

# Microsoft Power Tools for Data Analysis #8

## Power Query Group By

### Notes from Video:

#### Table of Contents:

1. What Does Power Query Group By feature do?.....	2
2. Concept of “Group By” is to group records together based on a condition or criteria :.....	3
3. Where is Group By feature located in Power Query?.....	3
4. Examples of PivotTable, SUMIFS, Data Model PivotTable and SQL Code to see that Power Query Group By is VERY Similar .....	4
1) PivotTable .....	4
2) SUMIFS Function .....	4
3) DAX & Data Model PivotTable .....	5
4) SQL Code .....	5
5. Example of Simple Power Query Group By Command for Total Sales .....	6
6. Example of using Group by to get Tables of Grouped Records, or Matching Records.....	8
7. Standard Deviation Custom Column Based on Grouped Records Table .....	10
8. Joined Text Items from Unique List formula in Custom Column Based on Grouped Records Table.....	11
9. Table.Group Function Third Argument Can List Multiple New Columns with Formulas.....	12
10. Power Query Group By to Help Rank Sales for Each Product.....	13
11. Group By More Than One Column.....	15
12. Picture of the Final Three Reports Created in This Video After Data in Table is Updated .....	16
13. Table.Group 4th argument to consider Sort & Group By Consecutive Occurances .....	17
14. Duplicate or Reference a Query?.....	18
15. Gear Icon in Applied Step disappears .....	18

1. **What Does Power Query Group By feature do? :**

- 1) The Group By feature will:
  - i. Takes one or more columns and will deliver a Unique List of items for a single column or a unique set of combination of items from for two or more columns and
  - ii. Can make an aggregate calculation for each row in the delivered Unique List, or deliver all the rows associated with each row in the delivered Unique List
- 2) Group By Power Query Feature is similar to:
  - i. PivotTables
    - 1. Where we get a Unique List of items in the Rows Area and make an Aggregate Calculation for each row in that Unique List.
  - ii. SUMIFS and COUNTIFS (and other similar functions)
    - 1. Where the function makes an aggregate calculation for each row in a column filled with a Unique List.
  - iii. SQL Group By
    - 1. Where the SQL code allows you to create a unique list of items and aggerate for each row.
- 3) Example of Group By in Power Query:

**What "Group By" Does:**

**1) Extract Unique List from Column/Columns**

**Sales Table:**

Date	Product	SalesRep	Sales
2/29/2020	Quad	Chauntel	1379.01
2/25/2020	Carlota	Tyrone	1018.27
2/2/2020	Bellen	Chauntel	2162.58
2/21/2020	Quad	Gigi	73.39
2/19/2020	Sunshine	Pham	2205.1
2/23/2020	Sunshine	Sioux	112.79
2/27/2020	Carlota	Tyrone	1171.23
2/12/2020	Carlota	Chauntel	617.22
2/12/2020	Bellen	Gigi	937.34
2/15/2020	Carlota	Chauntel	1562.88
2/12/2020	Bellen	Sioux	496.82
2/11/2020	Quad	Tyrone	2177.72
2/14/2020	Sunshine	Chauntel	2394.35
2/21/2020	Sunshine	Gigi	1097.95
2/20/2020	Quad	Pham	1920.33
2/17/2020	Quad	Sioux	1108.04
2/2/2020	Carlota	Tyrone	1170.44
2/4/2020	Carlota	Chauntel	2418.86

**Group By Feature to get Report:**

Product	Total Sales	Deviation of Sales	Who Sold Product
Bellen	\$3,597	\$863	Chauntel, Gigi, Sioux
Carlota	\$7,959	\$616	Tyrone, Chauntel
Quad	\$6,658	\$821	Chauntel, Gigi, Tyrone, Pham, Sioux
Sunshine	\$5,810	\$1,061	Pham, Sioux, Chauntel, Gigi

**2) Make Aggregate Calculations on Specified Number Column (Sales)**

**OR**

**2) Make Aggregate Calculations on Specified Text Column (SalesRep)**

2. **Concept of “Group By”** is to group records together based on a condition or criteria :

Date	Product	SalesRep	Sales
2/2/2020	Bellen	Chauntel	2162.58
2/12/2020	Bellen	Gigi	937.34
2/12/2020	Bellen	Sioux	496.82

