Microsoft Power Tools for Data Analysis #16 & 17

Introduction to Power BI Desktop

Notes from Video:

Table of Contents

- Overview of Excel Power Pivot & Power BI Desktop?

Choice between: Excel Power Pivot & Power BI Desktop

Excel Power Pivot:

1. Power Query, Columnar Database, Relationships, DAX Formulas are almost identical in both.
2. PivotTable Report is what you want
3. Have Excel Worksheets to compliment Data Model PivotTable Reports that allow you freedom to:

a. Work in cells, not columns and tables
b. Have any of the other Excel features to compliment Data Model PivotTables

4. Familiar with Excel.
5. DAX Formula calculate more slowly in Excel because they are calculated with MDX, which uses only one processor at a time.
6. Hard to share Power Pivot Report.

Power BI Desktop:

1. Power Query, Columnar Database, Relationships, DAX Formulas are almost identical in both.
2. More varied Visualizations and Reports
3. Visualizations and Reports are interactive (one can filter the other)
4. You can publish Visualizations and Reports, so they can be consumed on any device.
5. Table DAX Formulas can be part of the Data Model as a Table.
6. DAX Formulas calculate more quickly in Power BI because they are calculated using DAX which allows parallel processors to work on calculations. This matters for big data.
7. PBID gets updates each month & sometimes we get DAX Formulas or other features that aren't in Excel (why we can't create Data Model in PBID & open in Excel, but reverse is possible).

- Approximate History of Power BI Desktop :
    i. Power Pivot with the Columnar Database, DAX Formulas, Relationships and Data Model PivotTables were debuted as an add-in to Excel in 2009
    ii. Power Query was debuted as an Add-in for Excel in 2013
    iii. Power View & Power Map were add-ins in Excel also
    iv. All these Power Tools were refined in Excel between 2009 and 2015.
    v. Then in 2015 Microsoft combined all these tools together in one tool and gave it away for free. This tool was called Power BI Desktop.
- Power BI Desktop is a free download that accomplished these goals:
    i. We can import, transform and clean data and load it to the Power BI Desktop Data Model.
    ii. We can build relationships between tables.
    iii. We can build DAX Measures, DAX Calculated Columns and DAX Tables for the Data Model.
    iv. There are tabs in our Power BI File and each tab contains a set of tables, visualizations, formatting, filters and other visual elements that together are a single Tab or Page.
    v. All of the Tabs (Pages) together are called the report.
    vi. The report can be published to powerbi.com.
    vii. From the published report at powerbi.com, we can:
        1. View and interact with the report on any device.
        2. Print parts of our report.
        3. Download ".pbix" file.
        4. Edit the report or create new reports.
        5. Get embed code to use in a web site.
        6. Export as a PowerPoint Slide Show.
        7. Share with others who have an e-mail.
        8. Pin Multiple Tabs from multiple Reports to an online Power BI Dashboard.

- Different Versions of Power BI (Different Power BI Products) Available from Microsoft :
  1) Microsoft web site link for Power BI Products:
     https://powerbi.microsoft.com/en-us/pricing/

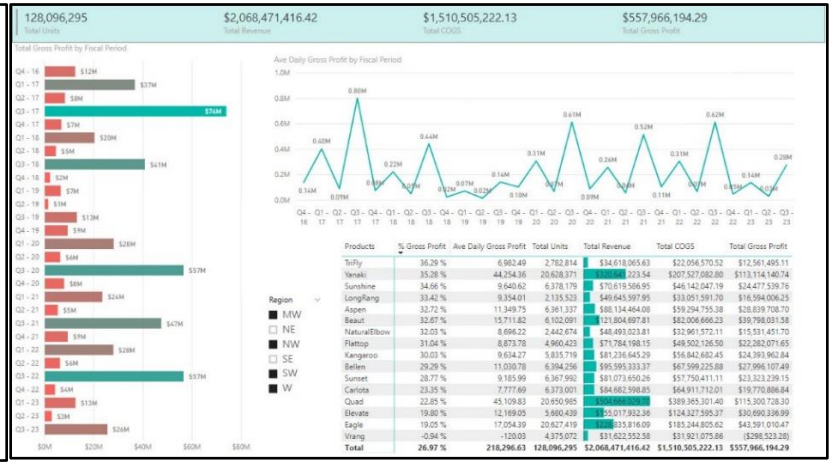
  2) Versions of Power BI (Different Power BI Products):
     i. Power BI Desktop
        1. Price: Free
        2. Connect to hundreds of data sources
        3. Clean and prepare data using visual tools
        4. Analyze and build reports with custom visualizations
        5. **Publish to the powerbi.com**
        6. **Can share ".pbix" file so others with the "free" Power BI Desktop can view.**
        7. **Create Web Embed Code you can post at Public web site.**
     ii. Power BI Pro
        1. $9.99, per user, per month
        2. Build dashboards that deliver a 360-degree, real-time view of the business
        3. Keep data up-to-date automatically, including on-premises sources
        4. Collaborate on shared data
        5. Audit and govern how data is accessed and used
        6. Package content and distribute to users with apps
        7. **Publish to powerbi.com and have others view on any device.**
        8. From published report viewers can download Data Model into Excel or as a ".pbix" file. (Implicit Measures will not download into an Excel file).
     iii. Power BI Premium
        1. Price: Negotiated per node, per month
        2. Share data with users inside and outside your organization without purchasing a per-user license
        3. Allocate, control, and manage your dedicated server capacity
        4. Unlock higher limits for Pro users with datasets up to 10 GB and refresh up to 48 times per day
        5. Access interactive and paginated reports online or use **Power BI Report Server for on-premises reporting**
        6. Deploy in the geographic region of your choice
- Download Power BI Desktop :
     i. Microsoft Download Page: https://www.microsoft.com/en-us/download/details.aspx?id=45331
        1. When you want to update each month, you have to re-download and re-install.
     ii. Microsoft Windows Store: https://powerbi.microsoft.com/en-us/desktop/
        1. This download will automatically update each month.
- Great YouTube Channels with more information :

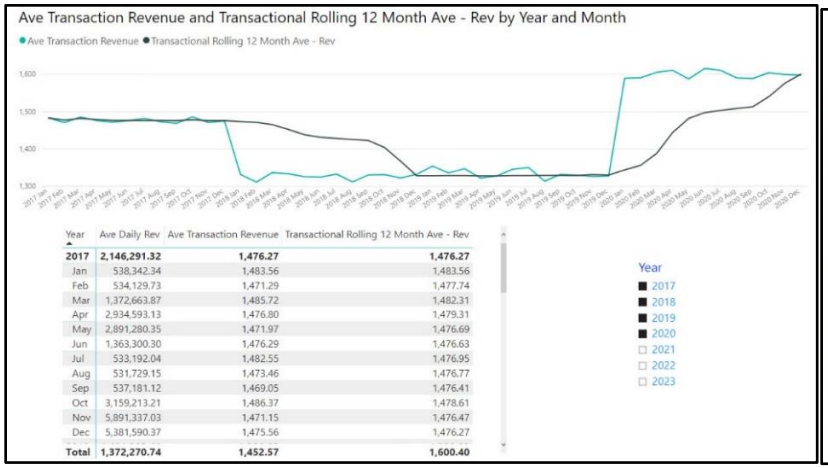
     Fellow YouTubers:
        1. Avi Singh - PowerBIPro: https://www.youtube.com/channel/UCRNmSv7mAPYiC0Y40TJijAw
        2. Guy In A Cube: https://www.youtube.com/channel/UCFp1vaKzpfvoGai0vE5VJ0w
        3. Curbal: https://www.youtube.com/channel/UCJ7UhloHSA4wAqPzyi6TOkw

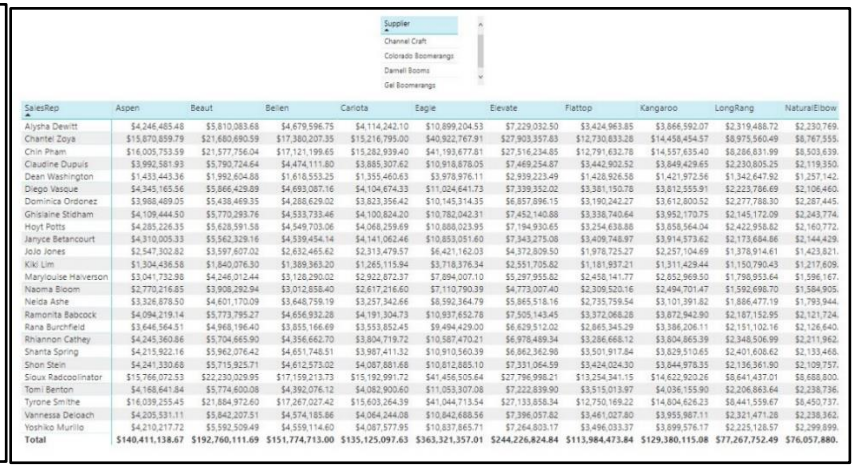
- DAX Formula Basics
    i. Types of DAX Formulas:
        1. **Calculated Columns**:
            i. Make Row-by-Row Calculations for a Table.
            ii. Used to Calculate: Numbers in a Fact Table, Attributes in Dimension Tables, Helper Columns.
        2. **Measures**:
            i. *Where we use Measures*: Calculation area of Visuals, In other DAX Formulas.
            ii. *Goal of Measures*: Measuring the health or performance of Entity. Examples: Total Revenue, Ave Daily Revenue
            iii. *Where we create Measures*: In Power BI Desktop, we select a Table in Data View or Report View, then in the Modeling Ribbon Tab, in the Calculations group we click the New Measure button.
        3. **Table Formulas**:
            i. Table formulas, like CALENDAR, CALCULATETABLE and FILTER deliver a table of values.
            ii. Examples:
                1. The DAX Table Function CALCULATETABLE can be used in a Measure.
                2. The DAX Table Function CALENDAR can be used to create a Date Table.
    ii. Conventions for DAX Formulas:
        1. For DAX Formulas, we must type our formula out in the Formula Bar.
        2. When you refer to a Field from a table (Column from a table), use the Table Name followed by the Field Name in Square Brackets. Example: **Table[Field Name]**
        3. When you use a Measure in a DAX Formula, use Measure Name in Square Brackets. Example: **[Measure]**. Why? So we know, when we see a Measure in a formula, that a hidden CALCULATE Function is present, and may have an impact on whether Context Transition occurs.
    iii. Tricks for creating DAX Formulas in Formula Bar:
        1. If you want to create the DAX Formula on different lines, use **Shift + Enter** to create a new line and **Tab** to indent the formula line.
        2. You can zoom in the Formula Bar by using Ctrl and Rolling the Wheel on your Mouse.
    iv. Row Context:
        1. In a Calculated Column or in an Iterator Function, the formula can see each row in the table and make a Row-By-Row Calculation.
    v. Filter Context:
        1. When a Measure is dropped into an Excel PivotTable or a Power BI Visualization, the Conditions/Criteria/Filters flow into the Measure and the Fact Table is filtered down to just the records that match the conditions of the calculation. This helps DAX Formulas to calculate quickly on Big Data.
        2. CALCULATE function can change the Filter Context, and it can perform Context Transition, converting Row Context into an equivalent Filter Context.
    vi. Explicit or Implicit Measures?
        1. Explicit = We Author Measure.
        2. Implicit Measure = Drag and drop into PivotTable or Power BI Visualization.
        3. **Do NOT use Implicit Measures. Use Explicit Measures.**
        4. In Power BI Desktop, if you use Implicit Measures and Publish Report to powerbi.com, then if you download into Excel, the Implicit Measures do not show up.
        5. Explicit Measures give us the control over the name, formatting and what functions are used in the Measure. Explicit Functions are easier because they allow us to use them over and over.

