Microsoft Power Tools for Data Analysis #14

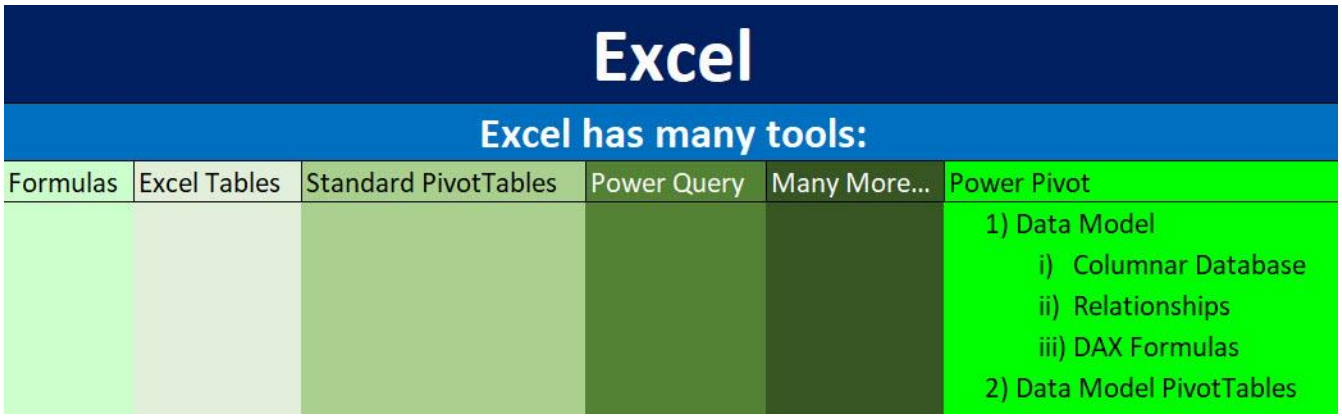
Power Pivot Into #2: Amazing Columnar Database:

Power Query & Power Pivot to Import Millions of Rows of Data into Excel For PivotTable Report

Notes from Video:

Table of Contents:

1. **Power Pivot comes in Office 365**
    1) It has been around in earlier versions, as far back as Excel 2010, but you either had to add it as an add-in or buy the correct version of Excel.
2. **Show Power Pivot Ribbon Tab in Excel**
    1) Click on the File Tab, then click on Options, then on the left, click on Add-ins, then in the Manage textbox dropdown, select “Com Add-ins”, then check the check Box for Power Pivot.
3. **Hierarchical Picture of Excel Power Pivot**
    1) Excel is a program with many tools
    2) Power Pivot is just one of the tools in Excel
    3) The tool Power Pivot has two main parts:
        i. Data Model
        ii. Data Model PivotTables
    4) The Data Model is made up of three main parts:
        i. Columnar Database
        ii. Relationships
        iii. DAX Formulas
    5) From the Data Model, we make Data Model PivotTables.
    6) Picture:



4. **Why the name Power Pivot?**
    1) Because Microsoft wanted to use the same amazing PivotTable user interface to drag and drop fields to make reports but with more Power.
    2) The “Power” part of the name means:
        i. We can make PivotTables from “Big Data”
        ii. We can make PivotTables from multiple Tables
        iii. We can use DAX Formulas, which can process over big data efficiently and which allows us more varied calculations than in a Standard PivotTable.
    3) The “Pivot” part of the name means we can use a PivotTable user interface, that we all know and love!

5. **Columnar Database (Part of Data Model)**

1) Columnar Database is part of the Data Model in both Excel Power Pivot and Power BI Desktop.
2) When you import tables to the Excel Power Pivot or Power BI Desktop Data Model, the data is stored in the Columnar Database.
3) **Define Columnar Database**
    i. Behind the scenes in RAM Memory Efficient Big Data Analytics Database
        1. **"Behind the scenes"** = When you open the Excel file, the Columnar Database is opened behind the scenes in RAM Memory.
        2. **"Efficient Big Data"** = Database encodes and compresses data and stores it in a structure that allows DAX Formulas to make calculations quickly and produces a small file size.
        3. **"Analytics Database"** = Database is specifically designed to work with the DAX Formula language to make calculations quickly on "Big Data".
4) **Columnar Database has many synonyms**, such as:
    i. Columnar database
    ii. VertiPaq engine
    iii. SSAS Tabular (SQL Server Analysis Services Tabular)
    iv. Storage Engine
    v. XVelocity analytics engine
5) **Overview of What Columnar Database does**
    i. Database takes a table with many columns and stores each column separately as a unique list of values and builds a map to help it reconstruct the table when needed for making DAX Formula calculations.
    ii. Why store data is separate columns with a unique list of items?
        1. Many calculations can be performed more quickly on individual columns with unique lists of values, rather than making calculation row by row.
        2. File size is reduced, and when the database is loaded into RAM Memory, less RAM is used, more is left for other tasks.
    iii. This means that for columns with only a few unique values, the file size reduction can be dramatic. (Number of unique items in a column = "Cardinality").
    iv. Here is a picture that helps to think about how data from a table is stored in the Columnar Database:

Columnar Database takes each column in the table and stores the columns separately, each as a unique list of items.