Date	Product	SalesRep	Sales
2/25/2020	Carlota	Tyrone	1018.27
2/27/2020	Carlota	Tyrone	1171.23
2/12/2020	Carlota	Chauntel	617.22
2/15/2020	Carlota	Chauntel	1562.88
2/2/2020	Carlota	Tyrone	1170.44
2/4/2020	Carlota	Chauntel	2418.86

Date	Product	SalesRep	Sales
2/29/2020	Quad	Chauntel	1379.01
2/21/2020	Quad	Gigi	73.39
2/11/2020	Quad	Tyrone	2177.72
2/20/2020	Quad	Pham	1920.33
2/17/2020	Quad	Sioux	1108.04

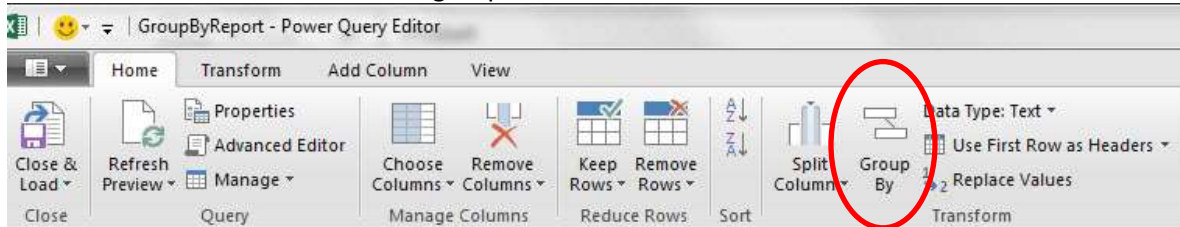
  

Date	Product	SalesRep	Sales
2/19/2020	Sunshine	Pham	2205.1
2/23/2020	Sunshine	Sioux	112.79
2/14/2020	Sunshine	Chauntel	2394.35
2/21/2020	Sunshine	Gigi	1097.95

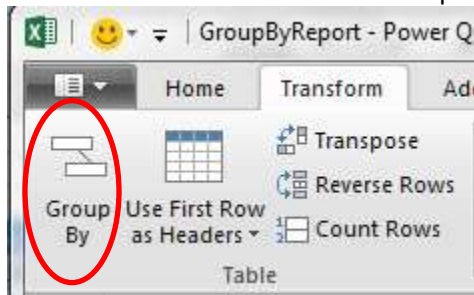
The Concept of Group By =  
Group Records together based  
a Condition or Criteria.

3. **Where is Group By feature located in Power Query?** :

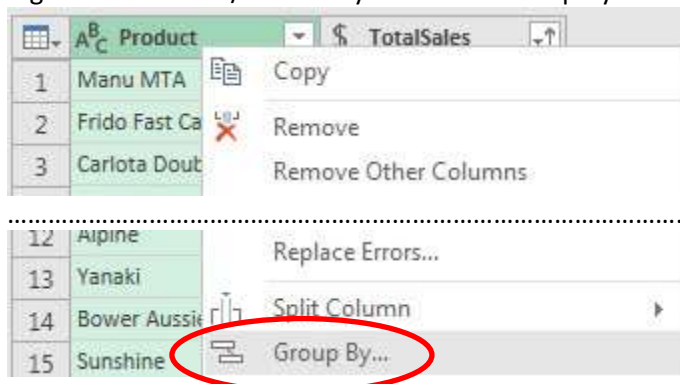
1) Home Ribbon Tab in the Transform group:



2) Transform Ribbon Tab in Table Group:



3) Right-Click Column/Columns you want to Group By:



4. Examples of PivotTable, SUMIFS, Data Model PivotTable and SQL Code to see that Power Query Group By is VERY Similar :

1) PivotTable :

Standard PivotTable: 1) Unique List of Product Names & 2) Two Aggregate Calculations:		
Product	Sum of Sale	StdDev of Sales
Bellen	\$3,597	\$863
Carlota	\$7,959	\$616
Quad	\$6,658	\$821
Sunshine	\$5,810	\$1,061
<b>Grand Total</b>	<b>\$24,024</b>	<b>\$750</b>

2) SUMIFS Function :

