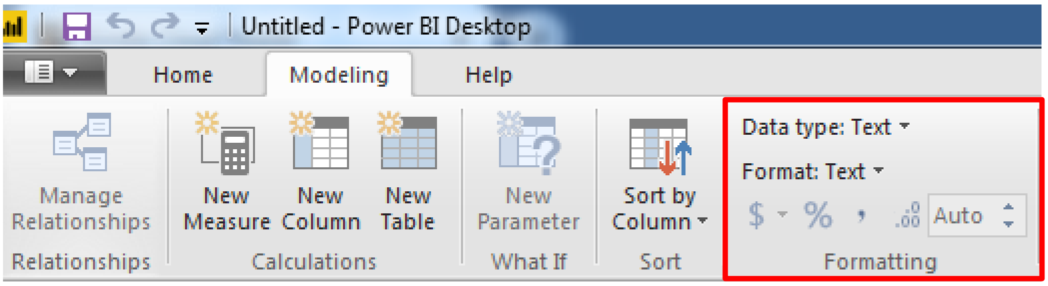


Formatting columns

While loading data, Power BI detects and chooses correct data type for each column, which sometimes may need a fix or alteration purposely. Other than that, Power BI lets you specify a display format for your columns when used in reports and visuals.

You can apply this formatting to any columns in Data view window from **Modeling -> Formating -> Format** option.

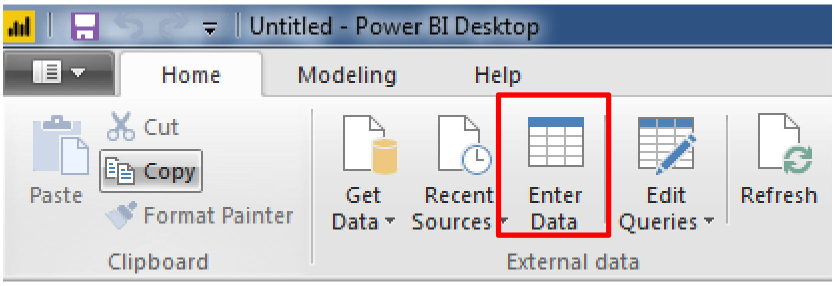
The options depend upon the data type of column you select when accessing this feature.



Manually type in data

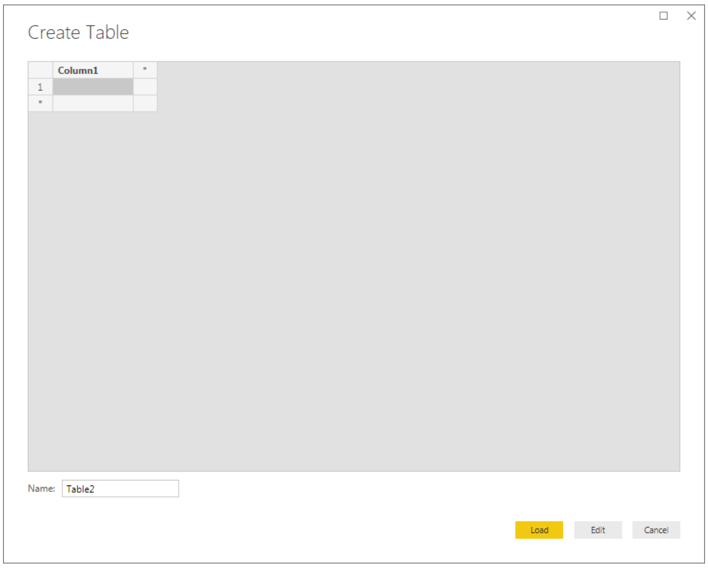
With Power BI Desktop, you can enter data directly and use that data in your reports and visualizations. For example, you can copy portions of a workbook or web page, then paste it into Power BI Desktop.

To enter data directly, select Enter Data from the Home ribbon.



Let’s try an example.

Click on Enter Data button, which opens following dialog box.