- List of Charts and Visualizations for your Dashboard (Review from prerequisite classes Busn 216 & 218) :
    i. What do Visualizations do?
        1. Visually portray quantitative data (number data).
        2. Give a **quick impression** of the number data.
        3. Create a picture that can communicate more quickly than just the numbers alone.
        4. Allow you to see **patterns or trends** that you may not be able to see looking at just numbers.
        5. Allows you to make relative comparisons more quickly than if you are using a table
    ii. Types of Visualizations:
        1. ***Tables***: Field Names in First Row and Records in Rows. Use when you want to see the individual numbers rather than a quick visual impression.
        2. ***Matrix***: Cross Tabulated Table with Row and Column Criteria and an intersecting calculation based on Row and Column Criteria.
        3. ***Column***: Use to compare differences across categories. Height of column conveys number.
        4. ***Bar***: Use to compare differences across categories. Length of bar conveys number.
        5. ***Stacked Column/Bar***: Good for displaying crosstabulation, emphasis on horizontal axis categories.
        6. ***Clustered Column/Bar***: Good for displaying crosstabulation, emphasis on legend categories.
        7. ***Histogram***: Chart used for counting numbers between a lower and upper limit. No gap between column indicates that there are no numbers between the upper and lower limit.
        8. ***Line***: Use to show trend for a number variable over a category such as time.
        9. ***Combination Chart***: Combine different chart types such as Column and Line.
        10. ***X-Y Scatter***: Used to show relationship between two number variables (x and y variables).
        11. ***Break Even Chart***: Specific type of X-Y Scatter Chart that shows the break-even cross over lines for Revenue and Costs.
        12. ***Bubble Chart***: Method of visualizing 3 variables in a 2-dimentional chart.
        13. ***Cards*** : Text box that can display Measures
        14. ***Maps***: Used for geographic data, like sales by zip code, sates, or country.
    iii. Effective Charts
        1. Include number data AND labels for the number data.
        2. No “Chart Junk”.
            i. Chart Junk means chart elements like:
                1. Unnecessary Repetition.
                2. Chart elements that do not contribute to the message.
                3. Chart elements that make the chart look busy:
                    i. Too many different colors
                    ii. Patterns that are distracting.
                4. 3-D effects that are not necessary and can be misleading
    iv. Dashboards:
        1. Dashboard:
            i. Data Visualization that presents useful information and metrics and will update automatically when new data become available.
            ii. Dashboards may contain: Tables, Charts, Data Validation, Pictures, Other visualizations of Data
        2. Effective Dashboards:
            i. Presents timely summary data, metrics or key performance indicators (KPI).
            ii. Metrics/KPIs should be useful for the user/decision maker
            iii. Dashboard should inform rather than overwhelm
            iv. Should call attention to unusual metrics/KPIs that require attention or are of interest

- Picture of Final Dashboards Created in this Power BI Project :

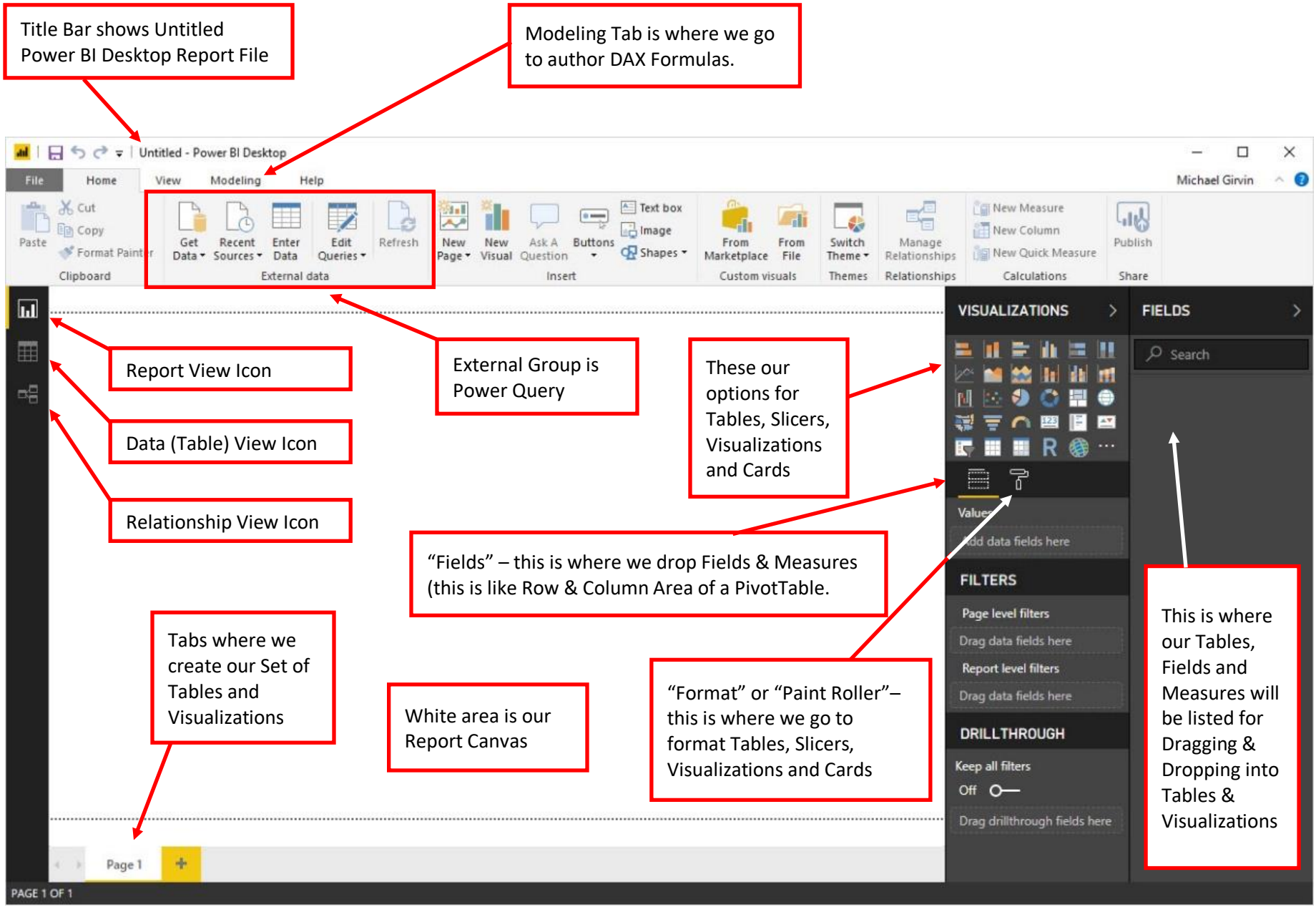




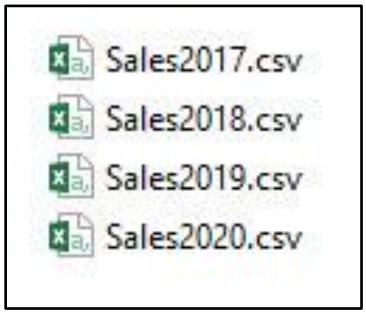


- Overriding Steps for our Project:
    1. Import, Clean Transform Data using Power Query.
    2. Create Date Table with DAX Formulas.
    3. Create Relationships.
    4. Create DAX Measures.
    5. Hide Items (Columns, Tables, Measures) that are needed as part of Data Modeling Steps, but not needed in Reporting View.
    6. Power Query to Refine Data Model by Editing Queries.
    7. Create Dashboards (Visualizations, Tables, Slicers) on as many Tabs as we want.
    8. Publish Power BI ".pbix" File.
    9. Refresh Data.

- Specific steps For Project :
  1) Open a blank Power BI File and the Power BI Window looks like this:



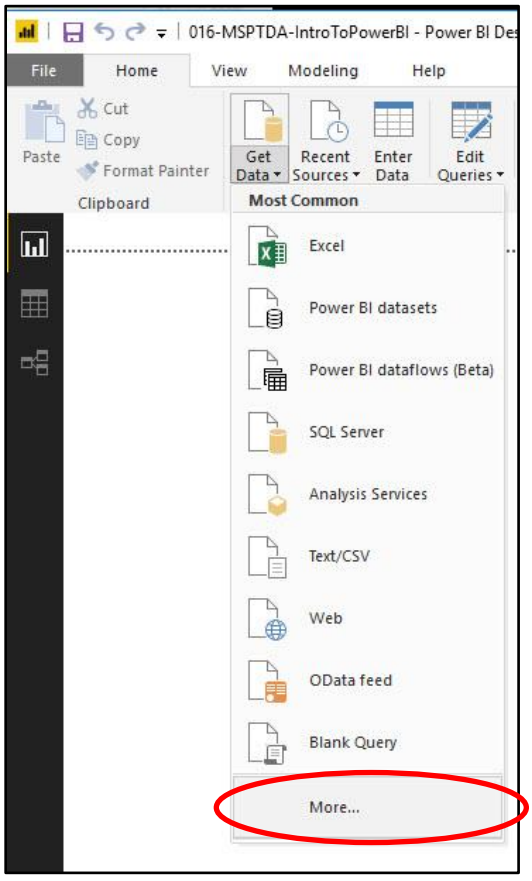
2) Save As keyboard in Power BI Desktop is the same as in Office: F12. We will name the Power BI File: "016-MSPTDA-IntroToPowerBI".
3) Import Multiple CSV Files. The Fact Table Data we need to import is coming from multiple CSV (Comma Separated Values) Files. We need to import these tables and append them into a single Fact Table. The initial files we need to import and append look like this:



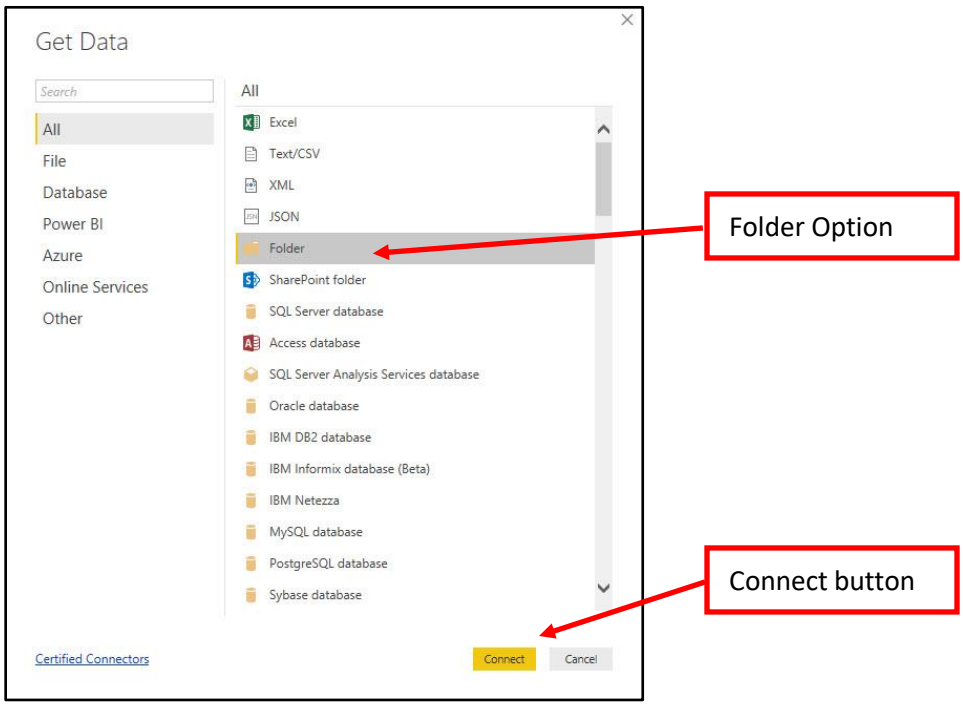
4) The individual CSV files look like this:

	A	B	C	D	E	F	G
1	ISO Date	ProductID	SalesRep	UnitsSold	Discount	COGSTotal	
2	20171228	1	2	7	0.055	131.0582355	
3	20170118	7	4	95	0.4375	675.4277925	
4	20171012	12	2	72	0.4375	708.2377406	
5	20171212	1	1	4	0	74.89042027	
6	20171109	6	15	6	0.055	60.91124705	
7	20170124	7	1	72	0.4375	511.9031691	
8	20171119	10	6	8	0.055	76.829026	
257498	20170419	1	2	72	0.4375	1348.027565	
257499	20170428	2	22	96	0.4375	953.3091909	
257500							

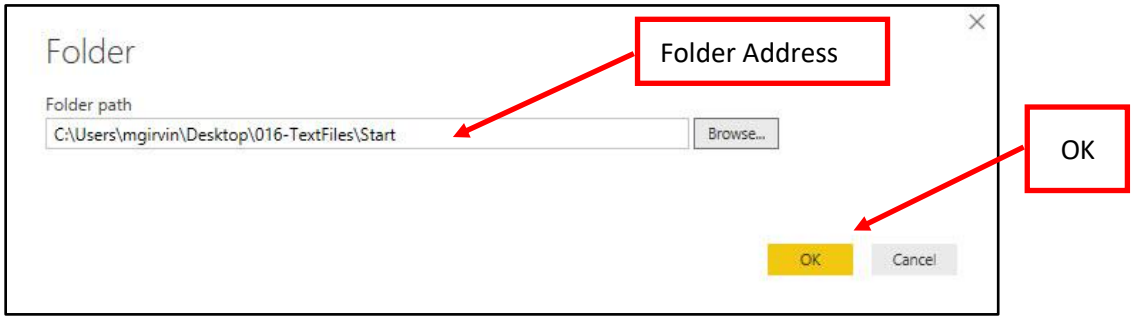
5) To start our Import & Transformation of the CSV files into a single Appended Proper Data Set, we go to the Home Ribbon Tab, then in the "External data" group, we click the Get Data dropdown arrow, then click on "More…"



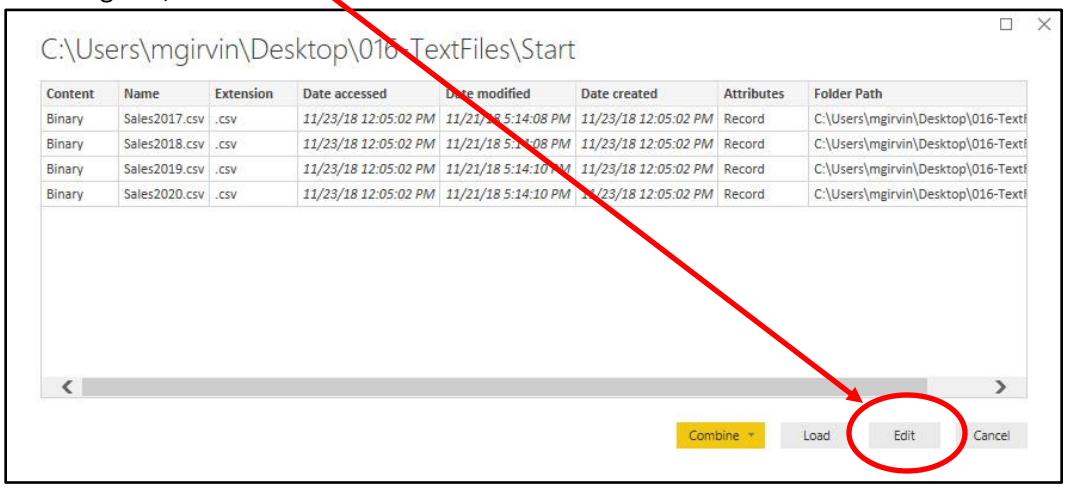
6) In the Get Data dialog box, select the option for “Folder”, then click the “Connect” button.



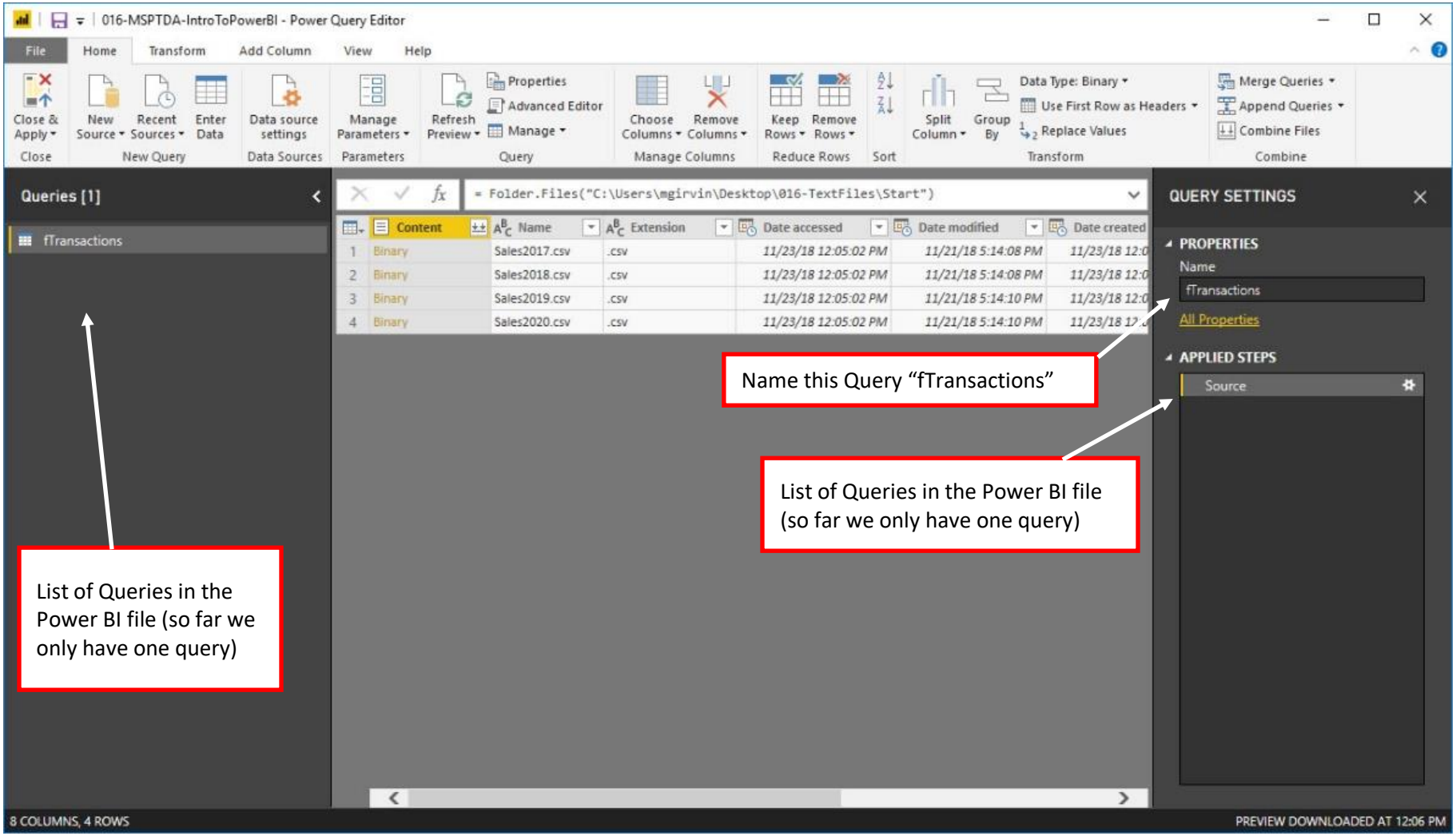
7) In the Folder dialog box, navigate to the correct “Start” Folder that contains the initial four CSV files, then click OK.