SUMIFS Function & Two Array Formulas: Unique List of Product Names, 3 Calculations:			
Product	Total Sales	Standard Deviation of S	Who Sold This Product?
Bellen	\$3,597	\$863	Chauntel, Gigi, Sioux
Carlota	\$7,959	\$616	Tyrone, Chauntel
Quad	\$6,658	\$821	Chauntel, Gigi, Tyrone, Pham, Sioux
Sunshine	\$5,810	\$1,061	Pham, Sioux, Chauntel, Gigi
<b>Formulas:</b>	H35: =SUMIFS(fSales[Sales],fSales[Product],G35)	I35: ={STDEV.S(IF(fSales[Product]=G35,fSales[Sales]))}	J35: {=TEXTJOIN(", ",IF(FREQUENCY(IF(IF(fSales[Product]=G35,fSales[SalesRep],"")<>"",MATCH(IF(fSales[Product]=G35,fSales[SalesRep],""),IF(fSales[Product]=G35,fSales[SalesRep],""),0)),ROW(fSales[SalesRep])-ROW(fSales[#Headers],[SalesRep])),fSales[SalesRep],""))}

3) DAX & Data Model PivotTable :

Data Model PivotTable: 1) Unique List of Product Names & 2) Three DAX Measures:			
Product	Total Sales	STDEV.S	Who Sold Product?
Bellen	\$3,597		\$863 Chauntel, Gigi, Sioux
Carlota	\$7,959		\$616 Chauntel, Tyrone
Quad	\$6,658		\$821 Chauntel, Tyrone, Gigi, Pham, Sioux
Sunshine	\$5,810		\$1,061 Chauntel, Gigi, Pham, Sioux
<b>Grand Total</b>	<b>\$24,024</b>		<b>\$750 Chauntel, Tyrone, Gigi, Pham, Sioux</b>
<b>DAX Formulas:</b>	=SUM(fSales[Sales])	=STDEV.S(fSales[Sales])	=CONCATENATEX(VALUES(fSales[SalesRep]),fSales[SalesRep],", ")

4) SQL Code :

SQL Code to get: 1) Unique List of Product Names & 2) One Aggregate Calculations:	
<pre>SELECT Product, SUM(Quantity) AS TotalSales FROM fTransactions GROUP BY Product</pre>	
Product	TotalSales
Manu MTA	\$383,134
Frido Fast Catch	\$383,318
Carlota Doublers	\$387,333
Manu LD	\$388,103
Fire Aspen	\$3,135,583
Sunspot	\$3,145,017
GelFast	\$3,145,642
Darnell Tri Fly	\$4,171,419

## 5. Example of Simple Power Query Group By Command for Total Sales

1) After importing the Excel Table into the Power Query Editor:

- i. Select the Column you want to use to extract a unique list, in our example we selected the Product Column. Right-click Column Header and click on Group By to open the Group By dialog box.
- ii. In the “New column name” text box type, the name of the new column for the aggregate calculation, in our example we typed “Total Sales”.
- iii. In the “Operation” text box, from the dropdown list select the function you want, in our example we chose the “Sum” calculation.
- iv. In the “Column” (this is the column that the function will operate on) text box, from the dropdown list select the column you want the function to operate on, in our example we chose “Sales”.
- v. Click OK.

The screenshot displays the Power Query Editor interface. On the left, a data table is visible with columns: Date, Product, SalesRep, and Sales. The 'Product' column header is circled in red with the number '1'. A 'Group By' dialog box is open in the foreground. The dialog has two tabs: 'Basic' (selected) and 'Advanced'. Below the tabs, it says 'Specify the column to group by and the desired output.' There are three input fields: 'Group by' (set to 'Product'), 'New column name' (set to 'Total Sales', circled in red with '2'), 'Operation' (set to 'Sum', circled in red with '3'), and 'Column' (set to 'Sales', circled in red with '4'). At the bottom right of the dialog are 'OK' and 'Cancel' buttons. The 'OK' button is circled in red with the number '5'. On the right side of the editor, the 'PROPERTIES' pane is visible, showing the name 'GroupBy-ProductReport'.