Name: Table2

Use Power Query

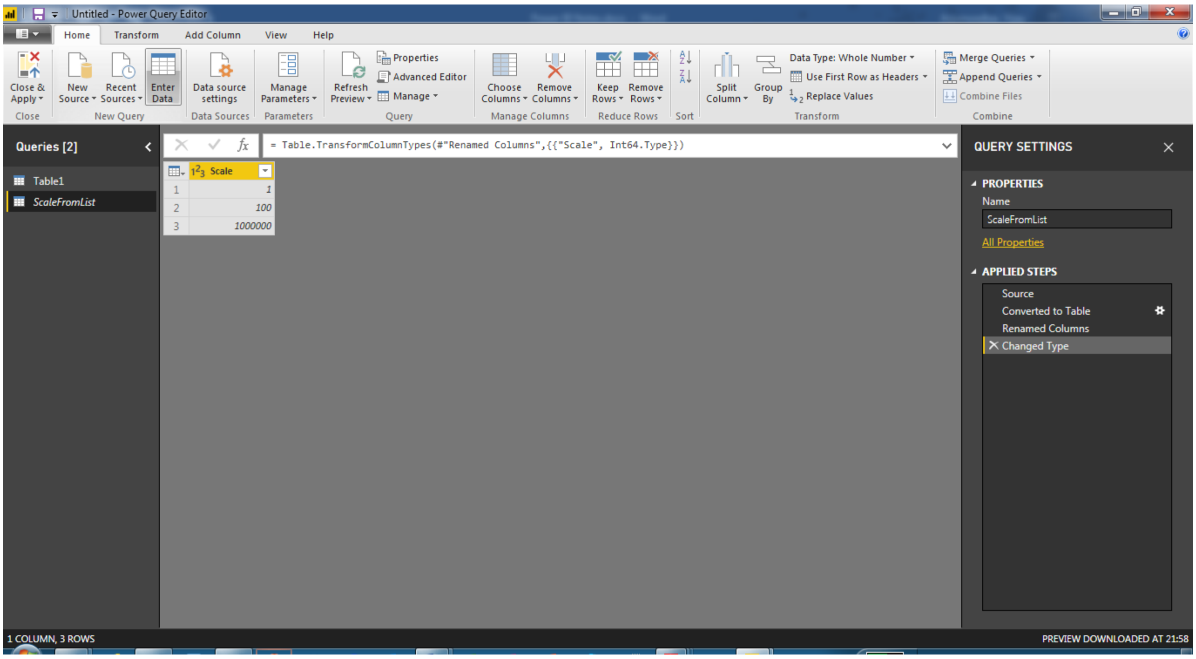
Power Query enables you to create tables using pure M. For example, you can create a table from a list, from records or by using a table construct.

Let's try an example.

1. In Power Query Editor, select **Home -> New Query -> New Source -> Blank Query.**
2. Rename the query to **ScaleFromList** and disable its loading.
3. In the formula bar, type the following code:

   = { 1, 1000, 1000000 }
4. Select List Tools Transform -> Convert -> To Table.
5. In the To Table dialog box, accept the default settings and click OK.
6. Double-click on the Column1 header and type Scale.
7. Change the data type of the Scale column to Whole Number.

The result should look like follows.



```
= Table.TransformColumnTypes(#"Renamed Columns",{{"Scale", Int64.Type}})
```

	Scale
1	1
2	100
3	1000000

Name: ScaleFromList

Applied steps:
- Source
- Converted to Table
- Renamed Columns
- Changed Type

1 COLUMN, 3 ROWS

PREVIEW DOWNLOADED AT 21:58

You may refer for more on M Language using the following link.

https://docs.microsoft.com/en-us/powerquery-m/power-query-m-reference

Create calculated columns, calculated tables, and measures

In most cases, after you import data, you need to create calculated columns, tables and measures to supplement the analysis. Power BI supports a powerful language called DAX (Data Analysis Expressions), which has similar syntax like Excel functions to allow you to create these.

You can learn more about DAX here: https://docs.microsoft.com/en-us/power-bi/desktop-quickstart-learn-dax-basics

Let’s try some DAX using following steps.

1. In a new Power BI instance, connect to the Orders1.xlsx file from **…\PBI-Training\Day 2** folder and switch to Data view.
2. Rename the Table1 to Orders.
3. We will add a few calculated columns and measures to this data set.
4. Begin by clicking the New Column button on the Home tab.
5. In the formula bar, replace the text with below:

   Year = YEAR(Orders[Date])

6. This will create a new Year column extracting year from the order date column.
7. Next create a Weekday column to store weekday number from the date.