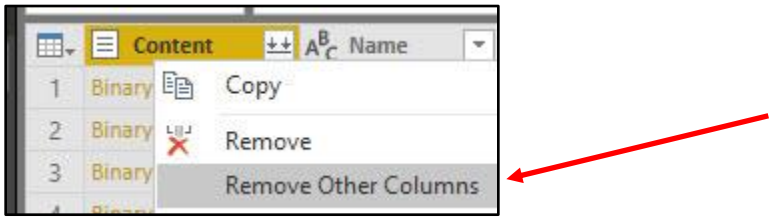
8) In the next dialog box, click “Edit”.



9) Power Query Editor Window. The below picture is what the Power Query window looks like in Power BI Desktop. On the left we see our list of queries we have created in our Power BI file (so far we only have one query). In the middle we see the list of files and attributes about each file. On the right, in the Query Settings Pane, we see the Query name and the list of Applied Steps (so far we only have one step). Name this Query "fTransactions". The name of the Query is also the name of the Appended Table that will be loaded into the Columnar Database in the Data Model in this Power BI Desktop File.



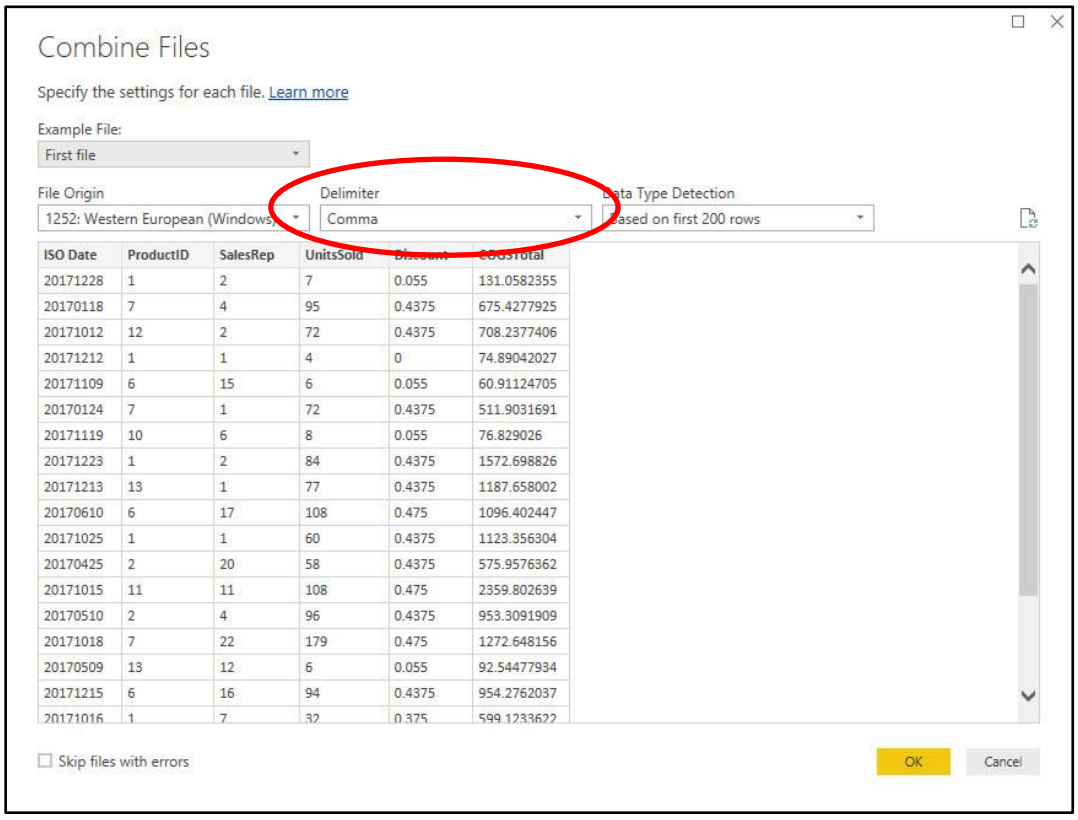
10) If we look at the Extensions Column in the previous picture and consider that the folder that contains these files will only have ".csv" files and that each of the extensions are in lowercase letters, we do not need to make any transformations on the Extensions column like we did earlier in this class.
11) Combine Files in Power Query. The Content Column is the column that contains a new ".csv" file for each row. We do not need any of the remining columns with file attributes, and so to remove all columns except for the Content Column, we right-click the Content Column and click on Remove Other Columns.



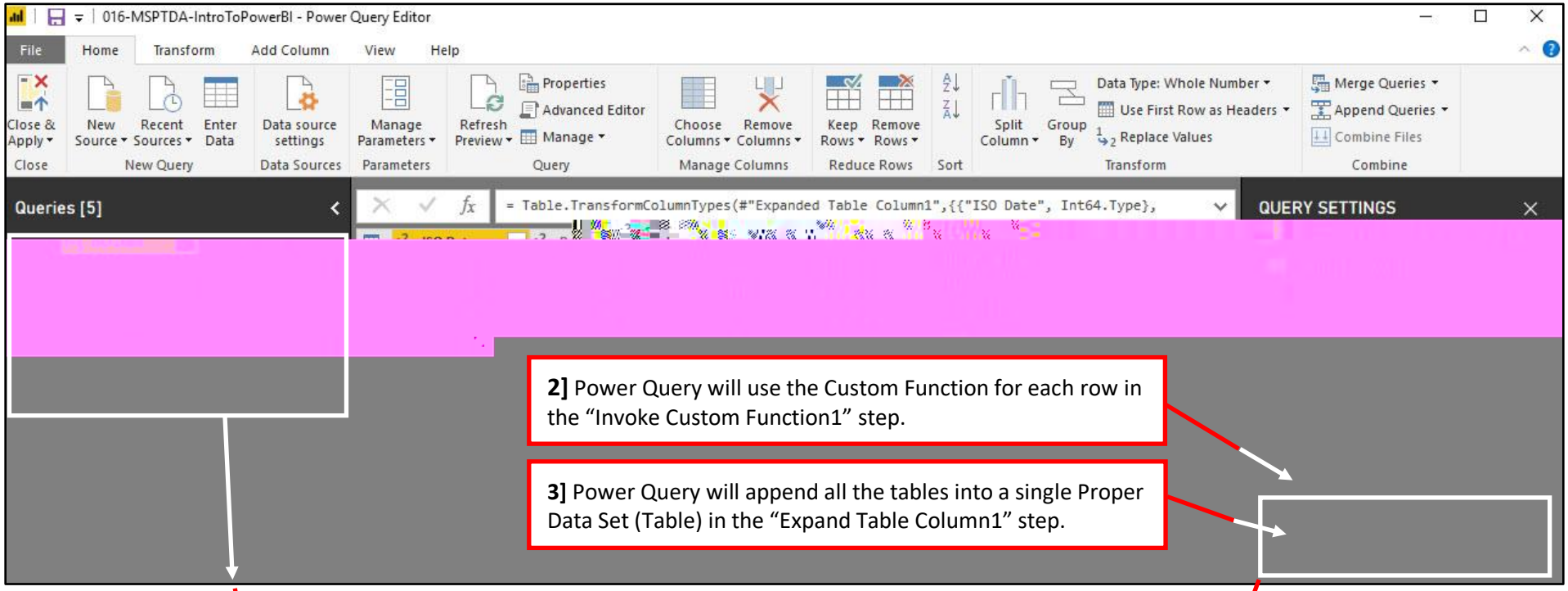
12) The Double Downward Pointing Arrow button at the Top-Right of the Content Column is the "Combine Files" button. When we click this button, Power Query will ask us what the Delimiter is, and then, after we indicate what the delimiter is, it will build a custom Power Query Function that will be used for each row in the Content column to extract the text file and convert it into a Proper Data Set. Power Query will then use the Custom Function on each row to create a Proper Data Set, and then it will append all the tables into a single Proper Data Set (Table).

Double Downward Pointing Arrow button at the Top-Right of the Content Column is the "Combine Files" button. When we click it, Power Query performs many steps and create a final Appended Table.

13) After clicking the "Combine Files" button, the next dialog box asks what Delimiter the files use. For us, the correct Delimiter is a comma. After you select the Delimiter, click the OK button.



14) After we select our Delimiter and click the OK button:
    1] Power Query will build a custom Power Query Function that can be used for each row in the Content column to extract the text file as a Proper Data Set
    2] Power Query will use the Custom Function for each row
    3] Power Query will append all the tables into a single Proper Data Set (Table)
    4] Power Query will perform a number of steps in order to append the tables, including the last step to add the correct Data Type to each column.

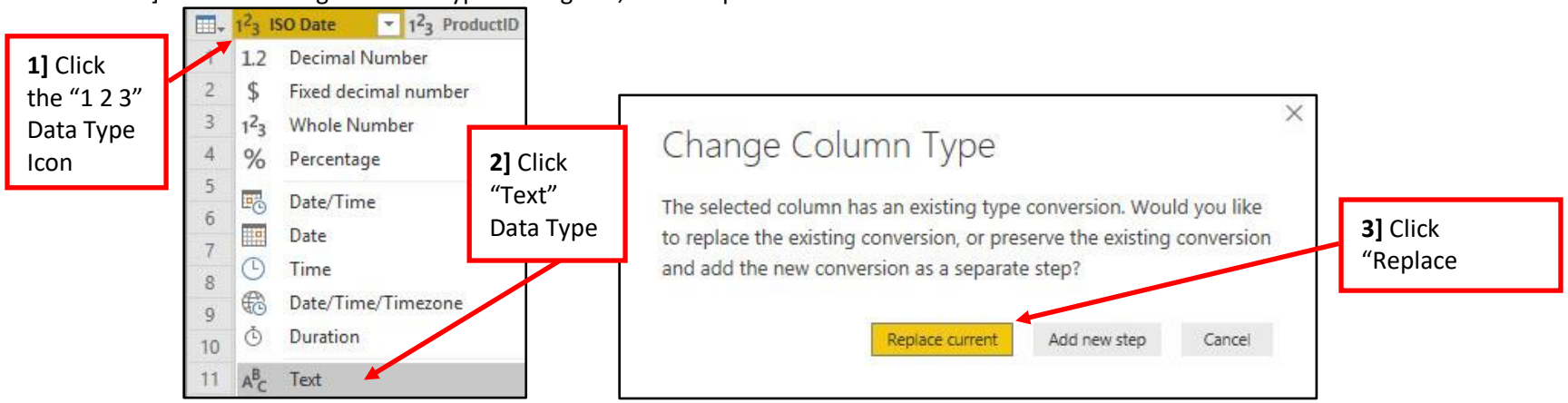


15) The top three rows of the final Appended fTransactions table is pictured below. But we need to perform two remaining data cleaning tasks for this table: we must 1) convert the ISO Date to a proper date and 2) then round the COGS Numbers.