ABC 123	Date	ABC 123	Product	ABC 123	SalesRep	ABC 123	Sales
1	2/29/2020 12:00:00 AM	Quad	Chauntel		1379.01		
2	2/25/2020 12:00:00 AM	Carlota	Tyrone		1018.27		
3	2/2/2020 12:00:00 AM	Bellen	Chauntel		2162.58		
4	2/21/2020 12:00:00 AM	Quad	Gigi		73.39		
5	2/19/2020 12:00:00						
6	2/23/2020 12:00:00						
7	2/27/2020 12:00:00						
8	2/12/2020 12:00:00						
9	2/12/2020 12:00:00						
10	2/15/2020 12:00:00						
11	2/12/2020 12:00:00						
12	2/11/2020 12:00:00						
13	2/14/2020 12:00:00						
14	2/21/2020 12:00:00						
15	2/20/2020 12:00:00						
16	2/17/2020 12:00:00						
17	2/2/2020 12:00:00						
18	2/4/2020 12:00:00						

- vii. The Group By Process results in a table with a unique list and the aggregate calculation, Sum, as seen below:
1. The Product Column shows a unique list of products.
  2. The Total Sales column shows the aggregate amount for each Product.
  3. Applied Steps shows the name of the new step.
  4. The Formula Bar shows the Table.Group Function.
    - i. The first argument of the function lists the previous Applied Step. This is the table that the Table.Group function is acting on to make the transformation.
    - ii. The second argument contains a list, in List Syntax (curly brackets), of the column name / columns names we are trying to extract a unique list from.
    - iii. The third argument contains the calculation. This argument contains a List within a List, where each List within the Parent List has details of the new column with the calculation. The three details listed for each new column are:
      1. The new column name, in our example it shows "Total Sales".
      2. The calculations to make in each row. In our example is shows: each List.Sum([Sales])
        - i. This argument must have the word "each", in lowercase, before the calculation so that the calculation can be made in each row.
        - ii. Following the word "each" is the calculation or value you want to make in each row.
          1. Some of the aggregate functions available are:
            - i. List.Sum
            - ii. List.StandardDeviation
            - iii. List.Min
          2. Other possible functions could be: Text.Combine or List.Distinct
      3. The Data Type, in our example it shows: type number, for Decimal Data Type.
      4. The Total Sales column

1) The Product Column shows a unique list of products.

4) Table.Group: 1) Previous Step, 2) Unique List Column, 3) List within List of New Columns & Calculations.

2) The Total Sales column shows the aggregate amount for each Product.

3) Applied Step

The screenshot displays a data tool interface. At the top, the formula bar contains the following DAX-like function: `= Table.Group(Source, {"Product"}, [{"Total Sales", each List.Sum([Sales]), type number}])`. Below the formula bar is a data table with the following content:

	Product	Total Sales
1	Quad	6658.49
2	Carlota	7958.9
3	Bellen	3596.74
4	Sunshine	6658.49

On the right side, the 'Query Settings' panel is visible, showing the 'APPLIED STEPS' section with 'Grouped Rows' selected.

## 6. Example of using Group by to get Tables of Grouped Records, or Matching Records

- 1) To edit our Group By calculations, we can Double Click the “Grouped Rows” step in the Applied Step list to open up the Group By dialog box.
  - i. The Advanced dialog button allows us to:
    1. Group By more than one column (we will not do that in this example).
    2. Add new columns with aggregate calculations.
  - ii. By Clicking the “Add Aggregation” button we can add a new column with a new calculation.
  - iii. The New column name should be typed out as “All Rows”. By Choosing the Operation “All Rows”, we can return a table of grouped or matching records for each row in our Unique List Table. The advantage of choosing “All Rows” is that once we have a table of grouped records for each row in our Unique List Table, we can make a calculation that we would like. This is especially important because the list of calculations in the Operations dropdown is limited.

**Group By** ✕

Basic  Advanced

Specify the columns to group by and one or more outputs:

Group by  
Product ▾  
Add grouping

New column name	Operation	Column
Total Sales	Sum ▾	Sales ▾
All Rows	All Rows ▾	▾

Add aggregation

OK Cancel



- 2) After we click OK in the Group By dialog box, we can see (in the picture below) that a new column named “All Rows”, which contains a table of grouped records for each row in our Unique List Table. Notice that for Carlota, the table of grouped records can be seen in the lower left corner of the Power Query Editor. The advantage of the “All Rows” option is that we can now add Custom Columns to make calculations based only on the records in the table of grouped records for each row in our Unique List Table.

