Highline Excel 2016 Class 22: How To Build Data Model & DAX Formulas in Power Pivot

Table of Contents

Which Versions of Excel Contain PowerPivot?

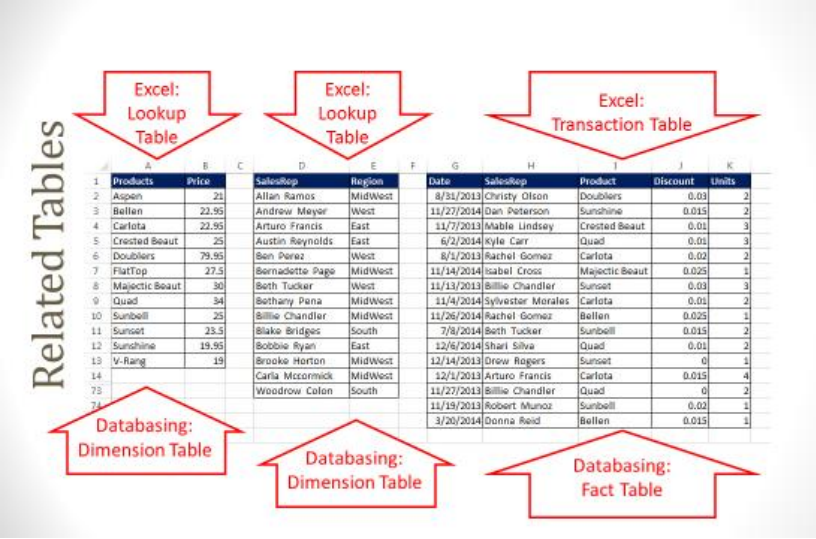
1) Versions of Excel 2013 contain PowerPivot:
    - Office 2013 Professional Plus
    - Stand Alone Excel
    - Office 365 (E3 or E4 editions)
2) Versions of Excel 2016 contain PowerPivot:
    - Office 2016 Professional
    - Stand Alone Excel
    - Office 365 Professional Plus editions

Power Pivot is a COM add-in that you must enable

1) File, Options, Add-ins, COM add-in, check box for Power Pivot.

Reminder about Terminology for Tables in a Data Model

Examples from data set not seen in this video:



What is Data Modeling?

1) Import Data into Power Pivot Data Model as Proper Data Sets (Tables):
    - Using Power Query to Clean, Transform and Import data.
    - "Add to Data Model" button in the Power Pivot Ribbon Tab if data is small & is in an Excel Sheet.
2) Create Relationships between Dimension Tables & Fact Tables.
3) Create DAX formulas:
    1. DAX Measures to use in Values area of PivotTable.
       and/or
    2. Calculated Columns to use as criteria for Row/Column/Filter/Slicer area of PivotTable or for use in DAX Measure.
4) Hide Tables and Fields that are not used in PivotTables.
5) Create PivotTables & Pivot Charts based on Data Model.
6) Refresh Data Model when source data changes.
7) Edit Data Model as necessary.

Power Pivot Data Model's Columnar Database

1) Power Pivot's Data Model does not store imported tables in in an Excel sheet or in a table format.
2) Power Pivot's Data Model has a behind the scenes Columnar Database where all data is stored.
3) When you import a table into the Data Model, each field in the imported table is stored separately with a unique list of values for the field. There is a sort of "map" that allows the database to reconstruct the original table and all of the records.
4) The Columnar Database is a behind the scenes In-Memory (RAM) Database.
    - RAM = Random Access Memory.
    - The number of unique values in any one field determines the amount of RAM that is used.
    - The Columnar Database allows you to import large data sets (millions of rows) that would not fit in an Excel sheet. You can safely handle 100 million rows.
5) The Columnar Database stores data efficiently and can dramatically reduce file size.
6) The Columnar Database is designed to work with DAX Formulas to calculate quickly on Big Data.
7) Example of Columnar Database, where each field is stored in a separate column with a unique list of values only:

Proper Data Set

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

Import into Data Model ==>>

Columnar Database:

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

Synonyms for Columnar Database:

- Columnar database
- Data Model
- PowerPivot Database stored in an Excel workbook
- PowerPivot xVelocity engine
- PowerPivot engine
- XVelocity analytics engine
- VertiPaq

Power Pivot Data Model's DAX Formulas

1) DAX = Data Analysis Expressions = formulas you can build in Data Model.
2) DAX formulas are specifically designed to work with Columnar Database and Relationships to calculate efficiently on Big Data.
3) There are many more DAX functions than in a normal PivotTable. We have new functions like RELATED, SUMX, SAMEPERIODLASTYEAR and CALCULATE.
4) When you create DAX Formulas they appear in PivotTable Field List and can be dragged and dropped into PivotTable.
5) Convention for creating DAX Formulas:
    - When you refer to a Field in a Table use the Table Name & the Field Name enclosed in square brackets (same as Excel Table Formula Nomenclature).
    - When referring to a Measure use the Measure Name enclosed in square brackets.
6) Two Types of DAX Formulas:
    1. Measures
    2. Calculated Columns
7) When you are creating your DAX formula next to the table (Calculated Column) or below the tables (Measures), the DAX formulas must be typed in the Formula Bar.

DAX Calculated Columns

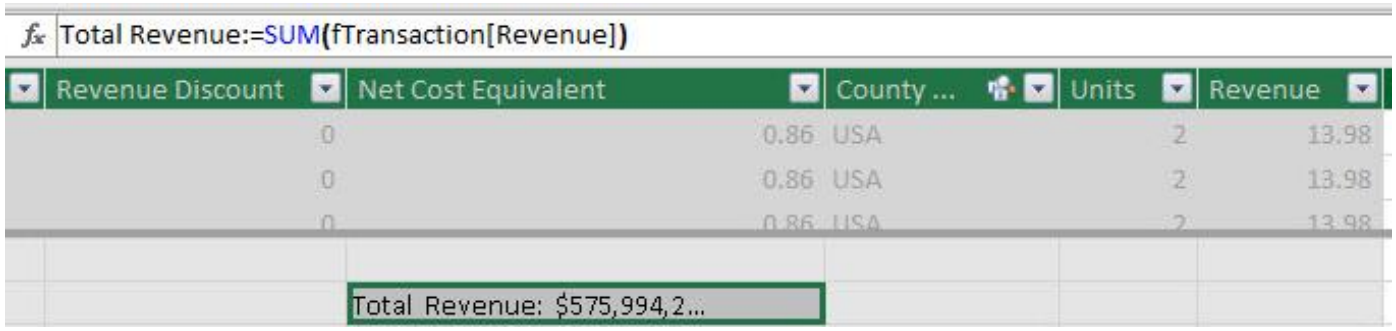
1) "Helper Columns" that are added to the Tables in the Data Model.
2) Calculated Columns can extend the content of the table such as:
    - Examples of new fields that extend the content: Month Name or Fiscal Quarter.
    - When you have a Calculated Column that extended the table's content, the Calculated Column will appear in the PivotTable Field List and you can drag and drop into the Row / Column / Filter / Slicer area of a PivotTable.
3) Calculated Columns can be used to calculate numbers such as Revenue, which in turn is used in a DAX Measure.
    - This is especially helpful if you have more than 1.04 million rows of records, which cannot fit into an Excel Sheet. By using the Data Model and a Calculated Column, we can easily create a helper column to lookup a price and calculate revenue for each record.
4) DAX Calculated Column formulas:
    - Must be create in the Formula Bar above the table.
    - Look similar to Excel Table Formula Nomenclature formulas in that they use the Table Name & the Field Name enclosed in square brackets, called field reference or column reference.
    - There are no "Cell References" in either Calculated Column or Excel Table Formula Nomenclature.
    - **When Calculated Columns are calculated/evaluated:**
        1. Calculated Columns are calculated/evaluated when the column is created or the Data Model is refreshed.
    - When you create a Calculated Column, the values are stored in the Column Database in RAM. The more unique values there are, the more RAM used.
    - DAX Calculated Columns calculate row-by-row in a Data Model Table using "Row Context" to calculate the answer for each record in the table.
5) **Row Context:**
    - Row Context simply means that field reference (column reference) calculates a different answer for each row based on the data in the row that the formula sits in. For example: for the field reference, "fTransactions[Unit]", the formula knows to get the units for each particular row.

DAX Measures

1) Measures are formulas created to use in:
    - The Values area of the Data Model PivotTable.
    - Other Measures.
    - Sometimes they are used in Calculated Columns.
2) You create or edit Measures in either:
    - Measured Grid below Data Model Table.
    - Measure dialog box: Power Pivot Ribbon Tab, Calculation group, Measure drop-down arrow, New Measure.
3) DAX Measure formulas:
    - Whenever you refer to a field in a Table, called either a column reference or field reference, you use the Table Name & the Field Name enclosed in square brackets, like: fTransactions[Unit].
    - Whenever you refer to another Measure, use the Measure Name enclosed in square brackets.
    - Add Number Formatting so that whenever you drag your Measure into a PivotTable the Number Formatting will appear.
    - In a PivotTable Field List and in Diagram View, Measures appear with a function icon.
    - **When Measures are calculated/evaluated:**
        1. Measures are calculated/evaluated when the formula is dragged into the Values area of a PivotTable or when the criteria is changed or the PivotTable is Refreshed.
    - Unlike Calculated Columns, Measures do not store any internal values in RAM. The values are generated when the Measure is dragged into the Values area of a PivotTable or when the criteria is changed or the PivotTable is Refreshed.
    - Measures make an aggregate calculation based on the criteria from the PivotTable and/or from inside the formula and calculates an answer for each cell in the PivotTable. The criteria from the PivotTable and/or from inside the formula is called the “Filter Context”.
4) **Filter Context:**
    - Filter Context simply means that a Measure can “see” the criteria from the Row/Column/Filter/Slicer area of a PivotTable or from within the formula. The criteria cause the underlying Columnar Database to become “filtered” down to only the records that match the criteria before the final answer is calculated.
5) Advantages of DAX Measures over Standard PivotTable calculations and/or Excel Spreadsheet formulas:
    - DAX Measures calculate quickly over millions of rows of data.
    - You can create the formula one time and can use it in as many Data Model PivotTables as you want.
    - You add Number Formatting to the formula and it follows the formula around.
    - There are many new DAX functions like SAMEPERIODLASTYEAR which we don’t have in a Standard PivotTable or in an Excel Spreadsheet.
    - DAX formulas are easy to edit in one location. When editing is done, all locations where the formula is used are updated.
    - DAX Measures, Relationship and the Columnar Database work together to make calculations in the PivotTable quickly.
6) Measures are referred to as “explicit” calculations.
7) **NOTE:** DAX Measures terminology:
    - In Excel 2010 & 2016 Microsoft uses the term “Measure” to refer to DAX formulas that you can use in the Values area of the PivotTable.
    - In Excel 2013 Microsoft uses the term “Calculated Field” to refer to DAX formulas that you can use in the Values area of the PivotTable.

Creating Measure in Measure Grid

1) Choose the table in the Data Model whose Field List you want the Measure to appear in.
2) Click in cell below table.
3) Type Measure Name followed by the “assignment operator” := (Colon, Equal Sign).
4) Your cursor will automatically jump up to the Formula Bar.
5) Create formula.
6) Add Number Formatting from the Formatting group in the Manage Data Model Home Ribbon Tab.
7) Example (details later in this project):



Implicit vs. Explicit calculations in a PivotTable

1) Implicit DAX Measures
    - Drag and drop field into Data Model PivotTable/Power BI Values Area and the Data Model makes a Read Only Measure for you
    - MUCH Less Control when you use Implicit
        1. Number Field will use SUM Function
        2. Text Field uses COUNT
        3. You can't add Number Formatting to the Measure
        4. You can't Reuse Formula
        5. You can't change the name of the Measure
        6. If you right-click an Implicit Measure in the Values Area of the PivotTable you can change the "Summarize Values By", but then it adds yet another Implicit Measure
2) Explicit DAX Measure
    - You create the formula, choose the name, add the Number Formatting and use it over and over.
    - You create formula in Measured Grid, or Measure dialog box
    - MUCH More Control when you use Explicit
        1. You choose what functions go into your Measure
        2. You can name the Measure
        3. You can apply Number Formatting that will follow the Measure around when it is reused.
        4. You can use the formula over and over.
        5. You won't have a bunch of extra Measures created by right-click, "Summarize Values By".

DAX Functions seen in this video:

1) MONTH: Calculates Month Number from Date.
2) FORMAT: Formats a values with a Custom Number Format and converts to text.
3) YEAR: Calculates Year Number from Date.
4) ROUNDUP: Rounds up to a certain digit.
5) IF: delivers on of two items of the same Data Type based on a logical test.
6) ROUND: Standard Rounding rule.
7) RELATED: Looks up an item in a row and through a relationship delivers a related value (like Exact Match VLOOKUP).
8) SUM: adds numbers.
9) SUMX: iterates a DAX formula over a table, row-by-row (Row Context), & then adds the resultant values.
10) DIVIDE: Can divided two numbers and deliver a DAX BLANK if an error occurs.
11) CALCULATE: Changes the Filter Context for a Measure based on criteria in Filter argument.
12) SAMEPERIODLASTYEAR: Retrieves an amount for same period last year based on the criteria in a Pivot.
13) BLANK: Delivers an empty cell that is not considered text or number and won't interfere with data type.

DAX Calculated Column or DAX Measure to calculate Total Revenue?

1) **DAX Calculated Column for calculating revenue for each record in the Fact Table (we see how to create this later in the project). Example demonstrated later in the prject:**

```
=ROUND(RELATED(dProducts[Retail Price])*(1-fTransactions[Revenue Discount])*fTransactions[Units],2)
```

- DAX Calculated Column for Revenue stores the column's unique values in the Columnar Database:
    1. If there are a few unique values, not much RAM space used
    2. If there are many unique values, more RAM space used.
- DAX Calculated Columns actually calculate an answer for each record in the column when the Calculated Column is created or when the Data Model is Refreshed.

2) **DAX Measure for Total Revenue (we see how to create this later in the project). Example demonstrated later in the prject:**

```
=SUMX(fTransactions,ROUND(RELATED(dProducts[Retail Price])*(1-fTransactions[Revenue Discount])*fTransactions[Units],2))
```

- DAX Measure does NOT store the values in RAM
- DAX Measure gets calculated only when you drop it into PivotTable OR if you change the criteria in the Row / Column / Filter / Slicer area. It is calculated by CPU – Central Processing Unit.

3) Which one to use?
    - It depends in part on how many unique values there are.
    - If Data Model is working slow, you may need to test which one works more quickly.

Criteria in a Data Model PivotTables

1) If you have a choice between a field that is in both a Dimension Table and Fact Table, Drag criterion from the Dimension Tables to the Row/Column/Filter/Slicer area of the PivotTable.
2) Using Criteria from Dimension Tables rather than Fact Tables helps the DAX Formulas to calculate more quickly.

