Microsoft Power Tools for Data Analysis #03:

Power Query Introduction: Transform & Load Data in Excel & Power BI

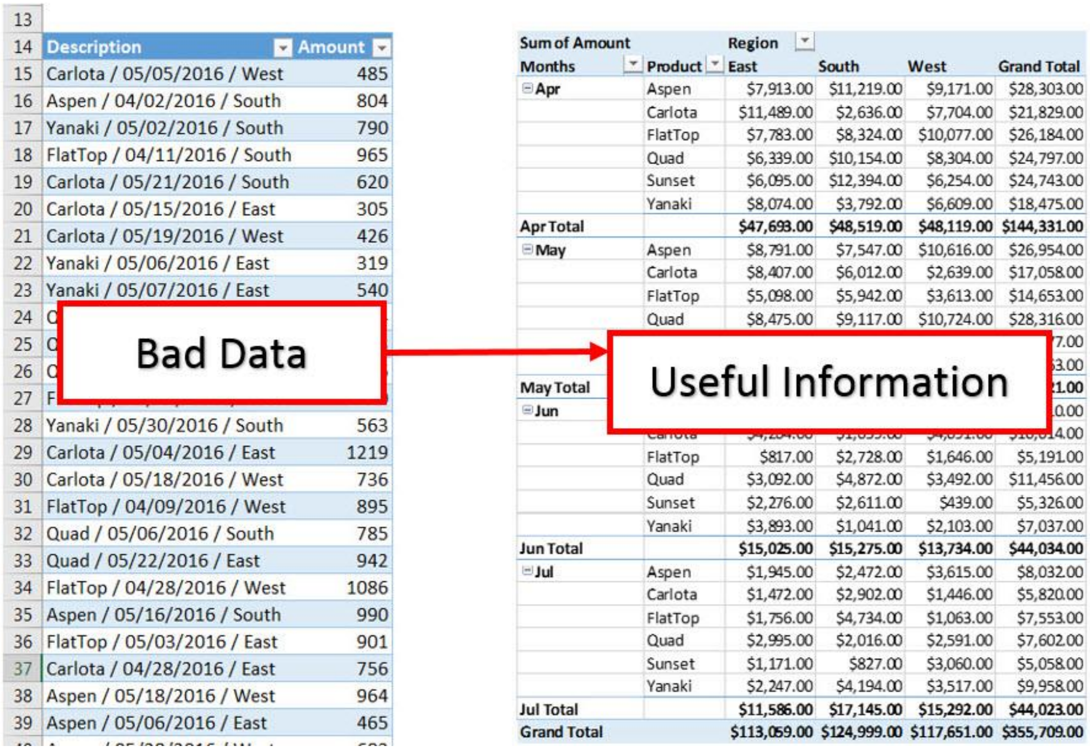
Introduction to Power Query: ETL Master Tool

Notes from Video:

Table of Contents:

In this video there are three examples. The written Description and pictures from the video are shown below.

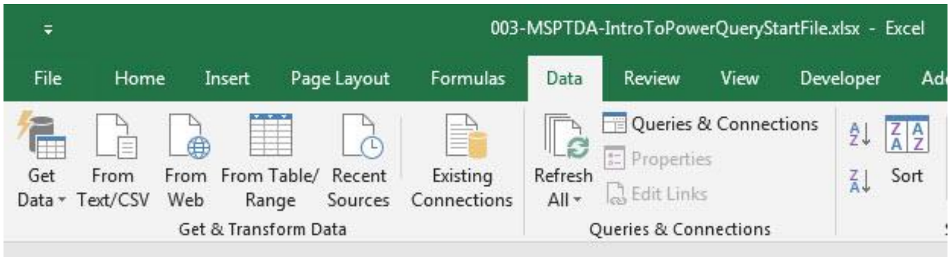
1. **Example 1: Clean and Transform Data in Excel. Look at Excel Power Query User Interface & M Code. Look at Locations to Load Data. Edit, Delete and Add Steps to Power Query Solution**.
   1) Goal is to go from a non-proper data set into a proper data set that will allow us to create a Standard PivotTable Report, as seen in this picture:



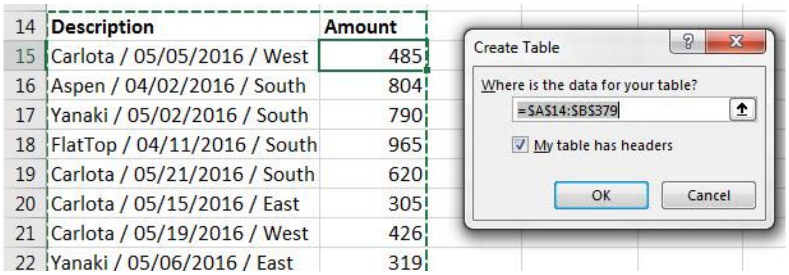
13		
14	Description	Amount
15	Carlota / 05/05/2016 / West	485
16	Aspen / 04/02/2016 / South	804
17	Yanaki / 05/02/2016 / South	790
18	FlatTop / 04/11/2016 / South	965
19	Carlota / 05/21/2016 / South	620
20	Carlota / 05/15/2016 / East	305
21	Carlota / 05/19/2016 / West	426
22	Yanaki / 05/06/2016 / East	319
23	Yanaki / 05/07/2016 / East	540
24	Q	
25	Q	
26	Q	
27	F	
28	Yanaki / 05/30/2016 / South	563
29	Carlota / 05/04/2016 / East	1219
30	Carlota / 05/18/2016 / West	736
31	FlatTop / 04/09/2016 / West	895
32	Quad / 05/06/2016 / South	785
33	Quad / 05/22/2016 / East	942
34	FlatTop / 04/28/2016 / West	1086
35	Aspen / 05/16/2016 / South	990
36	FlatTop / 05/03/2016 / East	901
37	Carlota / 04/28/2016 / East	756
38	Aspen / 05/18/2016 / West	964
39	Aspen / 05/06/2016 / East	465

Sum of Amount		Region			
Months	Product	East	South	West	Grand Total
Apr	Aspen	$7,913.00	$11,219.00	$9,171.00	$28,303.00
	Carlota	$11,489.00	$2,636.00	$7,704.00	$21,829.00
	FlatTop	$7,783.00	$8,324.00	$10,077.00	$26,184.00
	Quad	$6,339.00	$10,154.00	$8,304.00	$24,797.00
	Sunset	$6,095.00	$12,394.00	$6,254.00	$24,743.00
	Yanaki	$8,074.00	$3,792.00	$6,609.00	$18,475.00
Apr Total		**$47,693.00**	**$48,519.00**	**$48,119.00**	**$144,331.00**
May	Aspen	$8,791.00	$7,547.00	$10,616.00	$26,954.00
	Carlota	$8,407.00	$6,012.00	$2,639.00	$17,058.00
	FlatTop	$5,098.00	$5,942.00	$3,613.00	$14,653.00
	Quad	$8,475.00	$9,117.00	$10,724.00	$28,316.00
					7.00
					3.00
May Total					1.00
Jun					0.00
					4.00
	FlatTop	$817.00	$2,728.00	$1,646.00	$5,191.00
	Quad	$3,092.00	$4,872.00	$3,492.00	$11,456.00
	Sunset	$2,276.00	$2,611.00	$439.00	$5,326.00
	Yanaki	$3,893.00	$1,041.00	$2,103.00	$7,037.00
Jun Total		**$15,025.00**	**$15,275.00**	**$13,734.00**	**$44,034.00**
Jul	Aspen	$1,945.00	$2,472.00	$3,615.00	$8,032.00
	Carlota	$1,472.00	$2,902.00	$1,446.00	$5,820.00
	FlatTop	$1,756.00	$4,734.00	$1,063.00	$7,553.00
	Quad	$2,995.00	$2,016.00	$2,591.00	$7,602.00
	Sunset	$1,171.00	$827.00	$3,060.00	$5,058.00
	Yanaki	$2,247.00	$4,194.00	$3,517.00	$9,958.00
Jul Total		**$11,586.00**	**$17,145.00**	**$15,292.00**	**$44,023.00**
Grand Total		**$113,059.00**	**$124,999.00**	**$117,651.00**	**$355,709.00**

   2) In Office 365, Excel Power Query is everything that you see in the Get & Transform Data and Queries &Connections groups in the Data Ribbon Tab:



   3) **Convert the Proper Data Set to an Excel Table** : To bring data from an Excel Worksheet into the Power Query Editor, you must first Convert the Proper Data Set to an Excel Table and then name that table smartly:
      i. Use Ctrl + T to convert the Proper Data Set to an Excel Table, , as seen in this picture:



14	Description	Amount
15	Carlota / 05/05/2016 / West	485
16	Aspen / 04/02/2016 / South	804
17	Yanaki / 05/02/2016 / South	790
18	FlatTop / 04/11/2016 / South	965
19	Carlota / 05/21/2016 / South	620
20	Carlota / 05/15/2016 / East	305
21	Carlota / 05/19/2016 / West	426
22	Yanaki / 05/06/2016 / East	319

Create Table

Where is the data for your table?

=$A$14:$B$379

☑ My table has headers

ii. Use Alt, J, T, A to jump to the Table Name text box in the Properties group in the Table Tools Design Ribbon Tab, as seen in this picture:

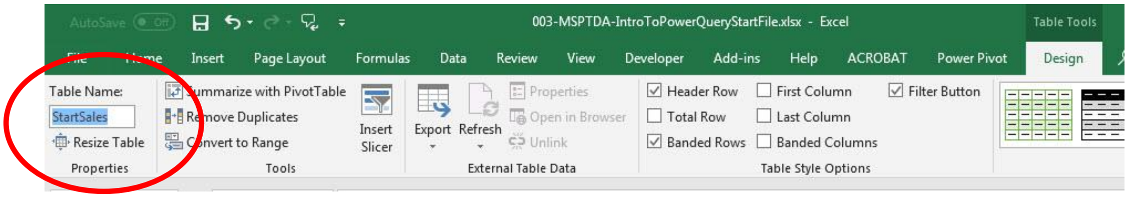
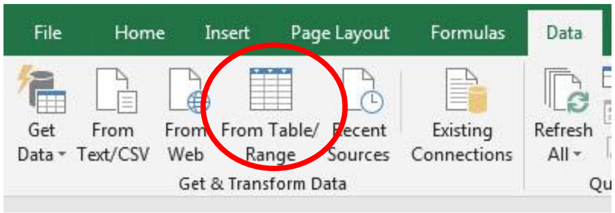


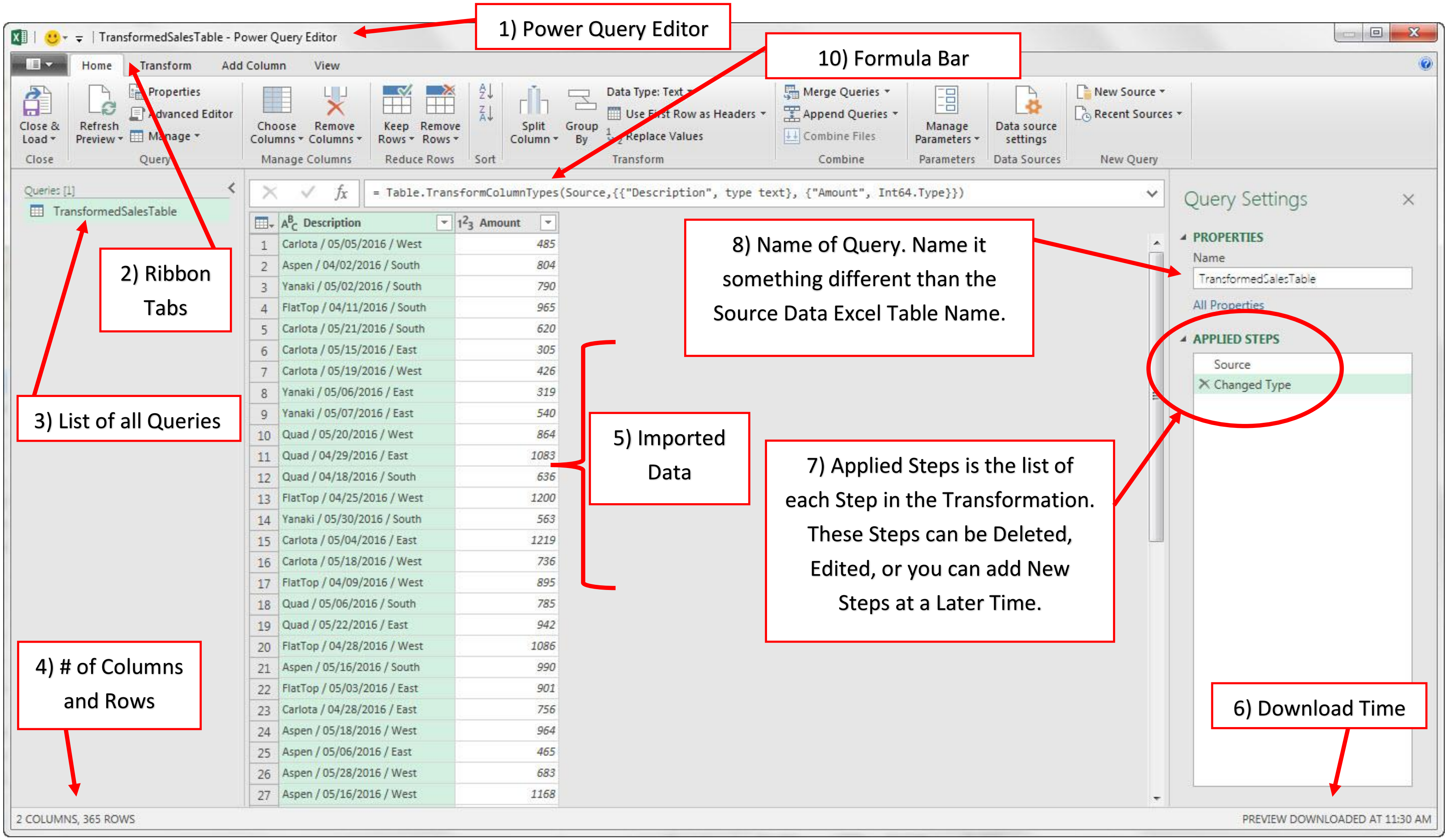
Table Name: StartSales

☑ Header Row ☐ First Column ☑ Filter Button
☐ Total Row ☐ Last Column
☑ Banded Rows ☐ Banded Columns

4) **From Table / Range button**: To bring an Excel Table into the Power Query Editor, use the From Table / Range button in the Get & Transform Data group in the Data Ribbon Tab, as seen in this picture:



5) **Power Query Editor Window**: Your data will open in a new window called the Power Query Editor Window. Some of the key features are listed below:



1) Power Query Editor

2) Ribbon Tabs

3) List of all Queries

4) # of Columns and Rows

5) Imported Data

6) Download Time

7) Applied Steps is the list of each Step in the Transformation. These Steps can be Deleted, Edited, or you can add New Steps at a Later Time.

8) Name of Query. Name it something different than the Source Data Excel Table Name.

10) Formula Bar

```
= Table.TransformColumnTypes(Source,{{"Description", type text}, {"Amount", Int64.Type}})
```

	Description	Amount
1	Carlota / 05/05/2016 / West	485
2	Aspen / 04/02/2016 / South	804
3	Yanaki / 05/02/2016 / South	790
4	FlatTop / 04/11/2016 / South	965
5	Carlota / 05/21/2016 / South	620
6	Carlota / 05/15/2016 / East	305
7	Carlota / 05/19/2016 / West	426
8	Yanaki / 05/06/2016 / East	319
9	Yanaki / 05/07/2016 / East	540
10	Quad / 05/20/2016 / West	864
11	Quad / 04/29/2016 / East	1083
12	Quad / 04/18/2016 / South	636
13	FlatTop / 04/25/2016 / West	1200
14	Yanaki / 05/30/2016 / South	563
15	Carlota / 05/04/2016 / East	1219
16	Carlota / 05/18/2016 / West	736
17	FlatTop / 04/09/2016 / West	895
18	Quad / 05/06/2016 / South	785
19	Quad / 05/22/2016 / East	942
20	FlatTop / 04/28/2016 / West	1086
21	Aspen / 05/16/2016 / South	990
22	FlatTop / 05/03/2016 / East	901
23	Carlota / 04/28/2016 / East	756
24	Aspen / 05/18/2016 / West	964
25	Aspen / 05/06/2016 / East	465
26	Aspen / 05/28/2016 / West	683
27	Aspen / 05/16/2016 / West	1168

Name: TransformedSalesTable

APPLIED STEPS:
- Source
- Changed Type

2 COLUMNS, 365 ROWS

PREVIEW DOWNLOADED AT 11:30 AM

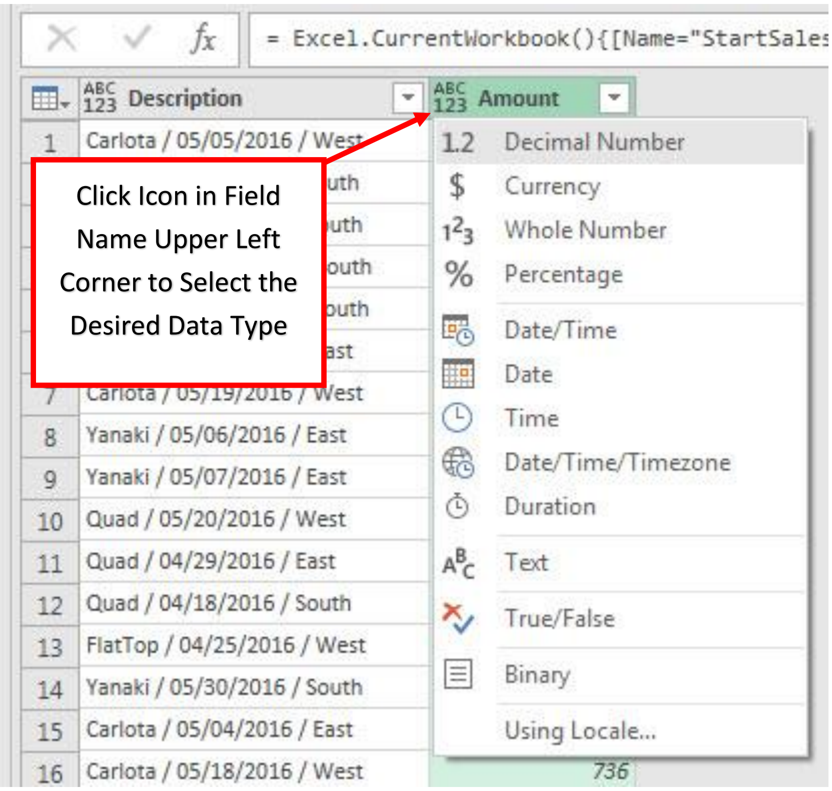
6) **Data Types in Power Query**. Unlike Excel, we must properly Defined each Field with a Data Type. If we do not define the correct Data Type, for example a dollar amount as Currency, then some of the calculations in Power Query, Excel, Power Pivot and Power BI Desktop will not work correctly.