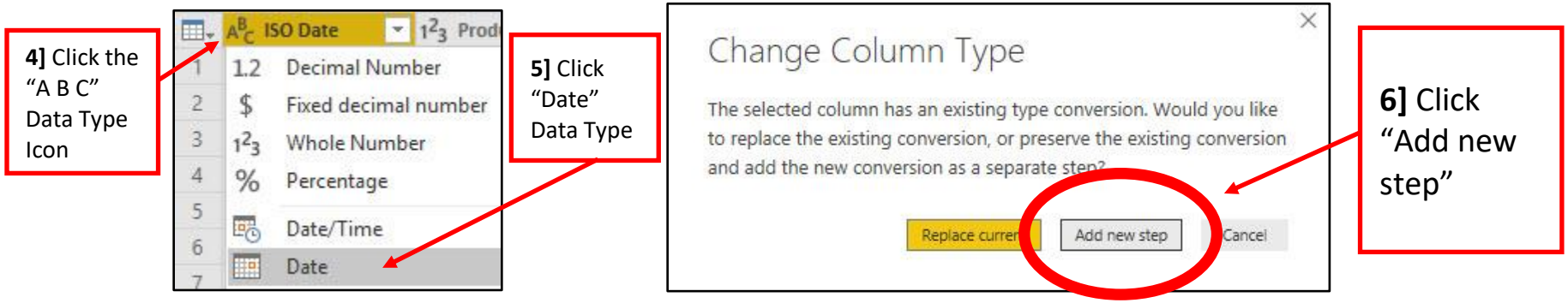
	ISO Date	ProductID	SalesRep	UnitsSold	Discount	COGSTotal
1	20171228	1	2	7	0.055	131.05823
2	20170118	7	4	95	0.4375	675.42775
3	20171012	12	2	72	0.4375	708.23774

16) Convert the ISO Dates to Proper Dates :

1] Click the "1 2 3" Data Type Icon in the ISO Date Column Header Upper-Left
2] From the dropdown click the Data Type "Text"
3] In the "Change Column Type" dialog box, click "Replace Current"

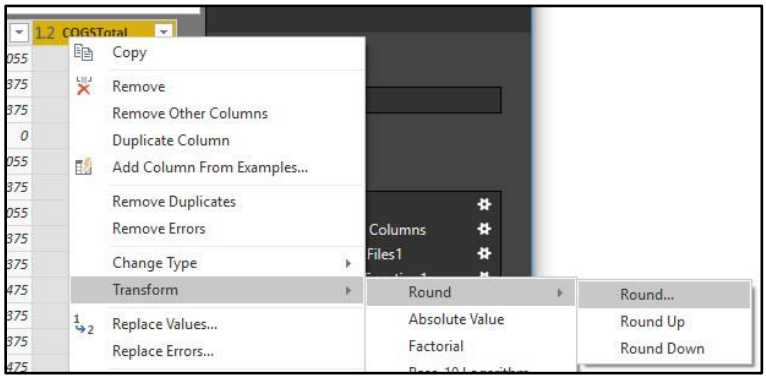


4] Click the "A B C" Data Type Icon in the ISO Date Column Header Upper-Left
5] From the dropdown click the Data Type "Date"
6] In the "Change Column Type" dialog box, click "Add New Step"

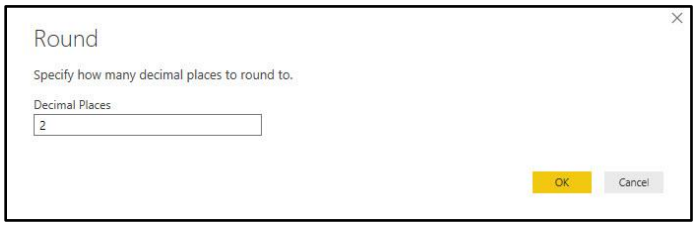


17) Double Click the “ISO Date” Field Name, Type new Field Name “Date”, then hit Enter.

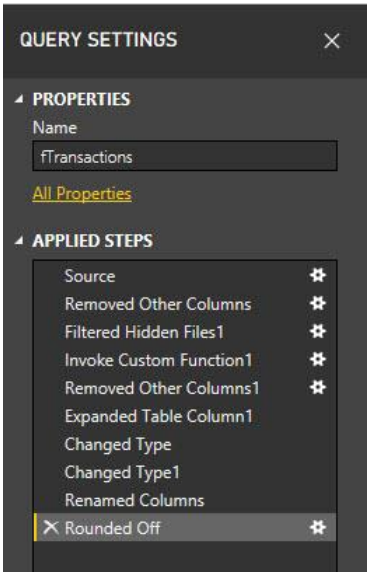
18) Round COGS Field: Right-click COGSTotal Field Name, point to “Transform”, Then “Round”, then click on “Round…”



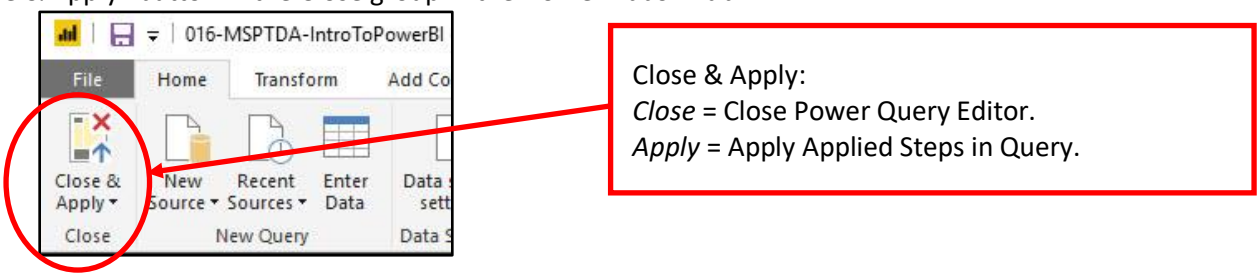
19) In the Round dialog box, type 2, then click OK.



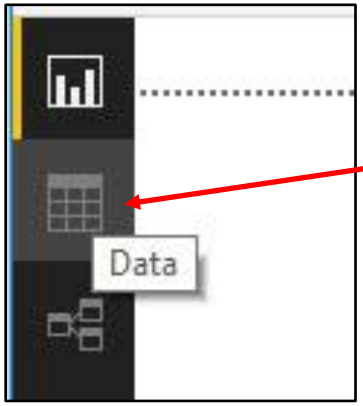
20) In the Power Query Applied Steps List, here are the Final Steps for our Import and Transformation:



21) Close & Apply. To Load the Appended Table to the Power BI Columnar Database in the Data Model, click the “Close & Apply” button in the Close group in the Home Ribbon Tab.



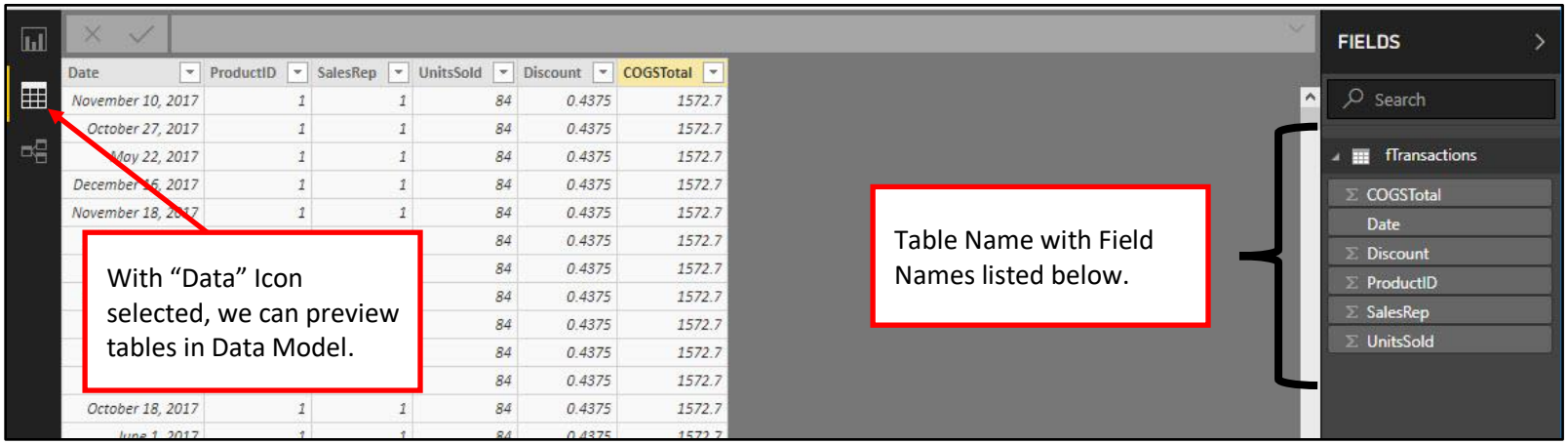
22) Data Icon. In the Power Query Editor Window, on the Far-Left Side, click the “Data” Icon Button (Really this Icon should be called “Tables Icon”).



In the Power BI Window, on the Far-Left Side, click the “Data” Icon Button.

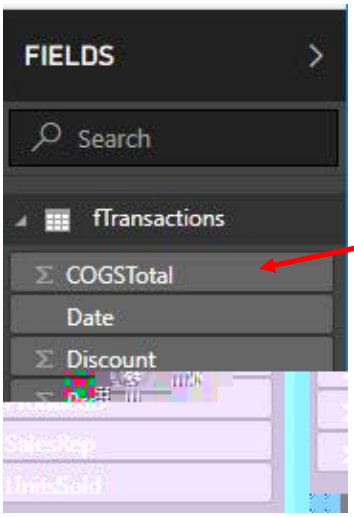
This Button should have been named “Tables” because this is the button we click to get a preview of the tables that are loaded to the Power BI - Data Model - Columnar Database.

23) This is what the Loaded, Appended fTransactions Table (Proper data Set) looks like:



24) Important Notes about Date & Number Fields in Power BI Tables.