Calendar Table (Dimension Table)

1) Why Calendar Table and not "Group by Date" feature?
    - By using a Calendar Table, we gain these advantages:
        1. With a Calendar Table we can use "Time Intelligence" DAX Functions like SAMEPERIODLASTYEAR. SAMEPERIODLASTYEAR and other Time Intelligence DAX Functions require a Calendar Table and do not work with the grouping feature.
        2. We can create date categories such as Fiscal Quarter that cannot be created with the Grouping feature in a PivotTable.
        3. When we use a Calendar Table (Dimension Table) with a One-to-Many-Relationship with the Fact Table rather than the Calculated Columns that are added to the Fact Table with the Grouping feature, DAX Formulas can calculate more quickly.
2) Requirements for a Calendar Table:
    - The first field in a Calendar Table has to have a unique list of all the dates from earliest to latest with no missing dates (even if sales were not made on a particular date).
    - Calculated Columns are added to the Calendar Table in order to create other fields that provide date items like: Month Name, Fiscal Quarter, Fiscal Year.

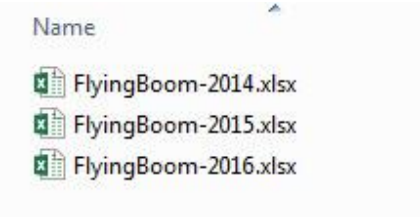
Advantage of Power Pivot Data Model Columnar Database & Relationships & DAX Measures when you have Big Data.

How Data Model can calculate quickly on big data:

1) When a criterion from a Dimension Table is added to the PivotTable the underlying Dimension Table is filter so that the record with the criterion is removed. One record is filtered out. This makes sense because a Dimension Table is the "One Side" in the One-to-Many Relationship.
2) In turn the filter from the Dimension Table is passed along to the Fact Table and the underlying Fact Table is filtered so that all the records with the criterion are removed. Many records are filtered out, which makes the Fact Table smaller. This makes sense because a Fact Table is the "Many Side" in the One-to-Many Relationship.
3) After all the criteria in the PivotTable pass along the "filters" to the Fact Table, the Fact Table is filtered to a smaller size.
4) The DAX Formulas can work more quickly over a smaller Fact Table.

Data Modeling Step 1: Power Query to Clean, Transform & Import Fact Tables

1) Excel files with data from 2014-2016 that sit in the folder named "Start". Each file has over 800,000 records. Each file has a single sheet with a proper data set of transactions for the year. We will import these using Power Query and create a single Fact Table in the Data Model.



2) Example of Transitional (Fact) Table for 2014:

	A	B	C	D	E	F	G
1	Date	Product	Revenue Discount	Net Cost Equivalent	County Code	Units	
2	8/18/2014	Aspen	0	1.01	UKR	2	
3	7/21/2014	Aspen	0	0.98	ZMB	1	
4	12/20/2014	Yanaki	0	1.1	DZA	3	
5	12/2/2014	Bower Aussie Round	0	0.97	USA	2	
6	12/12/2014	Fire Aspen	0	1.1	AUS	36	
7	5/13/2014	Carlota	0	0.98	DEU	1	
8	3/14/2014	Alpine	0	0.97	CHE	3	
9	1/6/2014	Majestic Beaut	0	1.05	FJI	3	

3) Example of Dimension Table for Country Name. This a Proper Data Set stored in an Excel Table with the name dCountry. The first field contains a unique list of Country Codes and the second field has country names.

	A	B
1	CountyCode	Country
2	AFG	Afghanistan
3	ALB	Albania
4	APR	Aprine
126	ZMB	Zambia
127	ZWE	Zimbabwe
128		
129		

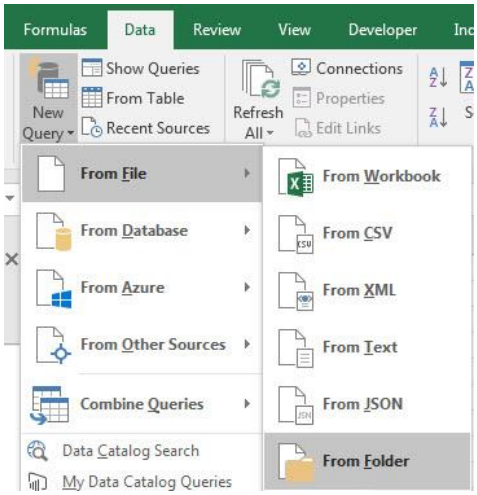
Topics | dCountry | dProduct | Sheet1

4) Example of Dimension Table for Products. This a Proper Data Set stored in an Excel Table with the name dProduct. The first field is a unique list of Product names and remaining fields have data for Retail Price, Standard Cost and Category for each Product.

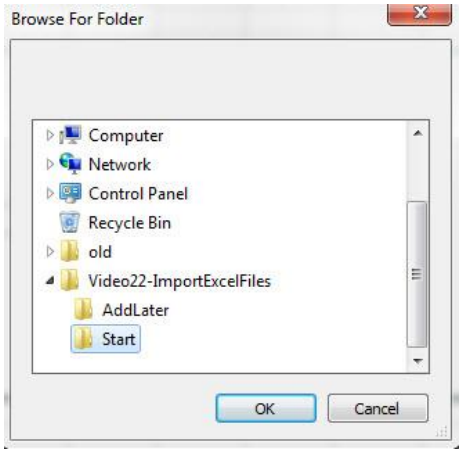
	A	B	C	D
1	Product	Retail Price	Standard Cost	Category
2	Alpine	23.95	10.09	Beginner
3	Aspen	23.95	9.48	Beginner
4	Bellen	26.99	10.75	Beginner
22	Sunspot	14.5	6	Novelty
23	Yanaki	25.99	10.02	Intermediate
24				
25				

Topics | dCountry | dProduct | Sheet1

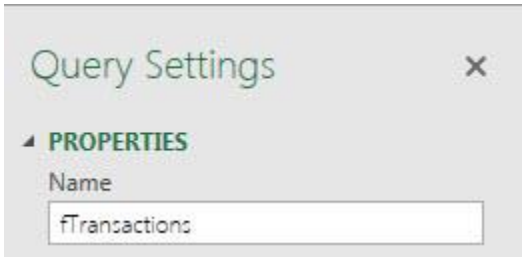
5) We need to import Excel workbooks from the different years and create a single table of transactions. Data Ribbon, Get & Transform group, New Query, From File, From Folder:



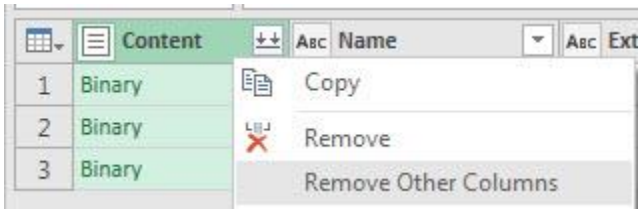
6) Browse to the Start Folder that is inside the Video22-ImportExcelFiles:



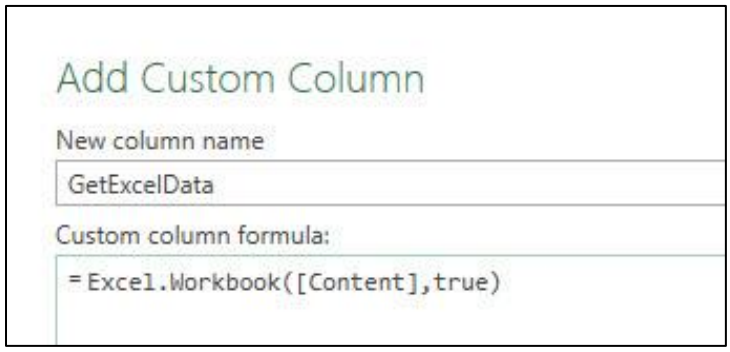
7) Once in the Power Query editor, name the query smartly. The name of the Query will also be the name of the Fact Table in the Data Model:



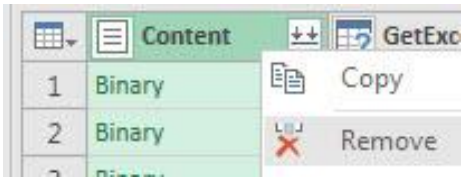
8) We will never have any files besides ".xlsx" files in our folder so we do not need to filter the Extension column. We don't need any of the other columns, so we right-click the Content column and click on Remove Other Columns.



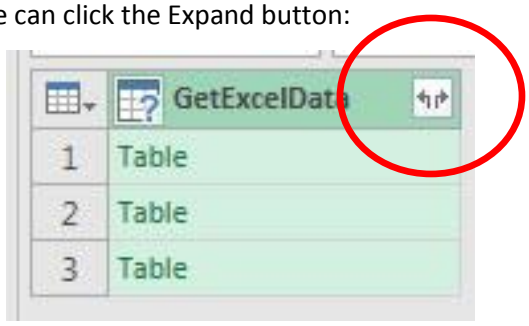
9) To extract the data from the single sheet in each Excel file and promote the first row that contains field names to actual Field Names in the Data Model Fact Table we Add a New Column and create the formula as seen below (review last video if you need more detail):



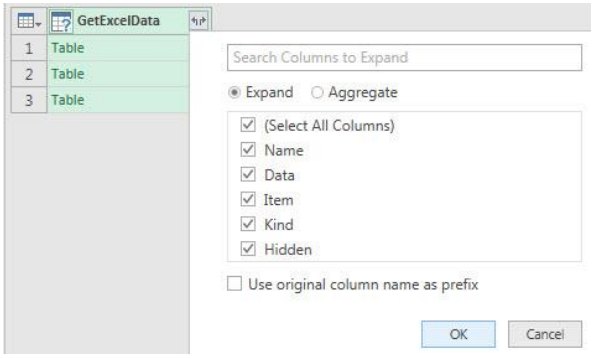
10) We no longer need the Content Column so we right-click and Remove:



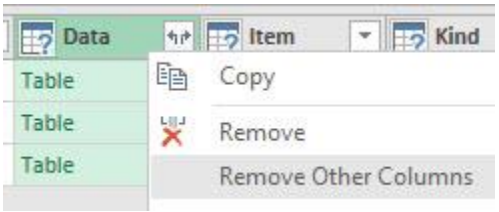
11) Now we can click the Expand button:



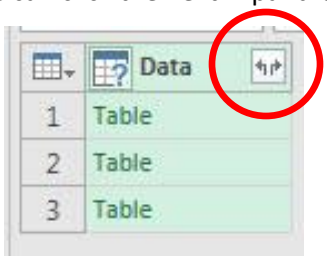
12) Uncheck “Use original column name as prefix”:



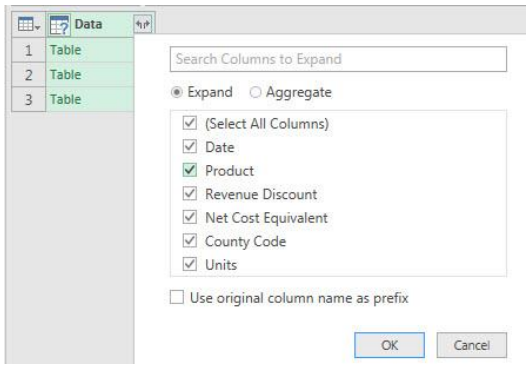
13) Unlike earlier videos in the class, with these Excel files we do not have any other objects besides the ***One Sheet in Each Workbook***, so we do not need to filter any columns, and we can simply right-click the Data column and point to Remove Other Columns.



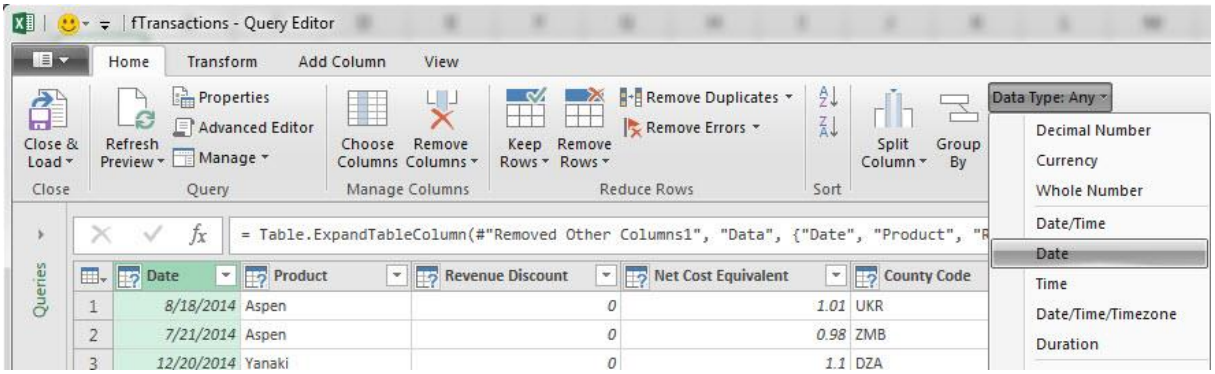
14) Now we can click the next Expand button:



15) Uncheck “Use original column name as prefix”:



16) For each column we need to change the Data Type to match the data. In this screen shot we are selecting the Date column and changing the Data Type to “Date”.