i. Here is a list of the Data Types in Power Query:

Data Types in Power Query	Short Definition
Decimal Number	Max 15 digits
Fixed Decimal Number	Max 4 decimals to right of decimal
Whole Number	No digits to right of decimal
Date/Time	Date and Time together
Date	Just Date
Time	Just Time
Date/Time/Timezone	Same as Date and Time
Duration	Length of Time
Text	Text - max length 268,435,456 Unicode characters
Ture/False	Ture/False

Data Types in Power Query - with Long Definition
Decimal Number– Represents a 64 bit (eight-byte) floating point number. It's the most common number type and corresponds to numbers as you usually think of them. Although designed to handle numbers with fractional values, it also handles whole numbers. The Decimal Number type can handle negative values from -1.79E +308 through -2.23E -308, 0, and positive values from 2.23E -308 through 1.79E + 308. For example, numbers like 34, 34.01, and 34.000367063 are valid decimal numbers. The largest value that can be represented in a Decimal Number type is 15 digits long. The decimal separator can occur anywhere in the number. The Decimal Number type corresponds to how Excel stores its numbers.
Fixed Decimal Number– Has a fixed location for the decimal separator. The decimal separator always has four digits to its right and allows for 19 digits of significance. The largest value it can represent is 922,337,203,685,477.5807 (positive or negative). The Fixed Decimal Number type is useful in cases where rounding might introduce errors. When you work with many numbers that have small fractional values, they can sometimes accumulate and force a number to be slightly off. Since the values past the four digits to the right of decimal separator are truncated, the Fixed Decimal type can help you avoid these kinds of errors. If you're familiar with SQL Server, this data type corresponds to SQL Server's Decimal (19,4), or the Currency Data type in Power Pivot.
Whole Number– Represents a 64 bit (eight-byte) integer value. Because it's an integer, it has no digits to the right of the decimal place. It allows for 19 digits; positive or negative whole numbers between -9,223,372,036,854,775,808 (-2^63) and 9,223,372,036,854,775,807 (2^63-1). It can represent the largest possible number of the various numeric data types. As with the Fixed Decimal type, the Whole Number type can be useful in cases where you need to control rounding.
Date/Time – Represents both a date and time value. Underneath the covers, the Date/Time value is stored as a Decimal Number Type. So you can actually convert between the two. The time portion of a date is stored as a fraction to whole multiples of 1/300 seconds (3.33 ms). Dates between years 1900 and 9999 are supported.
Date – Represents just a Date (no time portion). When converted into the model, a Date is the same as a Date/Time value with zero for the fractional value.
Time – Represents just Time (no Date portion). When converted into the model, a Time value is the same as a Date/Time value with no digits to the left of the decimal place.
Date/Time/Timezone– Represents a UTC Date/Time. Currently, it's converted into Date/Time when loaded into the model.
Duration – Represents a length of time. It's converted into a Decimal Number Type when loaded into the model. As a Decimal Number type it can be added or subtracted from a Date/Time field with correct results. As a Decimal Number type, you can easily use it in visualizations that show magnitude.
Text - A Unicode character data string. Can be strings, numbers, or dates represented in a text format. Maximum string length is 268,435,456 Unicode characters (256 mega characters) or 536,870,912 bytes.
True/False– A Boolean value of either a True or False.

ii. In Power Query, each Field has an Icon in the upper left corner that we can click and then select the correct Definition of Data Type for the Field, as seen in this picture:

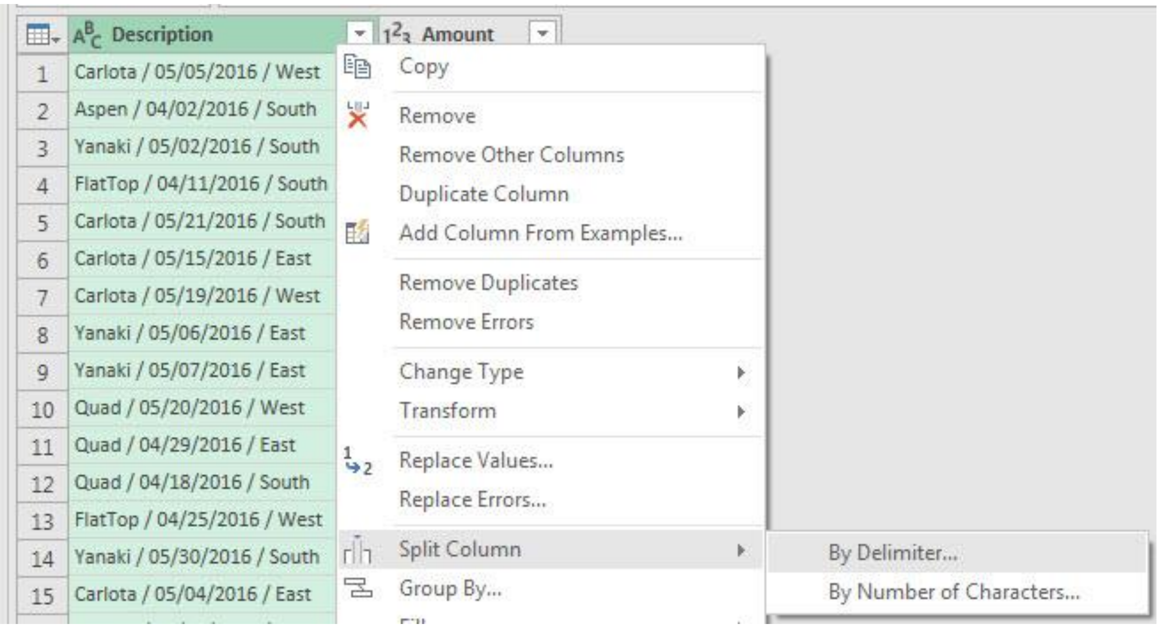


= Excel.CurrentWorkbook(){[Name="StartSales

Click Icon in Field Name Upper Left Corner to Select the Desired Data Type

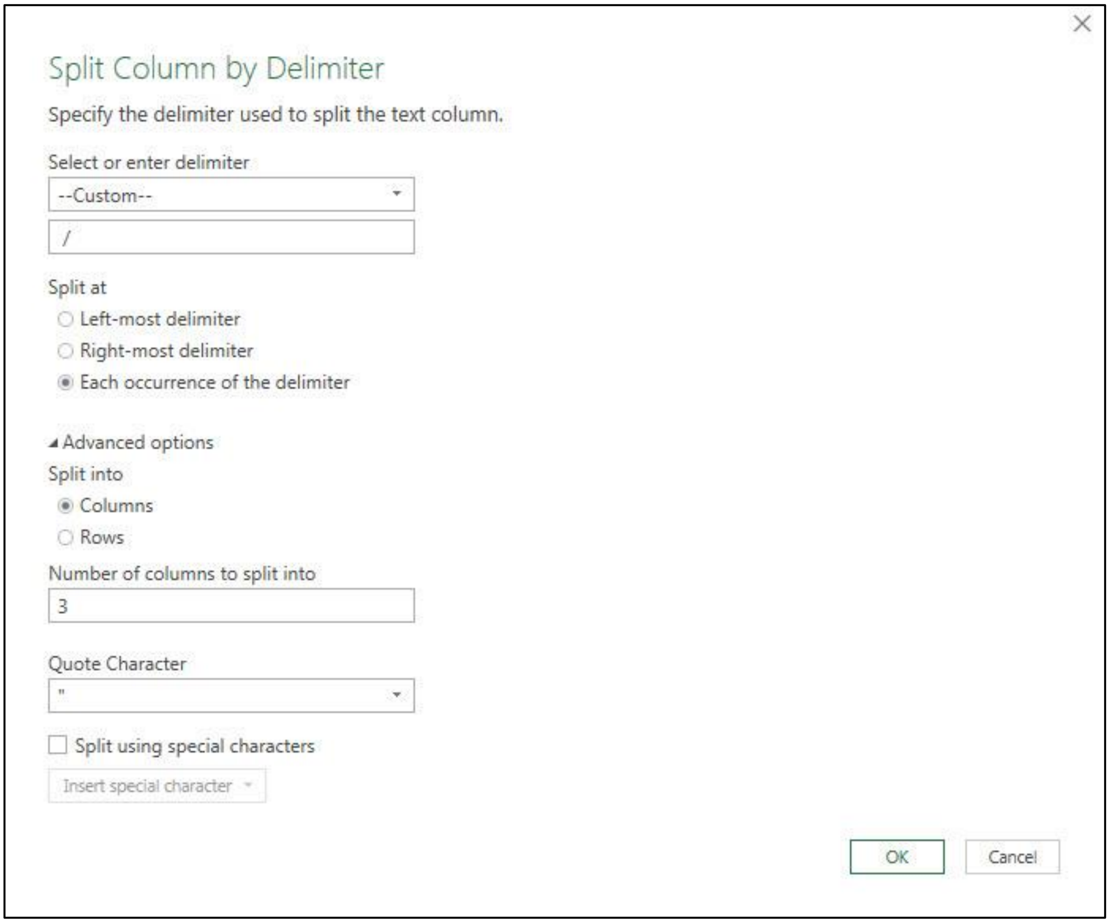
	Description	Amount
1	Carlota / 05/05/2016 / West	
	uth	
	uth	
	outh	
	outh	
	ast	
7	Carlota / 05/19/2016 / West	
8	Yanaki / 05/06/2016 / East	
9	Yanaki / 05/07/2016 / East	
10	Quad / 05/20/2016 / West	
11	Quad / 04/29/2016 / East	
12	Quad / 04/18/2016 / South	
13	FlatTop / 04/25/2016 / West	
14	Yanaki / 05/30/2016 / South	
15	Carlota / 05/04/2016 / East	
16	Carlota / 05/18/2016 / West	736

7) **To Split a Column by a Delimiter**, we can right-click the Field Name, point to Split Column, then click on By Delimiter, as seen in this picture:



	Description	Amount
1	Carlota / 05/05/2016 / West	
2	Aspen / 04/02/2016 / South	
3	Yanaki / 05/02/2016 / South	
4	FlatTop / 04/11/2016 / South	
5	Carlota / 05/21/2016 / South	
6	Carlota / 05/15/2016 / East	
7	Carlota / 05/19/2016 / West	
8	Yanaki / 05/06/2016 / East	
9	Yanaki / 05/07/2016 / East	
10	Quad / 05/20/2016 / West	
11	Quad / 04/29/2016 / East	
12	Quad / 04/18/2016 / South	
13	FlatTop / 04/25/2016 / West	
14	Yanaki / 05/30/2016 / South	
15	Carlota / 05/04/2016 / East	

8) **Here is the Split Column By Delimiter dialog Box**:



Split Column by Delimiter

Specify the delimiter used to split the text column.

Select or enter delimiter

--Custom--

/

Split at

○ Left-most delimiter
○ Right-most delimiter
● Each occurrence of the delimiter

Advanced options

Split into

● Columns
○ Rows

Number of columns to split into

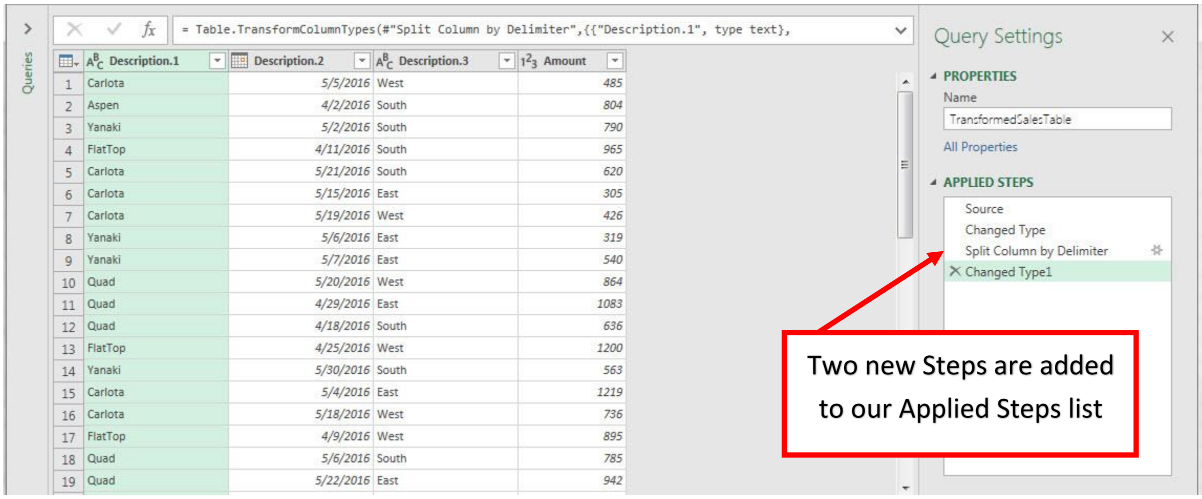
3

Quote Character

"

□ Split using special characters

9) After we split, the Data Set looks like this and two steps have been added:



```
= Table.TransformColumnTypes(#"Split Column by Delimiter",{{"Description.1", type text},
```

	Description.1	Description.2	Description.3	Amount
1	Carlota	5/5/2016	West	485
2	Aspen	4/2/2016	South	804
3	Yanaki	5/2/2016	South	790
4	FlatTop	4/11/2016	South	965
5	Carlota	5/21/2016	South	620
6	Carlota	5/15/2016	East	305
7	Carlota	5/19/2016	West	426
8	Yanaki	5/6/2016	East	319
9	Yanaki	5/7/2016	East	540
10	Quad	5/20/2016	West	864
11	Quad	4/29/2016	East	1083
12	Quad	4/18/2016	South	636
13	FlatTop	4/25/2016	West	1200
14	Yanaki	5/30/2016	South	563
15	Carlota	5/4/2016	East	1219
16	Carlota	5/18/2016	West	736
17	FlatTop	4/9/2016	West	895
18	Quad	5/6/2016	South	785
19	Quad	5/22/2016	East	942

Query Settings

PROPERTIES

Name

TransformedSalesTable

All Properties

APPLIED STEPS

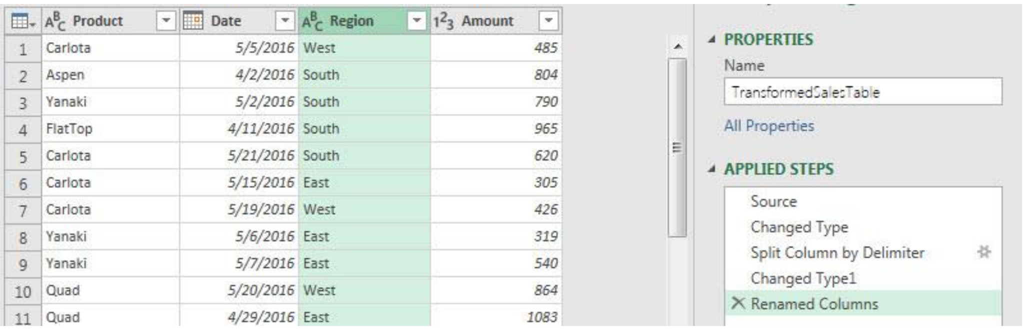
- Source
- Changed Type
- Split Column by Delimiter
- Changed Type1

Two new Steps are added to our Applied Steps list

10) To rename Fields, Select Field Name and then hit the F2 Key, type the new name and hit enter.

11) You can use the Arrow Keys to move to the next Field and rename the next Field.

12) After we renamed the first three Fields, the Transformed Data Set looks like this, and there are five steps in the Applied Steps list, as seen in this picture:



	Product	Date	Region	Amount
1	Carlota	5/5/2016	West	485
2	Aspen	4/2/2016	South	804
3	Yanaki	5/2/2016	South	790
4	FlatTop	4/11/2016	South	965
5	Carlota	5/21/2016	South	620
6	Carlota	5/15/2016	East	305
7	Carlota	5/19/2016	West	426
8	Yanaki	5/6/2016	East	319
9	Yanaki	5/7/2016	East	540
10	Quad	5/20/2016	West	864
11	Quad	4/29/2016	East	1083

PROPERTIES

Name

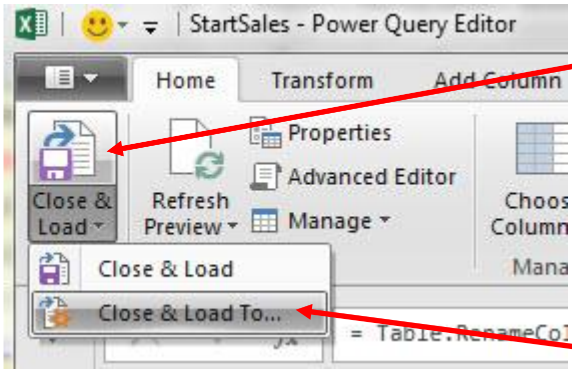
TransformedSalesTable

All Properties

APPLIED STEPS

- Source
- Changed Type
- Split Column by Delimiter
- Changed Type1
- Renamed Columns

13) **"Close & Load To…"**. Now that we have our Transformed Data Set, we can Close the Power Query Editor and Load it to one of five locations using the "Close & Load To…" option in the Close & Load drop-down arrow in the Close group in the Home Ribbon Tab, as seen in this picture:

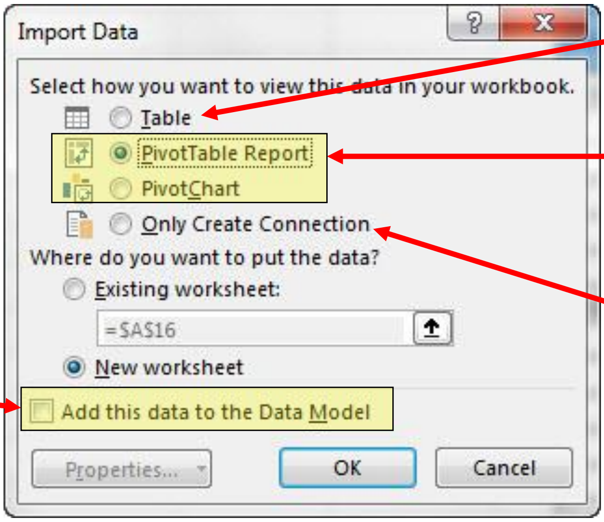


Clicking just Close & Load will send it to an Excel Sheet.