1] Below is a picture of the fTransactions Table and Field Names:



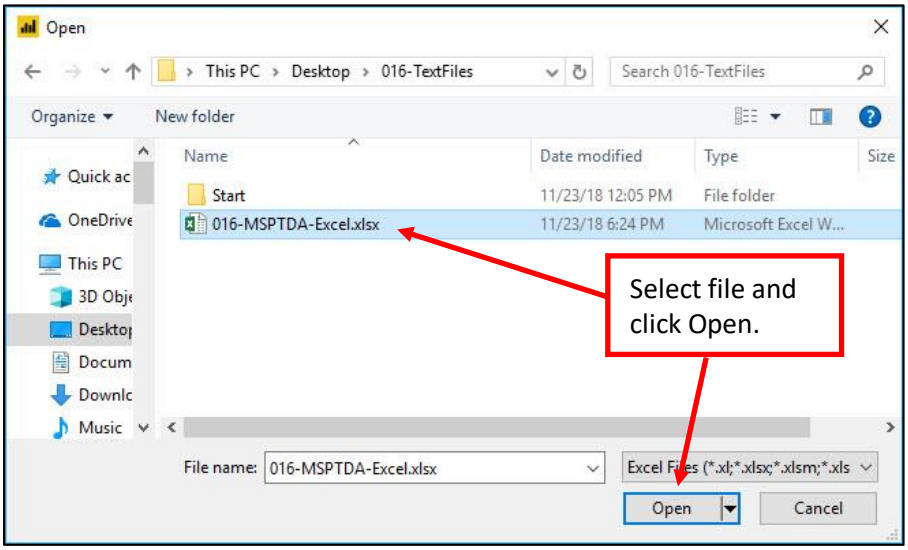
1] The Sigma Icon (SUM Icon) is mis-leading because it looks similar to the Icon for a Measure in Power Pivot.
2] In Power BI Desktop, this Sigma Icon means that there are numbers in these fields.
3] These Number Fields should NOT be dragged to Reports for summary calculations because it will create Implicit Measures.

2] Taking a close-up look of the Table Name and Field Names, we notice:

1. A Table Icon appears to the left of the Table Name.
2. The Date Field does not show an Icon to the Left of the Field Name.
    i. Date Fields from Fact Tables should not be dragged into Tables and Visuals because each time you do this a Hidden Date Table is created by Power BI Desktop.
    ii. Hidden Date Tables are NOT efficient because every time you drag a Date Field from Fact Table to Tables and Visuals you run the risk of creating multiple unnecessary Date Tables.
    iii. As we will see (and as we saw last video), it is much more efficient to create our own specific Date table with our own specific date logic; and more importantly, here in Power BI Desktop, by creating a single Date table and Marking it as a Date table, we avoid the build-up of multiple unnecessary Date tables.
    iv. Later we will have to **hide** any Date Fields in the Fact Table from the Report View, just as we hid the Date Field in the Fact Table in Excel Power Pivot.
    v. Rule for Date Fields from Fact Tables: We do not want to drag and drop Date Fields from Fact Tables into Tables or Visualizations.
3. The Fields that contain Numbers have a misleading Sigma Icon (SUM Icon) on the left of the Field Name. In Power BI Desktop this icon means that there are numbers in the column. This Icon is misleading for two reasons:
    i. The first reason the Sigma Icon is misleading is because it is similar to the Icon used in Excel Power Pivot for Measures. This means that the Sigma icon in Excel Power Pivot indicates a “Measure”, but in Power BI Desktop it indicates that the Field contains Numbers.
    ii. The second reason it is misleading is because it indirectly suggests that a Power BI Report Builder should drag these columns into the Tables and Visualizations that are being built. If we drag theses number Fields into Tables and Visualizations, an Implicit Measure will be created. As we Studied in last video, Implicit Measures are inefficient.
    iii. Later we will have to hide the Fact Table Number Fields from the Report View, just as we hide the Fact Table Number Fields from Client View in Excel Power Pivot.
    iv. Rule for Number Fields with Sigma Icons: We do not want to drag and drop Number Fields from Fact Tables into Tables or Visualizations.

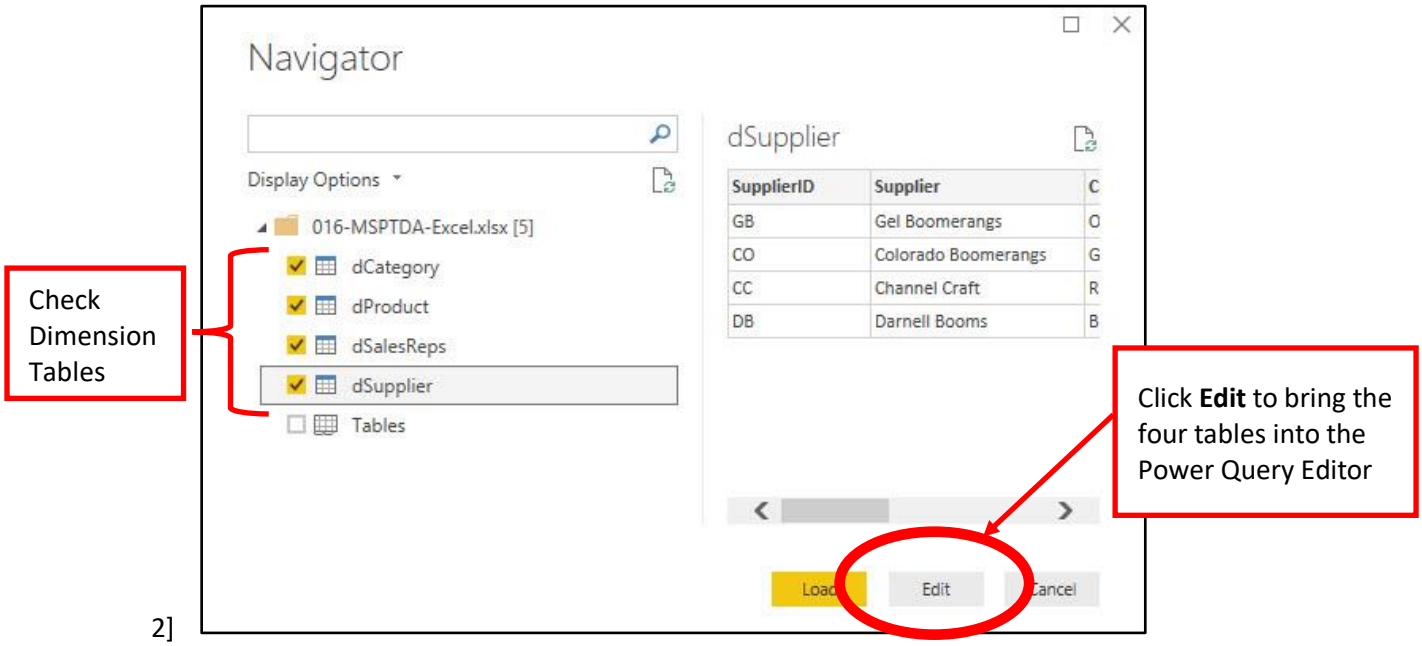
26) To import the Dimension Tables from the Excel Workbook, we go to the Power BI Desktop Home Ribbon Tab, then in the "External data" group, we click the Get Data dropdown arrow, then click on the Excel option (first item in list).

27) In the Open dialog box, select the Excel Workbook file named "016-MSPTDA-Excel.xlsx". Click Open.



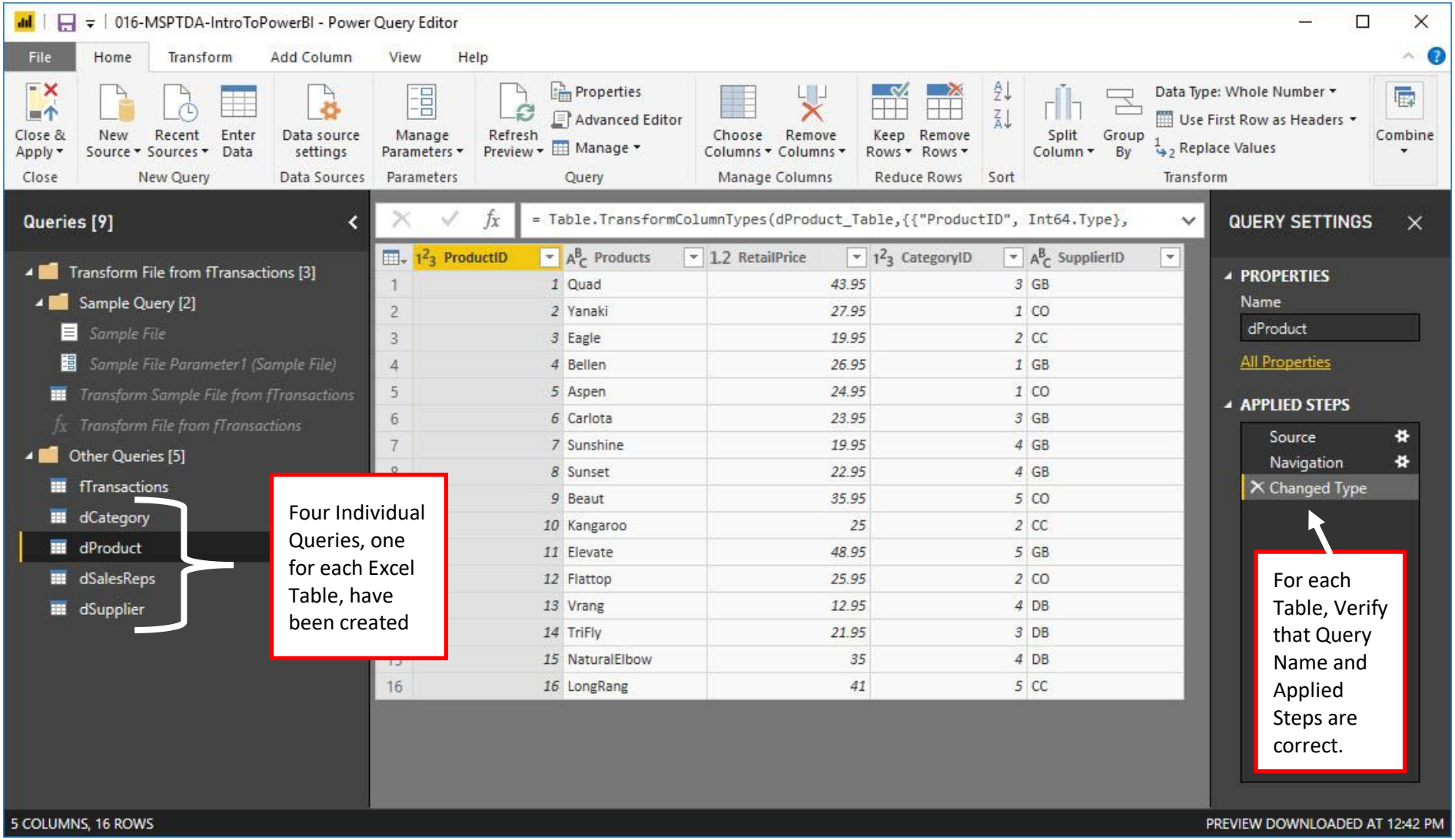
28) In the Navigator dialog box, check the Excel Objects that you want. For this project check the four Dimension Tables. Then click Edit to bring the four tables into the Power Query Editor.