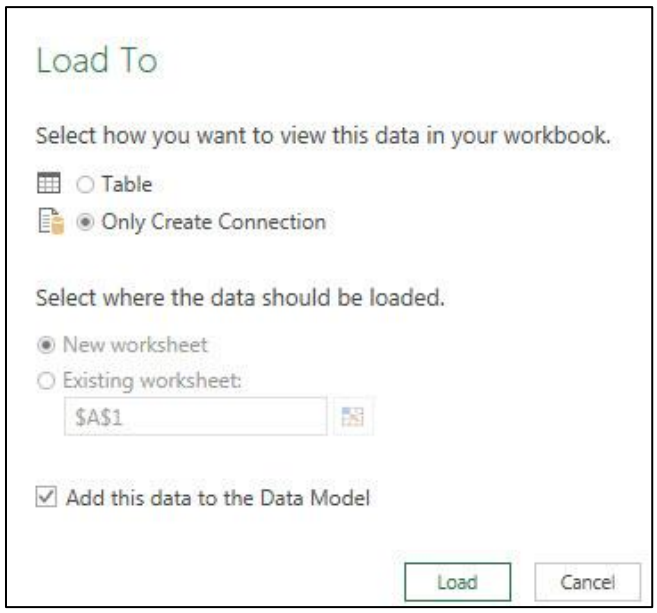
17) For each field we changed the Data Type:

- Date = Date
- Product = Text
- Revenue Discount = Decimal Number
- Net Cost Equivalent = Decimal Number
- Country Code = Text
- Units = Whole Number

	Date	Product	Revenue Discount	Net Cost Equivalent	County Code	Units
1	8/18/2014	Aspen	0	1.01	UKR	2
2	7/21/2014	Aspen	0	0.98	ZMB	1
3	12/20/2014	Yanaki	0	1.1	DZA	3
4	12/2/2014	Bower Aussie Round	0	0.97	USA	2
5	12/12/2014	Fire Aspen	0	1.1	AUS	36

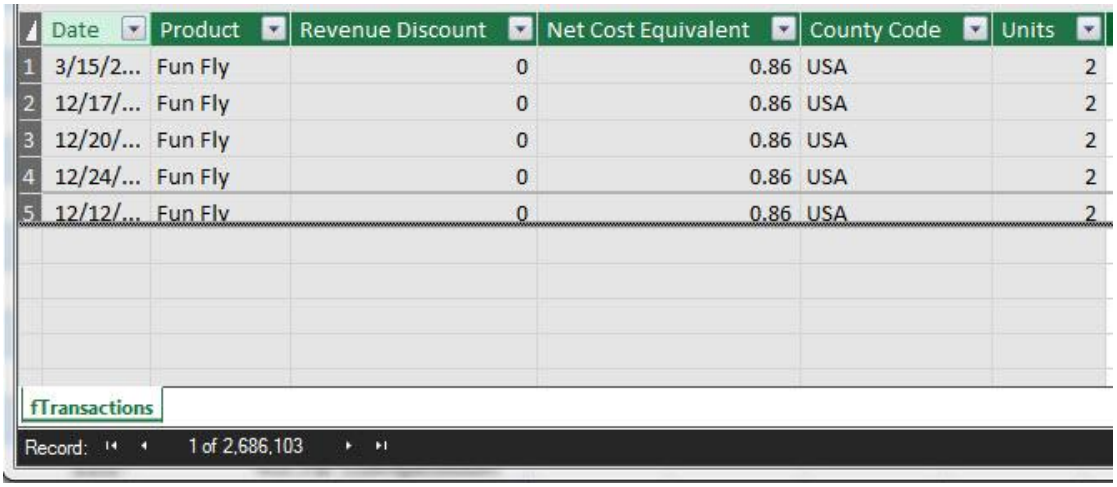
18) In the Home Ribbon Tab, click on Close and Load To:

19) In the Load dialog box, select “Only Create Connection” and “Add this to the Data Model”:



20) About 2.7 Millions Rows are Added to Data Model:

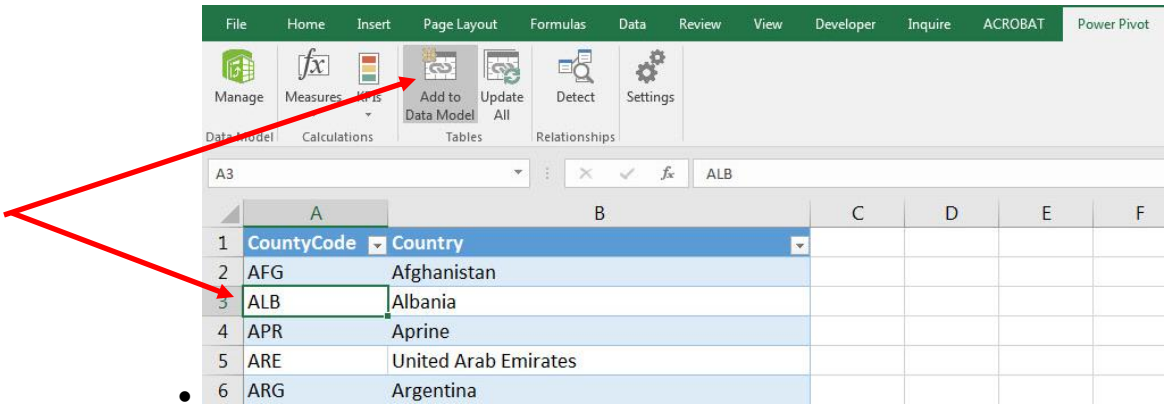
21) To see the Data Model, click the Manage Data Model button in the Power Pivot Ribbon Tab.



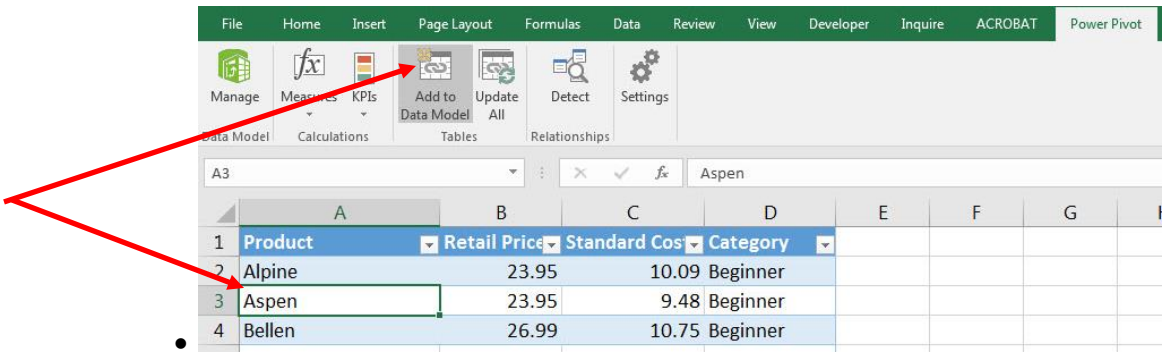
	Date	Product	Revenue Discount	Net Cost Equivalent	County Code	Units
1	3/15/2...	Fun Fly	0	0.86	USA	2
2	12/17/...	Fun Fly	0	0.86	USA	2
3	12/20/...	Fun Fly	0	0.86	USA	2
4	12/24/...	Fun Fly	0	0.86	USA	2
5	12/12/...	Fun Fly	0	0.86	USA	2

Data Modeling Step 1: Import Dimension Tables from an Excel Sheet

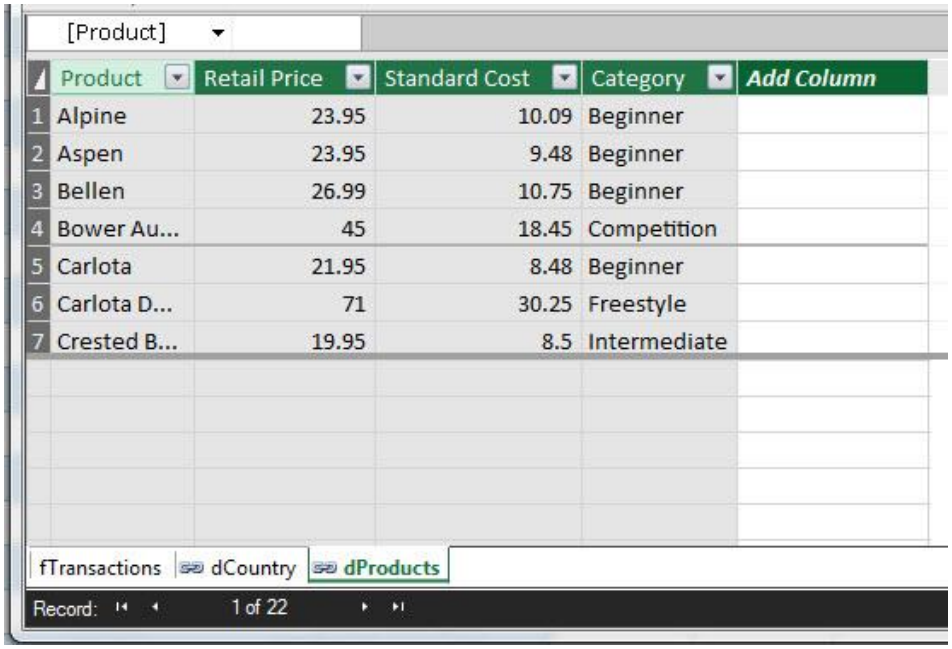
22) In the Excel Workbook file "Busn218-Video22Start.xlsm", on the sheet named "dCountry" click in a single cell in the dCountry Table, then click the "Add to Data Model" button in the Power Pivot Ribbon Tab.



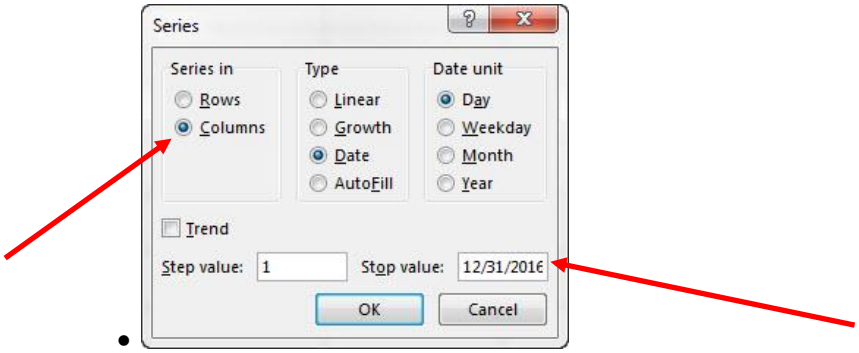
23) In the Excel Workbook file "Busn218-Video22Start.xlsm", on the sheet named "dProduct" click in a single cell in the dProduct Table, then click the "Add to Data Model" button in the Power Pivot Ribbon Tab.

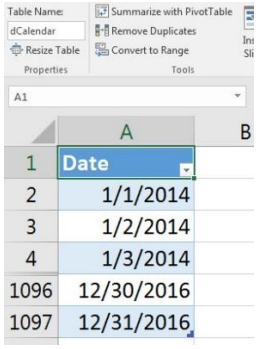


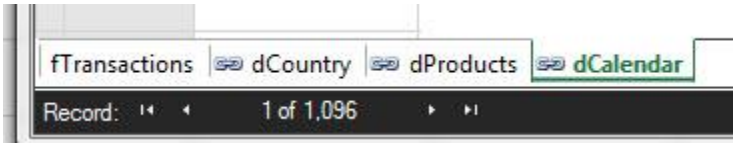
24) Data Model after importing the two Dimension Tables:



Data Modeling Step 1: Create Calendar Table in Excel & Import to Data Model

25) Why Calendar Table and not PivotTable Grouping feature?
    - We need to calculate Fiscal Quarter and use the SAMEPERIODLASTYEAR DAX function, neither of which works with the grouping feature.
26) Requirements for a Calendar Table:
    - The first field in a Calendar Table has to have a unique list of all the dates from earliest to latest.
    - Calculated Columns are added for: Month Name, Fiscal Quarter, Fiscal Year.
27) We must look at our source data and see what the earliest and latest dates are. For us, the earliest date in 1/1/2014 and the latest date is 12/31/2016.
28) In the Excel Workbook file "Busn218-Video22Start.xlsm" rename Sheet1 to "dCalendar".
29) In cell A1 Type Date and add Bold.
30) In cell A2 Type 1/1/2014
31) With cell A2 selected go to Home Ribbon Tab, Editing group, Fill drop-down arrow, and click on Series
32) In the Series dialog box complete as follows: 1) "Series in" should be "Columns", and 2) "Stop value" should be "12/31/2016":
    - 

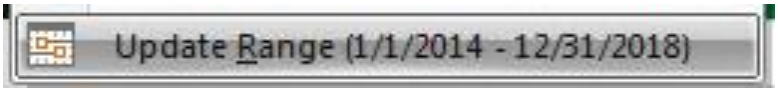
33) Widen Column A.
34) With a cell in the Date Field, convert the Proper Data Set to an Excel Table using the keyboard: Ctrl + T.
35) In the TablesTool Design Ribbon Tab, Properties group, click in the Table Name textbox and name the table "dCalendar".
36) Calendar Table in Excel:
    - 

37) With a cell in the Date Field, click the "Add to Data Model" button in the Power Pivot Ribbon Tab.
38) The Data Model now has four tables:
    - 

Steps to Create Automatic Calendar Table (Not Seen in Video)

1) Import your Fact Table into the Data Model.
    1. In the Data Model Window go to: Design Ribbon Tab, Calendar group, Date Table drop-down, Click New:

2) To update Automatic Calendar Table when new data is added to Fact Table, use the Dat Tbale drop down and point to Update Range:

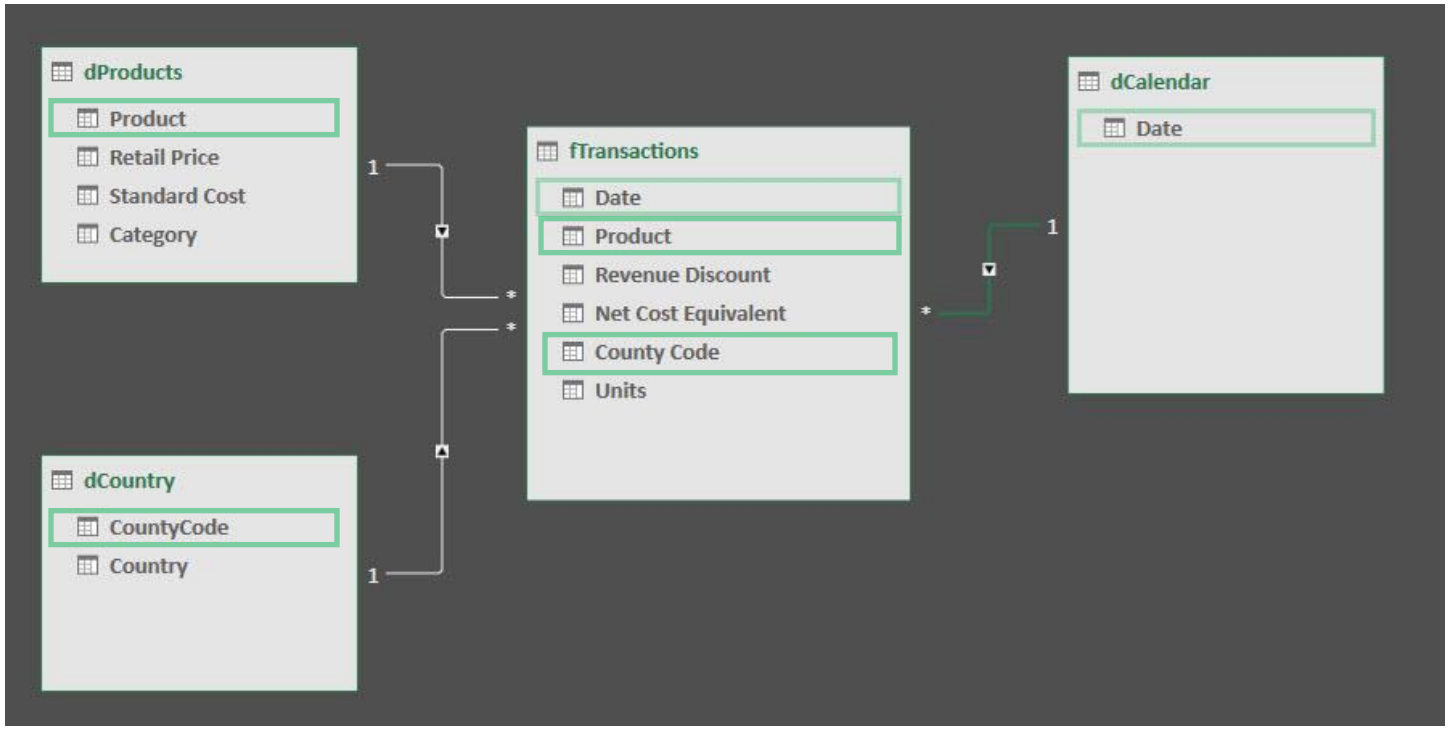


Data Modeling Step 2: Create Relationships between Related Tables

39) Click Diagram View button in View group: 