The screenshot shows the Power Query Editor interface. At the top, the formula bar contains the following M code:

```
= Table.Group(Source, {"Product"}, {"Total Sales", each List.Sum([Sales]), type number}, {"All Rows", each _, type table})
```

Below the formula bar is a summary table with the following data:

	Product	Total Sales	All Rows
1	Quad	6658.49	Table
2	Carlota	7958.9	Table
3	Bellen	3596.74	Table
4	Sunshine	6658.49	Table

In the lower-left corner, a detailed view of the source data for the 'Carlota' group is shown:

Date	Product	SalesRep	Sales
2/25/2020 12:00:00 AM	Carlota	Tyrone	1018.27
2/27/2020 12:00:00 AM	Carlota	Tyrone	1171.23
2/12/2020 12:00:00 AM	Carlota	Chauntel	617.22
2/15/2020 12:00:00 AM	Carlota	Chauntel	1562.88

On the right side, the 'Query Settings' pane is visible, showing the query name 'GroupBy-ProductReport' and the 'APPLIED STEPS' list, which includes 'Source' and 'Grouped Rows'.

## 7. Standard Deviation Custom Column Based on Grouped Records Table

- 1) To calculate Standard Deviation as a new column in our report, click on the Add Column Ribbon Tab, then in the General group, click on the Custom Column button. Then in the Custom Column dialog box, you can create the “New column name” and “Custom column formula”, as seen below. Notice that inside the List.StandardDeviation function, we have to list the column name with the Table in each row, “[All Rows]”, and then we have to list the column name that the function should work with, “[Sales]”. In the second picture below, we can see the new column added to our report calculating the standard deviation.

Custom Column

New column name  
Standard Deviation S

Custom column formula:  
= Number.Round(List.StandardDeviation([All Rows][Sales]),2)

Available columns:  
Product  
Total Sales  
All Rows

<< Insert

Learn about Power Query formulas

✓ No syntax errors have been detected.

OK Cancel

= Table.AddColumn("#Grouped Rows", "Standard Deviation S", each Number.Round(List.StandardDeviation([All Rows][Sales]),2))

	Product	Total Sales	All Rows	Standard Deviation S
1	Quad	6658.49	Table	821.26
2	Carlota	7958.9	Table	615.74
3	Bellen	3596.74	Table	863.14
4	Sunshine	6658.49	Table	1355.43

Query Settings ×

PROPERTIES

Name  
GroupBy-ProductReport

All Properties

APPLIED STEPS

Source  
Grouped Rows ✱  
Added Custom ✱

## 8. Joined Text Items from Unique List formula in Custom Column Based on Grouped Records Table

- 1) Using a Custom Column, you can create a formula, with the goal of listing a unique list of SalesRep names for each product. We can do this because for each row in our report we have a table of grouped records that lists a SalesRep name for each row in the table of grouped records. Using the Power Query function List.Distinct, we can get a unique list of names from the SalesRep column. Then we can use the result of the List.Distinct function in the first argument of the Text.Combine function, then list our delimiter in the second argument. The two pictures below show the formula in the Custom Column dialog box, and the second picture shows the final report.

The image shows two screenshots illustrating the process of creating a custom column. The first screenshot is the 'Custom Column' dialog box. It has a title 'Custom Column' and a field for 'New column name' containing 'Who Sold Products?'. Below it is a field for 'Custom column formula:' containing the formula: `=Text.Combine(List.Distinct([All Rows][SalesRep]),", ")`. A red box highlights the dialog box with the text '1) Custom Column dialog box, where we create our formula.' A red arrow points from this box to the dialog box. The second screenshot shows a report table with a new column. The table has columns: 'Product', '1.2 Total Sales', 'All Rows', 'Standard Deviation S', and 'Who Sold Products?'. The 'Who Sold Products?' column contains unique lists of sales rep names for each product. A red box highlights this column with the text '2) New column with calculation.' A red arrow points from this box to the column in the table.

1) Custom Column dialog box, where we create our formula.

2) New column with calculation.