   Weekday = WEEKDAY(Orders[Date])

8. Next create a Weekday name column from the date.

   WeekDayName = FORMAT(WEEKDAY(Orders[Date]),"DDDD")

9. Now, we will create a column based on conditions to determine order size based on Units column as follows:

   Order Size = IF(Orders[Units]<=25,"Small",IF(Orders[Units]<=50,"Medium","Large"))

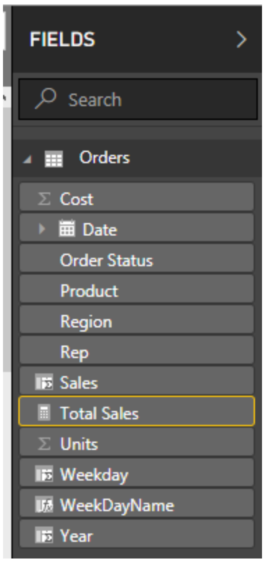
   This formula uses IF function to determine order size as Small, Medium or Large.
10. Finally, we will create a Sales column as a product of Units and Cost columns.

    Sales = Orders[Units] * Orders[Cost]

11. Next, we create a measure to calculate total sales by clicking New Measure button on the ribbon and entering following formula.

    Total Sales = SUM(Orders[Sales])

    Note that the Total Sales measure gets added to your fields list and not your table.

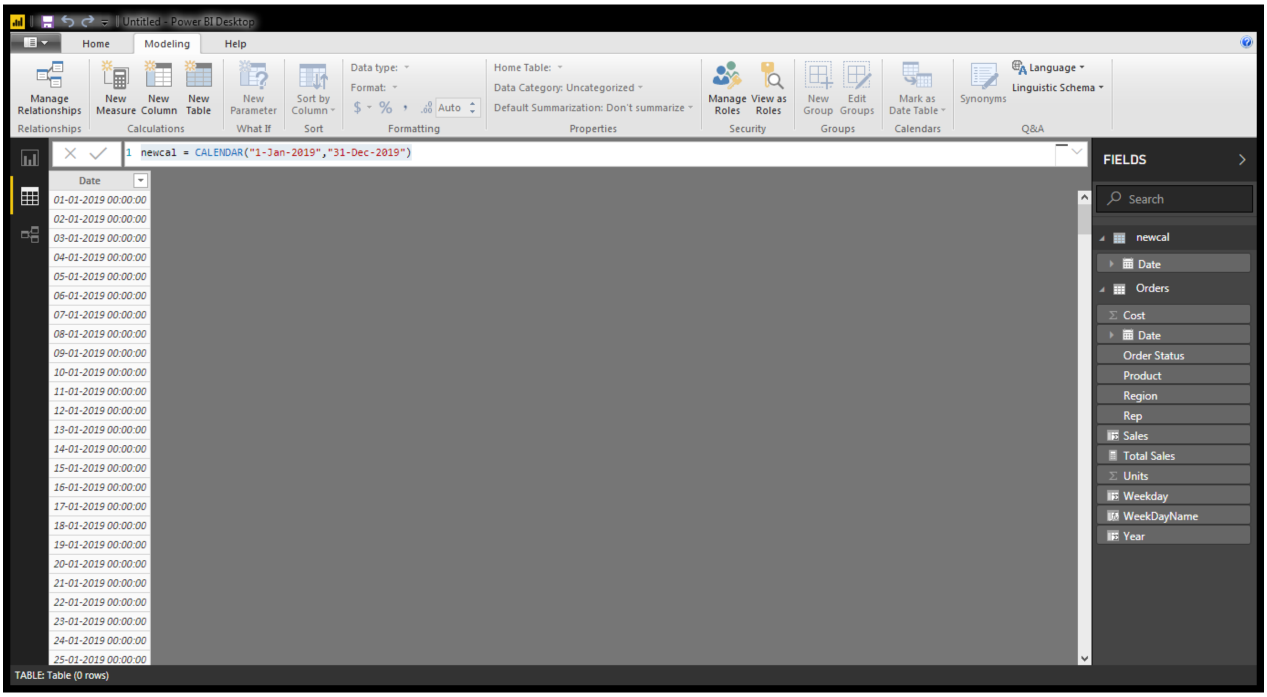


Similar to creating calculated columns and measures, we can create calculated tables such as Calendar tables and tables derived from existing tables.