40) Drag and drop fields to create a One-To-Many Relationships between Dimension Tables and Fact Tables:

- Product field in dProduct ➔ Product in fTransaction
- CountryCode field in dCountry ➔ Country Code in fTransaction
- Date field in dCalendar ➔ Date in fTransaction

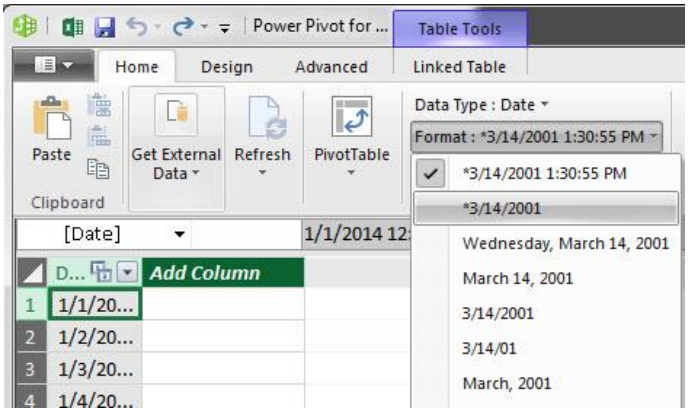


Data Modeling Step 3: Create DAX Calculated Columns in Calendar Table

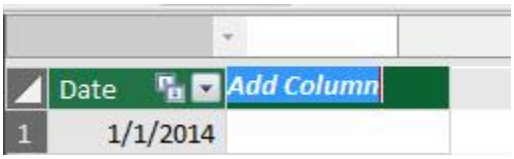
41) Click Data View button in View group: 

42) In Data View, click on dCalendar tab.

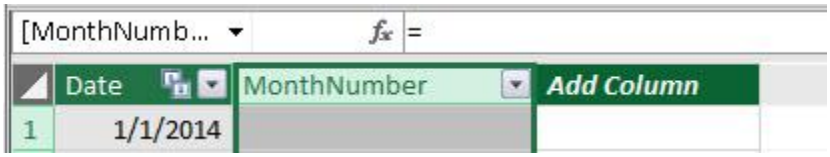
43) Click in first date, then in the Formatting group in the Manage Data Model Home Ribbon Tab, click the "Format:" button and then select Data Number Format from drop-down list:



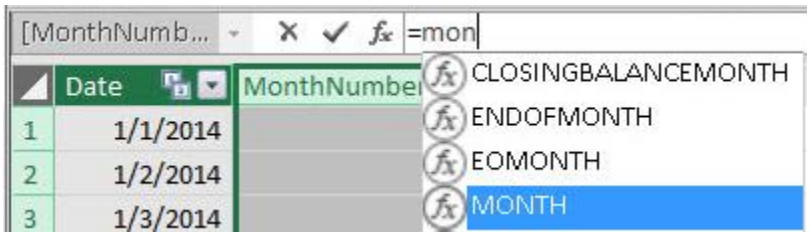
44) To create a new field in the Calendar Table, Double click "Add Column":



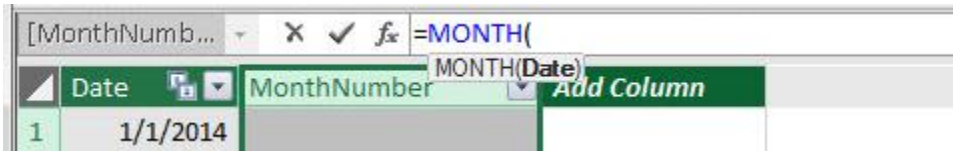
45) Type "Month Number" and hit Enter:



46) Type an equal sign and then the letters "mon". Notice that your cursor automatically jumps to the Formula Bar. Notice that there is a drop-down with possible DAX functions that you can use. Just as in Excel, when the function you want is highlighted in blue, use the Tab key to select the function.

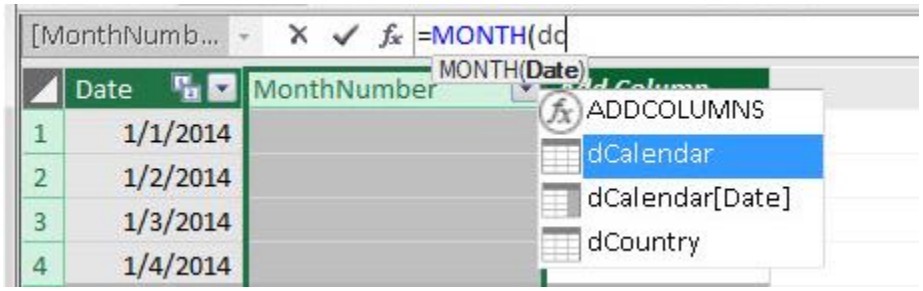


47) After you select function with Tab key, the MONTH function appears in the Formula Bar. This is the MONTH DAX Function. It works the same as the MONTH function in an Excel spreadsheet.

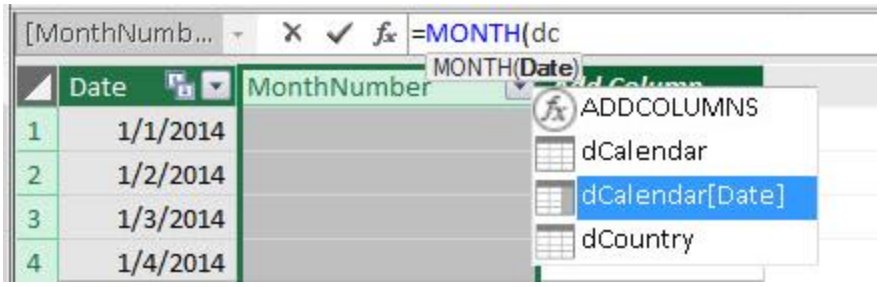


48) Type the first few letters of the Calendar Table name. Notice the drop down list that shows different icons:

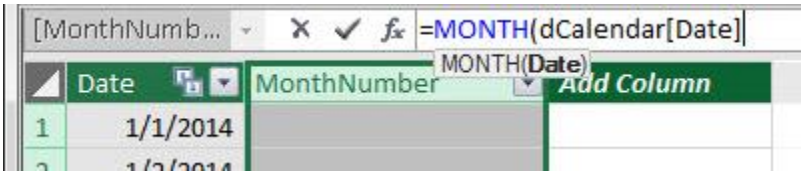
- f_x icon ➔ means a function.
- Table icon ➔ means the full Table.
- Table icon with a shaded column/field ➔ means a field in the Table.



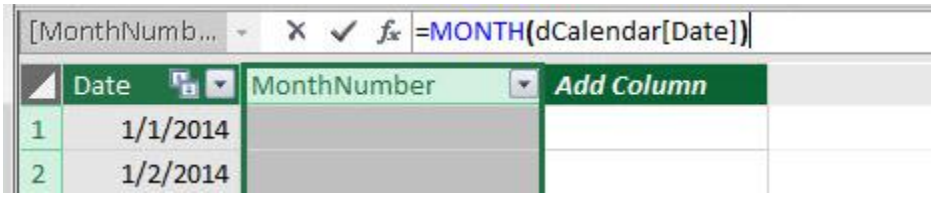
49) Arrow down to highlight the Date field in the dCalendar Table. Then hit the Tab key to select the Date field.



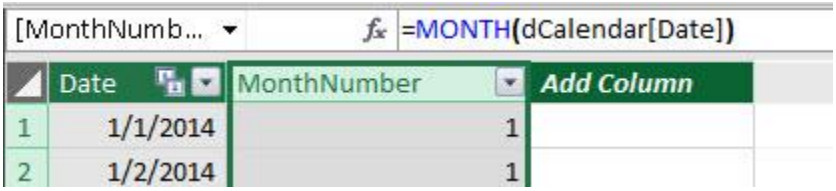
50) Our rule for referencing fields (called column references or field references) when we create DAX formulas is to always include both the Table Name and Field Name in square brackets.



51) Type Close parenthesis. Then hit the Enter key.



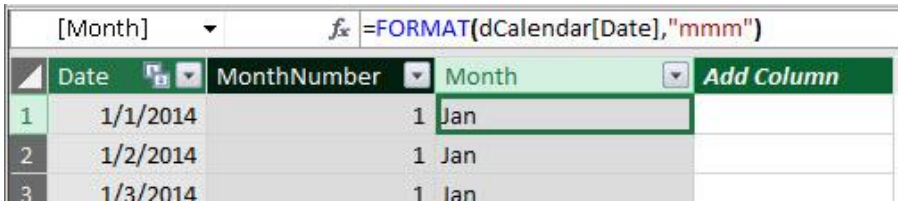
52) Our 1st DAX Calculated Column.



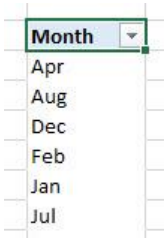
53) In Calculated Columns the formula calculated based on “Row Context”.

- The same formula appears in every cell in the column. Notice that there are no cell references in the formula. The Table Name and Field Name, dCalendar[Date] knows to look at the correct date in each row (record) because of “Row Context”, which is to say the field reference, dCalendar[Date], “sees” a new date in each row.

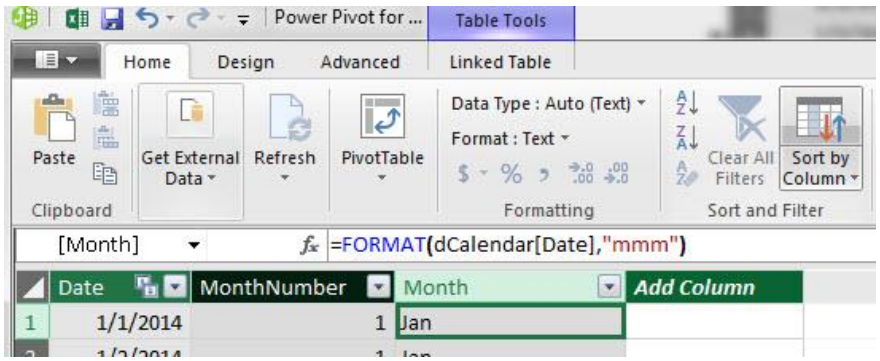
54) Now we create our 2nd Calculated Column to determine Month Name. Over in Excel we would use the TEXT Function with a formula like: =TEXT(A2,”mmm”). But over here in the DAX formula language the function name for adding a Custom Number Format to a number is FORMAT. Here is our formula for our DAX Calculated Column for Month Name:



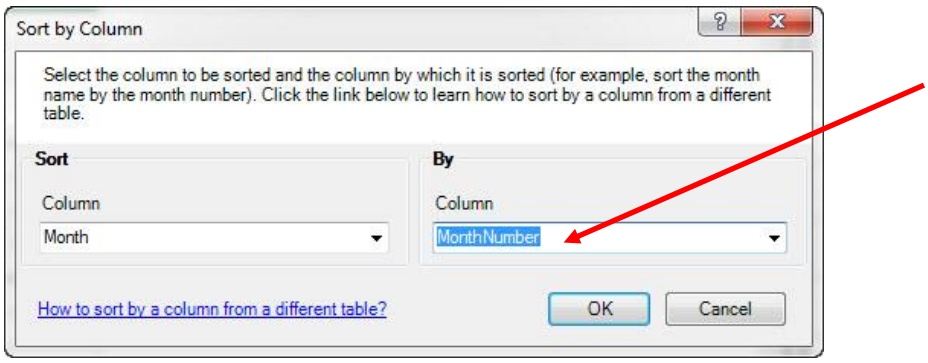
55) If we were to try to create a PivotTable based on this Month Column, the month names would be sorted alphabetically. It would look like this:



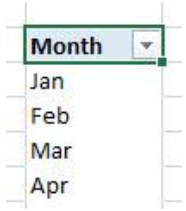
56) To solve this problem, we simply have to tell the Month field to sort based in the Month Number field (this is the reason we created the MonthNumber field). With a cell in the Month field selected, in the Sort & Filter group in the Manage Data Model Home Ribbon Tab, click the “Sort by Column” button:



57) In the Sort by Column dialog box set the “By column” to: MonthNumber:

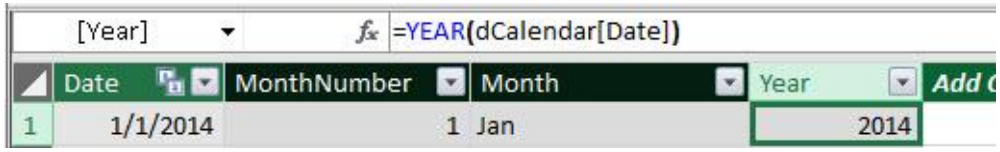


58) Now when you create a PivotTable based on this Month Column, the month sort correctly:



Month
Jan
Feb
Mar
Apr

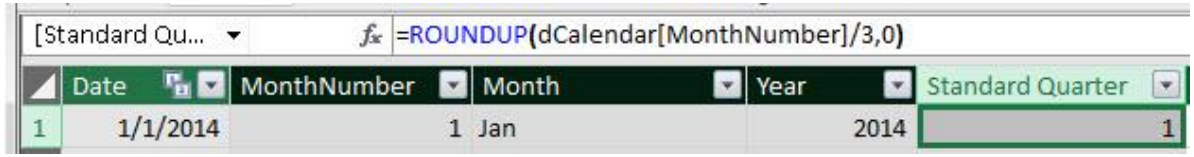
59) 3rd Calculated Column for Year:



[Year] =YEAR(dCalendar[Date])

	Date	MonthNumber	Month	Year	Add C
1	1/1/2014	1	Jan	2014	

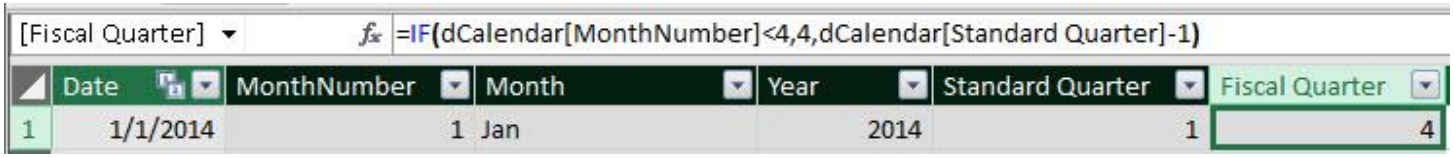
60) 4th Calculated Column for Standard Quarter (similar type of formula we learned about back in video #9):