"Close & Load To…" Allows us the Options of Different Load

14) In Excel, we can load our Transformed Data Set to five locations:
    i. An Excel Table on an Excel Worksheet (example shown in video).
    ii. A PivotTable Cache so that we can make a PivotTable (example shown in video)
    iii. A PivotTable Cache so that we can make a PivotTable (not shown in video)
    iv. Only Create Connection (example shown in video)
    v. Excel Power Pivot Data Model (example shown in video)

15) After we click the "Close & Load To…" option, we get the Import Data dialog box. Although in the video all the options were demonstrated, the final example is what we wanted: Load as a PivotTable Report to a New Worksheet:



Import Data

Select how you want to view this data in your workbook.

○ Table

● PivotTable Report

○ PivotChart

○ Only Create Connection

Where do you want to put the data?

○ Existing worksheet:

=$A$16

● New worksheet

☐ Add this data to the Data Model

Properties... OK Cancel

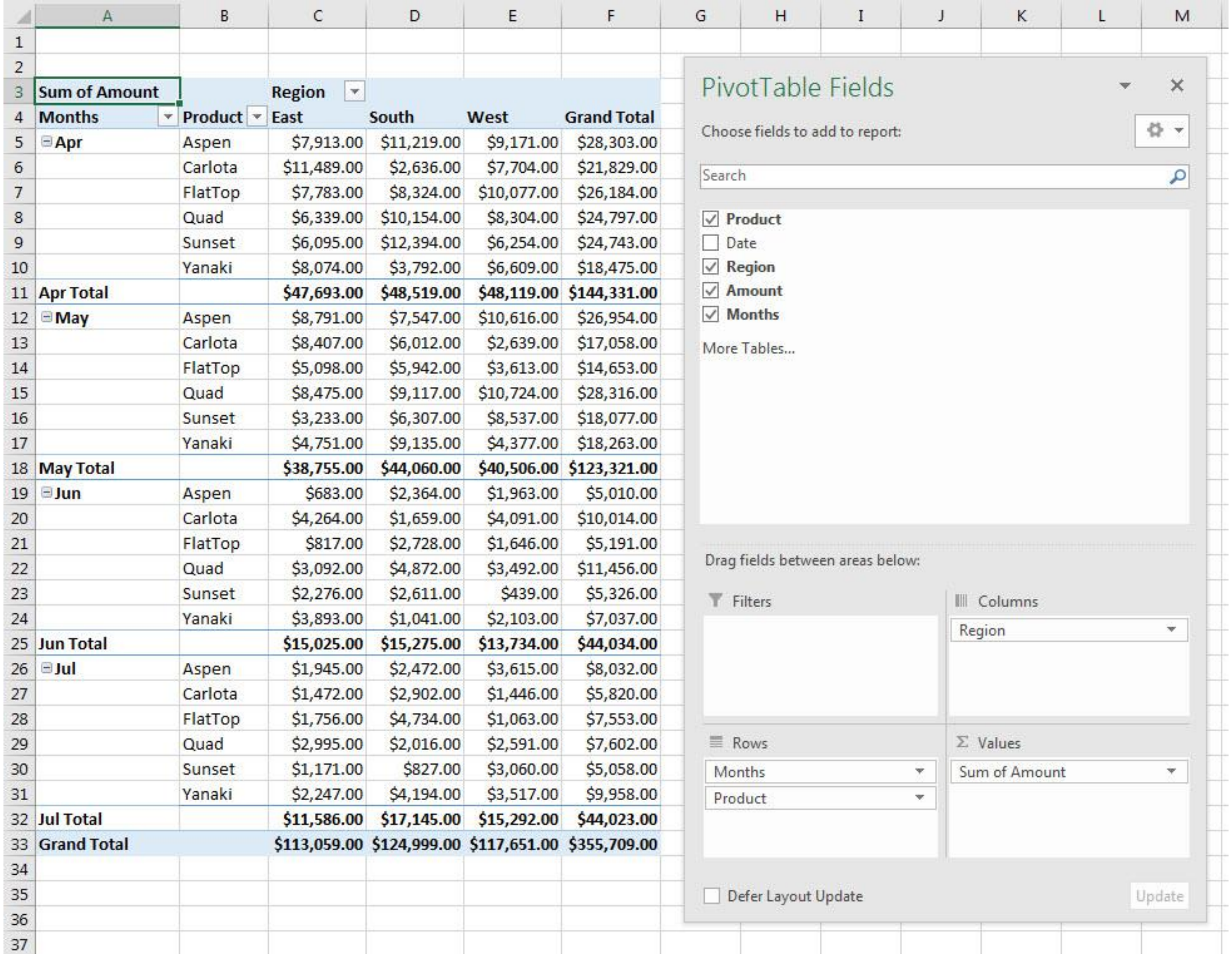
Loads Data to an Excel Sheet.

Loads Data to a PivotTable Cache.

Does not Load Data Anywhere.

Loads Data to the Excel Power Pivot Data Model.

16) Once you Load to PivotTable Report, you can build your PivotTable by dragging Fields from the Field List to the Row, Column, Filter and Values Area of the PivotTable Field List, as seen in this picture:



Sum of Amount		Region			
Months	**Product**	**East**	**South**	**West**	**Grand Total**
⊟**Apr**	Aspen	$7,913.00	$11,219.00	$9,171.00	$28,303.00
	Carlota	$11,489.00	$2,636.00	$7,704.00	$21,829.00
	FlatTop	$7,783.00	$8,324.00	$10,077.00	$26,184.00
	Quad	$6,339.00	$10,154.00	$8,304.00	$24,797.00
	Sunset	$6,095.00	$12,394.00	$6,254.00	$24,743.00
	Yanaki	$8,074.00	$3,792.00	$6,609.00	$18,475.00
Apr Total		**$47,693.00**	**$48,519.00**	**$48,119.00**	**$144,331.00**
⊟**May**	Aspen	$8,791.00	$7,547.00	$10,616.00	$26,954.00
	Carlota	$8,407.00	$6,012.00	$2,639.00	$17,058.00
	FlatTop	$5,098.00	$5,942.00	$3,613.00	$14,653.00
	Quad	$8,475.00	$9,117.00	$10,724.00	$28,316.00
	Sunset	$3,233.00	$6,307.00	$8,537.00	$18,077.00
	Yanaki	$4,751.00	$9,135.00	$4,377.00	$18,263.00
May Total		**$38,755.00**	**$44,060.00**	**$40,506.00**	**$123,321.00**
⊟**Jun**	Aspen	$683.00	$2,364.00	$1,963.00	$5,010.00
	Carlota	$4,264.00	$1,659.00	$4,091.00	$10,014.00
	FlatTop	$817.00	$2,728.00	$1,646.00	$5,191.00
	Quad	$3,092.00	$4,872.00	$3,492.00	$11,456.00
	Sunset	$2,276.00	$2,611.00	$439.00	$5,326.00
	Yanaki	$3,893.00	$1,041.00	$2,103.00	$7,037.00
Jun Total		**$15,025.00**	**$15,275.00**	**$13,734.00**	**$44,034.00**
⊟**Jul**	Aspen	$1,945.00	$2,472.00	$3,615.00	$8,032.00
	Carlota	$1,472.00	$2,902.00	$1,446.00	$5,820.00
	FlatTop	$1,756.00	$4,734.00	$1,063.00	$7,553.00
	Quad	$2,995.00	$2,016.00	$2,591.00	$7,602.00
	Sunset	$1,171.00	$827.00	$3,060.00	$5,058.00
	Yanaki	$2,247.00	$4,194.00	$3,517.00	$9,958.00
Jul Total		**$11,586.00**	**$17,145.00**	**$15,292.00**	**$44,023.00**
Grand Total		**$113,059.00**	**$124,999.00**	**$117,651.00**	**$355,709.00**

PivotTable Fields

Choose fields to add to report:

☑ Product
□ Date
☑ Region
☑ Amount
☑ Months

Drag fields between areas below:

Filters:

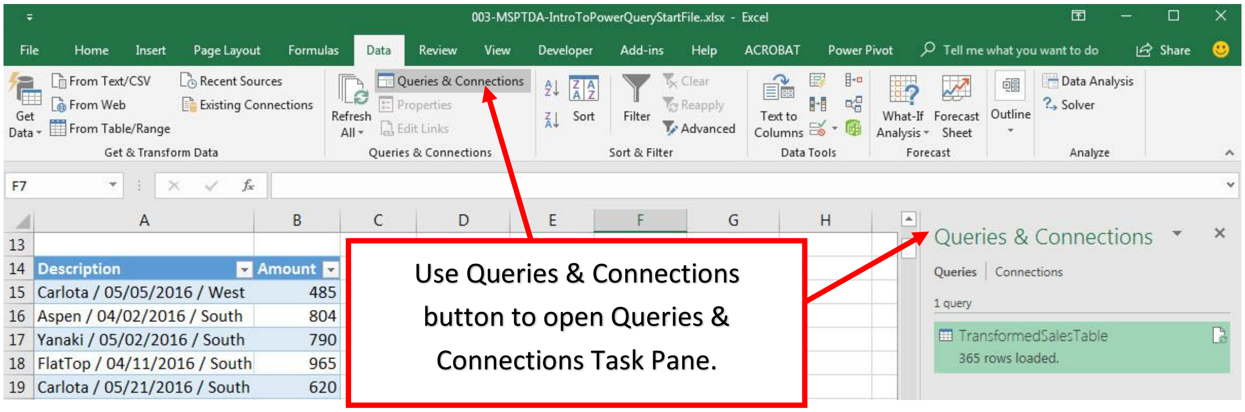
Columns: Region

Rows: Months, Product

Values: Sum of Amount

□ Defer Layout Update

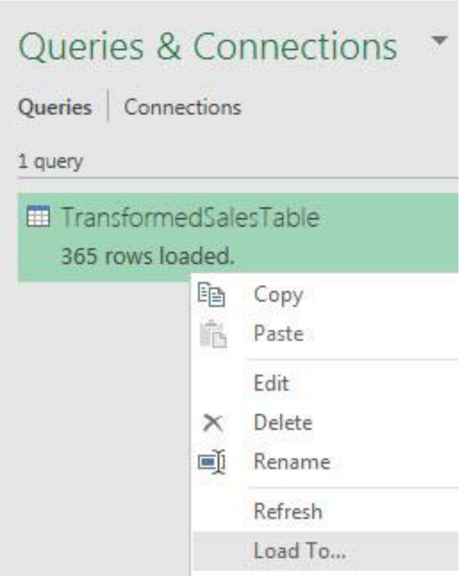
17) **When you want to Edit your Queries in Excel or change the "Load To" Location**, you use the Queries & Connections button in the Queries & Connections group in the Data Ribbon Tab to open the Queries & Connections Task Page, as seen in this picture:



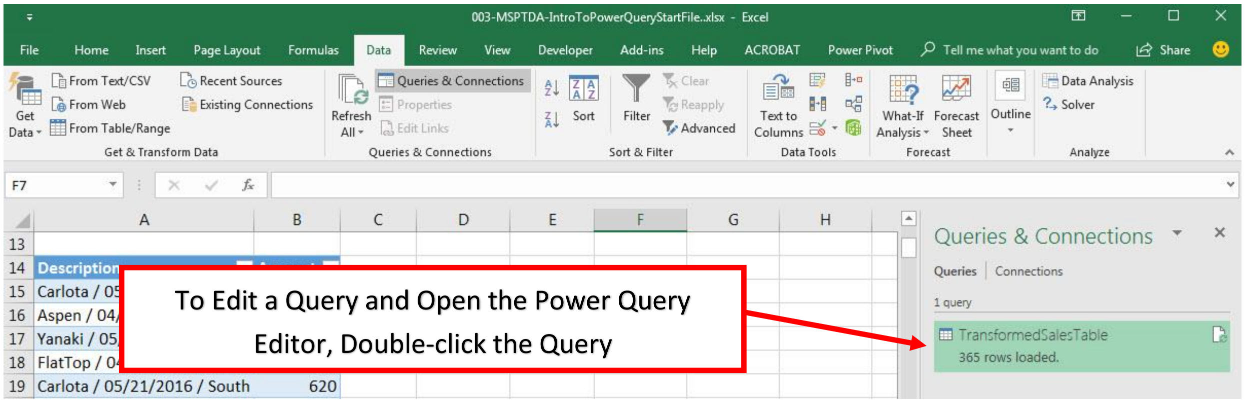
	A	B
13		
14	Description	Amount
15	Carlota / 05/05/2016 / West	485
16	Aspen / 04/02/2016 / South	804
17	Yanaki / 05/02/2016 / South	790
18	FlatTop / 04/11/2016 / South	965
19	Carlota / 05/21/2016 / South	620

Use Queries & Connections button to open Queries & Connections Task Pane.

18) **To Change the Location of where you Load the Data** (like we saw demonstrated many times in the video), Right-click the Query Listed in the Queries & Connections Task Pane and click on "Load To…".



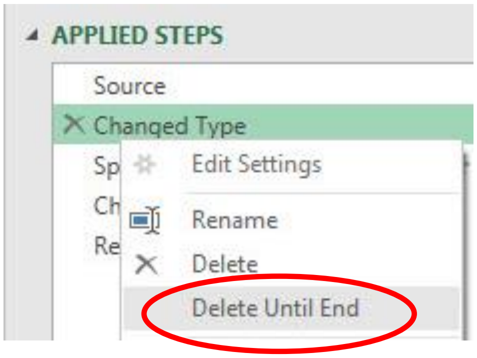
19) **To Edit a Query**, Right-click the Query Listed in the Queries & Connections Task Pane and then click on Edit, or simply Double-click the Query.



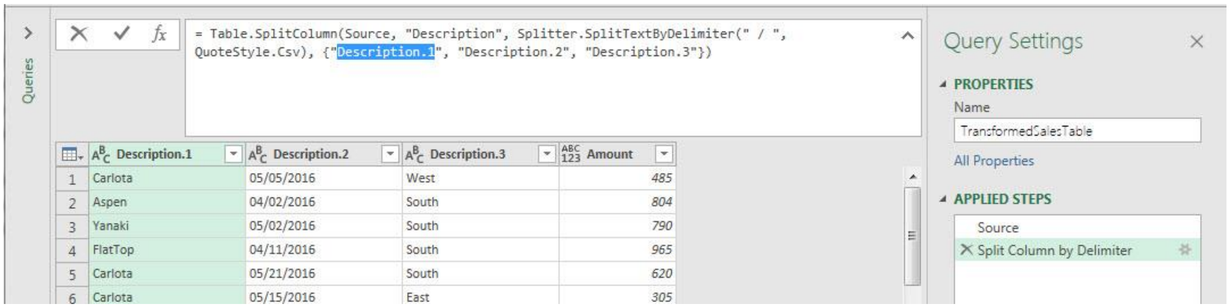
	A	B
13		
14	Description	
15	Carlota / 05	
16	Aspen / 04/	
17	Yanaki / 05/	
18	FlatTop / 04	
19	Carlota / 05/21/2016 / South	620

To Edit a Query and Open the Power Query Editor, Double-click the Query

20) To Delete the Transformation Steps #2 to #5 that we created earlier, right-click Step #2 and click on Delete Until End, as seen in this picture:



21) **To Edit a step in the Formula Bar**, select the step in the Applied Steps list, then edit the Power Query Function (M Code) in the formula bar as seen in this picture where we changed the default Field Names in the Table.SplitColumn Function:



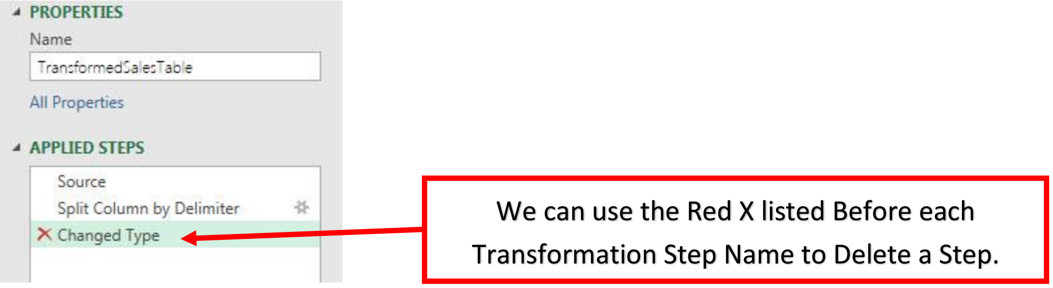
```
= Table.SplitColumn(Source, "Description", Splitter.SplitTextByDelimiter(" / ",
QuoteStyle.Csv), {"Description.1", "Description.2", "Description.3"})
```

	Description.1	Description.2	Description.3	Amount
1	Carlota	05/05/2016	West	485
2	Aspen	04/02/2016	South	804
3	Yanaki	05/02/2016	South	790
4	FlatTop	04/11/2016	South	965
5	Carlota	05/21/2016	South	620
6	Carlota	05/15/2016	East	305

22) To learn about particular Power Query Functions, we can search Google and go to the Microsoft Web Site. In the video we searched for information about the Table.SplitColumn Function.