	ABC 123 Product	1.2 Total Sales	All Rows	ABC 123 Standard Deviation S	ABC 123 Who Sold Products?
1	Quad	6658.49	Table	821.26	Chauntel, Gigi, Tyrone, Pham, Sioux
2	Carlota	7958.9	Table	615.74	Tyrone, Chauntel
3	Bellen	3596.74	Table	863.14	Chauntel, Gigi, Sioux
4	Sunshine	6658.49	Table	1355.43	Pham, Sioux, Chauntel, Gigi

## 9. Table.Group Function Third Argument Can List Multiple New Columns with Formulas :

- 1) Rather than use the Group By feature and the "All Rows" option to create a column with a table of grouped records for each row in our Unique List Table, and then create Custom Columns, you can edit the M Code and the Table.Group function's third argument. The third argument contains a List within a List, where each List within the Parent List has details of the new column with the calculation. The three details you must list for each calculation are: 1) The name of the new column, 2) The formula and 3) the Data Type. The below picture shows how we can make the calculation for Total Sales, Standard Deviation and the Unique List of Sales Rep Who Sold Each Product with a single Table.Group function and a single Grouped Rows Applied Steps.

	Product	Total Sales	Standard Deviation S	Who Sold Products?
1	Quad	6658.49	821.26	Chauntel, Gigi, Tyrone, Pham, Sioux
2	Carlota	7958.9	615.74	Tyrone, Chauntel
3	Bellen	3596.74	863.14	Chauntel, Gigi, Sioux
4	Sunshine	6658.49	1355.43	Pham, Sioux, Chauntel, Gigi

```
= Table.Group(Source, {"Product"}, {"Total Sales", each List.Sum([Sales]), type number}, {"Standard Deviation S", each Number.Round(List.StandardDeviation([Sales]),2), type number}, {"Who Sold Products?", each Text.Combine(List.Distinct([SalesRep]),", " ), type text})
```

## 10. Power Query Group By to Help Rank Sales for Each Product

- 1) Because there is no ranking function in Power Query, we have to go through a number of steps to create a rank. Here are the steps:
  - i. Import the fSalesPowerQuery table, name the query and then group by the Product column to get Total Sales for each Product.
  - ii. Using the Filter dropdown arrow for the Total Sales column, Sort Descending, Z to A.
  - iii. In the Add Column Ribbon Tab, click on the dropdown for Index Column, and then click on "From 1".
  - iv. You should see the four following steps as shown in the Applied argument contains a List within a List, where each List within the parent List has details of the new column with the calculation. Steps list below:

The screenshot shows the Power Query Editor window titled "RankProductSales - Power Query Editor". The ribbon is set to "Add Column", and the "Index Column" dropdown is selected. A red box highlights the "Index Column" dropdown with the text "3) Index Column, From 1". Below the ribbon, a data table is displayed with columns: Product, TotalSales, and Index. The data is sorted by TotalSales in descending order. A red arrow points from the "Index" column header to a red box containing the text "2) Sort Total Sales Descending." On the right side, the "Query Settings" pane is visible, showing the "APPLIED STEPS" list: Source, Grouped Rows, Sorted Rows, and Added Index (highlighted).

	Product	TotalSales	Index
1	Carlota	7958.9	1
2	Sunshine	6658.49	2
3	Quad	6658.49	3
4	Bellen	3596.74	4

- v. If ranking ties with different numbers is okay, then all you need to do is to Close and Load at this point. But if you want tie values to have the same rank, then we need to create a few extra steps.

- 2) Then select the Total Sales column and use the Group By feature. In the Group By dialog box create two new columns, one for “All Rows” and one for the Min function on the Index Column, as seen here:

- 3) The conceptual trick to this is that the Grouped Records for tie values will have different Index numbers, but if we use the Min function to pick out the smallest Index number, we can get the same rank for tie values. You can see in the first picture below, the table with the match records for the value 6658.49, lists the numbers 2 and 3. But the Min function will deliver only the 2 value. In the second picture, you can see that after we expand the All Rows column, the rank of 2 is repeated for each tie and the value, 3596.74 has a rank of 4.

	1.2 TotalSales	All Rows	1.2 Rank
1	7958.9	Table	1
2	6658.49	Table	2
3	3596.74	Table	4

1) The table with the match records for the value 6658.49, lists the numbers 2 and 3.

Product	TotalSales	Index
Sunshine	6658.49	2
Quad	6658.49	3

2) After we expand the All Rows column, the rank of 2 is repeated for each tie and the value, 3596.74 has a rank of 4.