[Standard Qu... =ROUNDUP(dCalendar[MonthNumber]/3,0)

	Date	MonthNumber	Month	Year	Standard Quarter
1	1/1/2014	1	Jan	2014	1

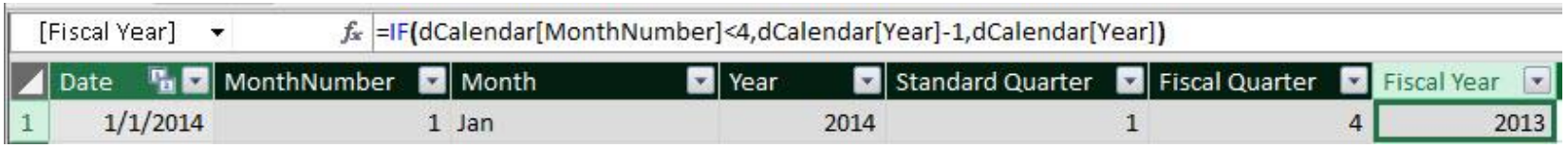
61) 5th Calculated Column for Fiscal Quarter, where our 4th Quarter is January, February and March (similar type of formula we learned about back in video #9):



[Fiscal Quarter] =IF(dCalendar[MonthNumber]<4,4,dCalendar[Standard Quarter]-1)

	Date	MonthNumber	Month	Year	Standard Quarter	Fiscal Quarter
1	1/1/2014	1	Jan	2014	1	4

62) 6th Calculated Column for Fiscal Year, where the first three months of the year must be associated with the previous year's financial statement reporting period (similar type of formula we learned about back in video #9):



[Fiscal Year] =IF(dCalendar[MonthNumber]<4,dCalendar[Year]-1,dCalendar[Year])

	Date	MonthNumber	Month	Year	Standard Quarter	Fiscal Quarter	Fiscal Year
1	1/1/2014	1	Jan	2014	1	4	2013

63) 7th Calculated Column for Fiscal Year & Quarter Label, (similar type of formula we learned about back in video #9):

- [Fiscal Period] =dCalendar[Fiscal Year]&" - Q"&dCalendar[Fiscal Quarter]

	Date	MonthNumber	Month	Year	Standard Quarter	Fiscal Quarter	Fiscal Year	Fiscal Period	Add Column
1	1/1/2014	1	Jan	2014	1	4	2013	2013 - Q4	

64) Here is a picture of our finished Calendar Table:

	Date	MonthNumber	Month	Year	Standard Quarter	Fiscal Quarter	Fiscal Year	Fiscal Period
1	1/1/2014	1	Jan	2014	1	4	2013	2013 - Q4
2	1/2/2014	1	Jan	2014	1	4	2013	2013 - Q4
3	1/3/2014	1	Jan	2014	1	4	2013	2013 - Q4
4	1/4/2014	1	Jan	2014	1	4	2013	2013 - Q4
5	1/5/2014	1	Jan	2014	1	4	2013	2013 - Q4

Data Modeling Step 3: Create DAX Calculated Columns in Fact Table for Revenue:

65) Our Fact Table has over two million records and we need to calculate the revenue for each transaction based on the price for each Product, the Revenue Discount and the Units (see table below). If we were in Excel, this would be very difficult to do and any formula that we created would take a long time to calculate. Luckily over here in the Data Model, we can use the DAX formula language in either a Calculated Column, or a Measure. As mentioned in the previous section (DAX Calculated Column or DAX Measure to calculate Total Revenue?), deciding which to use depends on how many records, how many unique values there are in each column and ultimately timing different methods). We will see how to make this particular calculation both ways.

	Date	Product	Revenue Discount	Net Cost Equivalent	County Code	Units	Add Column
1	3/15/2014	Fun Fly	0	0.86	USA	2	
2	12/17/2014	Fun Fly	0	0.86	USA	2	
3	12/20/2014	Fun Fly	0	0.86	USA	2	
4	12/24/2014	Fun Fly	0	0.86	USA	2	
5	12/12/2014	Fun Fly	0	0.86	USA	2	
6	12/10/2014	Fun Fly	0	0.86	USA	2	
7	11/25/2014	Fun Fly	0	0.86	USA	2	

fTransactions | dCountry | dProducts | dCalendar

Record: 1 of 2,686,103

66) The first thing we need to do is lookup the Retail Price for each product. There is no VLOOKUP in the DAX formula language. The RELATED DAX function is the function we use to do an Exact Match lookup to lookup the Product name in the dProduct table and retrieve the Retail Price for each product.

- How Related works:
    1. Because our goal is to retrieve the Retail Price in the dProducts table, we must enter the column reference dProducts[Retail Price] into the RELATED function.
    2. Through "Row Context", the RELATED function "sees" the Product name in each row of the fTransaction table.
    3. Because there is a One-to-Many Relationship between the two tables, RELATED goes over to the dProducts table and finds a match in the first column and brings back the Retail Price to the correct row in the fTransaction table.

- [Revenue] =RELATED(dProducts[Retail Price])

	Date	Product	Revenue Discount	Net Cost Equivalent	County Code	Units	Revenue	Add Column
1	3/15/2014	Fun Fly	0	0.86	USA	2	6.99	
2	12/17/2014	Fun Fly	0	0.86	USA	2	6.99	

67) Next we need to multiply the price times one minus the discount to get the Revenue after the discount is applied:

- [Revenue] =RELATED(dProducts[Retail Price])*(1-fTransactions[Revenue Discount])

	Date	Product	Revenue Discount	Net Cost Equivalent	County Code	Units	Revenue	Add Column
2069621	12/12/2015	Fun Fly	0.46	0.89	ITA	49	3.7746	
2069622	11/30/2015	Fun Fly	0.46	0.95	CRI	56	3.7746	

68) Now we multiply units times revenue after discount:

- [Revenue] =RELATED(dProducts[Retail Price])*(1-fTransactions[Revenue Discount])*fTransactions[Units]

	Date	Product	Revenue Discount	Net Cost Equivalent	County Code	Units	Revenue	Add Column
968764	2/20/2014	Bellen	0.48	0.85	KEN	72	1010.5056	
968765	6/20/2014	Bellen	0.48	1.01	DEU	72	1010.5056	

69) Finally, we must use the DAX ROUND function (works same as Excel's ROUND) to round to the penny:

- [Revenue] =ROUND(RELATED(dProducts[Retail Price])*(1-fTransactions[Revenue Discount])*fTransactions[Units],2)

	Date	Product	Revenue Discount	Net Cost Equivalent	County Code	Units	Revenue	Add Column
968764	2/20/2014	Bellen	0.48	0.85	KEN	72	1010.51	
968765	6/20/2014	Bellen	0.48	1.01	DEU	72	1010.51	

70) Notes about this formula:

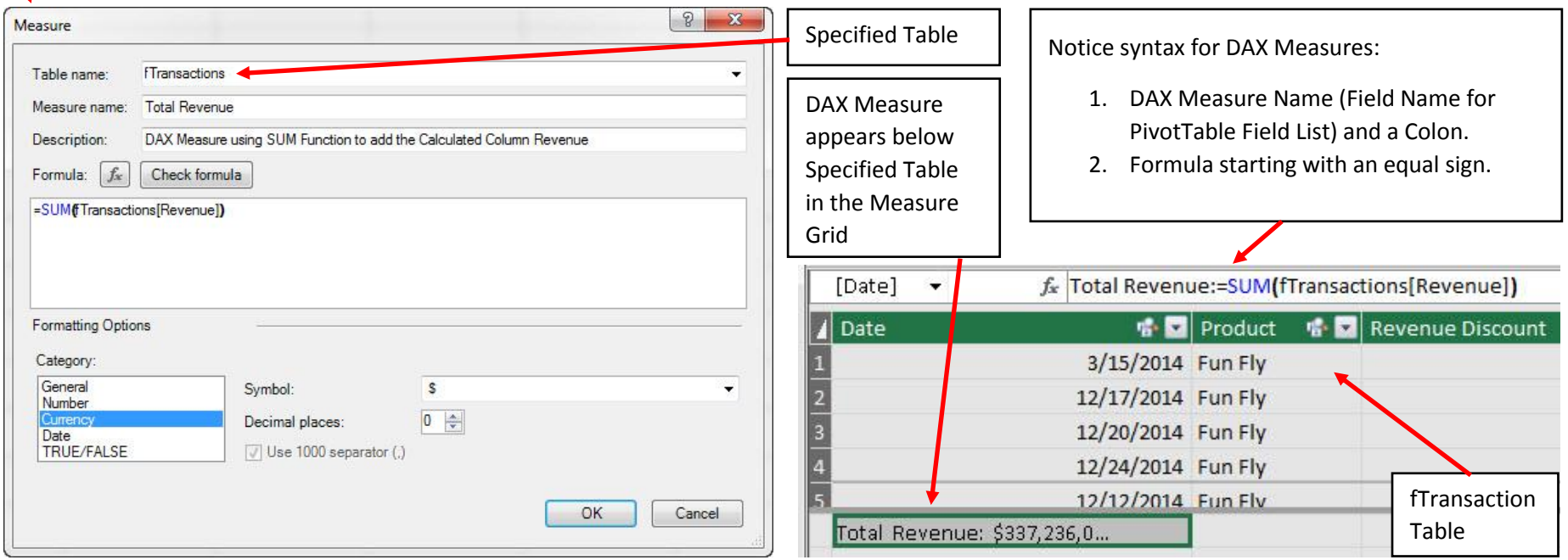
- The Calculated Column uses Row Context to calculate the Revenue amount for each row in the fTransaction table
- Because this is a Calculated Column, a unique list of revenue amounts is stored in a separate column in the in-RAM-memory Columnar Database. If this were a very large data set and we were having issues with slow calculations, we might consider not creating this Calculated Column, but instead creating a DAX Measure with the SUMX function for Total Revenue (example later…).
- We are not going to drop this new Calculated Column field, "Revenue" from the PivotTable Field List into the Values area of the PivotTable. Instead, we are going to create a Measure to add the total revenue and then use that Measure in our PivotTables or in other Measures. To this end, later in this project, we will hide the Revenue Calculated Column so that it does not appear in our PivotTable Field List.

Data Modeling Step 3: Create DAX Measures

71) Now we want to create a DAX Measure that will calculate Total Revenue.
72) There are two ways to create a DAX Measure:
    1. Measure Dialog Box.
    2. Measure Grid below Fact Table.

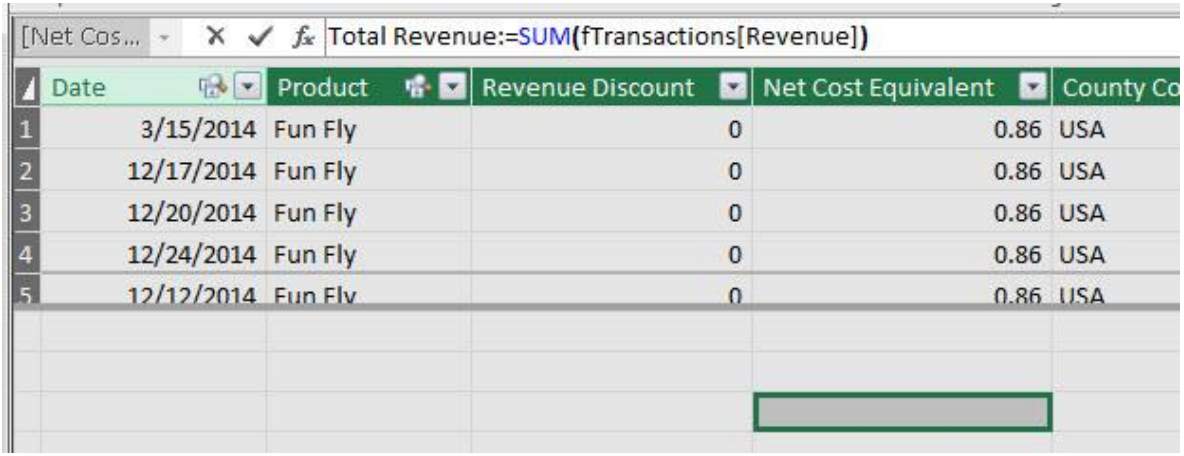
Measure Dialog Box

73) ***Measure Dialog Box Method***:
    1. To create your formula in the Measure Dialog Box: PowerPivot Ribbon Tab, Measure drop-down arrow, New Measure.
    2. When you are done and click OK in the dialog box and the DAX Measure appears below the Specified Table:

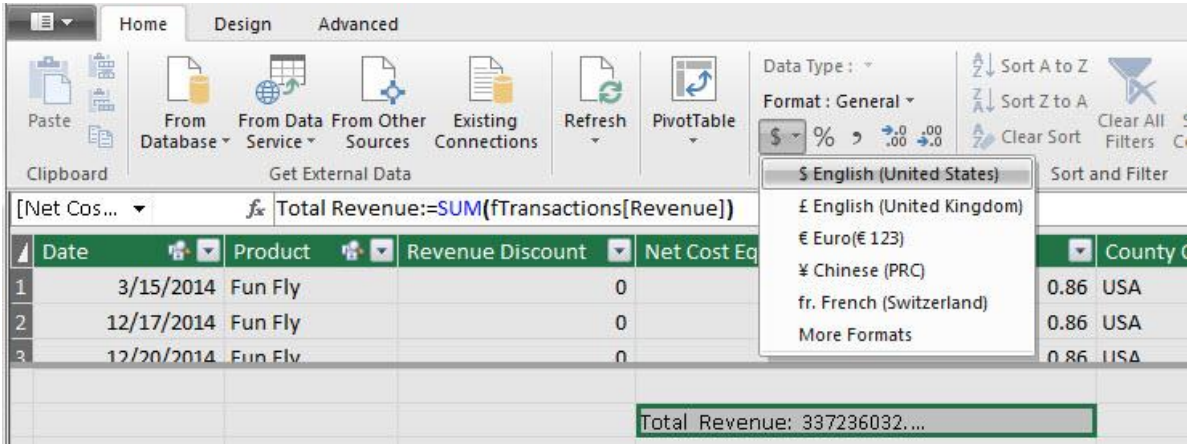


74) ***Second Method***:

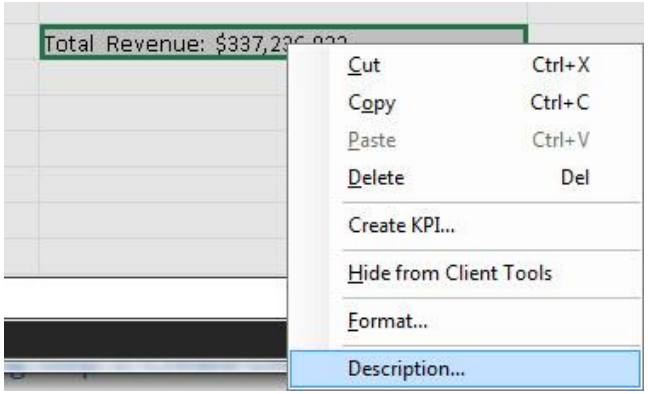
1. In the Measure Grid below the table in which you want the Measure to appear, click in cell.
2. Click in the Formula Bar.
3. Type your Measure.