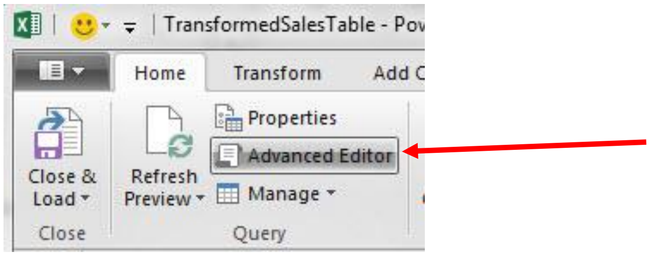
23) If you encounter an Error after editing steps, it is often caused because the edit you made impacted a later step. In the video, when we edited the Field Names this cause later steps to not recognize the newly named Fields. This means that when we edit steps we have to be aware of how the edits may effect later steps.

24) If we need to delete a single step, we can use the Red X listed before each Transformation Step Name.

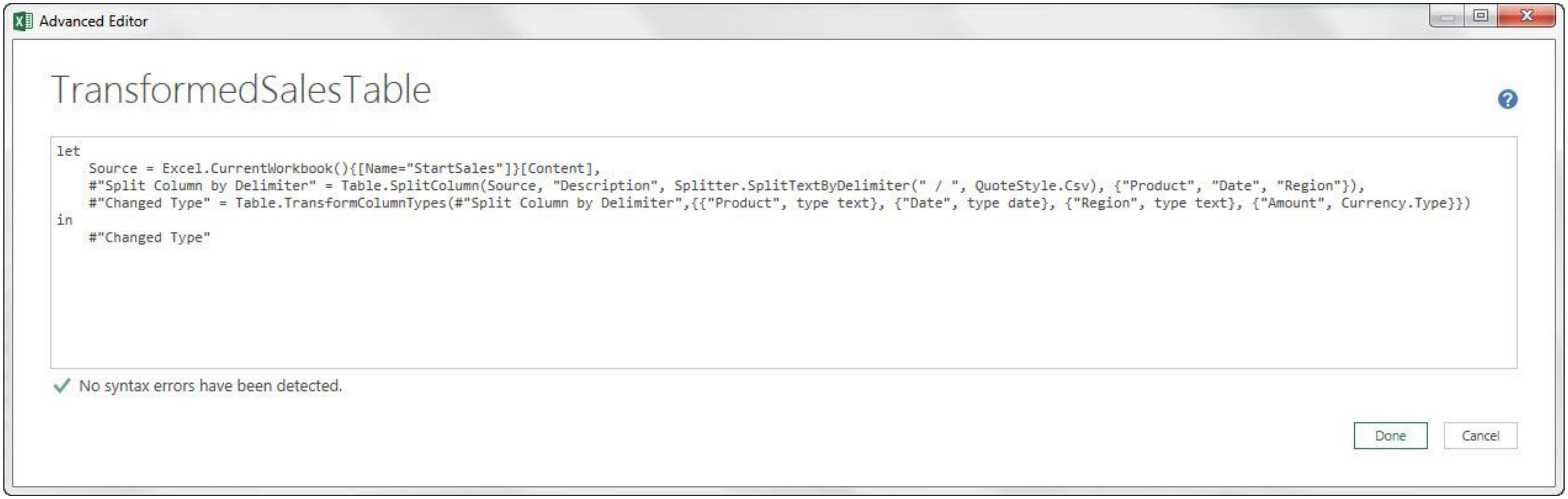


We can use the Red X listed Before each Transformation Step Name to Delete a Step.

25) **Advanced Editor for M Code**: If you want to view and or edit the full M Code, you can use the Advanced Editor button in the Query group in the Home Ribbon Tab, as seen in this picture:



26) Below are a few notes about the syntax of M Code as seen in the Advanced Editor:
    i. M Code is a Case Sensitive, Function Based Language.
    ii. The code starts with the text (in all lower case) “let”.
    iii. In the blow picture, we can see that there are three Transformation Steps listed.
    iv. Each Transformation Step starts with the name of the step.
        1. If the name has no spaces, then you simply type the name, like Source.
        2. If the name has one or more spaces, you must type a pound sign, open double quotes, the name you want and then close double quotes, like: #”Split Column by Delimiter”.
    v. After the name of each Transformation Step, there is an equal sign.
    vi. After the equal sign for each step, you have your Power Query Function / Functions, like Excel.CurrentWorkbook.
    vii. At the end of each Transformation Step, you type a comma, except for the last step. The last step does not require a comma.
    viii. After the last step, the word “in’ is listed.
    ix. What follows the “in” is the End Result or Output or Final Transformation. The Final Transformation is always the name of the last Transformation Step. This Final Transformation is what you Load to your desired location.



```
let
    Source = Excel.CurrentWorkbook(){[Name="StartSales"]}[Content],
    #"Split Column by Delimiter" = Table.SplitColumn(Source, "Description", Splitter.SplitTextByDelimiter(" / ", QuoteStyle.Csv), {"Product", "Date", "Region"}),
    #"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"Product", type text}, {"Date", type date}, {"Region", type text}, {"Amount", Currency.Type}})
in
    #"Changed Type"
```

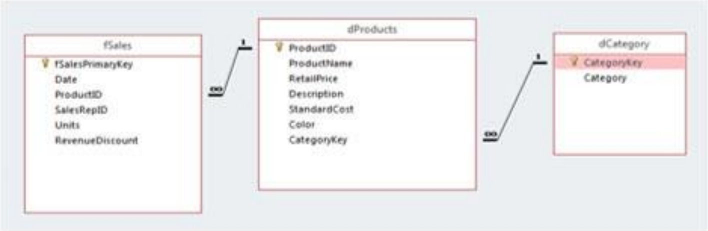
✓ No syntax errors have been detected.

27) After all your transformations in Power Query are done and you load the data and build a report, you can refresh the Power Query Transformation and the Final Report by using the Refresh All button in the Queries & Connections group in the Data Ribbon Tab (we did not see an example of using this button in the video), but is the easiest way if you want to refresh all the queries and reports at the same time. Alternatively, you can right-click any one query or PivotTable report and click on Refresh.

2. **Example 2: ETL Data from Access into Excel Power Pivot Data Model**.

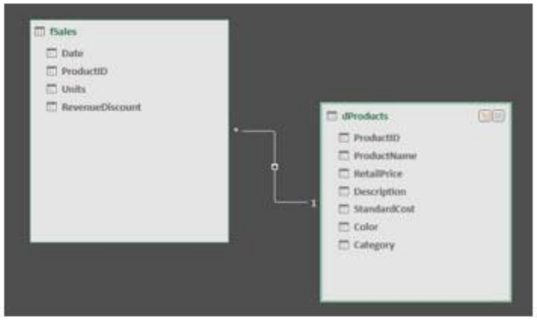
   1) **Our Goal** is to Extract Data from an Access Relational Database and transform it into the Start of a Start Schema Data Model in the Excel Power Pivot Data Model.

Access Database

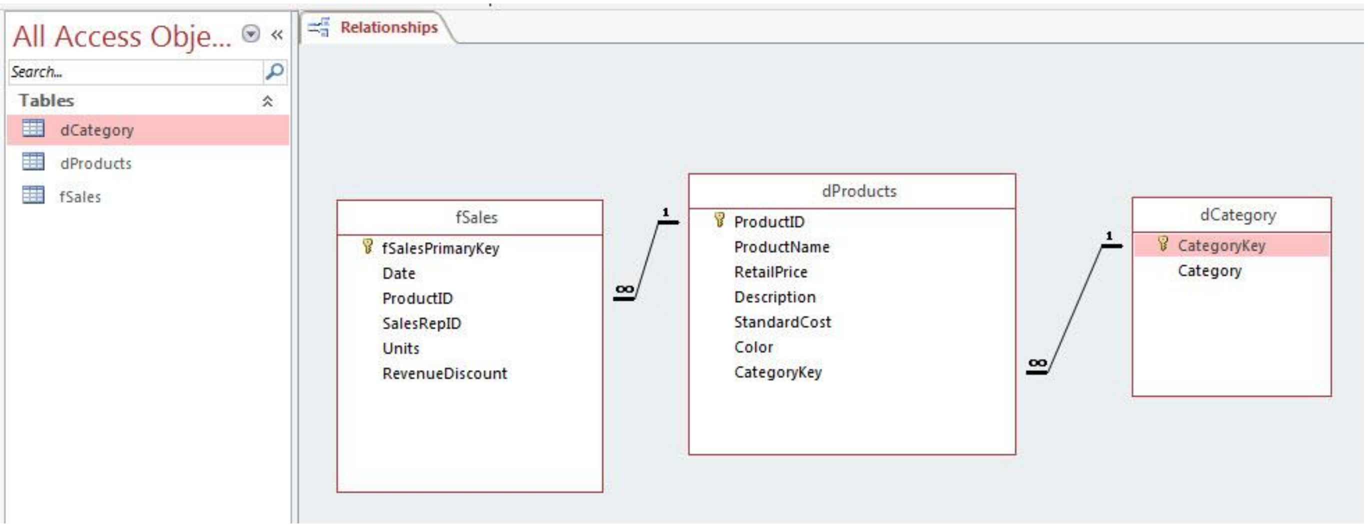


Into ==>>

Power Pivot Data Model in Star Schema

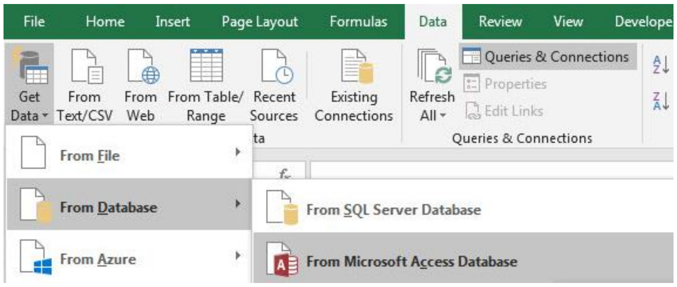


   2) **Here is a picture of the data that is in an Access database**, and we need to Extract the data from the Access Database and import, transform and load the data into the Excel Power Pivot Data Model:



   3) What is great about using Power Query to important data from a Relational Database, like the Access Relational Database like in the above picture, is that when you import the tables, Power Query will import that tables and the Relationships. The fact that the Relationships are also imported is that we can then use the Relationships to help make our Transformations in Power Query easier to complete.

4) **Import From Access**. In Excel Power Query we can import parts of an Access Database by going to the Data Ribbon Tab, then in the Get & Transform group, click the Get Data Dropdown arrow, then point to From Database, then click on From Microsoft Access Database.

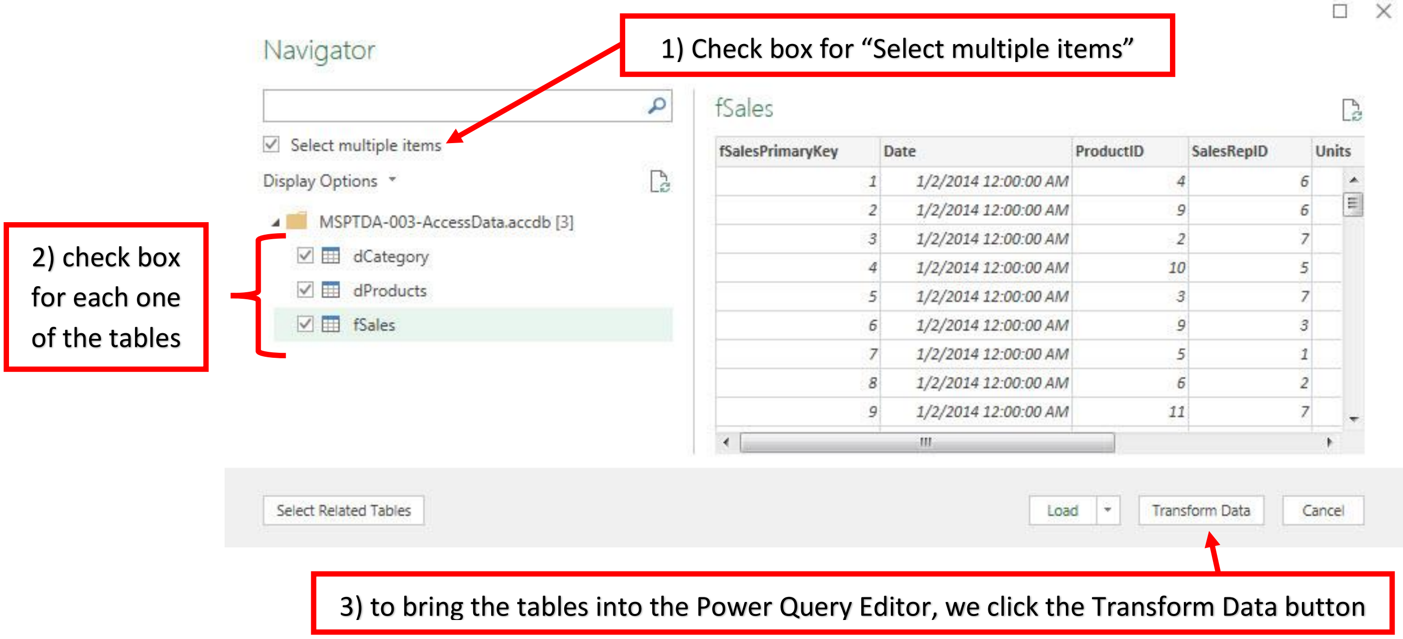


5) **Navigation Window**. After we click on From Microsoft Access Database, we will see the Navigation Window. To import the three tables, we:

i. Check the check box for "Select multiple items".

ii. Then we check the check box for each one of the tables.

iii. Then to bring the tables into the Power Query Editor, we click the Transform Data button



1) Check box for "Select multiple items"

☑ Select multiple items

2) check box for each one of the tables

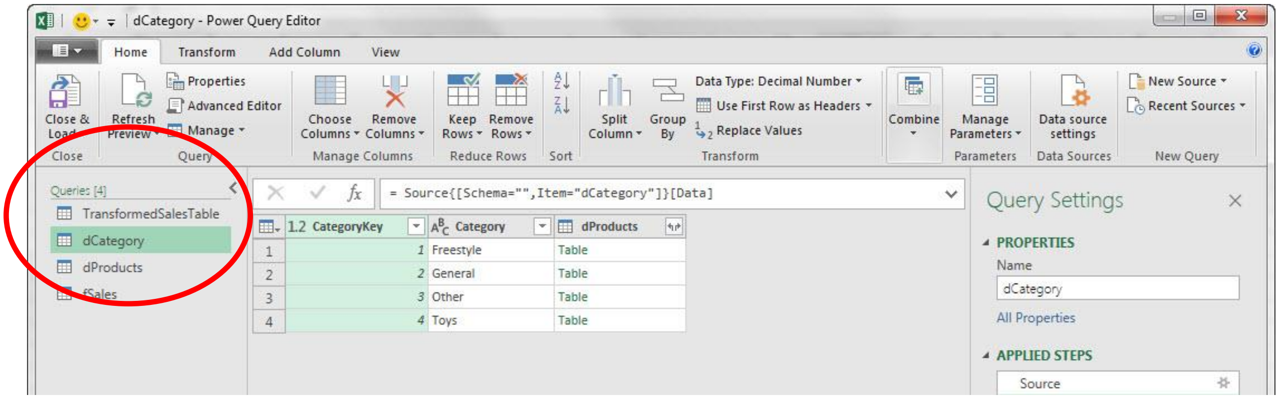
MSPTDA-003-AccessData.accdb [3]
☑ dCategory
☑ dProducts
☑ fSales

fSales

fSalesPrimaryKey	Date	ProductID	SalesRepID	Units
1	1/2/2014 12:00:00 AM	4	6	
2	1/2/2014 12:00:00 AM	9	6	
3	1/2/2014 12:00:00 AM	2	7	
4	1/2/2014 12:00:00 AM	10	5	
5	1/2/2014 12:00:00 AM	3	7	
6	1/2/2014 12:00:00 AM	9	3	
7	1/2/2014 12:00:00 AM	5	1	
8	1/2/2014 12:00:00 AM	6	2	
9	1/2/2014 12:00:00 AM	11	7	

3) to bring the tables into the Power Query Editor, we click the Transform Data button

6) **List of All Queries**. When the Power Query Editor opens we can see all of our Queries listed on the Left: the previous Query we created named "TransformedSalesTable", as well as the three new tables from Access.