1. Begin by clicking on New Table button in the Modeling Tab of ribbon.
2. Type following formula in the formula bar.

   newcal = CALENDAR("1-Jan-2019","31-Dec-2019")

3. This will create a table named newcal for year 2019 and add it to your data model as shown below.



```
newcal = CALENDAR("1-Jan-2019","31-Dec-2019")
```

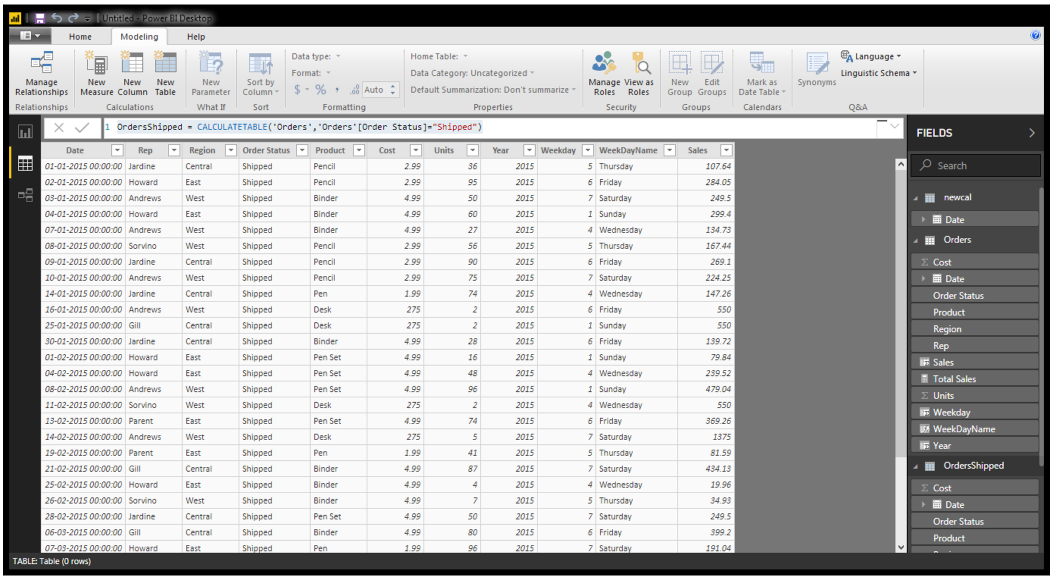
Date
01-01-2019 00:00:00
02-01-2019 00:00:00
03-01-2019 00:00:00
04-01-2019 00:00:00
05-01-2019 00:00:00
06-01-2019 00:00:00
07-01-2019 00:00:00
08-01-2019 00:00:00
09-01-2019 00:00:00
10-01-2019 00:00:00
11-01-2019 00:00:00
12-01-2019 00:00:00
13-01-2019 00:00:00
14-01-2019 00:00:00
15-01-2019 00:00:00
16-01-2019 00:00:00
17-01-2019 00:00:00
18-01-2019 00:00:00
19-01-2019 00:00:00
20-01-2019 00:00:00
21-01-2019 00:00:00
22-01-2019 00:00:00
23-01-2019 00:00:00
24-01-2019 00:00:00
25-01-2019 00:00:00

TABLE: Table (0 rows)

4. To create a table as subset of rows from an existing table, try the following.
5. Click New Table button again and enter the following in the formula bar

   OrdersShipped = CALCULATETABLE('Orders','Orders'[Order Status]="Shipped")

6. This will create a table named OrdersShipped that will contain only rows with Shipped order status.



```
OrdersShipped = CALCULATETABLE('Orders','Orders'[Order Status]="Shipped")
```