Original Table:

Sales	Sales Rep	Region
$54.00	Jo	West
$26.00	Nina	East
$54.00	Jo	South
$57.00	Kip	West
$22.00	Gigi	West
$59.00	Gigi	South
$95.00	Kip	East
$99.00	Kip	South
$51.00	Nina	South
$49.00	Nina	East
$12.00	Jo	East
$30.00	Jo	East
$20.00	Nina	West
$92.00	Kip	West
$73.00	Gigi	South

You can picture a Columnar Database like this:

Sales
$54.00
$26.00
$57.00
$22.00
$59.00
$95.00
$99.00
$51.00
$49.00
$12.00
$30.00
$20.00
$92.00
$73.00

Sales Rep
Jo
Nina
Kip
Gigi

Region
West
East
South

6) **Columnar Database performs "Vertipaq" Compression when a Table is imported**
    i. Steps is VertiPaq Compression:
        1. Read source dataset.
        2. For each Raw Data Column in the table, a unique list of values is determined, and that list is encoded, compressed and stored as a separate unit (memory block with a certain file size).
        3. Relationships are also stored in the Columnar Database.
        4. DAX Calculated Column are also calculated and compressed, but these columns are compressed last in the process and sometimes are not stored as efficiently as Raw Data Columns.
        5. An internal "map" is created that helps the Columnar Database engine can use to reconstruct the original table when it is needed to make a calculation.
            i. This map helps DAX Formulas to work quickly.
                i. If the calculation is on a single column, the map works quickly because it is working on a single unique list of items, a single memory black.
                ii. If the calculation is on a few columns, the map must work between multiple columns and will take longer than a single column calculation.
                iii. If the calculation is complex, at some point the map will actually materialize a full table in order to make the calculation. These calculations can sometimes take a long time.
    ii. The Vertipaq Compression uses many different techniques to encode and compress the data. It depends on such things as data types, variation in the data, mathematical patterns and more. Some of the techniques are proprietary to Microsoft and we cannot know the exact process.
        1. Some of the techniques used are:
            i. **Value Encoding**. For integer values, the Vertipaq engine may determine mathematical patterns that can help encode and compress to save the data in a smaller file size. This is called "Value Encoding".
            ii. **Dictionary Encoding**. For strings, the Vertipaq engine may build a dictionary of unique values with a first column of lookup integer values, and then replace the values in the original column with the integer lookup values. This is called "Dictionary Encoding". This can reduce file size. A picture of this is seen here:

Original Column:

Sales Rep
Jo
Nina
Jo
Kip
Gigi
Gigi
Kip
Kip
Nina
Nina
Jo
Jo
Nina
Kip
Gigi

Integer Index:

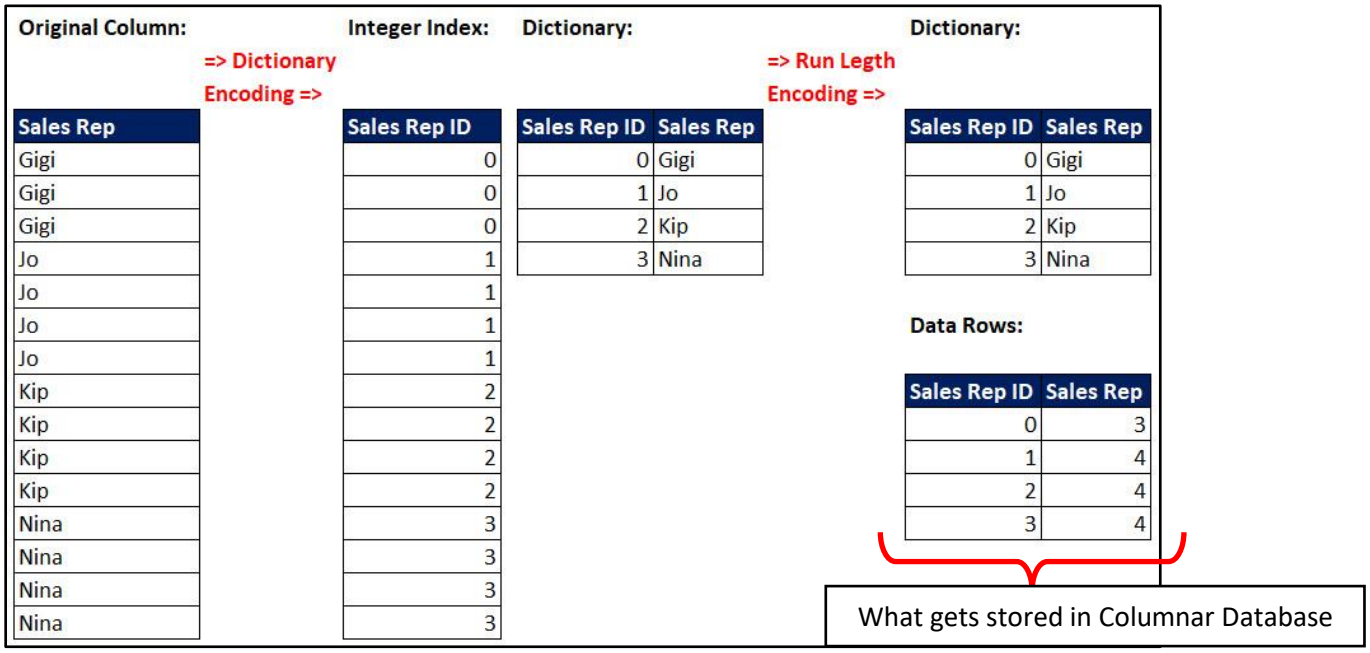
Sales Rep ID
0
1
0
2
3
3
2
2
1
1
0
0
1
2
3

Dictionary:

Sales Rep ID	Sales Rep
0	Jo
1	Nina
2	Kip
3	Gigi

What gets stored in Columnar Database

iii. **Run Length Encoding**. After Value Encoding or Dictionary Encoding is performed, Run Length Encoding may be able to further reduce the file size. This method tries to reduce file size by avoiding repeat values. If there is a pattern of many consecutive repeated values, Run Length Encoding will keep the Dictionary, but rather than having an Index Column is build a table that simple counts the repeated values, as seen in the below picture.