Queries [4]
- TransformedSalesTable
- dCategory
- dProducts
- fSales

```
= Source{[Schema="",Item="dCategory"]}[Data]
```

	1.2 CategoryKey	A^B_C Category	dProducts
1	1	Freestyle	Table
2	2	General	Table
3	3	Other	Table
4	4	Toys	Table

Name: dCategory

Applied Steps: Source

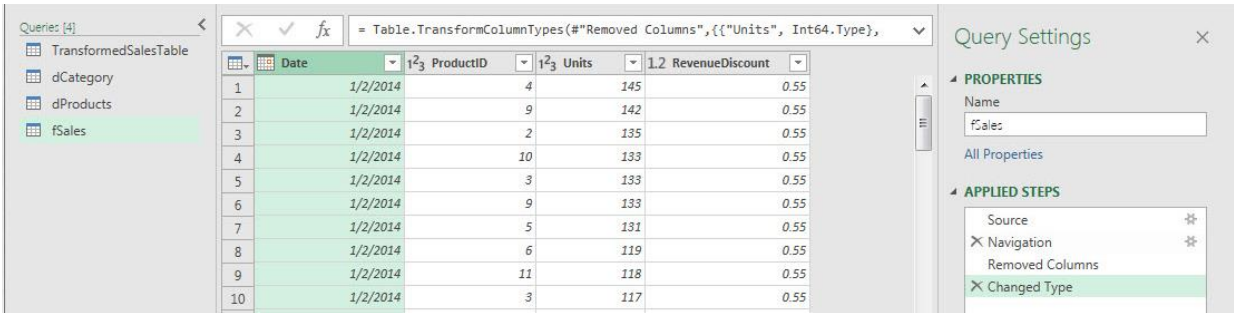
7) **For the fSales Table transformation**, we need to:

   i. Selected the fSales Table in the list of Queries on the left.

   ii. Removed the three unnecessary fields (fSalesPrimaryKey, SalesRepID, dProducts).

   iii. Changed the Data Types on the Units and ProductID fields to Whole Number and the Data Type on the Date field to Date.

   iv. This picture shows the transformation for the fSales Table:



Queries [4]
- TransformedSalesTable
- dCategory
- dProducts
- fSales

```
= Table.TransformColumnTypes(#"Removed Columns",{{"Units", Int64.Type},
```

	Date	ProductID	Units	RevenueDiscount
1	1/2/2014	4	145	0.55
2	1/2/2014	9	142	0.55
3	1/2/2014	2	135	0.55
4	1/2/2014	10	133	0.55
5	1/2/2014	3	133	0.55
6	1/2/2014	9	133	0.55
7	1/2/2014	5	131	0.55
8	1/2/2014	6	119	0.55
9	1/2/2014	11	118	0.55
10	1/2/2014	3	117	0.55

Query Settings

PROPERTIES

Name: fSales

All Properties

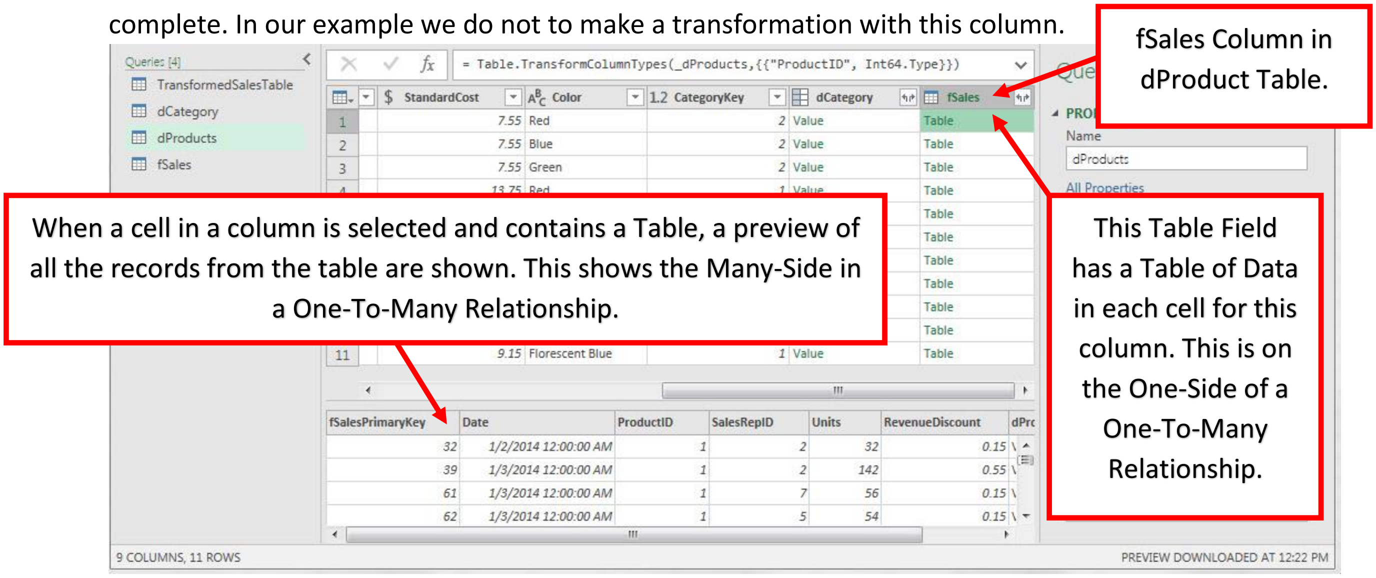
APPLIED STEPS
- Source
- Navigation
- Removed Columns
- Changed Type

8) **For the dProduct Table transformation**, we started by:

   i. Selected the dProduct Table in the list of Queries on the left.

   ii. Changed the Data Type on the ProductID field to Whole Number.

9) **Example of a Table Column**. If we look at the fSales Column in the dProduct Table, we can see an example of a Field that contains a Table in each row. This is because there is a One-To-Many Relationship between the dProduct Table (One-Side) and the fSales Table (Many-Side). In the picture below, you can see the first cell in the fSales Field is selected, and below there is a preview of all the records from the fSales Table for the product in that row. When we import from a Relational Database, a Table Field is always created like this and can make transformations easier to complete because all the data from the other side of the relationship is available for any transformation we may need to complete. In our example we do not to make a transformation with this column.



fSales Column in dProduct Table.

When a cell in a column is selected and contains a Table, a preview of all the records from the table are shown. This shows the Many-Side in a One-To-Many Relationship.

This Table Field has a Table of Data in each cell for this column. This is on the One-Side of a One-To-Many Relationship.

Queries [4]
- TransformedSalesTable
- dCategory
- dProducts
- fSales

```
= Table.TransformColumnTypes(_dProducts,{{"ProductID", Int64.Type}})
```

	StandardCost	Color	CategoryKey	dCategory	fSales
1	7.55	Red	2	Value	Table
2	7.55	Blue	2	Value	Table
3	7.55	Green	2	Value	Table
4	13.75	Red	1	Value	Table
					Table
					Table
					Table
					Table
					Table
					Table
11	9.15	Florescent Blue	1	Value	Table

fSalesPrimaryKey	Date	ProductID	SalesRepID	Units	RevenueDiscount	dPrc
32	1/2/2014 12:00:00 AM	1	2	32	0.15	\
39	1/3/2014 12:00:00 AM	1	2	142	0.55	\
61	1/3/2014 12:00:00 AM	1	7	56	0.15	\
62	1/3/2014 12:00:00 AM	1	5	54	0.15	\

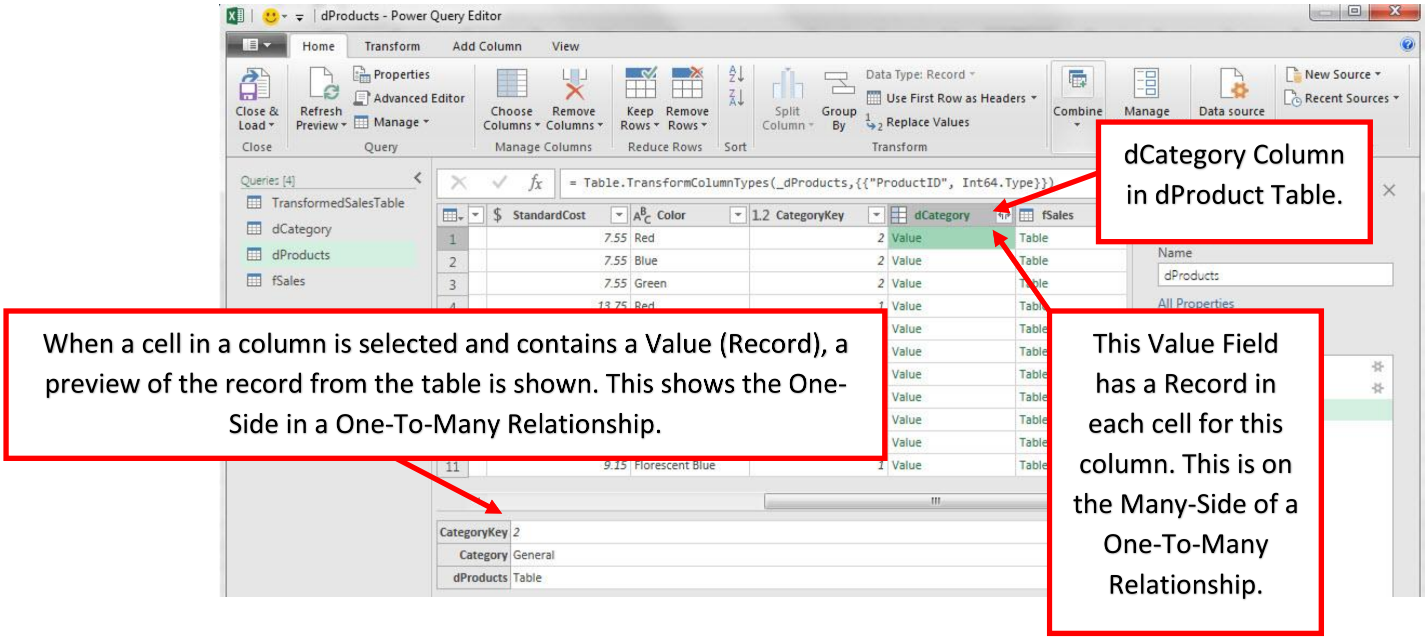
Name: dProducts

All Properties

9 COLUMNS, 11 ROWS

PREVIEW DOWNLOADED AT 12:22 PM

10) **Example of a Record Column (Value)**. If we look at the dCategory Column in the dProduct Table, we can see an example of a Field that contains a Record in each row (called Value). This is because there is a One-To-Many Relationship between the dProduct Table (Many-Side) and the dCategory Table (One-Side). In the picture below, you can see the first cell in the dCategory Field is selected, and below there is a preview of the one record from the dCategory Table for the product in that row. When we import from a Relational Database, a Value Field is always created like this and can make transformations easier to complete because the data from the other side of the relationship is available for any transformation we may need to complete. In our example we need to pull the Category Data from the dCategory Table over to the dProduct Table.



```
= Table.TransformColumnTypes(_dProducts,{{"ProductID", Int64.Type}})
```

	StandardCost	Color	CategoryKey	dCategory	fSales
1	7.55	Red	2	Value	Table
2	7.55	Blue	2	Value	Table
3	7.55	Green	2	Value	Table
4	13.75	Red	1	Value	Table
				Value	Table
				Value	Table
				Value	Table
				Value	Table
				Value	Table
				Value	Table
11	9.15	Florescent Blue	1	Value	Table

CategoryKey	2
Category	General
dProducts	Table

dCategory Column in dProduct Table.

When a cell in a column is selected and contains a Value (Record), a preview of the record from the table is shown. This shows the One-Side in a One-To-Many Relationship.

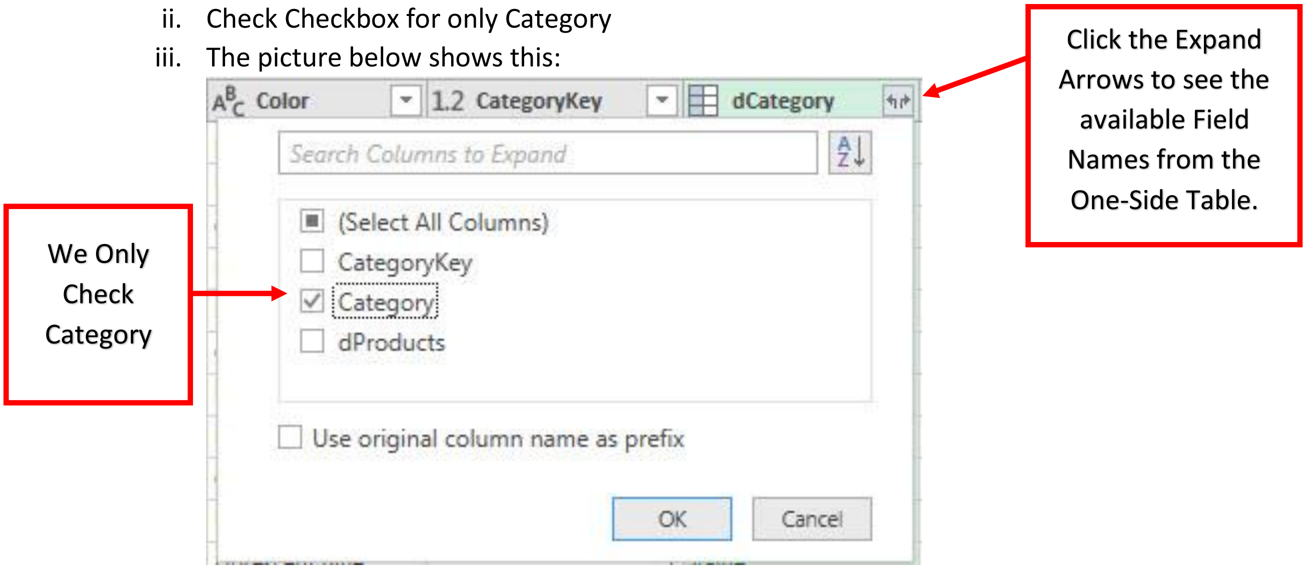
This Value Field has a Record in each cell for this column. This is on the Many-Side of a One-To-Many Relationship.

11) **Expand Arrow**. To Pull the Category Name for each Product in the dProduct Table, we need to:

i. Click the Expand Arrows

ii. Check Checkbox for only Category

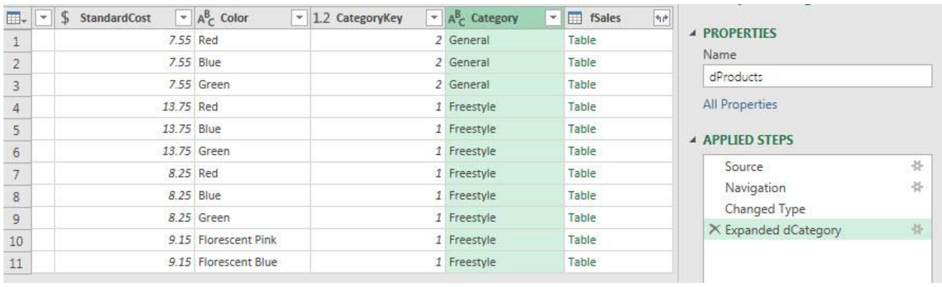
iii. The picture below shows this:



Click the Expand Arrows to see the available Field Names from the One-Side Table.

We Only Check Category

iv. After pulling the Categories from the dCategory Table, we can see the new Category Field in the dProduct Table. This helps us to accomplish our goal of creating one Dimension dProduct Table that we can use to create a Relationship with the fSales table and create the beginnings of a Star Schema data Model.



		StandardCost	Color	CategoryKey	Category	fSales
1		7.55	Red	2	General	Table
2		7.55	Blue	2	General	Table
3		7.55	Green	2	General	Table
4		13.75	Red	1	Freestyle	Table
5		13.75	Blue	1	Freestyle	Table
6		13.75	Green	1	Freestyle	Table
7		8.25	Red	1	Freestyle	Table
8		8.25	Blue	1	Freestyle	Table
9		8.25	Green	1	Freestyle	Table
10		9.15	Florescent Pink	1	Freestyle	Table
11		9.15	Florescent Blue	1	Freestyle	Table

PROPERTIES

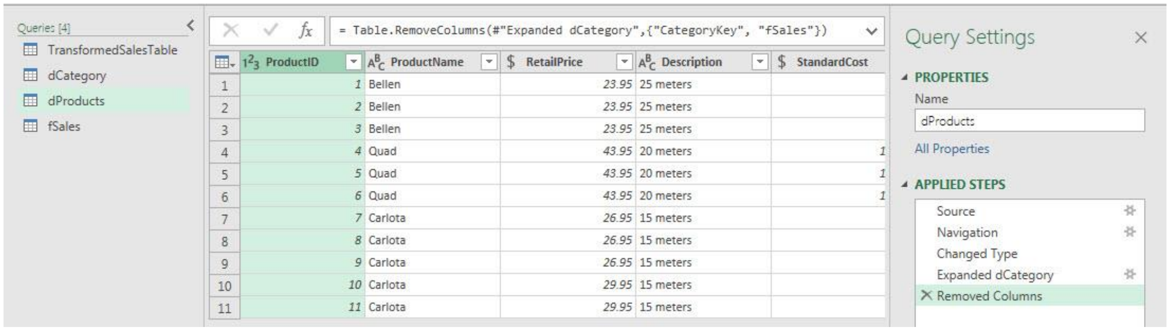
Name: dProducts

All Properties

APPLIED STEPS

- Source
- Navigation
- Changed Type
- Expanded dCategory

12) To Complete the transformation of the dProducts Table, we remove the fSales and CetgeoryKey Fields. The final set of Applied Steps are listed below:



Queries [4]

- TransformedSalesTable
- dCategory
- dProducts
- fSales

```
= Table.RemoveColumns(#"Expanded dCategory",{"CategoryKey", "fSales"})
```

	ProductID	ProductName	RetailPrice	Description	StandardCost
1	1	Bellen	23.95	25 meters	
2	2	Bellen	23.95	25 meters	
3	3	Bellen	23.95	25 meters	
4	4	Quad	43.95	20 meters	1
5	5	Quad	43.95	20 meters	1
6	6	Quad	43.95	20 meters	1
7	7	Carlota	26.95	15 meters	
8	8	Carlota	26.95	15 meters	
9	9	Carlota	26.95	15 meters	
10	10	Carlota	29.95	15 meters	
11	11	Carlota	29.95	15 meters	

Query Settings

PROPERTIES

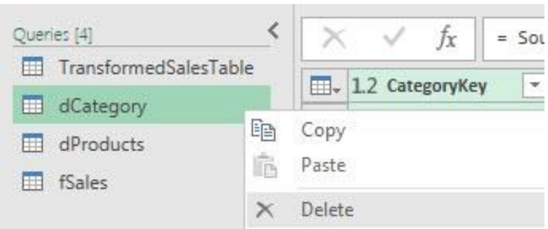
Name: dProducts

All Properties

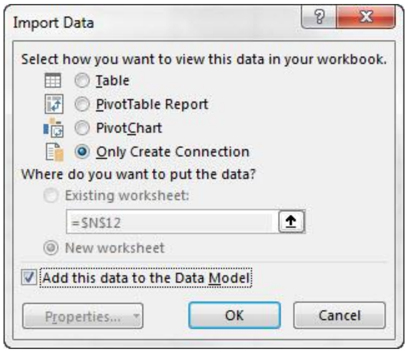
APPLIED STEPS

- Source
- Navigation
- Changed Type
- Expanded dCategory
- Removed Columns

13) **Delete Query**. When we imported the three tables from the Access Relational Database, we actually did not need to import the dCategory Table because that data was already imported in the Value Field in the dProduct Table (dCategory Column that contains the Category data that we expanded). For this reason, we can delete the Query we created to import the dCategory Table. To Delete the dCategory Table, right-click the list of Queries on the left side of the Power Query Editor and click on Delete. As shown below:



14) **Load to Data Model**. Using the "Close & Load To…" option in the Close & Load drop-down arrow in the Close group in the Home Ribbon Tab, open the Import Data Dialog Box and Select "Only Create Connection" and "Load to the Data Model", as shown here:



Import Data

Select how you want to view this data in your workbook.