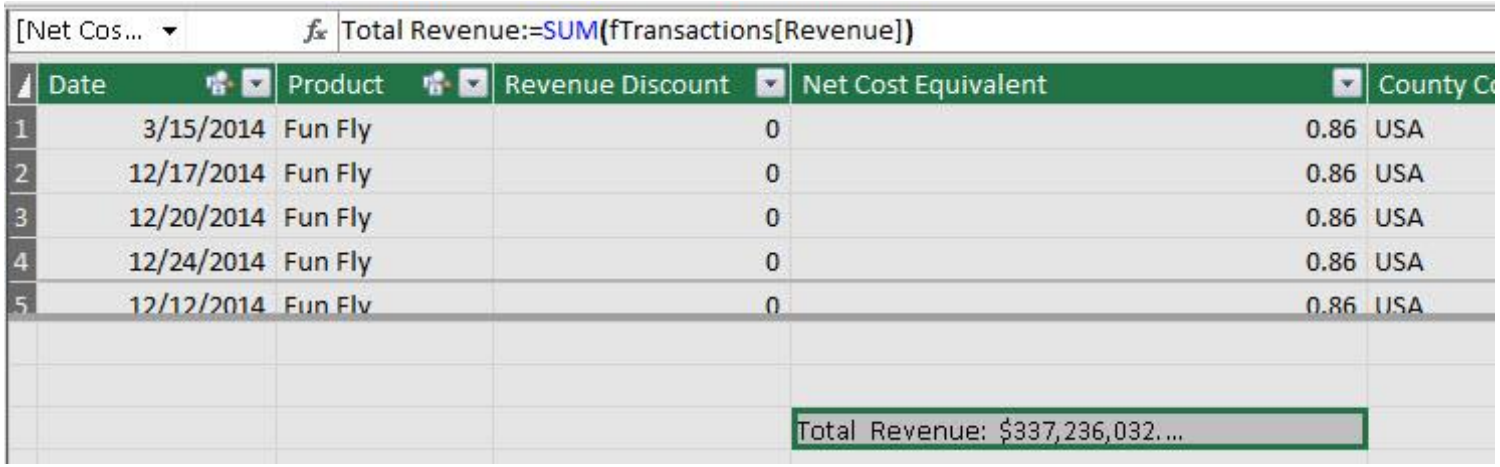
4. Hit Enter.
5. Use the Number Formatting buttons in the Format group in the Home Ribbon Tab of the Manage Data Model to apply Number Formatting.



6. You can add a description if you right-click the cell with the Measure and click on Description.



7. Done.

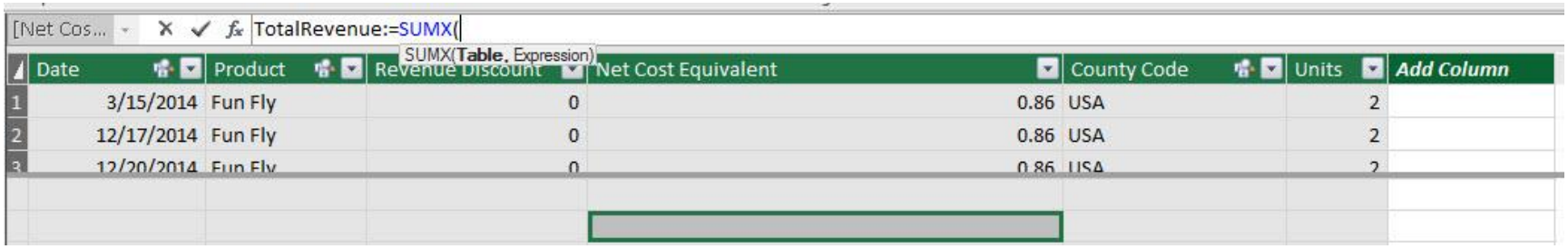


Data Modeling Step 3: Alternative Total Revenue Calculation: DAX Measure with SUMX

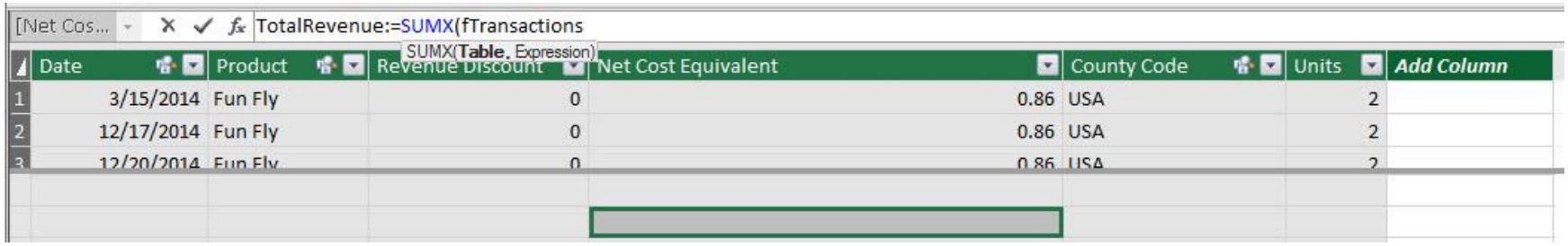
75) For comparison purposes, we now want to look at how to calculate Total Revenue if we did not use a Calculated Column. Remember: DAX Measures make aggregate calculations based on criteria. So how are we going to get a DAX Measure to iterate over the Fact Table and calculate using "Row Context"? A group of iterator functions are the answer! Functions like:
    1. SUMX, AVERAGEX, COUNTX, COUNTAX, MINX, MAXX ("X" functions).
    2. FILTER is also an iterative function (we don't get to use this function in this class).

78) With cell in Measure Grid, let's create the SUMX formula to add Total Revenue without a helper column:

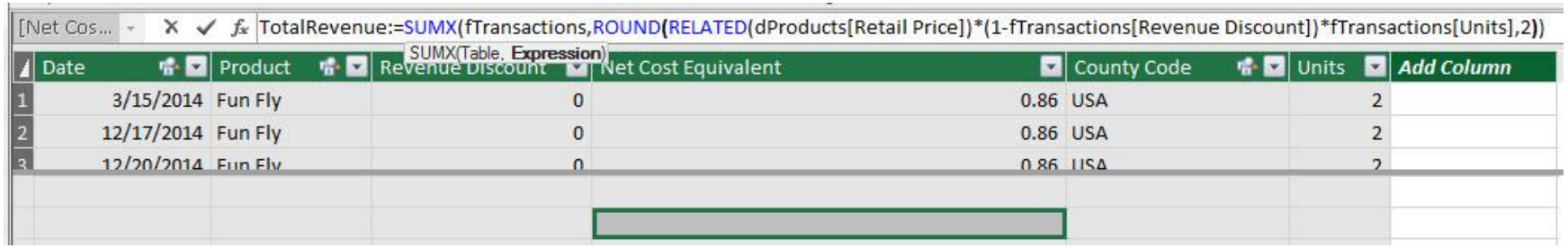
1. Use SUMX function:



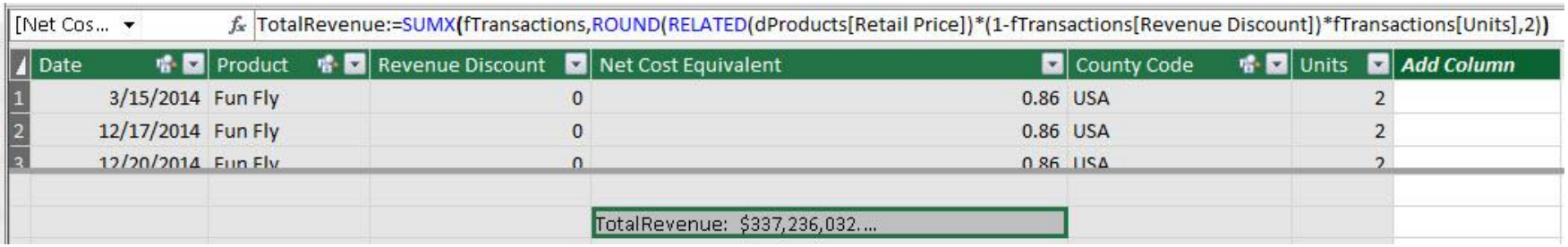
2. The SUMX needs to know which table it should "iterate" over, or which table it should apply "Row Context" to:



3. The Expression argument simply gets the formula that you would have used, had you done a Calculated Column (same exact formula we did when we created the Calculated Column for Revenue).



4. Apply Number Formatting and a Description and you are done:



79) Notes about this formula:

- The values created by the SUMX function when it iterates over the Fact Table are exactly the same as when we created the Calculated Column.
- We get the same number for Total Revenue whether we use the SUMX function or a Calculated Column and then a DAX Measure using the SUM function.
- Because there is no Calculated Column in the Fact Table, no values need to be stored in the Columnar Database and therefore nothing is stored in RAM.
- The values that the SUMX function creates and then adds are processed by the CPU – Central Processing Unit.
- This means there is a tradeoff and you decide:
    1. With Calculated Column you use more RAM.
    2. With SUMX you use more CPU when the Measure is dropped in PivotTable or when criteria is changed.
    3. For small data sets (even 10 million rows), it may not show any difference.
    4. If there appears to be slow calculation time, you may need to try both and see which works best.
    5. The larger the number of unique values:
        i. The more RAM will be used for Calculated Columns
        ii. The more the CPU will have to work with SUMX.
- None of our data sets in this class will show a significant difference in calculating times.

Data Modeling Step 3: More DAX Measures

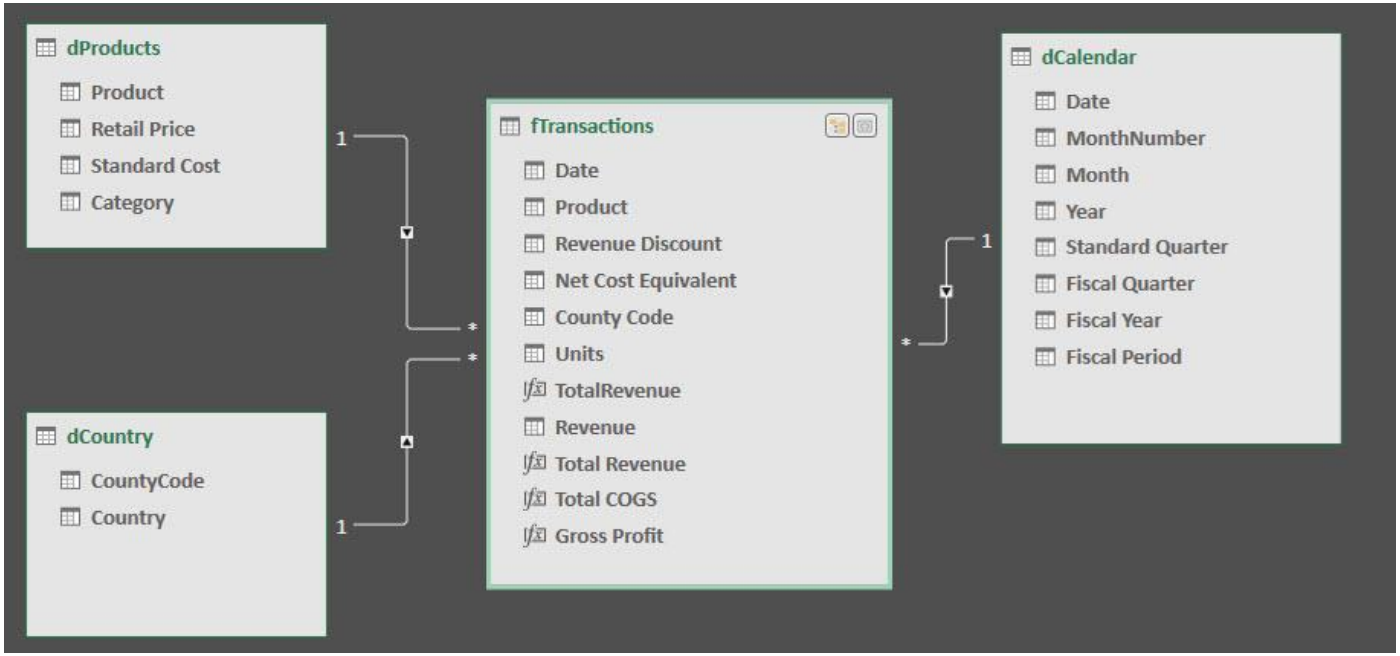
80) Create DAX Measure to calculate Total COGS:

- `Total COGS:=SUMX(fTransactions,ROUND(RELATED(dProducts[Standard Cost])*fTransactions[Net Cost Equivalent]*fTransactions[Units],2))`

81) Create DAX Measure to calculate Total Gross Profit. We are using our DAX Formula convention of always referring to Measures that are used in other Measures with square brackets only.

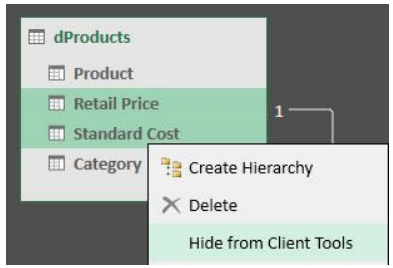
- `Gross Profit:=[Total Revenue]-[Total COGS]`

82) We can see the emerging Data Model in Diagram View:

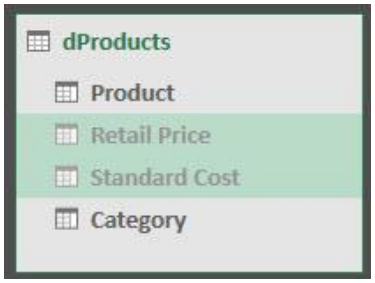
- 

Data Modeling Step 4: Hide Tables & Fields not used in PivotTables

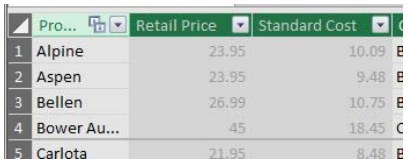
83) In Diagram View right click fields to hide and point to Hide from Client Tools.



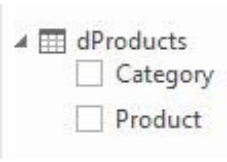
84) Afterwards, the fields are greyed out.