Sales Rep
Gigi
Gigi
Gigi
Jo
Jo
Jo
Jo
Kip
Kip
Kip
Kip
Nina
Nina
Nina
Nina

Sales Rep ID
0
0
0
1
1
1
1
2
2
2
2
3
3
3
3

Sales Rep ID	Sales Rep
0	Gigi
1	Jo
2	Kip
3	Nina

Sales Rep ID	Sales Rep
0	Gigi
1	Jo
2	Kip
3	Nina

Sales Rep ID	Sales Rep
0	3
1	4
2	4
3	4

iv. Value Encoding, Dictionary Encoding, and Run Length Encoding help reduce file size and help formulas calculate more quickly.

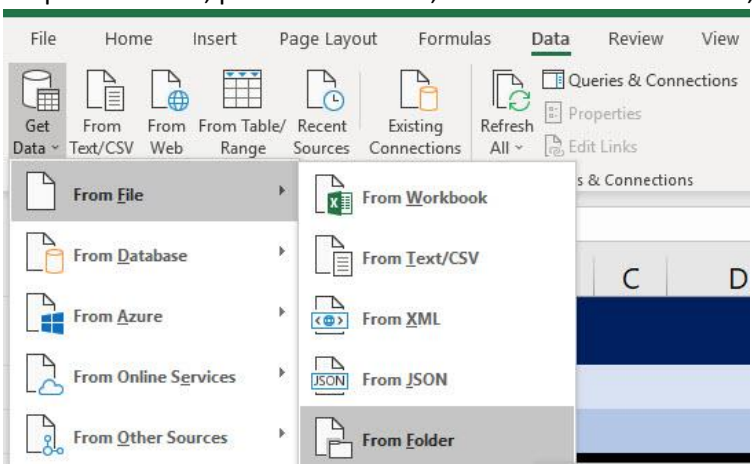
iii. **Cardinality**. Number of unique items in a column = “Cardinality”.
   1. It is the Cardinality that is the primary factor in determining the file size

iv. **Some Factors in Determining File Size**:
   1. Number of Columns.
   2. Cardinality of columns. The more unique items there are the more space the column will take up in file size.
      i. If you have numbers with many decimals, and you don’t need them, then round the numbers before you import them. This will result in a column with far fewer unique values and therefore the column will be compressed more and result in a smaller file size.
   3. If there is a pattern of consecutive repeated items in a column, file size may be reduced.
   4. Other techniques also…

6. **Example in this video: Power Query to Import Many Text Files into the Power Pivot Data Model Columnar Database, Create Data Model PivotTable with Implicit Measures:**

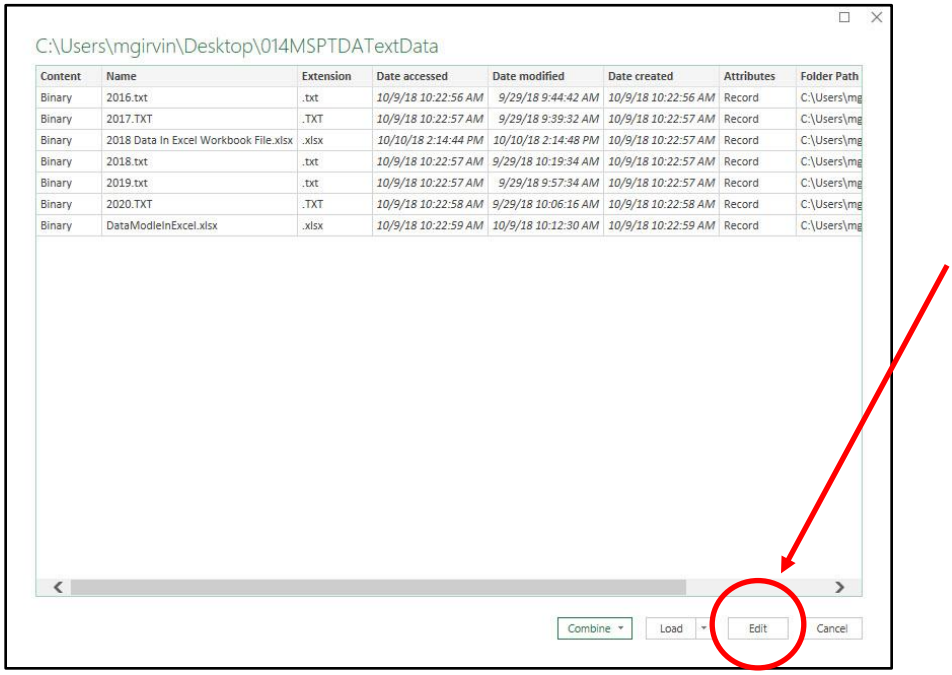
   Here are the Steps for the Project in this video:

   1) Using Power Query, go to the Data Ribbon Tab, Get & Transform Data group, Click the Get Data dropdown arrow, point to From File, then click on From Folder, as seen here:

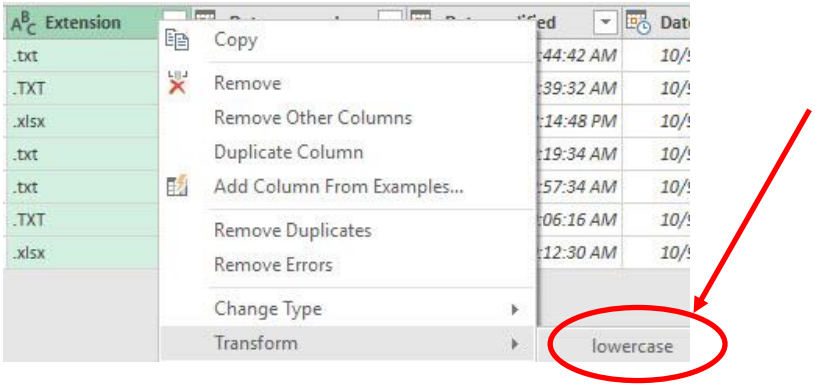


   2) Enter your own Folder Path to the downloaded folder.

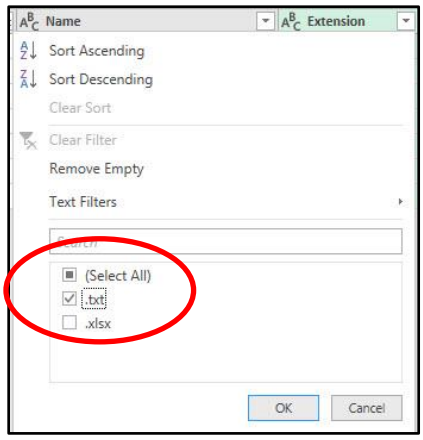
   3) In the next dialog box, click Edit, as seen here:



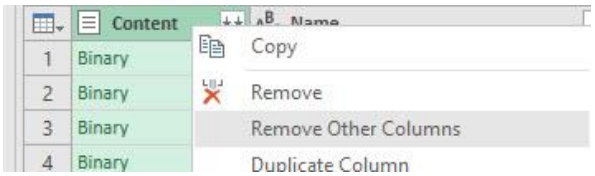
4) When the Power Query editor opens, name the Query “AllTextFilesInOneTable”.
5) Right-click the Extension column, point to Transform, then click on “lowercase”, as seen here:



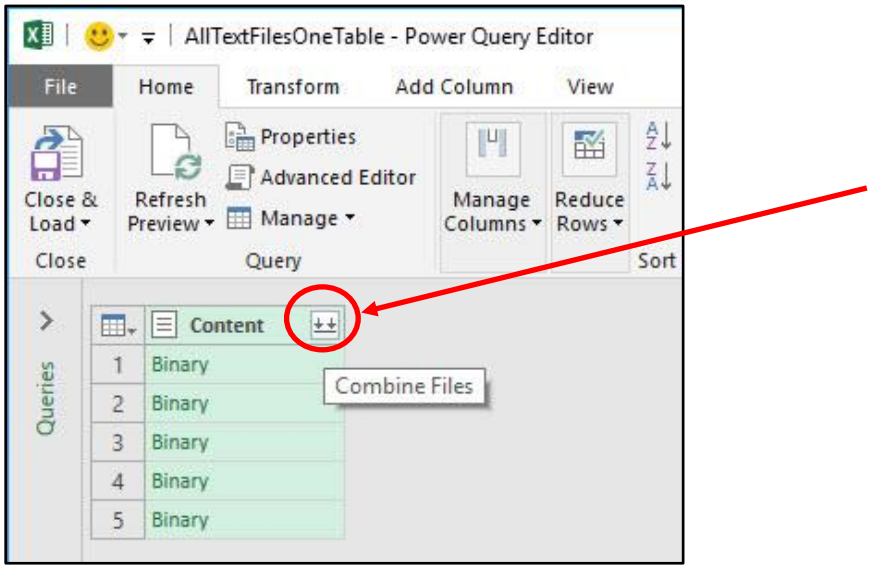
6) Using the Filter arrow in the Extensions column, filter to allow only “.txt” files to be imported, as seen here:



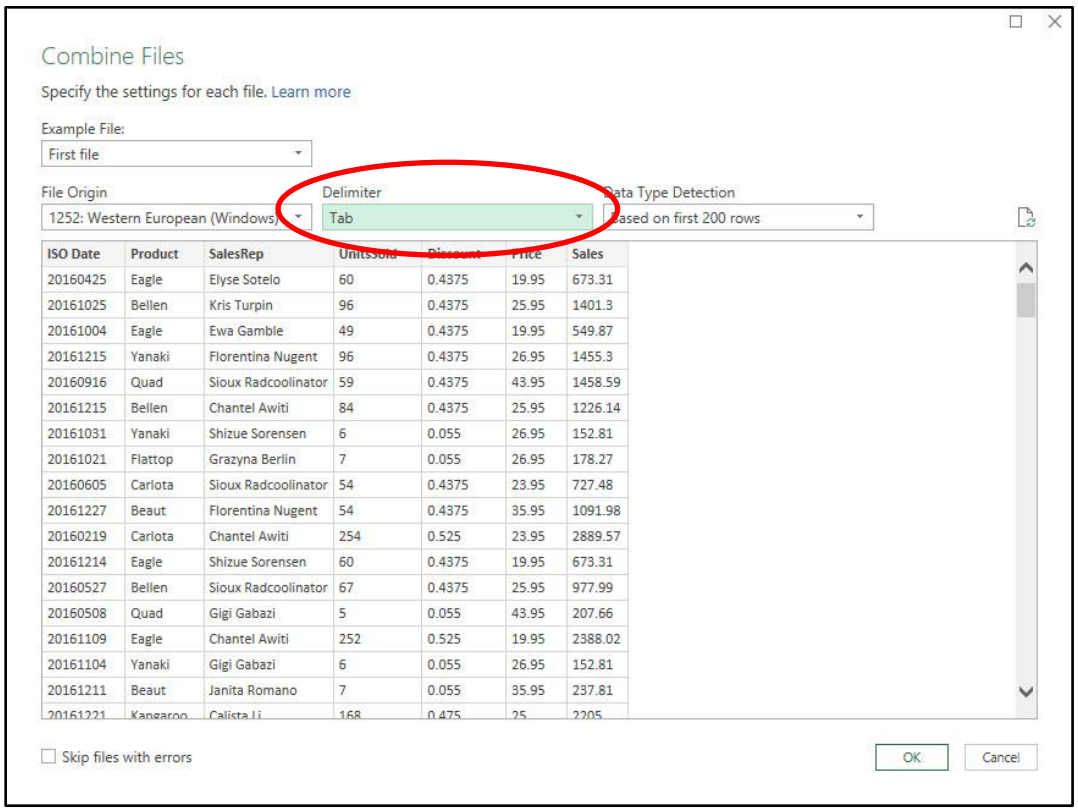
7) Right-click the Content column and click on “Remove Other Columns, as seen here:



8) To append all the text files into one table, click the “Combine Files” button, as seen here:

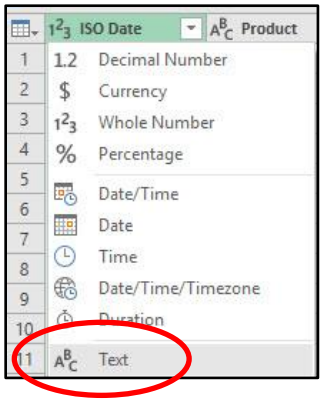


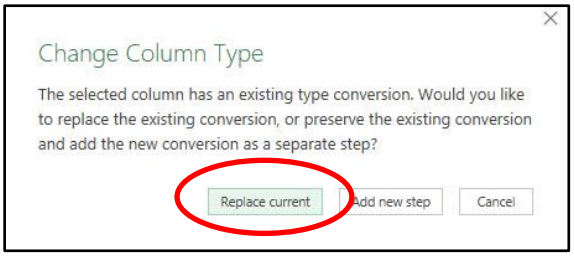
9) In Combine Files dialog box, select “Tab” as the Delimiter, then click OK.



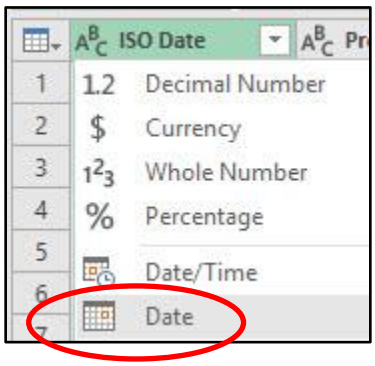
ISO Date	Product	SalesRep	UnitsSold	Discount	Price	Sales
20160425	Eagle	Elyse Sotelo	60	0.4375	19.95	673.31
20161025	Bellen	Kris Turpin	96	0.4375	25.95	1401.3
20161004	Eagle	Ewa Gamble	49	0.4375	19.95	549.87
20161215	Yanaki	Florentina Nugent	96	0.4375	26.95	1455.3
20160916	Quad	Sioux Radcoolinator	59	0.4375	43.95	1458.59
20161215	Bellen	Chantel Awiti	84	0.4375	25.95	1226.14
20161031	Yanaki	Shizue Sorensen	6	0.055	26.95	152.81
20161021	Flattop	Grazyna Berlin	7	0.055	26.95	178.27
20160605	Carlota	Sioux Radcoolinator	54	0.4375	23.95	727.48
20161227	Beaut	Florentina Nugent	54	0.4375	35.95	1091.98
20160219	Carlota	Chantel Awiti	254	0.525	23.95	2889.57
20161214	Eagle	Shizue Sorensen	60	0.4375	19.95	673.31
20160527	Bellen	Sioux Radcoolinator	67	0.4375	25.95	977.99
20160508	Quad	Gigi Gabazi	5	0.055	43.95	207.66
20161109	Eagle	Chantel Awiti	252	0.525	19.95	2388.02
20161104	Yanaki	Gigi Gabazi	6	0.055	26.95	152.81
20161211	Beaut	Janita Romano	7	0.055	35.95	237.81
20161221	Kangaroo	Calista Li	168	0.475	25	2205

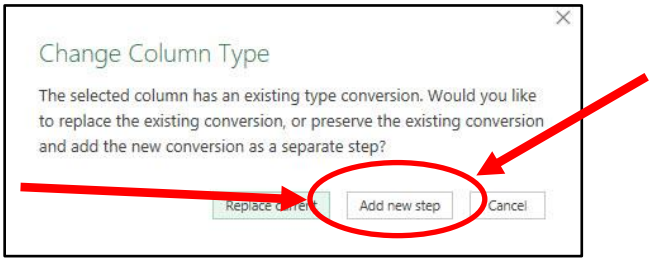
10) To convert the number ISO Dates to Real Dates, first, click Data Type Icon at top of Date column and select the Text Data Type, and then click the “Replace current” button in the next dialog box, as seen here: as seen here:



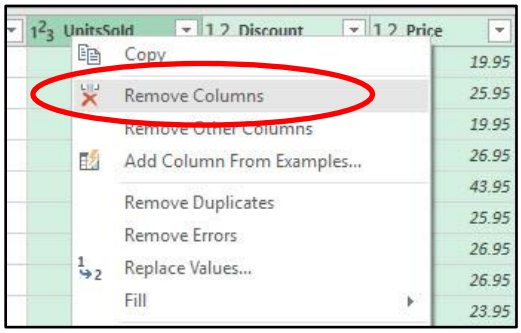


11) In the second step to convert ISO Dates, click Data Type Icon at top of Date column and select the Date Data Type, and then click the “Add new step” button in the next dialog box, as seen here: as seen here:

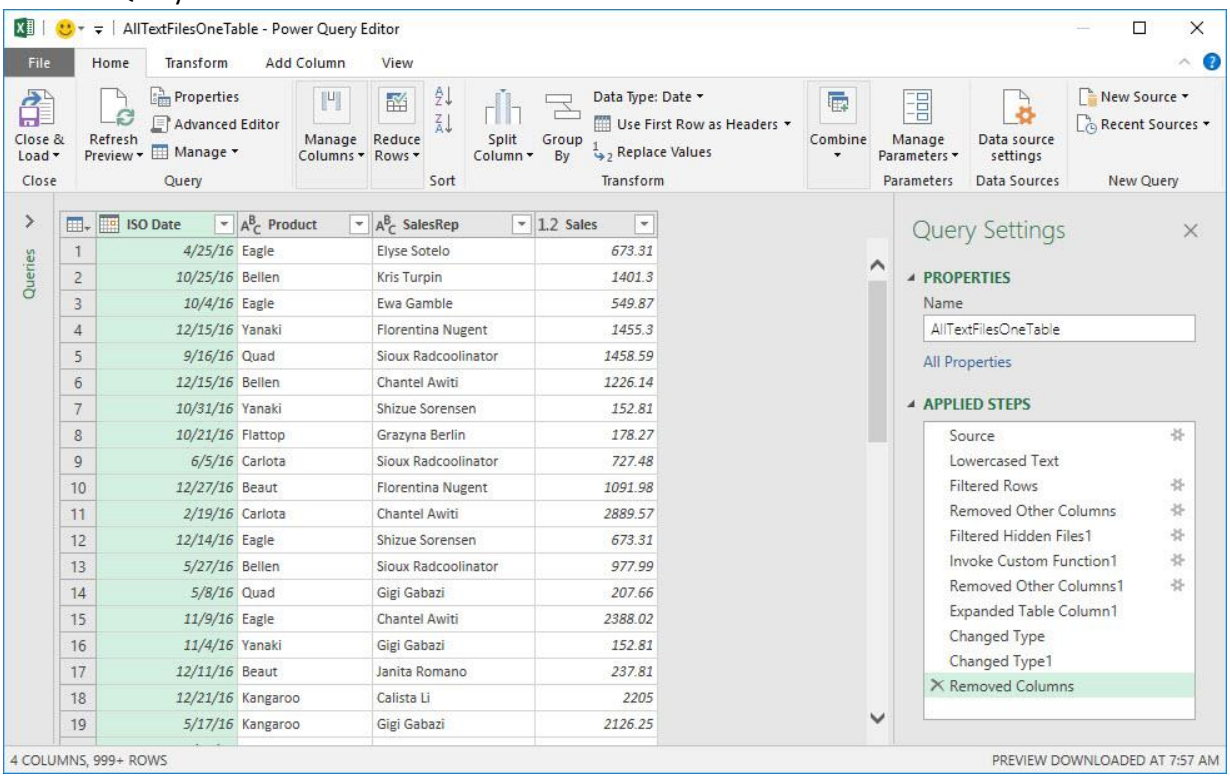




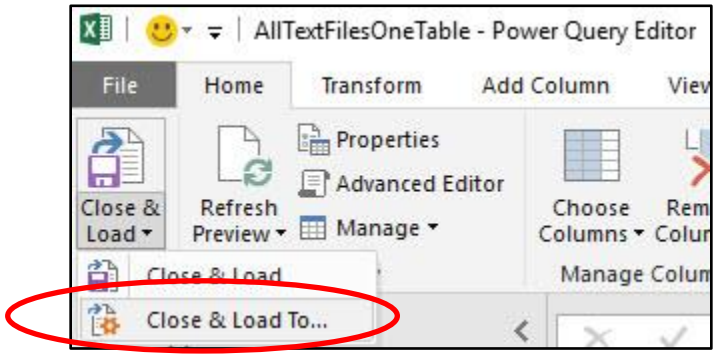
12) Selecting the UnitsSold, Discount and Price Columns, right-click any one of the three columns and click on “Remove” Columns”, as seen here:



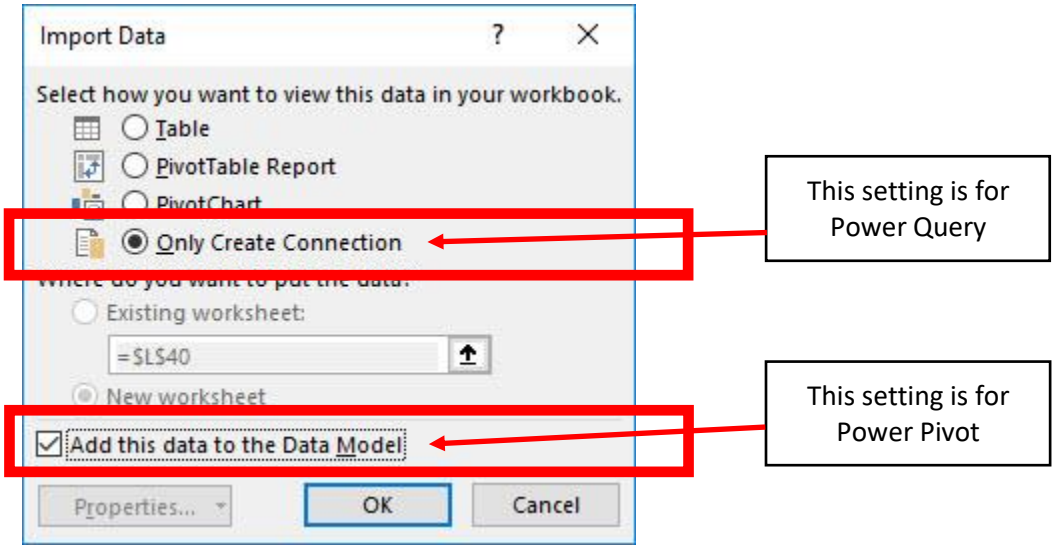
13) The final Query should look like this:



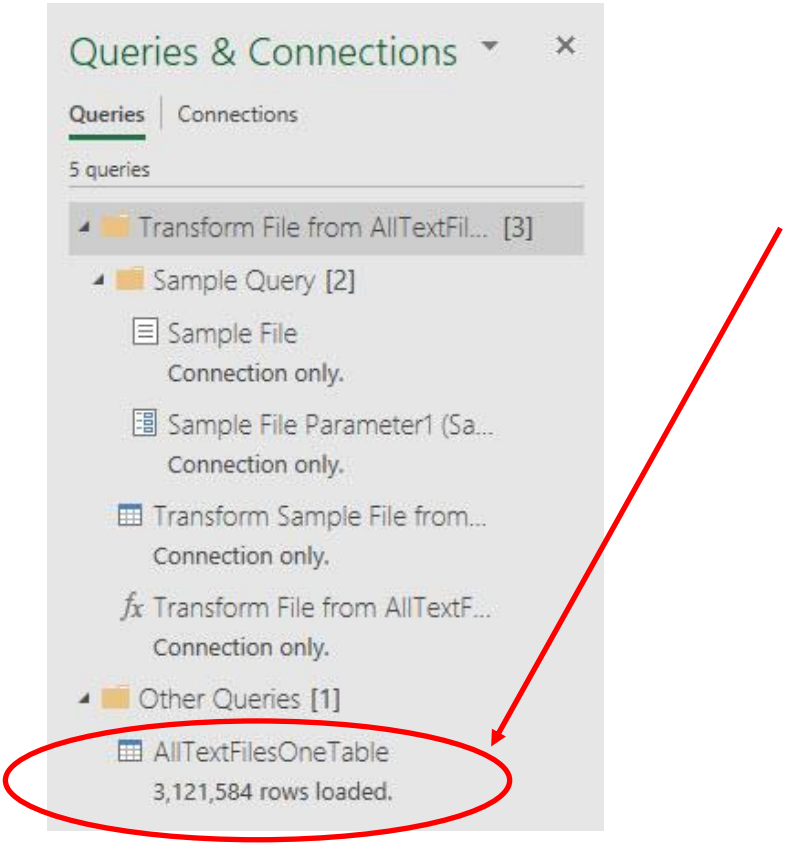
14) In the Home Ribbon Tab, Close group, click the dropdown arrow for Close & Load, then click on Close & Load To, as seen here:



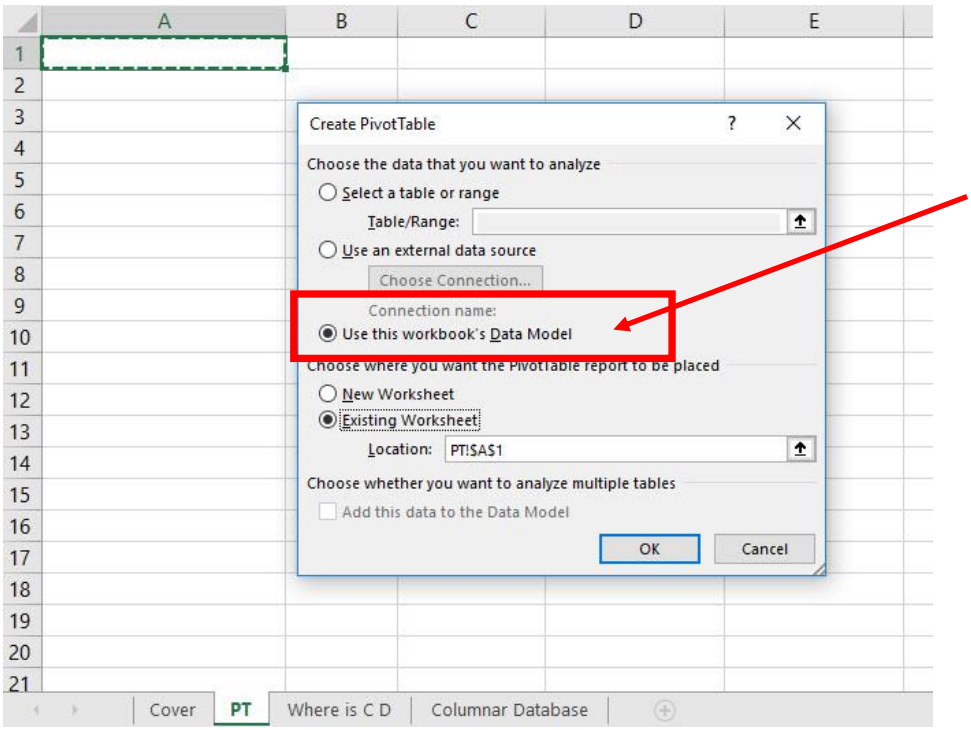
15) In the Import Data dialog box, select:
    i. "Only Create Connection" (this setting is for Power Query)
    and
    ii. "Add this data to the Data Model" (this setting is for Power Pivot), as seen here:



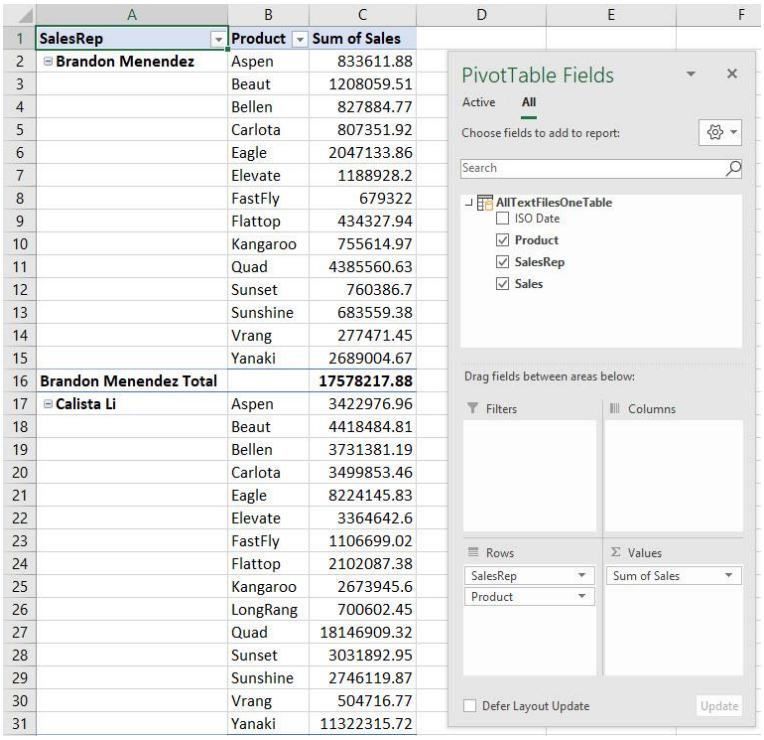
16) The Queries & Connections Pane shows the queries created by Power Query. The table loaded to the Power Pivot Data Model Columnar Database shows that 3,121,584 rows of data have been loaded.



17) Because we have a table in the Data Model, when we open the Create PivotTable dialog box, “Use this workbook’s Data Model” is selected, and we can make our Data Model PivotTable Report from the Data Model.



18) Because we are making a simple PivotTable with just a few criteria in the Row Area and a simple SUM calculation in the Values Area, we can use Implicit Measures (calculation for Sum of Sales).



SalesRep	Product	Sum of Sales
Brandon Menendez	Aspen	833611.88
	Beaut	1208059.51
	Bellen	827884.77
	Carlota	807351.92
	Eagle	2047133.86
	Elevate	1188928.2
	FastFly	679322
	Flattop	434327.94
	Kangaroo	755614.97
	Quad	4385560.63
	Sunset	760386.7
	Sunshine	683559.38
	Vrang	277471.45
	Yanaki	2689004.67
Brandon Menendez Total		17578217.88
Calista Li	Aspen	3422976.96
	Beaut	4418484.81
	Bellen	3731381.19
	Carlota	3499853.46
	Eagle	8224145.83
	Elevate	3364642.6
	FastFly	1106699.02
	Flattop	2102087.38
	Kangaroo	2673945.6
	LongRang	700602.45
	Quad	18146909.32
	Sunset	3031892.95
	Sunshine	2746119.87
	Vrang	504716.77
	Yanaki	11322315.72

7. **Implicit Measures**. As seen the above picture, and as demonstrated in the video, when we drag a Field from a Data Model Table into the Values area of the Data Model PivotTable, the calculation that is made is called an "Implicit Measure".
    1) Implicit Measures are hidden DAX Formulas that are automatically made when you drag a Field from a Data Model Table into the Values area of the Data Model PivotTable.
    2) There are multiple disadvantages to use Implicit Measure sin a Data Model PivotTable. We will discuss these disadvantages in full detail in MSPTDA Video #15 - Power Pivot Video Intro Video #3. However, Implicit Measures are perfectly okay to use when you are making a simple PivotTable report, like adding to get totals. In this case, the advantages of being able to quickly create the PivotTable outweigh the disadvantages. Much more in video MSPTDA Video #15 - Power Pivot Video Intro Video #3.