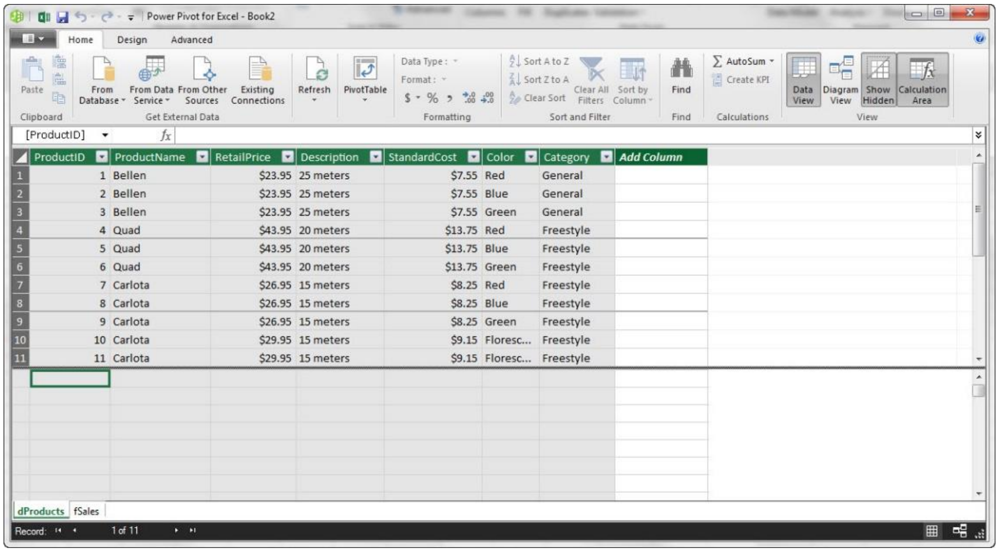
○ Table
○ PivotTable Report
○ PivotChart
● Only Create Connection

Where do you want to put the data?

○ Existing worksheet:
=$N$12
● New worksheet

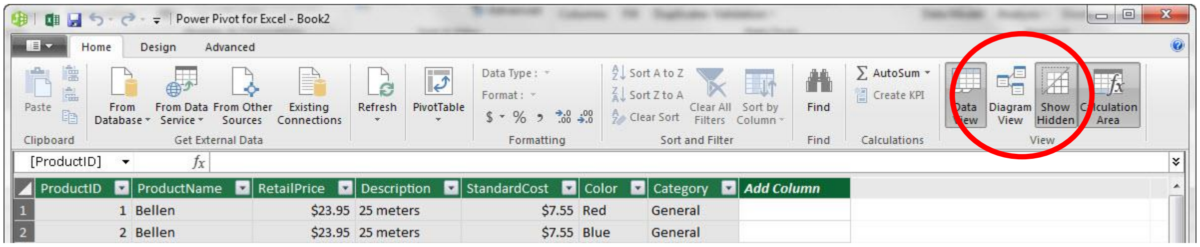
☑ Add this data to the Data Model

15) **Open Excel Power Pivot Data Model** in the Power Pivot Ribbon Tab, the Data Ribbon Tab or with the keyboard Alt, A, D, M. The Power Pivot Window should look like this:



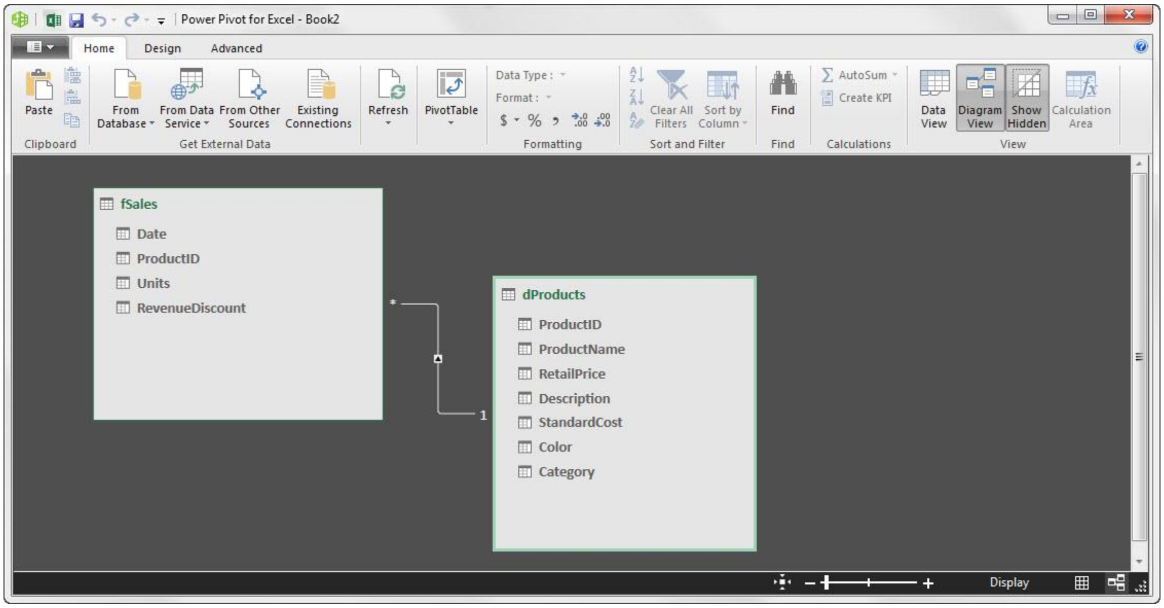
	ProductID	ProductName	RetailPrice	Description	StandardCost	Color	Category	Add Column
1	1	Bellen	$23.95	25 meters	$7.55	Red	General	
2	2	Bellen	$23.95	25 meters	$7.55	Blue	General	
3	3	Bellen	$23.95	25 meters	$7.55	Green	General	
4	4	Quad	$43.95	20 meters	$13.75	Red	Freestyle	
5	5	Quad	$43.95	20 meters	$13.75	Blue	Freestyle	
6	6	Quad	$43.95	20 meters	$13.75	Green	Freestyle	
7	7	Carlota	$26.95	15 meters	$8.25	Red	Freestyle	
8	8	Carlota	$26.95	15 meters	$8.25	Blue	Freestyle	
9	9	Carlota	$26.95	15 meters	$8.25	Green	Freestyle	
10	10	Carlota	$29.95	15 meters	$9.15	Floresc...	Freestyle	
11	11	Carlota	$29.95	15 meters	$9.15	Floresc...	Freestyle	

16) To See the Relationships View, click the Diagram View button in the View group in the Home Ribbon Tab



	ProductID	ProductName	RetailPrice	Description	StandardCost	Color	Category	Add Column
1	1	Bellen	$23.95	25 meters	$7.55	Red	General	
2	2	Bellen	$23.95	25 meters	$7.55	Blue	General	

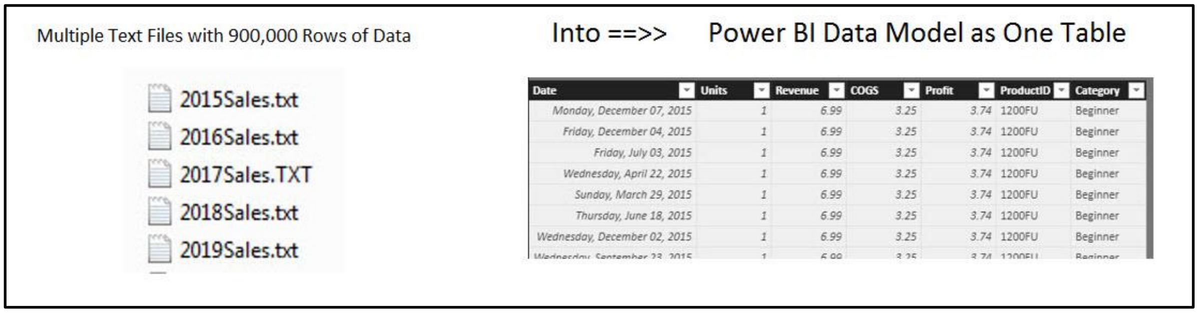
17) **Create Relationship & Star Schema Data Model**. In Diagram View you can move the tables, resize the tables and create a One-To-Many Relationship by dragging the ProductID Field from the dProducts Table to the fSales Table.



18) The above picture shows that we accomplished our goal of using Power Query to Extract data from an Access Relational Database, Transformed the Data in Power Query and then Loaded it to the Power Pivot data Model to the create a Relationship results in the start of a Star Schema Data Model.

3. **Example 3: ETL Data From Multiple Text Files into Power BI Desktop Data Model**.

1) **Goal is to 5 Different Text Files and Append into one Power BI Desktop Data Model Table**:



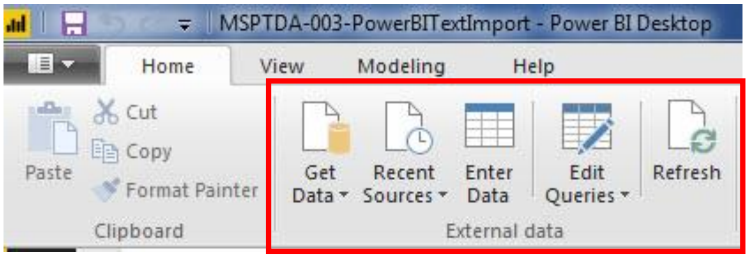
Multiple Text Files with 900,000 Rows of Data

- 2015Sales.txt
- 2016Sales.txt
- 2017Sales.TXT
- 2018Sales.txt
- 2019Sales.txt

Into ==>> Power BI Data Model as One Table

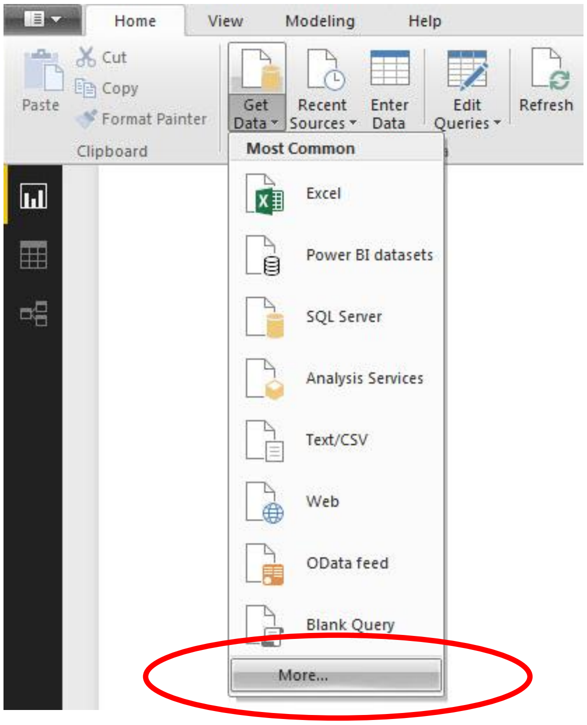
Date	Units	Revenue	COGS	Profit	ProductID	Category
Monday, December 07, 2015	1	6.99	3.25	3.74	1200FU	Beginner
Friday, December 04, 2015	1	6.99	3.25	3.74	1200FU	Beginner
Friday, July 03, 2015	1	6.99	3.25	3.74	1200FU	Beginner
Wednesday, April 22, 2015	1	6.99	3.25	3.74	1200FU	Beginner
Sunday, March 29, 2015	1	6.99	3.25	3.74	1200FU	Beginner
Thursday, June 18, 2015	1	6.99	3.25	3.74	1200FU	Beginner
Wednesday, December 02, 2015	1	6.99	3.25	3.74	1200FU	Beginner
Wednesday, September 23, 2015	1	6.99	3.25	3.74	1200FU	Beginner

2) **Power BI Desktop's Power Query is in External Data group in Home Ribbon**:

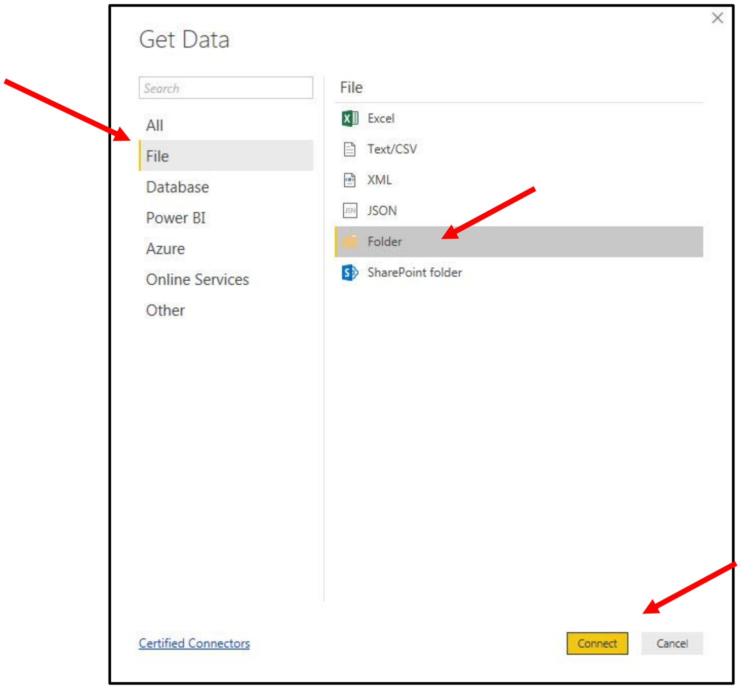


3)

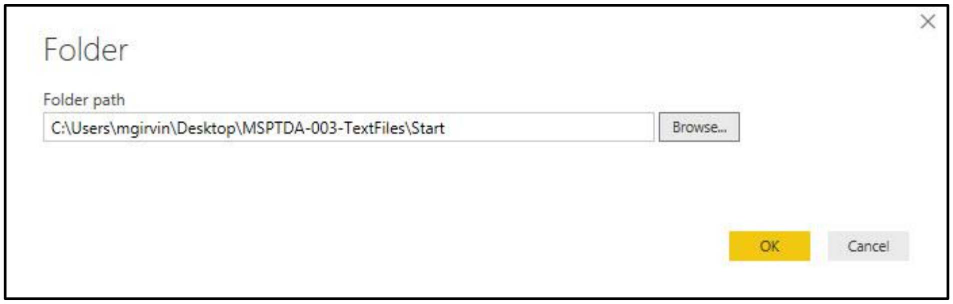
4) **Get Data**. To Import Data click the Get Data dropdown in the External Data group (Power Query) in the Home Ribbon Tab, and then click on More, as seen in this picture:



5) **From File, From Folder**. When the Get Data dialog box opens, in order to import multiple txt files, select File on the left and then Folder on the Right, then click the Connect button, as seen below:



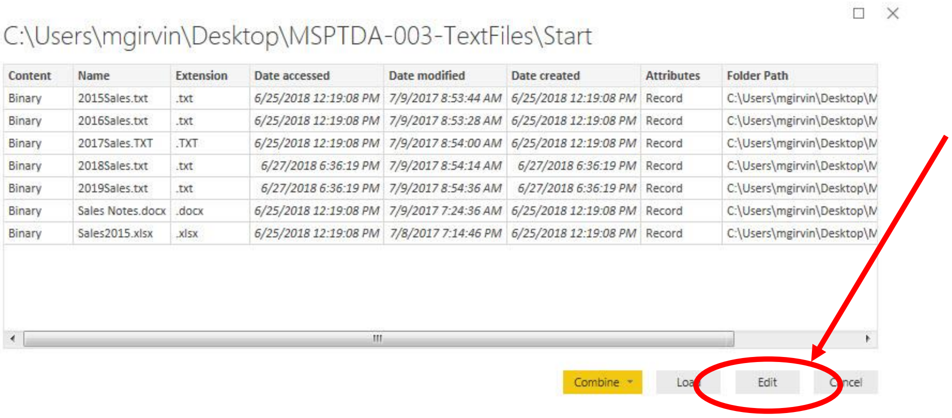
6) In the Folder path text box, use the Browse button to point Power Query to the Folder that contains the Text Files we want to import. In our example, the folder named "Start" contains out Text files that we need to import and append into a single table, as seen in this picture: .