Date	Rep	Region	Order Status	Product	Cost	Units	Year	Weekday	WeekDayName	Sales
01-01-2015 00:00:00	Jardine	Central	Shipped	Pencil	2.99	36	2015	5	Thursday	107.64
02-01-2015 00:00:00	Howard	East	Shipped	Pencil	2.99	95	2015	6	Friday	284.05
03-01-2015 00:00:00	Andrews	West	Shipped	Binder	4.99	50	2015	7	Saturday	249.5
04-01-2015 00:00:00	Howard	East	Shipped	Binder	4.99	60	2015	1	Sunday	299.4
07-01-2015 00:00:00	Andrews	West	Shipped	Binder	4.99	27	2015	4	Wednesday	134.73
08-01-2015 00:00:00	Sorvino	West	Shipped	Pencil	2.99	56	2015	5	Thursday	167.44
09-01-2015 00:00:00	Jardine	Central	Shipped	Pencil	2.99	90	2015	6	Friday	269.1
10-01-2015 00:00:00	Andrews	West	Shipped	Pencil	2.99	75	2015	7	Saturday	224.25
14-01-2015 00:00:00	Jardine	Central	Shipped	Pen	1.99	74	2015	4	Wednesday	147.26
16-01-2015 00:00:00	Andrews	West	Shipped	Desk	275	2	2015	6	Friday	550
25-01-2015 00:00:00	Gill	Central	Shipped	Desk	275	2	2015	1	Sunday	550
30-01-2015 00:00:00	Jardine	Central	Shipped	Binder	4.99	28	2015	6	Friday	139.72
01-02-2015 00:00:00	Howard	East	Shipped	Pen Set	4.99	16	2015	1	Sunday	79.84
04-02-2015 00:00:00	Howard	East	Shipped	Pen Set	4.99	48	2015	4	Wednesday	239.52
08-02-2015 00:00:00	Andrews	West	Shipped	Pen Set	4.99	96	2015	1	Sunday	479.04
11-02-2015 00:00:00	Sorvino	West	Shipped	Desk	275	2	2015	4	Wednesday	550
13-02-2015 00:00:00	Parent	East	Shipped	Pen Set	4.99	74	2015	6	Friday	369.26
14-02-2015 00:00:00	Andrews	West	Shipped	Desk	275	5	2015	7	Saturday	1375
19-02-2015 00:00:00	Parent	East	Shipped	Pen	1.99	41	2015	5	Thursday	81.59
21-02-2015 00:00:00	Gill	Central	Shipped	Binder	4.99	87	2015	7	Saturday	434.13
25-02-2015 00:00:00	Howard	East	Shipped	Binder	4.99	4	2015	4	Wednesday	19.96
26-02-2015 00:00:00	Sorvino	West	Shipped	Binder	4.99	7	2015	5	Thursday	34.93
28-02-2015 00:00:00	Jardine	Central	Shipped	Pen Set	4.99	50	2015	7	Saturday	249.5
06-03-2015 00:00:00	Gill	Central	Shipped	Binder	4.99	80	2015	6	Friday	399.2
07-03-2015 00:00:00	Howard	East	Shipped	Pen	1.99	96	2015	7	Saturday	191.04

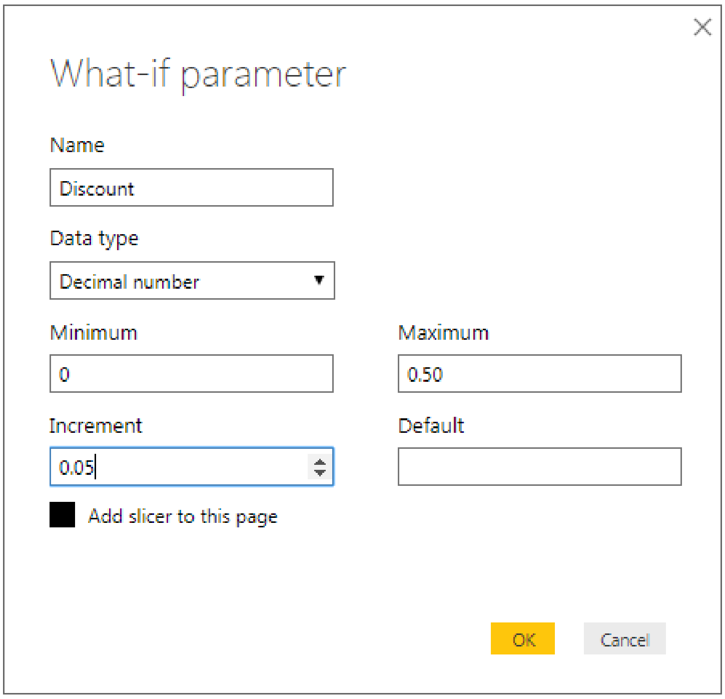
TABLE: Table (0 rows)

Use What if parameters

What if parameters interact with the variable as a slicer and visualize and quantify different key values in your reports.

Let's try an example.

1. Click on the New Parameter What if on the Modeling tab of ribbon.
2. In this example, we will create a discount parameter.
3. In the dialog box, fill in details as shown below.



What-if parameter

Name: Discount

Data type: Decimal number

Minimum: 0

Maximum: 0.50

Increment: 0.05

Default:

■ Add slicer to this page

4. Once you click OK, the parameter is added to your fields well, and a slicer also appears on the Report view.

Measure performance by using KPIs, gauges and cards

In BI, KPI (Key Performance Indicators) are very important and useful to let a user know the status of a business scenario in relation to something. E.g. actual sales v/s target sales.