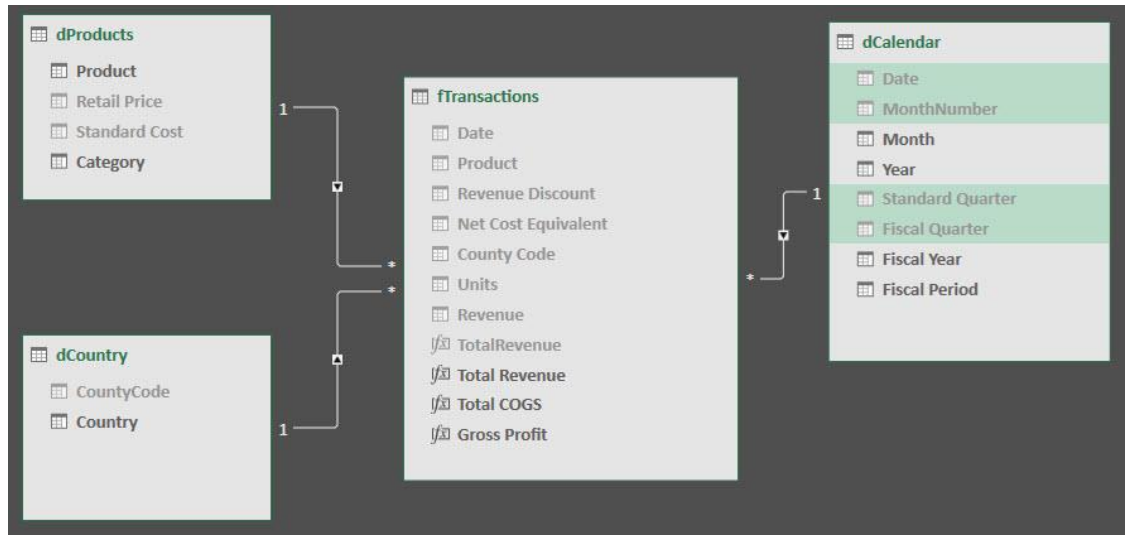
85) If you look at the table, the fields are greyed out.



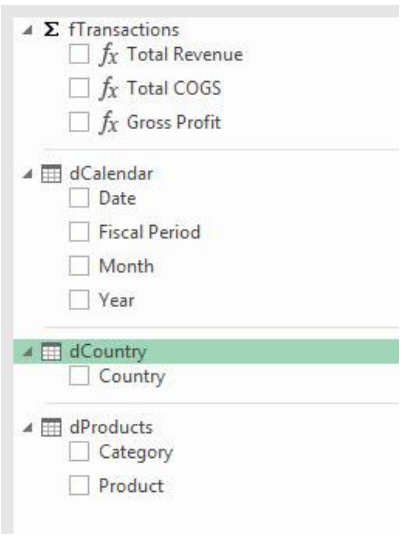
86) If you look at a PivotTable Field List that contains the table, the fields are hidden.



87) For our Data Model and Reports we want to hide all fields we will not need in the PivotTable:



88) The PivotTable Field List looks like this:



Data Modeling Step 5: Create PivotTables and Pivot Charts

89) PivotTable for Categories:

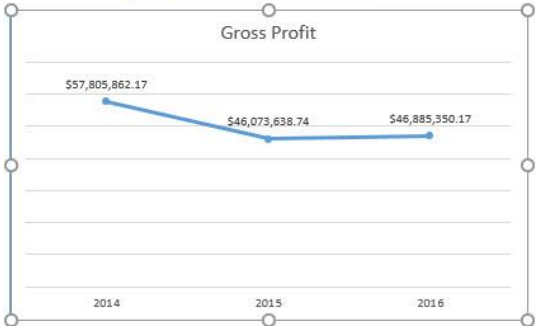
Category	Total Revenue	Total COGS	Gross Profit
Advanced	$47,068,502.99	$25,557,582.57	$21,510,920.42
Beginner	$128,947,457.40	$71,334,091.79	$57,613,365.61
Competition	$45,292,583.45	$26,367,405.10	$18,925,178.35
Freestyle	$26,777,447.52	$14,635,707.15	$12,141,740.37
Intermediate	$85,374,917.56	$45,894,782.02	$39,480,135.54
Novelty	$2,514,686.69	$1,421,175.90	$1,093,510.79
Grand Total	**$335,975,595.61**	**$185,210,744.53**	**$150,764,851.08**

90) PivotTable for Products:

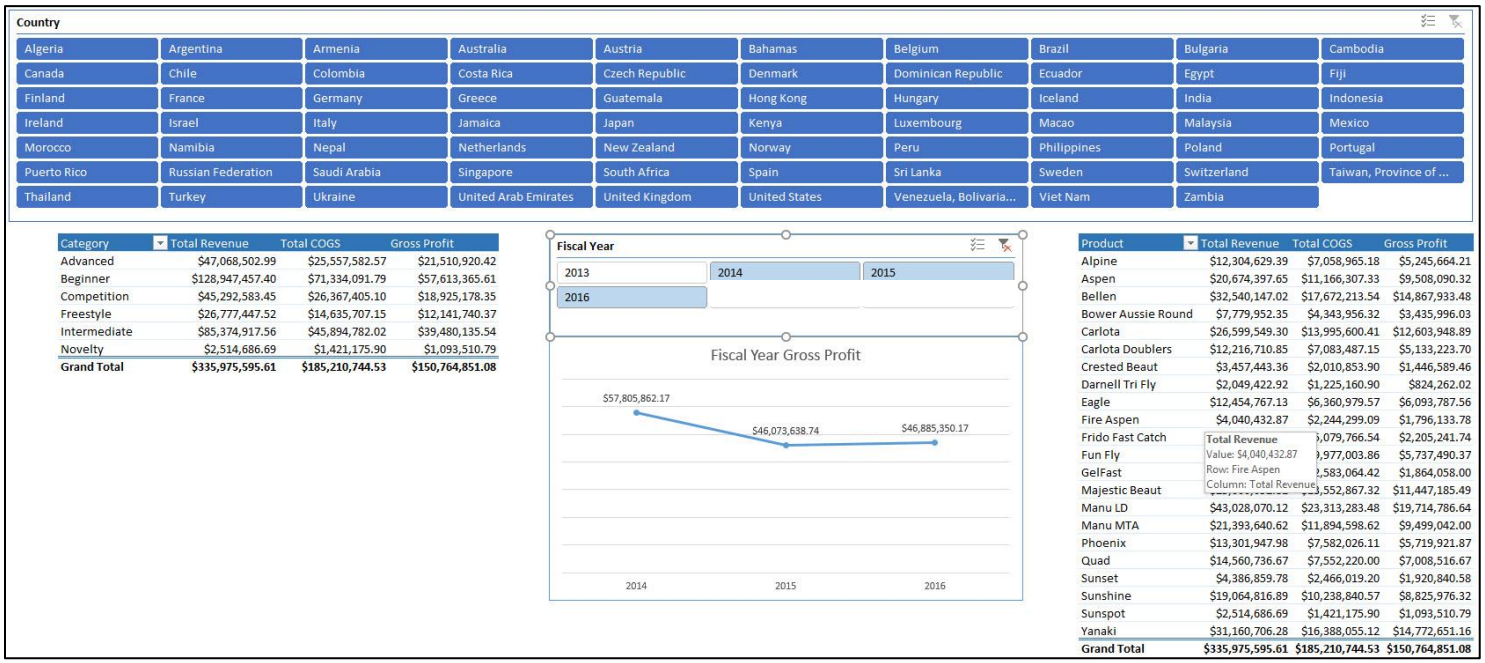
Product	Total Revenue	Total COGS	Gross Profit
Alpine	$12,304,629.39	$7,058,965.18	$5,245,664.21
Aspen	$20,674,397.65	$11,166,307.33	$9,508,090.32
Bellen	$32,540,147.02	$17,672,213.54	$14,867,933.48
Bower Aussie Round	$7,779,952.35	$4,343,956.32	$3,435,996.03
Carlota	$26,599,549.30	$13,995,600.41	$12,603,948.89
Carlota Doublers	$12,216,710.85	$7,083,487.15	$5,133,223.70
Crested Beaut	$3,457,443.36	$2,010,853.90	$1,446,589.46
Darnell Tri Fly	$2,049,422.92	$1,225,160.90	$824,262.02
Eagle	$12,454,767.13	$6,360,979.57	$6,093,787.56
Fire Aspen	$4,040,432.87	$2,244,299.09	$1,796,133.78
Frido Fast Catch	$7,285,008.28	$5,079,766.54	$2,205,241.74
Fun Fly	$15,714,494.23	$9,977,003.86	$5,737,490.37
GelFast	$4,447,122.42	$2,583,064.42	$1,864,058.00
Majestic Beaut	$25,000,052.81	$13,552,867.32	$11,447,185.49
Manu LD	$43,028,070.12	$23,313,283.48	$19,714,786.64
Manu MTA	$21,393,640.62	$11,894,598.62	$9,499,042.00
Phoenix	$13,301,947.98	$7,582,026.11	$5,719,921.87
Quad	$14,560,736.67	$7,552,220.00	$7,008,516.67
Sunset	$4,386,859.78	$2,466,019.20	$1,920,840.58
Sunshine	$19,064,816.89	$10,238,840.57	$8,825,976.32
Sunspot	$2,514,686.69	$1,421,175.90	$1,093,510.79
Yanaki	$31,160,706.28	$16,388,055.12	$14,772,651.16
Grand Total	**$335,975,595.61**	**$185,210,744.53**	**$150,764,851.08**

79) PivotTable and Chart for Fiscal Year Gross Profit:

Fiscal Year	Total Gross Profit
2014	$57,805,862.17
2015	$46,073,638.74
2016	$46,885,350.17
Grand Total	**$150,764,851.08**



80) Put all together on one sheet with a Slicer for Country and a Slicer for Fiscal Year:



Category	Total Revenue	Total COGS	Gross Profit
Advanced	$47,068,502.99	$25,557,582.57	$21,510,920.42
Beginner	$128,947,457.40	$71,334,091.79	$57,613,365.61
Competition	$45,292,583.45	$26,367,405.10	$18,925,178.35
Freestyle	$26,777,447.52	$14,635,707.15	$12,141,740.37
Intermediate	$85,374,917.56	$45,894,782.02	$39,480,135.54
Novelty	$2,514,686.69	$1,421,175.90	$1,093,510.79
Grand Total	**$335,975,595.61**	**$185,210,744.53**	**$150,764,851.08**

Product	Total Revenue	Total COGS	Gross Profit
Alpine	$12,304,629.39	$7,058,965.18	$5,245,664.21
Aspen	$20,674,397.65	$11,166,307.33	$9,508,090.32
Bellen	$32,540,147.02	$17,672,213.54	$14,867,933.48
Bower Aussie Round	$7,779,952.35	$4,343,956.32	$3,435,996.03
Carlota	$26,599,549.30	$13,995,600.41	$12,603,948.89
Carlota Doublers	$12,216,710.85	$7,083,487.15	$5,133,223.70
Crested Beaut	$3,457,443.36	$2,010,853.90	$1,446,589.46
Darnell Tri Fly	$2,049,422.92	$1,225,160.90	$824,262.02
Eagle	$12,454,767.13	$6,360,979.57	$6,093,787.56
Fire Aspen	$4,040,432.87	$2,244,299.09	$1,796,133.78
Frido Fast Catch		,079,766.54	$2,205,241.74
Fun Fly		,977,003.86	$5,737,490.37
GelFast		,583,064.42	$1,864,058.00
Majestic Beaut		,552,867.32	$11,447,185.49
Manu LD	$43,028,070.12	$23,313,283.48	$19,714,786.64
Manu MTA	$21,393,640.62	$11,894,598.62	$9,499,042.00
Phoenix	$13,301,947.98	$7,582,026.11	$5,719,921.87
Quad	$14,560,736.67	$7,552,220.00	$7,008,516.67
Sunset	$4,386,859.78	$2,466,019.20	$1,920,840.58
Sunshine	$19,064,816.89	$10,238,840.57	$8,825,976.32
Sunspot	$2,514,686.69	$1,421,175.90	$1,093,510.79
Yanaki	$31,160,706.28	$16,388,055.12	$14,772,651.16
Grand Total	**$335,975,595.61**	**$185,210,744.53**	**$150,764,851.08**

Data Modeling Step 6: Refresh Data Model when Source Data Changes

81) New Files added to folder:

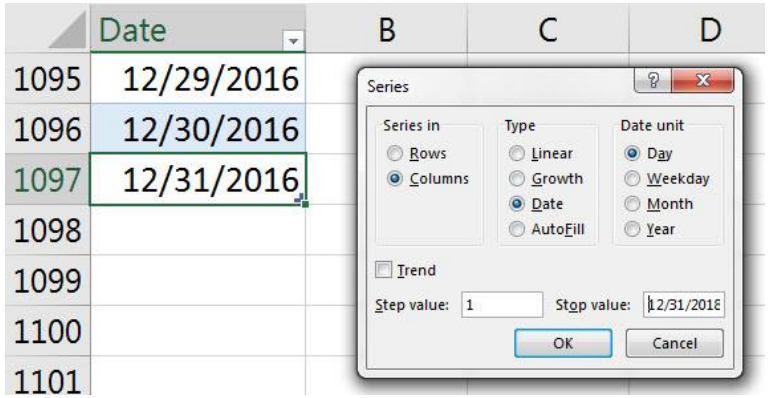
82) After we refresh Data Model with Ctrl + Alt + F5 we have 4.7 million records:

83) Chart for Fiscal Year did not update because Calendar Table in Excel only goes to 2016.