	A <sup>B</sup> <sub>C</sub> Product	1.2 TotalSales	1.2 Rank
1	Carlota	7958.9	1
2	Sunshine	6658.49	2
3	Quad	6658.49	2
4	Bellen	3596.74	4

### 11. Group By More Than One Column :

- 1) You can use the Group By feature on more than one column. When you do this, you create a Unique List of Combination of Items from the two columns, which serve as the two criteria for the aggregate calculations.
- 2) You can Group By two or more columns using one of three methods:
  - i. Highlight the two or more columns in the table and then click the Group By feature.
  - ii. Select the Advanced dialog button in the Group By dialog box and then select your columns.
  - iii. You can type the column names in quotes and in List Format in the first argument of the Table.Group function, as seen here:

```
= Table.Group(Source, {"Product", "SalesRep"},  
{{"Total Sales", each List.Sum([Sales]), type number},  
{{"Standard Deviation S", each Number.Round(List.StandardDeviation([Sales]),2), type number}})
```

12. Picture of the Final Three Reports Created in This Video After Data in Table is Updated

Sales Table:

Date	Product	SalesRep	Sales
2/22/2020	Quad	Chauntel	1795.01
2/6/2020	Carlota	Tyrone	78.81
2/18/2020	Bellen	Chauntel	2264.72
1/8/2020	Quad	Gigi	146.28
1/23/2020	Sunshine	Pham	1373.96
1/15/2020	Sunshine	Sioux	357.17
2/18/2020	Carlota	Tyrone	229.06
1/18/2020	Carlota	Chauntel	747.34
1/29/2020	Bellen	Gigi	2363.2
2/23/2020	Carlota	Chauntel	1598.23
1/25/2020	Bellen	Sioux	330.27
2/16/2020	Quad	Tyrone	1758.34
1/30/2020	Sunshine	Chauntel	1603.92
1/22/2020	Sunshine	Gigi	372.35
2/2/2020	Quad	Pham	348.87
1/25/2020	Quad	Sioux	2169.12
2/15/2020	Carlota	Tyrone	57.5
1/31/2020	Carlota	Chauntel	1762.01
1/6/2020	Quad	Chauntel	289.61
1/19/2020	Carlota	Tyrone	1252.39
2/21/2020	Bellen	Chauntel	2156.69
1/25/2020	Quad	Gigi	221.58
2/28/2020	Sunshine	Pham	83.72
2/22/2020	Sunshine	Sioux	1261.13
2/19/2020	Carlota	Tyrone	136.11
1/30/2020	Carlota	Chauntel	242.12
2/3/2020	Bellen	Gigi	110.12
2/11/2020	Carlota	Chauntel	109.17
2/18/2020	Bellen	Sioux	432.37
2/5/2020	Quad	Tyrone	622.97
1/21/2020	Sunshine	Chauntel	191.11
1/31/2020	Sunshine	Gigi	1883.49
2/3/2020	Quad	Pham	1558.09
1/10/2020	Quad	Sioux	358.61
1/15/2020	Carlota	Tyrone	1600.08
1/25/2020	Carlota	Chauntel	1617.58

Group By Feature to get Report:

Product	Total Sale	Standard Deviation S	Who Sold Products?
Quad	15950.7	901.56	Chauntel, Gigi, Tyrone, Pham, Sioux
Carlota	17174.77	796.93	Tyrone, Chauntel, Pham, Sioux
Bellen	10368.1	988.5	Chauntel, Gigi, Sioux, Tyrone
Sunshine	8883.68	652.41	Pham, Sioux, Chauntel, Gigi, Tyrone

Group By To Rank:

Product	TotalSale	Rank
Carlota	17174.77	1
Quad	15950.7	2
Bellen	10368.1	3
Sunshine	8883.68	4

Group By Two Columns:

Product	SalesRep	Total Sales	Standard Deviation S
Quad	Chauntel	2173.66	932.45
Carlota	Tyrone	6239.08	717.23
Bellen	Chauntel	4421.41	76.39
Quad	Gigi	2589.54	1177.1
Sunshine	Pham	1457.68	912.34
Sunshine	Sioux	1618.3	639.2
Carlota	Chauntel	6239.08	749.47
Bellen	Gigi	4421.41	1199.08
Bellen	Sioux	762.64	72.2
Quad	Tyrone	4538.78	796.14
Sunshine	Chauntel	3115.98	682.02
Sunshine	Gigi	2618.63	875.23
Quad	Pham	4120.99	946.16
Quad	Sioux	2527.73	1280.22
Carlota	Pham	2384.6	0
Carlota	Sioux	2312.01	1194.68
Bellen	Tyrone	762.64	46.51
Sunshine	Tyrone	73.09	0