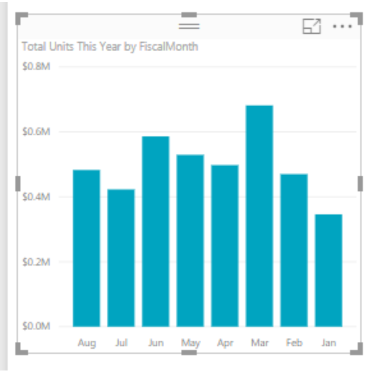
In this section, we will see how to create KPIs and present information as gauges and cards.

KPI Example

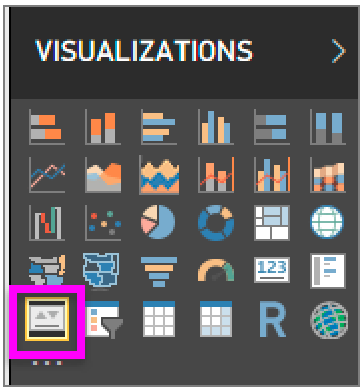
Open the **Retail Analysis Sample PBIX.pbix** from **…\PBI-Training\Day 3** folder in Power BI Desktop. We'll create a KPI that measures the progress we've made toward a sales goal.

1. Open the report in Report view and select the yellow tab to add a new page.
2. From the Fields pane, select Sales > Total Units This Year. This will be the indicator.
3. Add Time > FiscalMonth. This will be represent the trend.

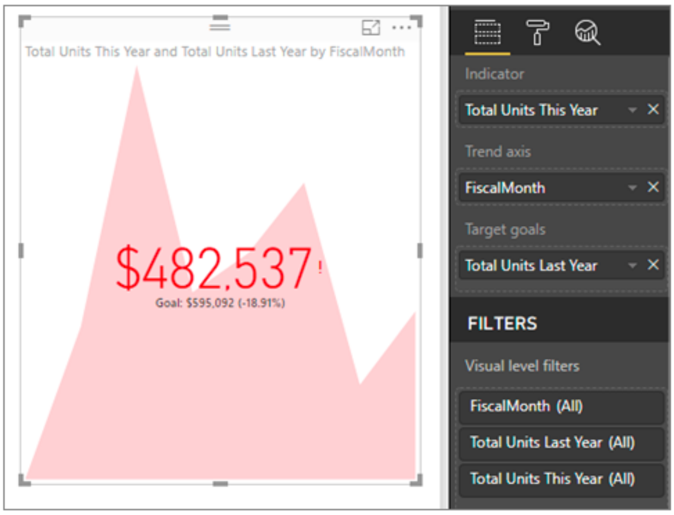
IMPORTANT: Sort the chart by FiscalMonth. Once you convert the visualization to a KPI, there is no option to sort.



4. Convert the visual to a KPI by selecting the KPI icon from the Visualization pane.



5. Add a goal. Add last year's sales as the goal. Drag **Total Units Last Year** to the Target goals field.



Total Units This Year and Total Units Last Year by FiscalMonth
$482,537
Goal: $595,092 (-18.91%)

Indicator

Total Units This Year

Trend axis

FiscalMonth

Target goals

Total Units Last Year

FILTERS

Visual level filters

FiscalMonth (All)

Total Units Last Year (All)

Total Units This Year (All)

6. Optionally, format the KPI by selecting the paint roller icon to open the Formatting pane.

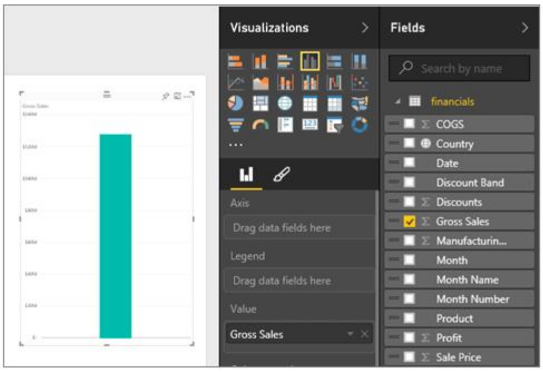
- Indicator - controls the indicator's display units and decimal places.
- Trend axis - when set to On, the trend axis is displayed as the background of the KPI visual.
- Goals - when set to On, the visual displays the goal and the distance from the goal as a percentage.
- Color coding > Direction - some KPIs are considered better for higher values and some are considered better for lower values. For example, earnings versus wait time. Typically a higher value of earnings is better versus a higher value of wait time. Select high is better and, optionally, change the color settings.