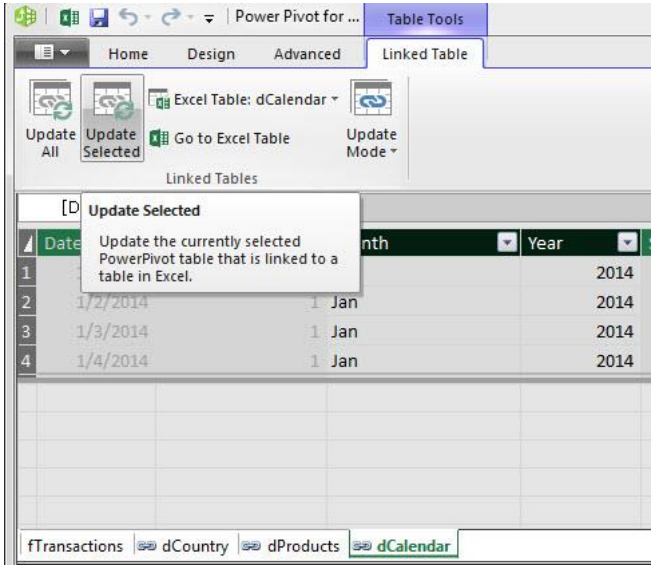
Fiscal Year	Total Gross Profit
2014	$57,805,862.17
2015	$46,073,638.74
2016	$46,885,350.17
Grand Total	$150,764,851.08

Data Modeling Step 7: Fix Calendar Table

84) In Excel we click in last cell in the Calendar Table, go to the Home Ribbon Tab, Edit group, Fill drop-down, Series, and in the Series dialog box we choose “Series in” as “Columns”, and we choose “Stop value” as “12/31/2018”:

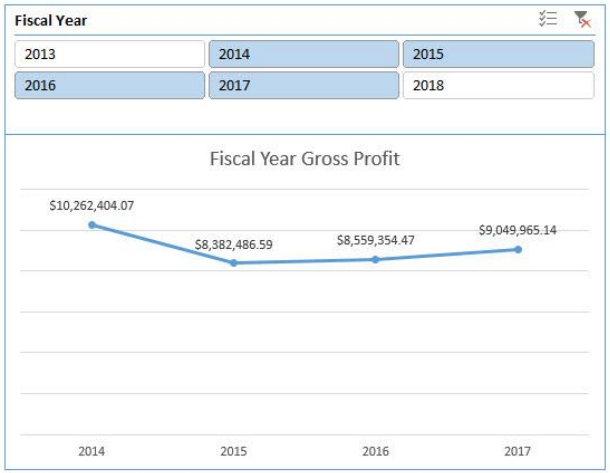


85) To update the Calendar Table in the Data Model we go to the Data Model, then we go to Data View, then click on the tab for dCalendar, then we go to the Linked Table Ribbon Tab, then we click on the Update Selected button.



Data Modeling Step 7: After Refreshing

Category	Total Revenue	Total COGS	Gross Profit
Advanced	$11,466,023.29	$6,294,457.87	$5,171,565.42
Beginner	$31,570,287.90	$17,641,542.57	$13,928,745.33
Competition	$10,898,810.77	$6,400,549.07	$4,498,261.70
Freestyle	$6,409,053.54	$3,536,573.03	$2,872,480.51
Intermediate	$20,723,401.43	$11,207,204.19	$9,516,197.24
Novelty	$614,821.59	$347,861.52	$266,960.07
Grand Total	**$81,682,398.52**	**$45,428,188.25**	**$36,254,210.27**



Product	Total Revenue	Total COGS	Gross Profit
Alpine	$3,016,415.69	$1,745,237.94	$1,271,177.75
Aspen	$5,032,898.75	$2,748,517.30	$2,284,381.45
Bellen	$7,976,660.20	$4,378,203.56	$3,598,456.64
Bower Aussie Round	$1,790,089.65	$1,005,051.70	$785,037.95
Carlota	$6,546,543.18	$3,482,300.63	$3,064,242.55
Carlota Doublers	$2,955,012.90	$1,725,845.35	$1,229,167.55
Crested Beaut	$845,960.89	$496,293.96	$349,666.93
Darnell Tri Fly	$483,025.75	$291,262.08	$191,763.67
Eagle	$3,028,760.01	$1,559,592.23	$1,469,167.78
Fire Aspen	$979,993.33	$547,798.40	$432,194.93
Frido Fast Catch	$1,744,300.85	$1,221,329.46	$522,971.39
Fun Fly	$3,835,876.32	$2,457,093.60	$1,378,782.72
GelFast	$1,104,850.76	$651,269.97	$453,580.79
Majestic Beaut	$6,089,819.85	$3,316,519.61	$2,773,300.24
Manu LD	$10,486,029.96	$5,746,659.47	$4,739,370.49
Manu MTA	$5,172,564.54	$2,901,169.62	$2,271,394.92
Phoenix	$3,227,778.22	$1,846,381.25	$1,381,396.97
Quad	$3,454,040.64	$1,810,727.68	$1,643,312.96
Sunset	$1,087,004.97	$621,728.32	$465,276.65
Sunshine	$4,678,868.01	$2,538,927.46	$2,139,940.55
Sunspot	$614,821.59	$347,861.52	$266,960.07
Yanaki	$7,531,082.46	$3,988,417.14	$3,542,665.32
Grand Total	**$81,682,398.52**	**$45,428,188.25**	**$36,254,210.27**

Data Modeling Step 7: Create new DAX Formulas and create New Report.

86) Create DAX Measure to calculate Gross Profit %:

- `Gross Profit %:=DIVIDE([Gross Profit],[Total Revenue])`

    1. DIVIDE DAX function. If there is an error, DIVIDE will show nothing in the cell (it actually delivers a “BLANK”, which is neither text or a number, but rather an empty cell).

87) We need to create a formula that can see the date Filter Context and get the amount from last year’s same period. The CALCULATE function is the DAX function that can change the Filter Context. By listing the Measure, [Gross Profit] and the DAX function SAMEPERIODLASTYEAR, the CALCULATE function knows to calculate the Gross Profit for the same period last year.

88) Create DAX Measure to calculate Gross Profit Amount for the Same Period Last Year:

- `GrossProfitAmountSamePeriodLastYear:=CALCULATE([Gross Profit],SAMEPERIODLASTYEAR(dCalendar[Date]))`

    1. CALCULATE DAX function can change the Filter Context for the Measure listed in the first argument. The Filter1, Filter2 arguments are the conditions that change the Filter Context.
    2. For more about the CALCULATE function see this video: [Excel 2013 PowerPivot Basics #10: CALCULATE function to Change Filter Context (14 Examples)](https://www.youtube.com/watch?v=kMMohkVk8Ds), https://www.youtube.com/watch?v=kMMohkVk8Ds

89) We need a DAX Measure to calculate the percentage change from one date period to the next. Because some of the earliest periods do not have a previous period to be compared to, we use the IF function with a logical test to check if the same period last year is zero.

90) Create DAX Measure to calculate % Gross Profit Change From Last Year:

-
```
% Gross Profit Change From Last Year:=IF([GrossProfitAmountSamePeriodLastYear]=0,BLANK(),[Gross Profit]/[GrossProfitAmountSamePeriodLastYear]-1)
```

    1. IF DAX function. Works same as Excel's IF function.
    2. BLANK DAX function:
        - BLANK = Empty Cell, Missing Value
        - BLANK is not a Zero Length Text String
        - BLANK is not an Error
        - BLANK function is important because we cannot use a Zero Length Text String (like we could in Excel), because in Power Query and Power Pivot, the field must contain one Data Type. Using a Zero Length Text String with a Gross Profit Number is not allowed because then there would be a Text & Number Data Type mixed together, and that is not allowed.

91) New Report:

-

Fiscal Period	Gross Profit	Gross Profit %	% Gross Profit Change From Last Year
2015 - Q1	$8,646,180.29	42.72 %	-23.31 %
2015 - Q2	$8,637,962.69	43.19 %	-24.97 %
2015 - Q3	$20,200,779.54	42.88 %	-23.77 %
2015 - Q4	$8,588,716.22	42.85 %	0.82 %
2016 - Q1	$8,652,597.69	42.97 %	0.07 %
2016 - Q2	$8,785,364.72	42.89 %	1.71 %
2016 - Q3	$20,378,256.63	42.95 %	0.88 %
2016 - Q4	$9,069,131.13	42.73 %	5.59 %
2017 - Q1	$9,120,299.37	42.66 %	5.41 %
2017 - Q2	$9,174,040.39	43.02 %	4.42 %
2017 - Q3	$21,499,224.89	42.98 %	5.50 %
2017 - Q4	$9,933,862.42	42.76 %	9.53 %
Grand Total	**$142,686,415.98**	**42.90 %**	**-5.36 %**

DAX Operators

Operators			
Category	**Operator**	**Symbol**	**Example**
Parenthesis	Precedence order	( )	
Arithmetic	Add	+	
	Subtract	-	
	Multiply	*	
	Divide	/	
	Exponent	^	
Comparative	Equal	=	
	Not Equal	<>	
	Great than	>	
	Great than or equal to	>=	
	Less than	<	
	Less than or equal to	<=	
Join	Concatenation	&	
Logical	AND	&& or AND function	[Product]="Quad && [Quantity] >20 AND([Product]="Quad , [Quantity])
	OR	|| or OR function	[Product]="Quad || [Quantity] >20 OR([Product]="Quad , [Quantity])
	Not Equal	! Or NOT function	! ([Product]="Quad") NOT([Product]="Quad")

Cumulative List of Keyboards Throughout Class:

1) Esc Key:
    i. Closes Backstage View (like Print Preview).
    ii. Closes most dialog boxes.
    iii. If you are in Edit mode in a Cell, Esc will revert back to what you had in the cell before you put the Cell in Edit mode.
2) **F2 Key** = Puts formula in Edit Mode and shows the rainbow colored Range Finder.
3) SUM Function: **Alt + =**
4) **Ctrl + Shift + Arrow** = Highlight column (Current Region).
5) **Ctrl + Backspace** = Jumps back to Active Cell
6) **Ctrl + Z** = Undo.
7) **Ctrl + Y** = Undo the Undo.
8) **Ctrl + C** = Copy.
9) **Ctrl + X** = Cut.
10) **Ctrl + V** = Paste.
11) **Ctrl + PageDown** =expose next sheet to right.
12) **Ctrl + PageUp** =expose next sheet to left.
13) **Ctrl + 1** = Format Cells dialog box, or in a chart it opens Format Chart Element Task Pane.
14) **Ctrl + Arrow**: jumps to the bottom of the "**Current Region**", which means it jumps to the last cell that has data, right before the first empty cell.
15) **Ctrl + Home** = Go to Cell A1.
16) **Ctrl + End** = Go to last cell used.
17) Alt keyboards are keys that you hit in succession. Alt keyboards are keyboards you can teach yourself by hitting the Alt key and looking at the screen tips.
    i. Create PivotTable dialog box: **Alt, N, V**
    ii. Page Setup dialog box: **Alt, P, S, P**
    iii. Keyboard to open Sort dialog box: **Alt, D, S**
18) **ENTER** = When you are in Edit Mode in a Cell, it will put thing in cell and move selected cell DOWN.
19) **CTRL + ENTER** = When you are in Edit Mode in a Cell, it will put thing in cell and keep cell selected.
20) **TAB** = When you are in Edit Mode in a Cell, it will put thing in cell and move selected cell RIGHT.
21) **SHIFT + ENTER** = When you are in Edit Mode in a Cell, it will put thing in cell and move selected cell UP.
22) **SHIFT + TAB** = When you are in Edit Mode in a Cell, it will put thing in cell and move selected cell LEFT.
23) **Ctrl + T** = Create Excel Table (with dynamic ranges) from a Proper Data Set.
    i. Keyboard to name Excel Table: **Alt, J, T, A**
    ii. **Tab** = Enter Raw Data into an Excel Table.
24) **Ctrl + Shift + ~ (`)** = General Number Formatting Keyboard.
25) **Ctrl + ;** = Keyboard for hardcoding today's date.
26) **Ctrl + Shift + ;** = Keyboard for hardcoding current time.
27) **Arrow Key** = If you are making a formula, Arrow key will “hunt” for Cell Reference.
28) **Ctrl + B** = Bold the Font
29) **Ctrl + * (on Number Pad)** or **Ctrl + Shift + 8** = Highlight Current Table.
30) **Alt + Enter** = Add Manual Line Break (Word Wrap)
31) **Ctrl + P =** Print dialog Backstage View and Print Preview
32) **F4 Key** = If you are in Edit mode while making a formula AND your cursor is touching a particular Cell Reference, F4 key will toggle through the different Cell References:
    i. **A1** = Relative
    ii. **A1** = Absolute or “Locked”

iii. **A$1** = Mixed with Row Locked (Relative as you copy across the columns AND Locked as you copy down the rows)
iv. **$A1** = Mixed with Column Locked (Relative as you copy down the rows AND Locked as you across the columns)

33) **Ctrl + Shift + 4** = Apply Currency Number Formatting
34) **Tab key** = When you are selecting a Function from the Function Drop-down list, you can select the function that is highlighted in blue by using the Tab key.
35) **F9 Key** = To evaluate just a single part of formula while you are in edit mode, highlight part of formula and hit the F9 key.
    i. If you are creating an Array Constant in your formula: Hit F9.
    ii. If you are evaluating the formula element just to see what that part of the formula looks like, REMEMBER: to Undo with Ctrl + Z.
36) **Alt, E, A, A** = Clear All (Content and Formatting)
37) Evaluate Formula One Step at a Time Keyboard: **Alt, M, V**
38) Keyboard to open Sort dialog box: **Alt, D, S**
39) **Ctrl + Shift + L** = Filter (or **Alt, D, F, F**) = Toggle key for Filter Drop-down Arrows
40) **Ctrl + N** = Open New File
41) **F12** = Save As (Change File Name, Location, File Type)
42) Import Excel Table into Power Query Editor: **Alt, A, P, T**
43) **Ctrl + 1 (When Chart element in selected):** Open Task Pane for Chart Element
44) **F4 Key** = If you are in Edit mode while making a formula AND your cursor is touching a particular Cell Reference, F4 key will toggle through the different Cell References:
    i. **A1** = Relative
    ii. **A1** = Absolute or "Locked"
    iii. **A$1** = Mixed with Row Locked (Relative as you copy across the columns AND Locked as you copy down the rows)
    iv. **$A1** = Mixed with Column Locked (Relative as you copy down the rows AND Locked as you across the columns)
45) Keyboard to open Scenario Manager = **Alt, T, E**
46) **Ctrl + Tab** = Toggle between Excel Workbook File Windows
47) **Ctrl + Shift + F3** = Create Names From Selection
48) **Ctrl + F3** = open Name Manager
49) **F3** = Paste Name or List of Names
50) **Alt + F4** = Close Active Window
51) **Window Key + Up Arrow** = Maximize Active Window
52) **Ctrl + Shift + Enter** = Keystroke to enter Array Formulas that: 1) have a function argument that requires it, or 2) whether or not you are entering the Resultant Array into multiple cells simultaneously.
53) **Ctrl + /** = Highlight current Array
54) Data Validation Dialog Box: **Alt, D, L**
55) **F11** = Create Chart on a new sheet
56) **Alt + F11** = Create Chart on currently selected sheet.
57) New Format Rule dialog box: **Alt, H, L, N**
58) Delete conditional Formatting Rule: **Alt, O, D, D**
59) Manage Rule dialog box keyboard: **Alt, O, D**
60) "Format values where this formula is true": **Alt, H, L, N, PageDown, Tab**
61) Shift + F11 = Insert a New Sheet
62) Ctrl + F1 = Toggle Ribbon Tabs on and off
63) Ctrl + Alt + F5 = Refresh All Data in Excel Workbook.

64) Zoom to Selection = **Alt, W, G**
65) Ctrl + F = Find
66) Ctrl + H = Find and Replace
67) Advanced Filter keyboard: Alt, A, Q or Alt, D, F, A