1] Note: There are multiple potential Excel Objects that can appear in this Navigator dialog box including Excel Table Objects and Sheet Objects. You will have to be aware of what objects are in the Excel Workbook File and select the objects that you want to import accordingly.



2]

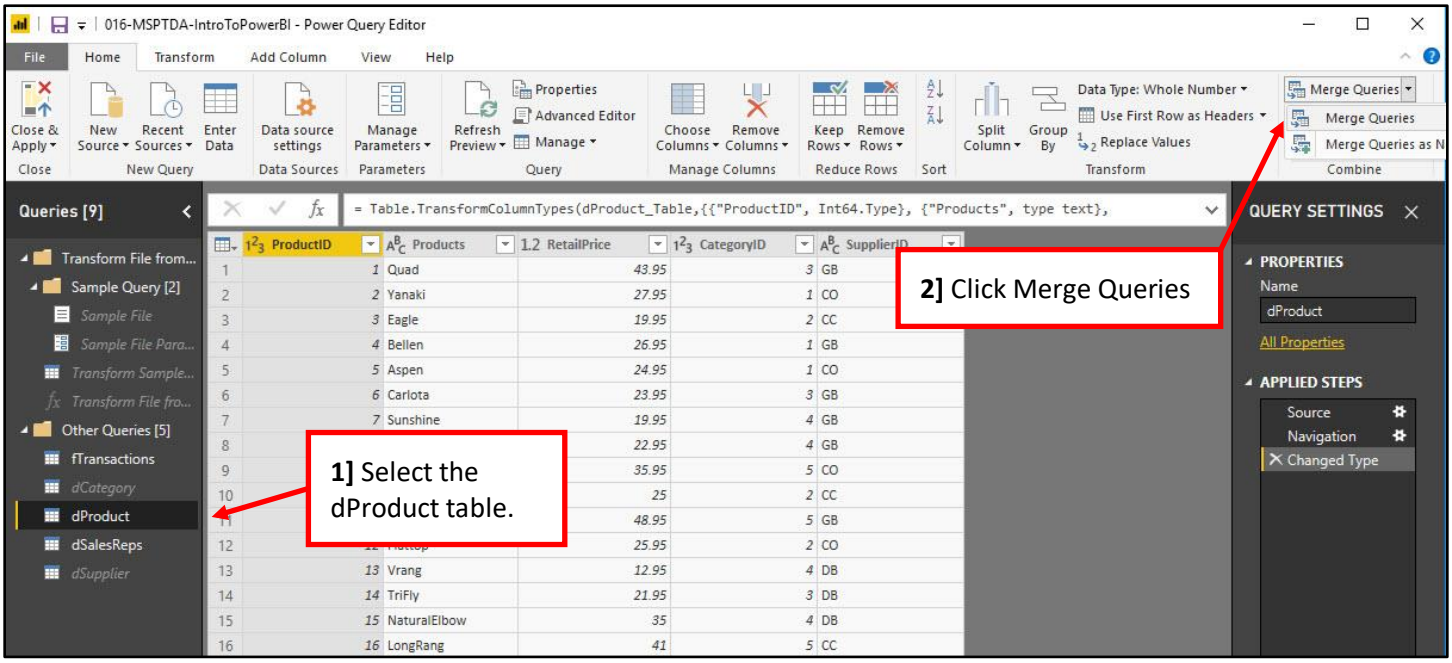
29) **Verify Dimension Tables imported correctly & have the correct Data Type for each Field.** In the Power Query Editor, we can see that there are four new Queries listed in the Queries Pane on the Left: dCategory, dProduct (the one selected in the picture), dSalesRep and dSupplier. On the Right in the Query Settings Pane, we can see the name of the Query and Applied Steps for the dProduct Query. One by one, you MUST check each Query to verify that the Query Name is adequate, and that the Applied Steps have added the correct Data Type for each column in each table. For each Query name, the inherited Excel Table name is sufficient. For example, "dProduct" is the inherited Excel Table name and that is a good name for the Query and for the resulting table that will be imported into the Data Model.



30) Merge Snow Flake Dimension Tables into dProduct Table. Follow these steps to Merge the dCategory and dSupplier tables into the d Product table:

1] Select the dProduct table in the Query Pane

2] In the Combine group, in the Home Ribbon Tab, click the Merge Queries dropdown, then from the dropdown menu click on Merge Queries.

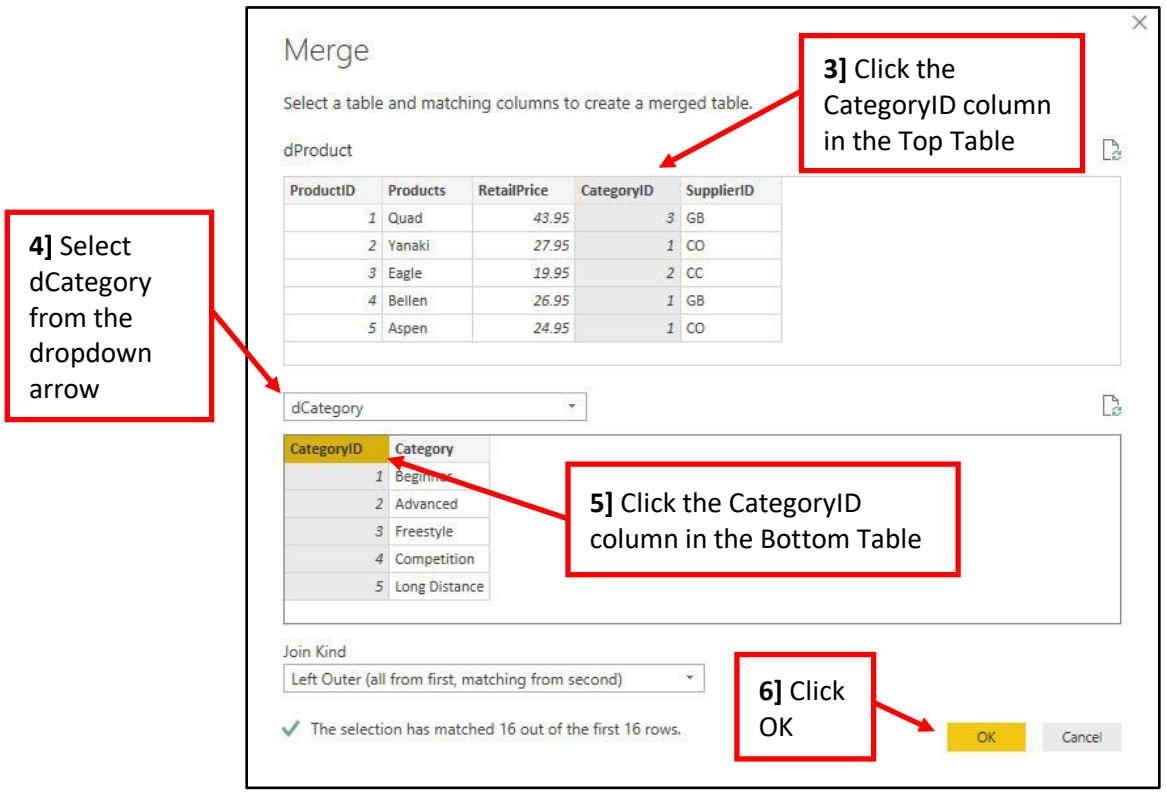


3] In the Merge dialog box click the CategoryID column in the Top Table

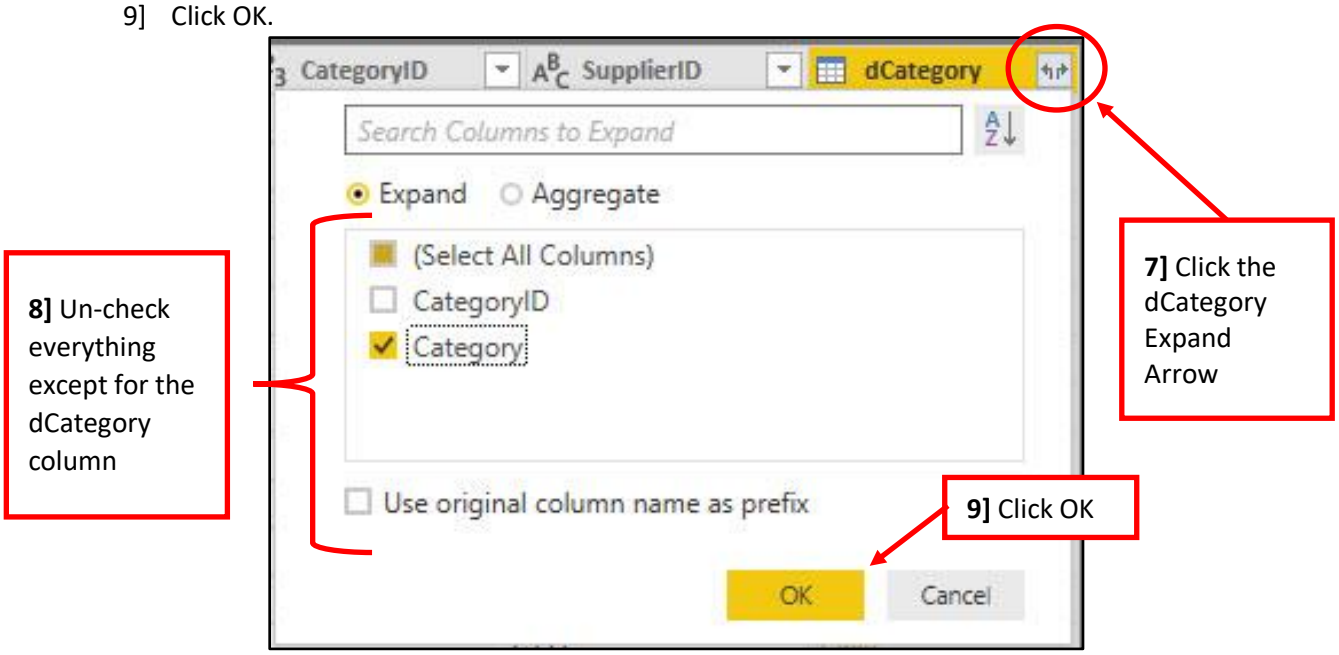
4] For the Bottom Table select dCategory from the dropdown arrow.

5] Click on the CategoryID column in the Bottom Table.