Gauges

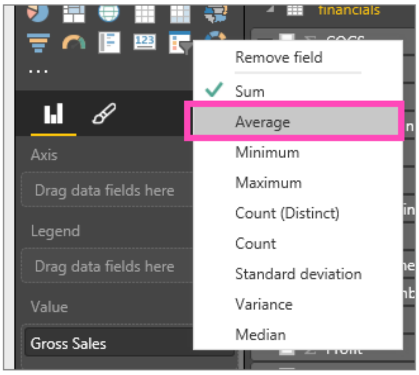
A radial gauge chart has a circular arc and displays a single value that measures progress toward a goal/KPI. The goal, or target value, is represented by the line (needle). Progress toward that goal is represented by the shading. And the value that represents that progress is shown in bold inside the arc. All possible values are spread evenly along the arc, from the minimum (left-most value) to the maximum (right-most value).

Let's try an example.

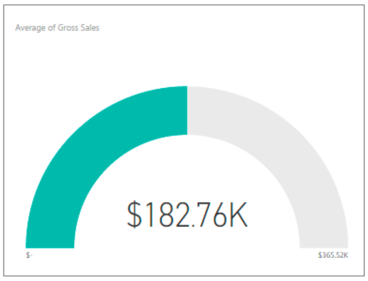
1. Open the Financial Sample.xlsx file from **…\PBI-Training\Day 3** folder.
2. In the Fields pane, select **Gross Sales**.



3. Change the aggregation to Average.

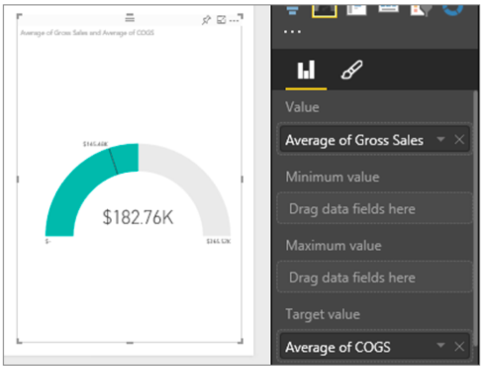


4. Select the Gauge icon to convert the Column Chart to a gauge.
5. By default, Power BI creates a Gauge chart where the current value (in this case, Average of Gross Sales) is assumed to be at the halfway point on the gauge. Since the Average Gross Sales is $182.76K, the start value (Minimum) is set to 0 and the end value (Maximum) is set to double the current value.



Set a target value

1. Drag COGS to the Target value well.
2. Change the aggregation to Average. Power BI adds a needle to represent our target value of $145.48K. Notice that we've exceeded our target.

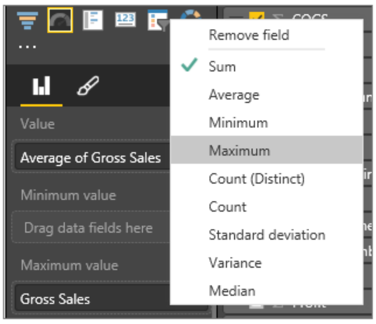


Value
Average of Gross Sales
Minimum value
Drag data fields here
Maximum value
Drag data fields here
Target value
Average of COGS

Set a maximum value

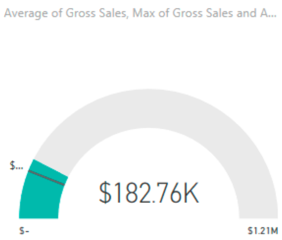
Power BI used the Value field to automatically set minimum (start) and maximum (end). But what if you want to set your own maximum value? Let's say that instead of using double the current value as the maximum possible value, you want to set it to the highest Gross Sales number in your dataset?

1. Drag Gross Sales from the Fields list to the Maximum Value well.
2. Change the aggregation to Maximum.



Value
Average of Gross Sales
Minimum value
Drag data fields here
Maximum value
Gross Sales

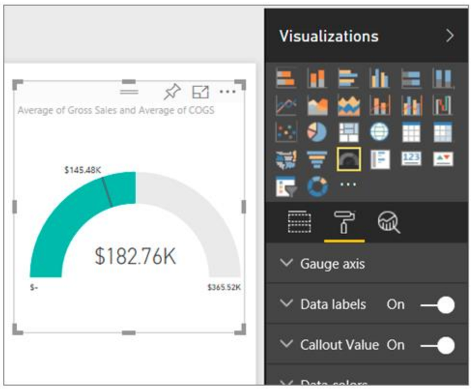
The gauge is redrawn with a new end value, 1.21 million in gross sales.