Folder path

C:\Users\mgirvin\Desktop\MSPTDA-003-TextFiles\Start

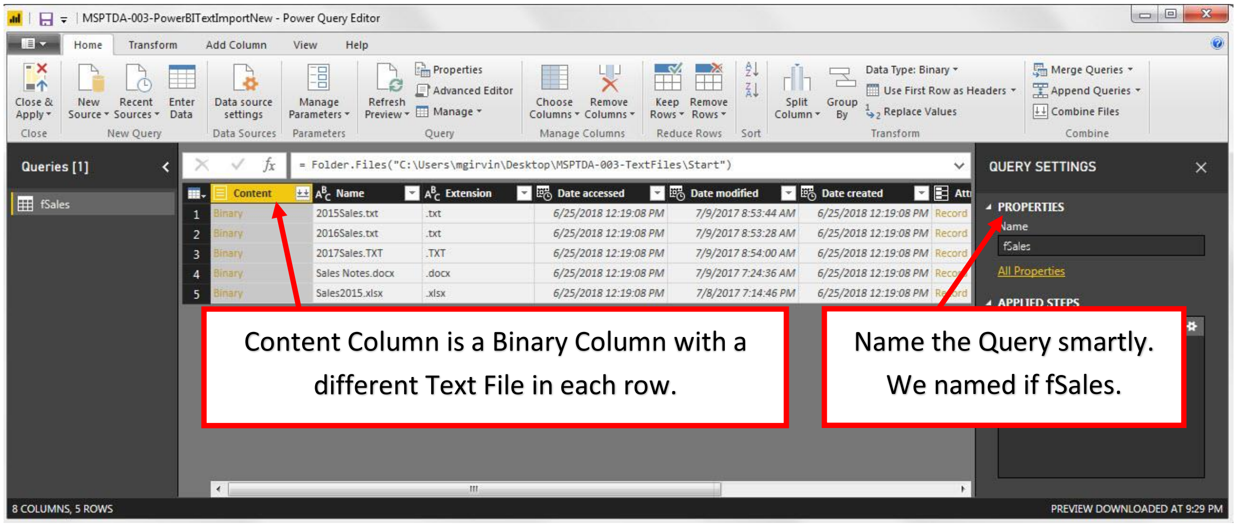
7) In the next dialog box, click the Edit button, as seen in this picture:



C:\Users\mgirvin\Desktop\MSPTDA-003-TextFiles\Start

Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
Binary	2015Sales.txt	.txt	6/25/2018 12:19:08 PM	7/9/2017 8:53:44 AM	6/25/2018 12:19:08 PM	Record	C:\Users\mgirvin\Desktop\M
Binary	2016Sales.txt	.txt	6/25/2018 12:19:08 PM	7/9/2017 8:53:28 AM	6/25/2018 12:19:08 PM	Record	C:\Users\mgirvin\Desktop\M
Binary	2017Sales.TXT	.TXT	6/25/2018 12:19:08 PM	7/9/2017 8:54:00 AM	6/25/2018 12:19:08 PM	Record	C:\Users\mgirvin\Desktop\M
Binary	2018Sales.txt	.txt	6/27/2018 6:36:19 PM	7/9/2017 8:54:14 AM	6/27/2018 6:36:19 PM	Record	C:\Users\mgirvin\Desktop\M
Binary	2019Sales.txt	.txt	6/27/2018 6:36:19 PM	7/9/2017 8:54:36 AM	6/27/2018 6:36:19 PM	Record	C:\Users\mgirvin\Desktop\M
Binary	Sales Notes.docx	.docx	6/25/2018 12:19:08 PM	7/9/2017 7:24:36 AM	6/25/2018 12:19:08 PM	Record	C:\Users\mgirvin\Desktop\M
Binary	Sales2015.xlsx	.xlsx	6/25/2018 12:19:08 PM	7/8/2017 7:14:46 PM	6/25/2018 12:19:08 PM	Record	C:\Users\mgirvin\Desktop\M

8) **Binary Column with Text Files**. When the Power Query Editor Opens be sure to give the Query a smart name in the Properties Name box, as seen in this picture below. Also, notice that Power Query brought in columns with retributes of each file in the folder. The columns show things like File Name and Extension. The Content Column is a Binary Column with a different Text File in each row.



Queries: fSales

`= Folder.Files("C:\Users\mgirvin\Desktop\MSPTDA-003-TextFiles\Start")`

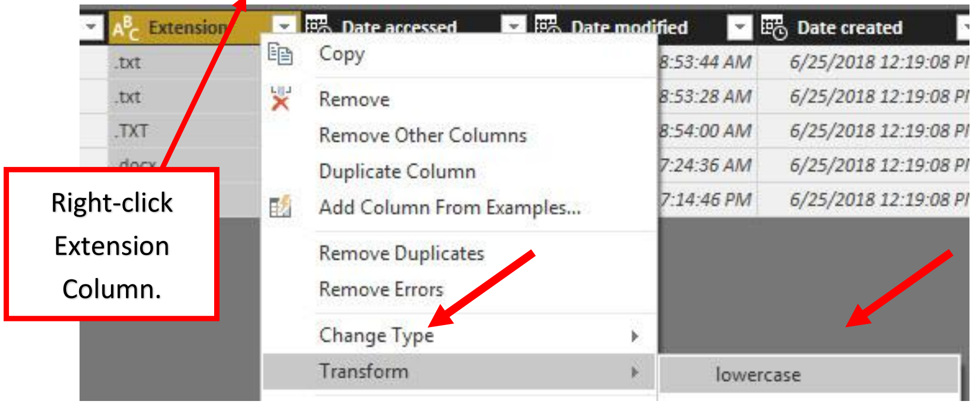
	Content	Name	Extension	Date accessed	Date modified	Date created	Att
1	Binary	2015Sales.txt	.txt	6/25/2018 12:19:08 PM	7/9/2017 8:53:44 AM	6/25/2018 12:19:08 PM	Record
2	Binary	2016Sales.txt	.txt	6/25/2018 12:19:08 PM	7/9/2017 8:53:28 AM	6/25/2018 12:19:08 PM	Record
3	Binary	2017Sales.TXT	.TXT	6/25/2018 12:19:08 PM	7/9/2017 8:54:00 AM	6/25/2018 12:19:08 PM	Record
4	Binary	Sales Notes.docx	.docx	6/25/2018 12:19:08 PM	7/9/2017 7:24:36 AM	6/25/2018 12:19:08 PM	Record
5	Binary	Sales2015.xlsx	.xlsx	6/25/2018 12:19:08 PM	7/8/2017 7:14:46 PM	6/25/2018 12:19:08 PM	Record

Name: fSales

Content Column is a Binary Column with a different Text File in each row.

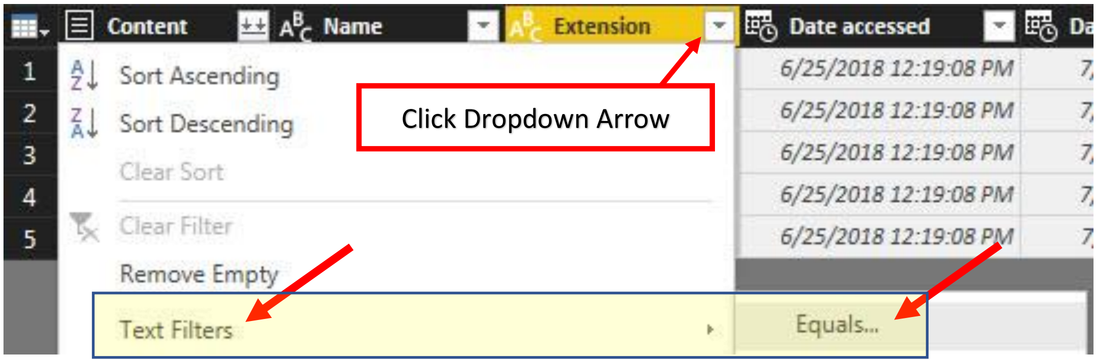
Name the Query smartly. We named if fSales.

9) **Convert to lowercase**. Because Power Query is Case Sensitive, and because we might get Text Files with extensions that have lowercase or uppercase letters, we need to convert all the Extensions to lowercase. We can do this by right-clicking the Extension Column, Point to Transform, then click on lowercase, as seen in this picture:



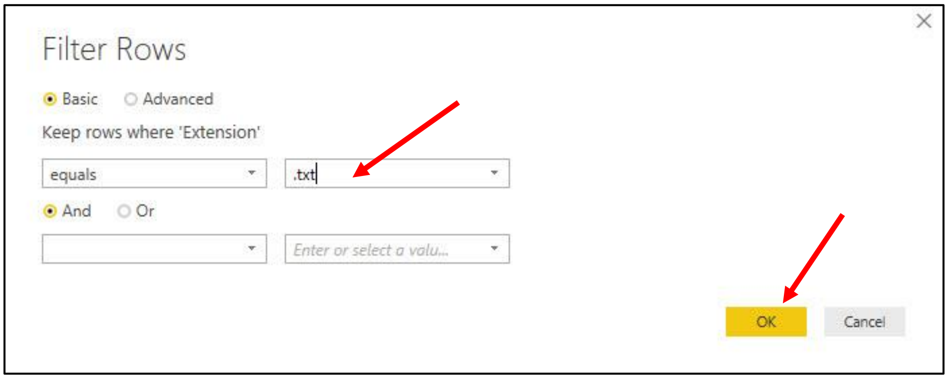
Right-click Extension Column.

10) **Filter ".txt"**. In order to only import the correct Text Files, we need to filter the Extension Column to only include files that contain ".txt". To do this, click the dropdown arrow in the Extension Column, then point to Text Filters and then click on Equals.



Click Dropdown Arrow

11) In the Filter Rows dialog box, enter ".txt" into the textbox for equals. Then click OK.



Filter Rows

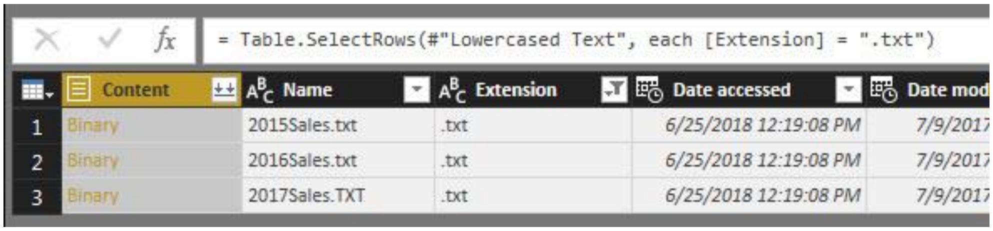
● Basic ○ Advanced

Keep rows where 'Extension'

equals	.txt

● And ○ Or

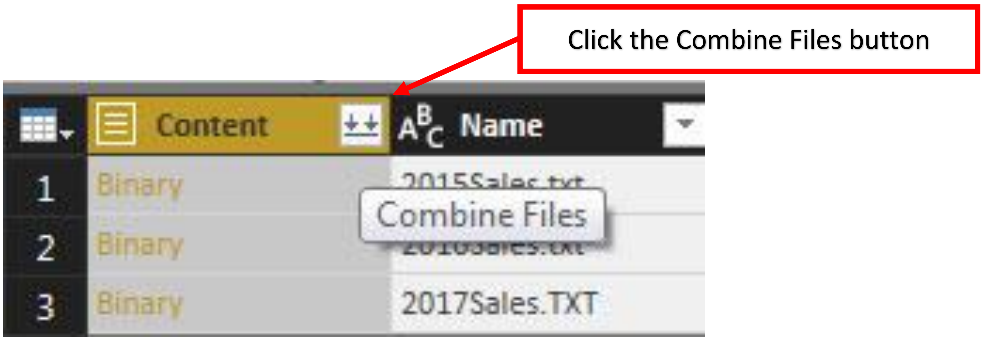
12) Now we can see that only the files containing Text Files are showing, as seen in this picture:



```
= Table.SelectRows(#"Lowercased Text", each [Extension] = ".txt")
```

	Content	Name	Extension	Date accessed	Date mod
1	Binary	2015Sales.txt	.txt	6/25/2018 12:19:08 PM	7/9/2017
2	Binary	2016Sales.txt	.txt	6/25/2018 12:19:08 PM	7/9/2017
3	Binary	2017Sales.TXT	.txt	6/25/2018 12:19:08 PM	7/9/2017

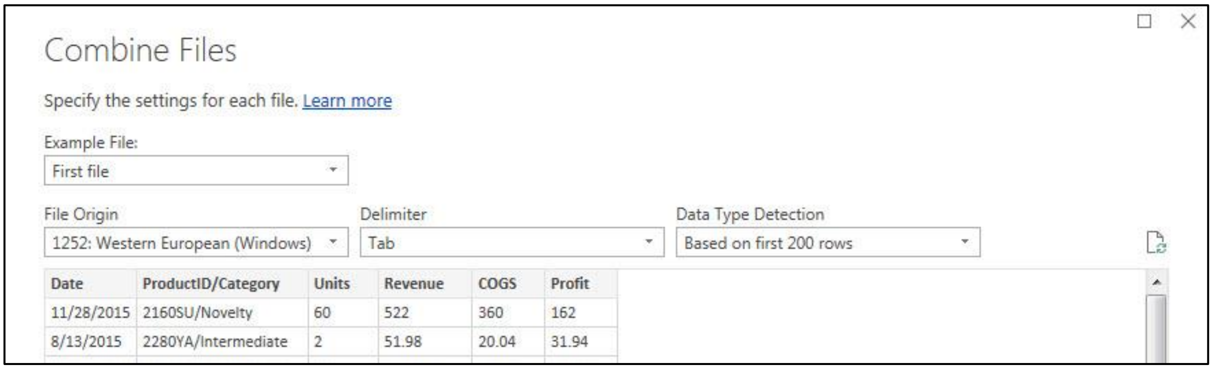
13) **Combine Files button**. To combine all the Text Files into one table, we need to Append the tables, one on top of the other. To do this we click the Double Downward Point Arrows, or Combine Files button, in the right corner of the Content Field, as seen in this picture:



Click the Combine Files button

	Content	Name
1	Binary	2015Sales.txt
2	Binary	2016Sales.txt
3	Binary	2017Sales.TXT

14) In the next Combine Files dialog box, we have to confirm Example File, File Origin, Delimiter and Data Type Detection , as seen in this picture:



Combine Files

Specify the settings for each file. Learn more

Example File: First file

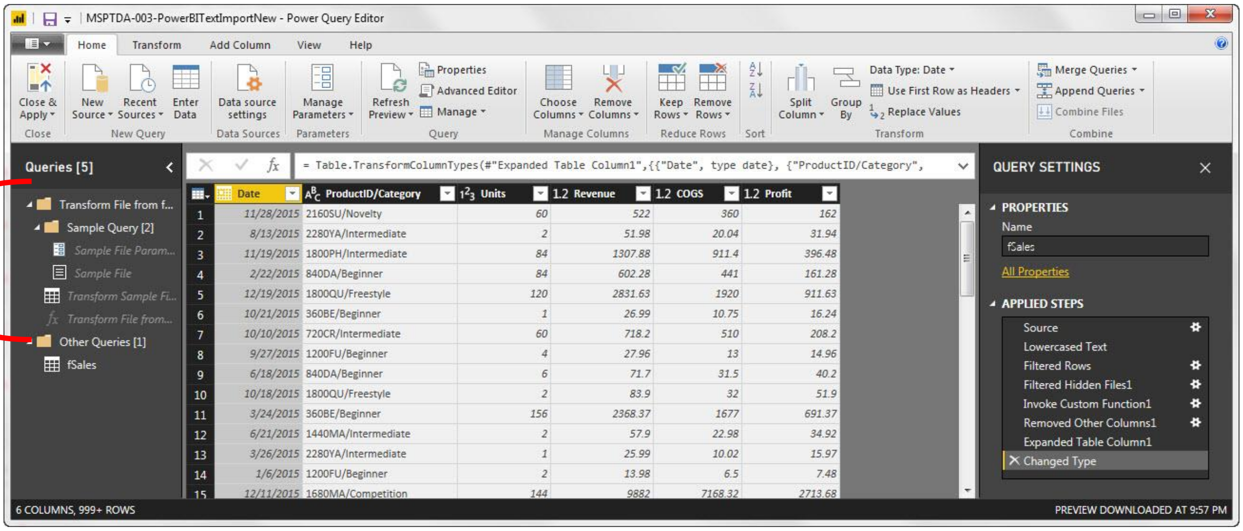
File Origin: 1252: Western European (Windows)

Delimiter: Tab

Data Type Detection: Based on first 200 rows

Date	ProductID/Category	Units	Revenue	COGS	Profit
11/28/2015	2160SU/Novelty	60	522	360	162
8/13/2015	2280YA/Intermediate	2	51.98	20.04	31.94

15) After we click OK in the Combine Files dialog box, Power Query builds a number of queries for us, and appends all the text file tables, one on top of each other, to create a single table, as seen in this picture:



Queries [5]
- Transform File from f...
  - Sample Query [2]
    - Sample File Param...
    - Sample File
  - Transform Sample Fi...
  - Transform File from...
- Other Queries [1]
  - fSales

```
= Table.TransformColumnTypes(#"Expanded Table Column1",{{"Date", type date}, {"ProductID/Category",
```

	Date	ProductID/Category	Units	Revenue	COGS	Profit
1	11/28/2015	2160SU/Novelty	60	522	360	162
2	8/13/2015	2280YA/Intermediate	2	51.98	20.04	31.94
3	11/19/2015	1800PH/Intermediate	84	1307.88	911.4	396.48
4	2/22/2015	840DA/Beginner	84	602.28	441	161.28
5	12/19/2015	1800QU/Freestyle	120	2831.63	1920	911.63
6	10/21/2015	360BE/Beginner	1	26.99	10.75	16.24
7	10/10/2015	720CR/Intermediate	60	718.2	510	208.2
8	9/27/2015	1200FU/Beginner	4	27.96	13	14.96
9	6/18/2015	840DA/Beginner	6	71.7	31.5	40.2
10	10/18/2015	1800QU/Freestyle	2	83.9	32	51.9
11	3/24/2015	360BE/Beginner	156	2368.37	1677	691.37
12	6/21/2015	1440MA/Intermediate	2	57.9	22.98	34.92
13	3/26/2015	2280YA/Intermediate	1	25.99	10.02	15.97
14	1/6/2015	1200FU/Beginner	2	13.98	6.5	7.48
15	12/11/2015	1680MA/Competition	144	9882	7168.32	2713.68