### 13. Table.Group 4th argument to consider Sort & Group By Consecutive Occurances

- 1) Table.Group has five arguments:
  - i. Microsoft Help:
    1. Table.Group(table as table, key as any, aggregatedColumns as list, optional groupKind as nullable number, optional comparer as nullable function) as table
  - ii. Excelisfun description:
    1. Table.Group(Table, GroupByColumns, ListAsListWithAggregations(NameAggregationType), GroupKind) \*\* The is a 5th argument, not discussed here.
  - iii. The fourth argument is GroupKind. There are two options for this:
    1. GroupKind.Global (default)
      - i. This does not consider sort or order in the Group By Column.
      - ii. A Unique List is determined from the Group By Column and then the Group By Aggregation is performed.
    2. GroupKind.Local
      - i. A local group is formed from a consecutive sequence of rows from an input table with the same key value. The sort determines how the items are grouped.
      - ii. The Group By Aggregation is performed based on consecutive occurrences of items in the Group By Column.
- 2) Example of Group By a Unique List & Group By Consecutive Occurances:
  - i. Consecutive Occurrence implies a certain sort. So the Group By for Consecutive Occurrence requires that you sort the column the way you want it before doing the Group By GroupKind.Local. You can add a step in the query to assure the correct order by sorting the Date Column.

```
= Table.Group(Source, {"TypeDay"}, {"Total Sales", each List.Sum([Sales]), type number}, {"NumberDays", each Table.RowCount(_), type number}))
```

Date	TypeDay	Sales
Friday, August 03, 2018	Workday	2386.5
Saturday, August 04, 2018	Workday	1048.36
Sunday, August 05, 2018	Weekend	106.75
Monday, August 06, 2018	Weekend	2880.13
Tuesday, August 07, 2018	Holiday	0
Wednesday, August 08, 2018	Workday	2462.57
Thursday, August 09, 2018	Donation	1993.07
Friday, August 10, 2018	Donation	1278.86
Saturday, August 11, 2018	Workday	1599.91
Sunday, August 12, 2018	Weekend	16
Monday, August 13, 2018	Weekend	1666.92
Tuesday, August 14, 2018	Workday	2094.19
Wednesday, August 15, 2018	Workday	108.03
Thursday, August 16, 2018	Workday	2577.19
Friday, August 17, 2018	Holiday	15.2
Saturday, August 18, 2018	Holiday	1904.77
Sunday, August 19, 2018	Holiday	0

GroupKind.Global (Default)  
Group By a Unique List

TypeDay	Total Sales	NumberDays
Workday	22268.63	14
Weekend	4669.8	4
Holiday	1919.97	5

GroupKind.Local (Default)  
Group By Consecutive Occurances

TypeDay	Total Sales	Count Rows
Workday	3434.86	2
Weekend	2986.88	2
Holiday	0	1
Workday	2462.57	1
Donation	3271.93	2
Workday	1599.91	1
Weekend	1682.92	2
Workday	4779.41	3
Holiday	1919.97	4
Workday	3316.61	2
Donation	3039.9	2
Workday	363.44	1

```
= Table.Group(Source, {"TypeDay"}, {"Total Sales", each List.Sum([Sales]), type number}, {"Count Rows", each Table.RowCount(_), type number}), GroupKind.Local)
```

Adding a Step to Assure the Sort Order

#### 14. Duplicate or Reference a Query?

- 1) You can right-click a query and copy the query in two ways:
  - i. **Duplicate** will copy the full M Code and create a new query. This is good when you want to copy the full code and change a few elements in the code.
  - ii. **Reference** will simply reference the original query. This is good when you do not want to alter nay of the code and you do not want to link to an changes in the first query. You can think of reference as a cell reference in a formula, where the cell reference will automatically update when the source item changes.

#### 15. Gear Icon in Applied Step disappears :

- 1) When you use a dialog box or a feature from the Power Query Dialog Box, the Applied Step that is created often has a Gear Icon next to the Applied Step name. This Gear Icon in Applied Step disappears when you change the M code to a legitimate M Code structure, but there is no equivalent option in the functions dialog box.