Use the format settings to manually set values

1. Remove Max of Gross Sales from the Maximum value well.

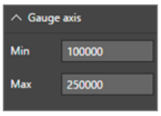
2. Open the formatting pane by selecting the paint roller icon.



Data labels On

Callout Value On

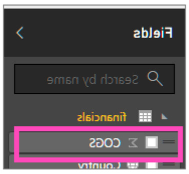
3. Expand the Gauge axis and enter values for Min and Max.



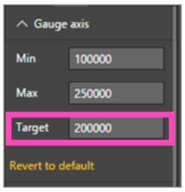
Min 100000

Max 250000

4. Remove the current target value by removing the checkmark next to COGS.



5. When the Target field appears under Gauge axis, enter a value.



Min 100000

Max 250000

Target 200000

Create hierarchies

In business data, hierarchies are common. E.g. Date data has hierarchy as follows:

- Year
    - Quarter
        - Month
            - Day

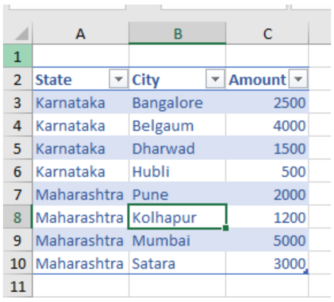
Power BI automatically recognizes date values when used in visuals and allows drill down and drill up operations.

For other scenarios, we need to define the hierarchies as required.

Let’s try an example.

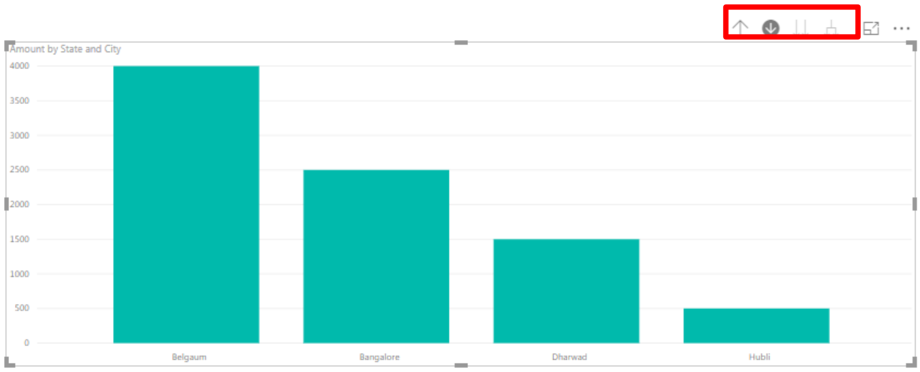
Consider following data which represents state, city and amount for some business transactions.

As we see, city belongs to state, so we can create a hierarchy to drill down from state to city when used in visuals.

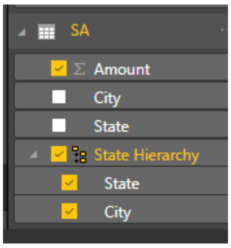


	A	B	C
1			
2	State	City	Amount
3	Karnataka	Bangalore	2500
4	Karnataka	Belgaum	4000
5	Karnataka	Dharwad	1500
6	Karnataka	Hubli	500
7	Maharashtra	Pune	2000
8	Maharashtra	Kolhapur	1200
9	Maharashtra	Mumbai	5000
10	Maharashtra	Satara	3000
11			

1. Open SA.xlsx file from …\PBI-Training\Day 3 folder and load it in Power BI.
2. On report canvas, drag the Amount field, followed by State. This create a column chart showing state wise amount division.
3. To allow drill down from State to City, create a hierarchy as follows.
    a. Right-click on State field and choose New hierarchy
    b. This creates a hierarchy named State Hierarchy and includes State field under it
    c. Now drag the City field and drop it on the State Hierarchy so it gets added under State.
    d. Next, uncheck the individual State field to remove it from visual, and instead check the State Hierarchy
    e. This will allow you to use the drill-down and drill-up option as shown in following figure.



The hierarchy appears in the Fields list as follows:



SA
- ☑ Σ Amount
- □ City
- □ State
- ☑ State Hierarchy
    - ☑ State
    - ☑ City