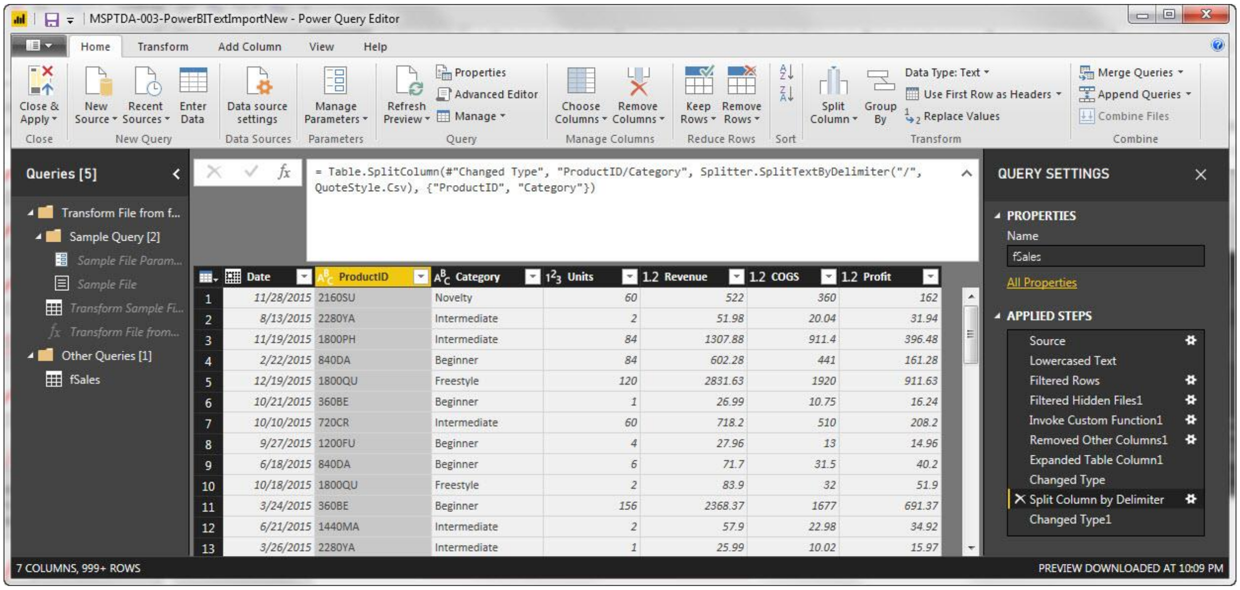
6 COLUMNS, 999+ ROWS

Name: fSales

APPLIED STEPS
- Source
- Lowercased Text
- Filtered Rows
- Filtered Hidden Files1
- Invoke Custom Function1
- Removed Other Columns1
- Expanded Table Column1
- Changed Type

Queries automatically created to Combine (Append) the Tables

16) **Split by Delimiter**. We can split the ProductID Column/Category Column in the same way we split and renamed the columns in our Excel Power Query Example #1: Split By Delimiter, then edit the M Code in the Formula Bar to change the column names. Here is a picture of the transformation:



Queries [5]
- Transform File from f...
  - Sample Query [2]
    - Sample File Param...
    - Sample File
  - Transform Sample Fi...
  - Transform File from...
- Other Queries [1]
  - fSales

```
= Table.SplitColumn(#"Changed Type", "ProductID/Category", Splitter.SplitTextByDelimiter("/",
QuoteStyle.Csv), {"ProductID", "Category"})
```

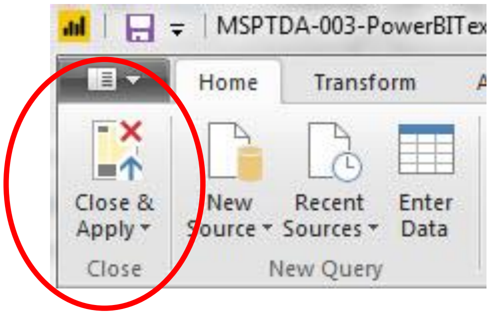
	Date	ProductID	Category	Units	Revenue	COGS	Profit
1	11/28/2015	2160SU	Novelty	60	522	360	162
2	8/13/2015	2280YA	Intermediate	2	51.98	20.04	31.94
3	11/19/2015	1800PH	Intermediate	84	1307.88	911.4	396.48
4	2/22/2015	840DA	Beginner	84	602.28	441	161.28
5	12/19/2015	1800QU	Freestyle	120	2831.63	1920	911.63
6	10/21/2015	360BE	Beginner	1	26.99	10.75	16.24
7	10/10/2015	720CR	Intermediate	60	718.2	510	208.2
8	9/27/2015	1200FU	Beginner	4	27.96	13	14.96
9	6/18/2015	840DA	Beginner	6	71.7	31.5	40.2
10	10/18/2015	1800QU	Freestyle	2	83.9	32	51.9
11	3/24/2015	360BE	Beginner	156	2368.37	1677	691.37
12	6/21/2015	1440MA	Intermediate	2	57.9	22.98	34.92
13	3/26/2015	2280YA	Intermediate	1	25.99	10.02	15.97

7 COLUMNS, 999+ ROWS

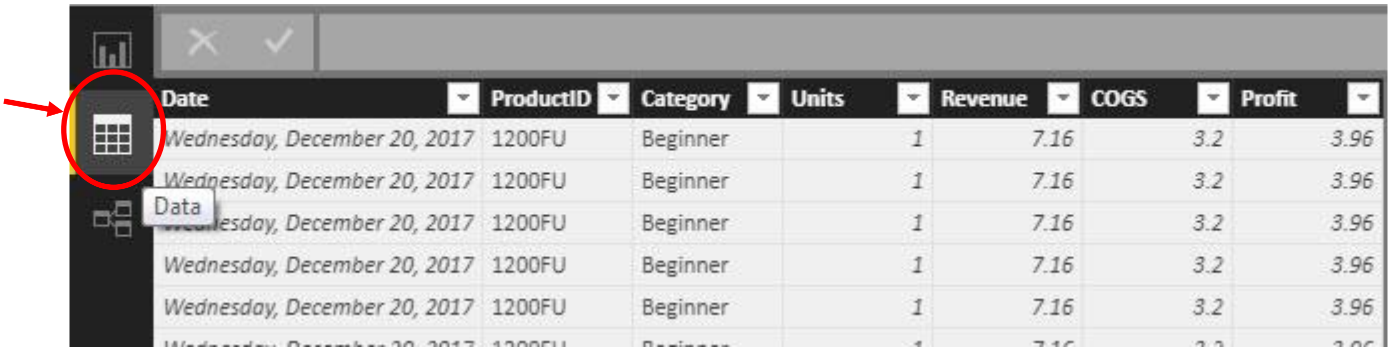
Name: fSales

APPLIED STEPS
- Source
- Lowercased Text
- Filtered Rows
- Filtered Hidden Files1
- Invoke Custom Function1
- Removed Other Columns1
- Expanded Table Column1
- Changed Type
- Split Column by Delimiter
- Changed Type1

17) **Close & Apply**. We can load the single table to the Power BI Desktop Data Model by clicking on the Close & Apply button in the Home Ribbon Tab , as seen in this picture:

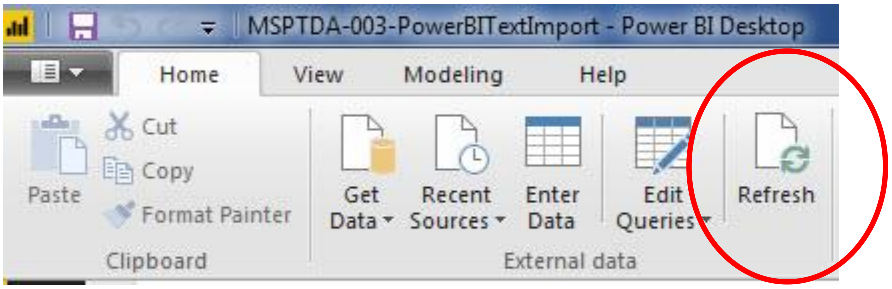


18) **Data Button to see Table**. On the Left side of Power BI Desktop, you can click the Data Button to see the Loaded Table. The table now has over 500,000 rows of data from the 2015, 2016 and 2017 Text Files.



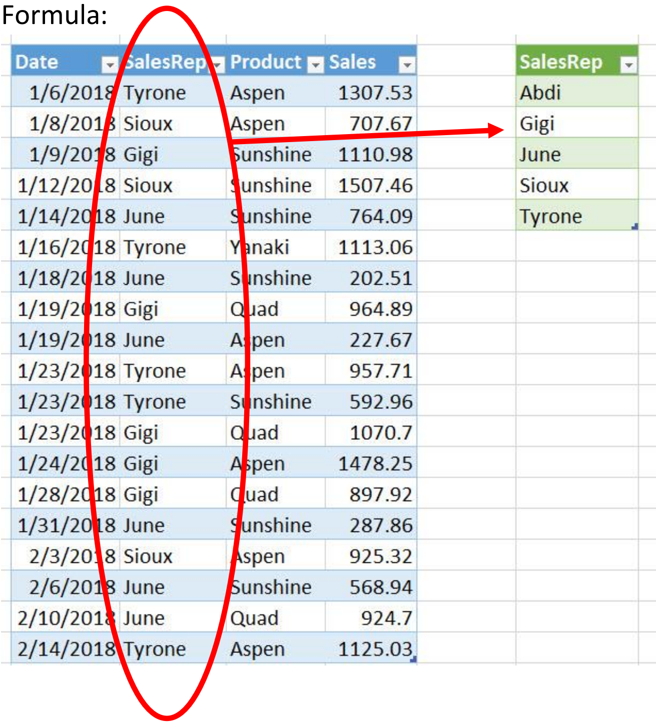
Date	ProductID	Category	Units	Revenue	COGS	Profit
Wednesday, December 20, 2017	1200FU	Beginner	1	7.16	3.2	3.96
Wednesday, December 20, 2017	1200FU	Beginner	1	7.16	3.2	3.96
Wednesday, December 20, 2017	1200FU	Beginner	1	7.16	3.2	3.96
Wednesday, December 20, 2017	1200FU	Beginner	1	7.16	3.2	3.96
Wednesday, December 20, 2017	1200FU	Beginner	1	7.16	3.2	3.96

19) **Refresh after new files added to folder**. After you copy and paste the 2018 and 2019 Text Files (from the Add Later Folder that was in the Zipped Download Data Folder) into the Start Folder, you can click the Refresh button in the External Data group in the Home Ribbon Tab and now the table will have over 900,000 rows of data.



4. **Example 4: Replace Complex Excel Array Formulas with Simple Power Query Solution**.

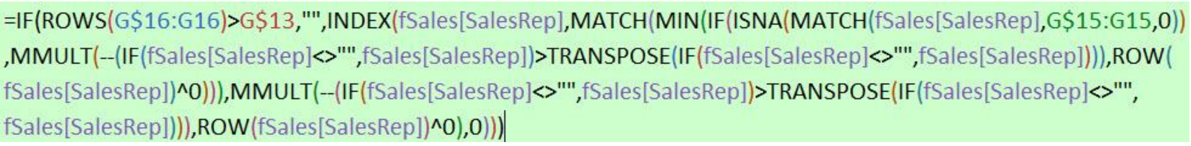
   1) Goal is to extract a unique list of values from the Sales Rep Column, without using an Excel Array Formula:



Date	SalesRep	Product	Sales
1/6/2018	Tyrone	Aspen	1307.53
1/8/2018	Sioux	Aspen	707.67
1/9/2018	Gigi	Sunshine	1110.98
1/12/2018	Sioux	Sunshine	1507.46
1/14/2018	June	Sunshine	764.09
1/16/2018	Tyrone	Yanaki	1113.06
1/18/2018	June	Sunshine	202.51
1/19/2018	Gigi	Quad	964.89
1/19/2018	June	Aspen	227.67
1/23/2018	Tyrone	Aspen	957.71
1/23/2018	Tyrone	Sunshine	592.96
1/23/2018	Gigi	Quad	1070.7
1/24/2018	Gigi	Aspen	1478.25
1/28/2018	Gigi	Quad	897.92
1/31/2018	June	Sunshine	287.86
2/3/2018	Sioux	Aspen	925.32
2/6/2018	June	Sunshine	568.94
2/10/2018	June	Quad	924.7
2/14/2018	Tyrone	Aspen	1125.03

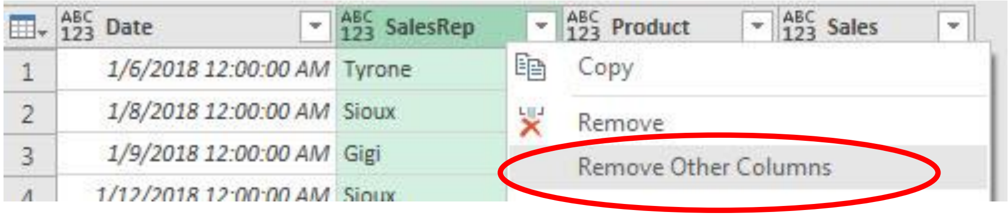
SalesRep
Abdi
Gigi
June
Sioux
Tyrone

   2) Here the Array Formula we are trying to avoid using:



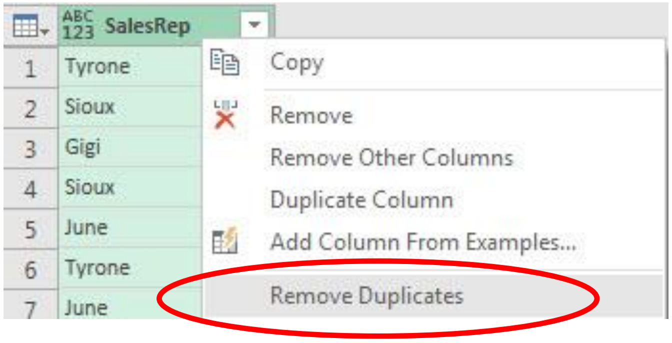
```
=IF(ROWS(G$16:G16)>G$13,"",INDEX(fSales[SalesRep],MATCH(MIN(IF(ISNA(MATCH(fSales[SalesRep],G$15:G15,0))
,MMULT(--(IF(fSales[SalesRep]<>"",fSales[SalesRep])>TRANSPOSE(IF(fSales[SalesRep]<>"",fSales[SalesRep]))),ROW(
fSales[SalesRep])^0))),MMULT(--(IF(fSales[SalesRep]<>"",fSales[SalesRep])>TRANSPOSE(IF(fSales[SalesRep]<>"",
fSales[SalesRep]))),ROW(fSales[SalesRep])^0),0)))
```

   3) After we import the Excel Table into Power Query, our first step is to Remove all columns except the Sales Rep Column. We can do this by right-clicking the Sales Rep Column and then clicking on Remove Other Columns, as seen in this picture:



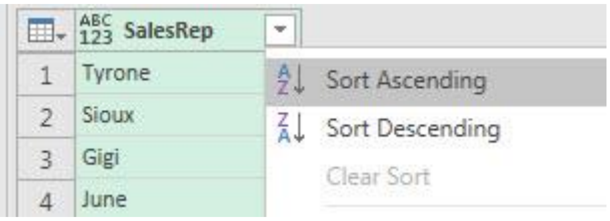
	Date	SalesRep	Product	Sales
1	1/6/2018 12:00:00 AM	Tyrone		
2	1/8/2018 12:00:00 AM	Sioux		
3	1/9/2018 12:00:00 AM	Gigi		
4	1/12/2018 12:00:00 AM	Sioux		

   4) To Remove the Duplicates in the Sale Rep column, right-click the Sales Rep Column and then click on Remove Duplicates, as seen in this picture:



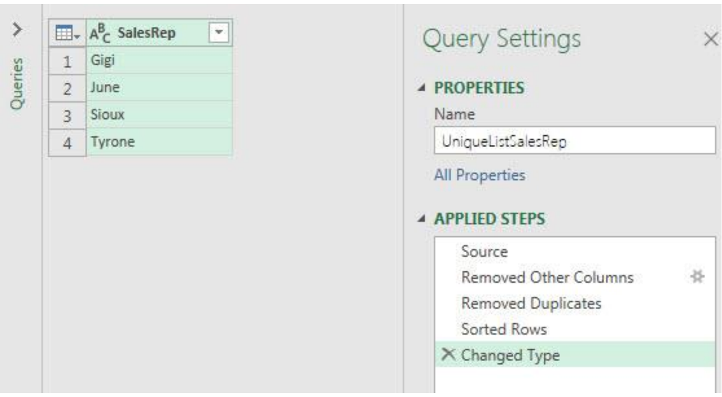
	SalesRep
1	Tyrone
2	Sioux
3	Gigi
4	Sioux
5	June
6	Tyrone
7	June

5) To Sort the column, click the dropdown arrow in the SalesRep Column and then click on A to Z.



	SalesRep
1	Tyrone
2	Sioux
3	Gigi
4	June

6) After changing the Data Type for the SalesRep Column, we can see the five Applied Steps.



	SalesRep
1	Gigi
2	June
3	Sioux
4	Tyrone

Name: UniqueListSalesRep

Source
Removed Other Columns
Removed Duplicates
Sorted Rows
Changed Type

7) We can click the Close & Load dropdown arrow and then select "Close & Load To…" and load it to our Excel Sheet.

Important Keyboards Seen in this Video:

1. Create Excel Table = **Ctrl + T**
2. Name an Excel Table = **Alt, J, T, A**
3. Highlight a Table in an Excel Worksheet: **Ctrl + *** (Number Pad) or **Ctrl + Shift + 8**
4. Rename a Query, a Transformation Step or a Field name in the Power Query Editor = **F2 Key**
5. Move to next Field in the Power Query Editor = **Arrow Key**
6. Open Excel data Model = **Alt, A, D, M**
7. Create PivotTable = **Alt, N, V**

This is what I (Mike excelisfun Girvin) like about Power Query:

1. Extract and Import Data from Multiple Sources
2. Clean and Transform Data into Desired Form
3. Load Data to:
    - An Excel Worksheet
    - Excel Power Pivot Data Model
    - Power BI Desktop Data Model
    - Connection Only
4. When the Data is Loaded, it can be Loaded as either:
    - The Final Useful Information
    - Data that will be used Later to Create Useful Information.
5. Use Power Query to Replace Complex Excel Array Formula Solutions.
6. Power Query is both in Excel and Power BI Desktop
7. Behind the scenes for every Power Query Solution is a Function Based, Case Sensitive Language called "M Code".
8. Creating a Power Query Solution can be done:
    - Using the Power Query User Interface
    - Manually typing the M Code
    - A Combination of Both
9. When you use the Power Query User Interface, Each Step in the Power Query Process is automatically Documented in two places:
    - Applied Steps Pane, where you see a list of steps
    - Advanced Editor, where the M Code is located
10. Each Step in the Power Query Process can be Edited in one of three places:
    - Applied Steps Pane, using the User Interface
    - The Formula Bar, where you can edit parts of each line of code
    - The Advanced Editor, where you have full control to edit all the M Code
11. After the Useful Information is Created we can:
    - Refresh the Useful Information when New Data Arrives
    - Go Back and Edit Any of the Steps in the Power Query Process