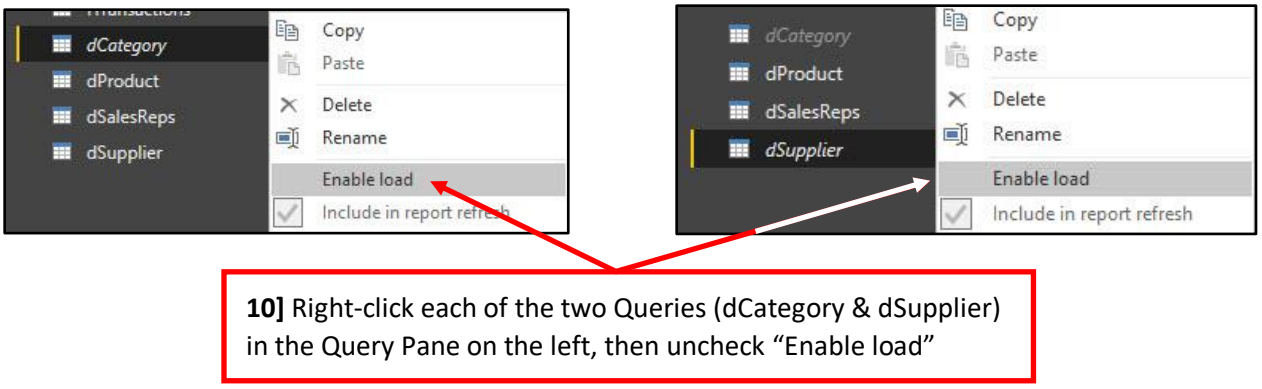
6] Click OK.



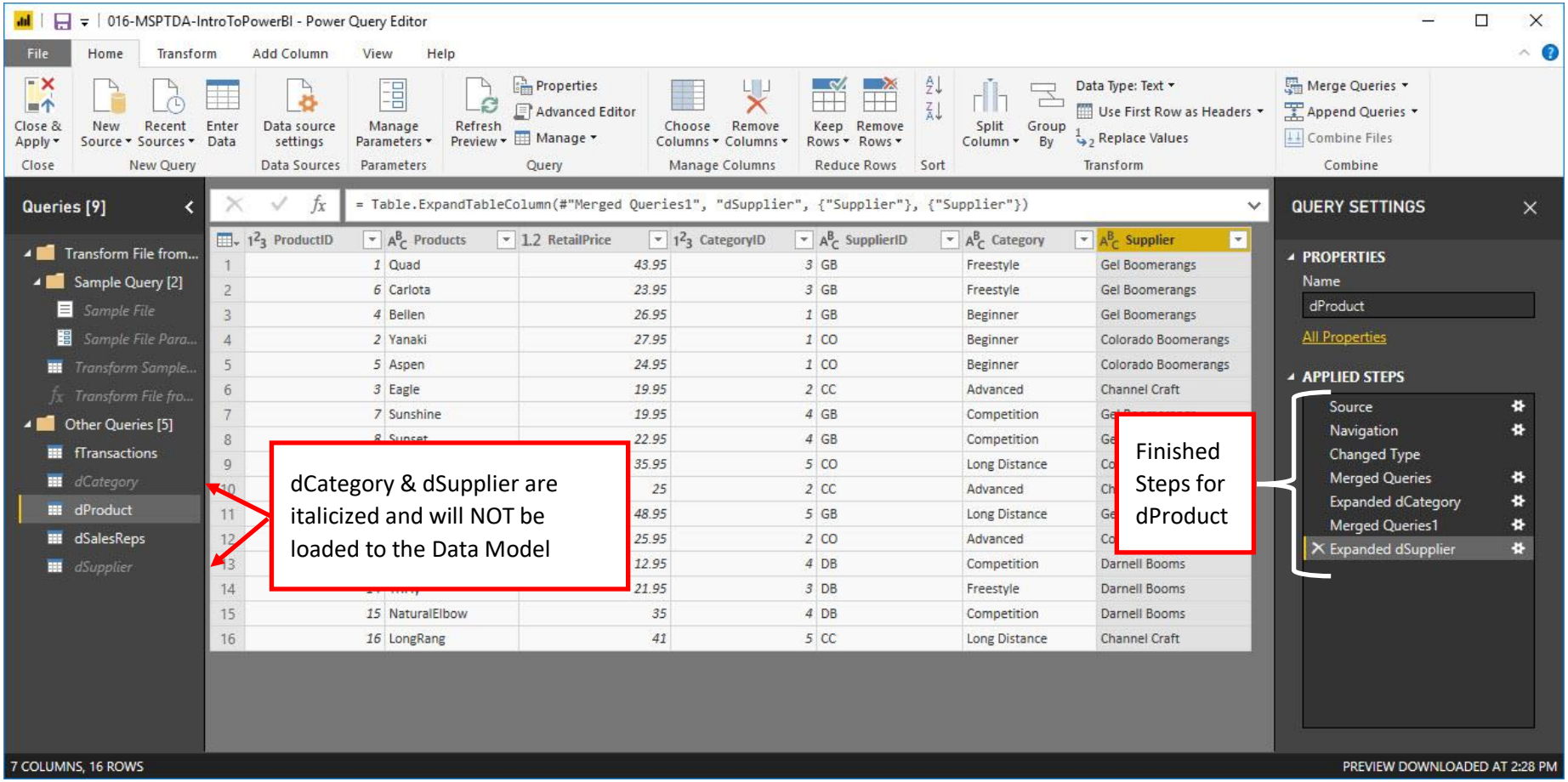
7] The Merge process added a new column named dCategory. Click the dCategory Expand Arrow.
8] Un-check everything except for the dCategory column.
9] Click OK.



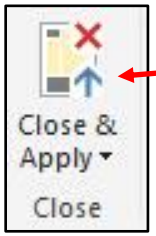
10] Because we ONLY need the dCategory & dSupplier tables for part of our Power Query Transformation, and because we do NOT want to load the two tables to the Data Model, we must right-click each of the two Queries (dCategory & dSupplier) in the Query Pane on the left, then uncheck “Enable load”.



31) Below is a picture of the Transformed dProduct table. On the left in the Queries Pane, we can see that there is one Fact Table (fTransactions), two Dimension Tables (dProduct & dSalesRep) and the dCategory & dSupplier, which are italicized and will NOT be loaded to the Data Model. On the right you can see the completed Applied Steps for the dProduct table.



32) To Close and Load the Dimension Tables to the Columnar Database in the Data Model, click the Close and Apply button in the Close group in the Power Query Home Ribbon Tab.

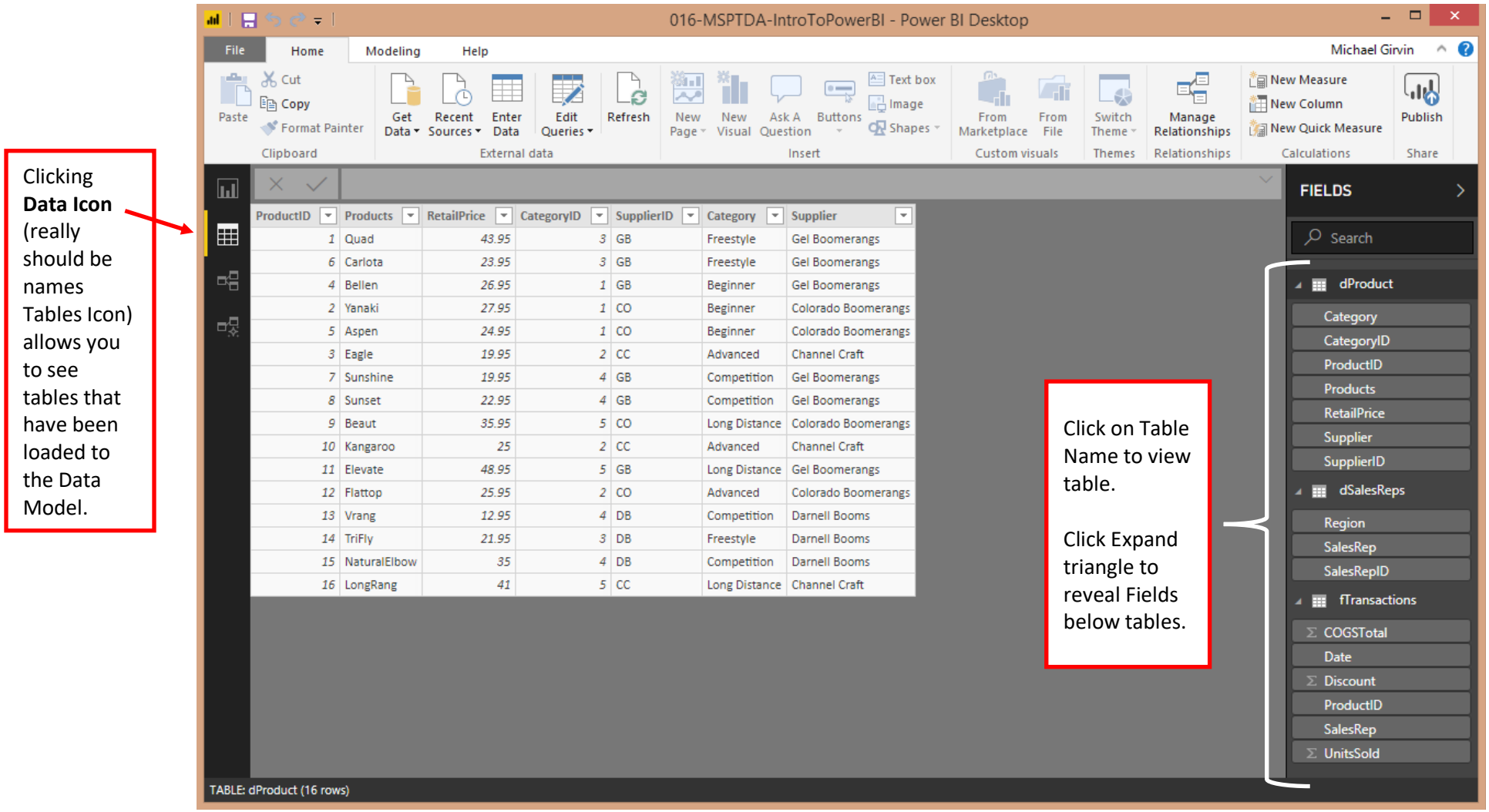


Clicking Close & Apply will apply Query changes, load tables to Data Model and close the Power Query window.

33) View Tables in the Power BI Desktop window :

1] To View the three tables loaded to the Data Model, click the Data Icon on the Far-Left in the Power BI Desktop window.

2] In the Fields Pane on the right, you can select tables to view.

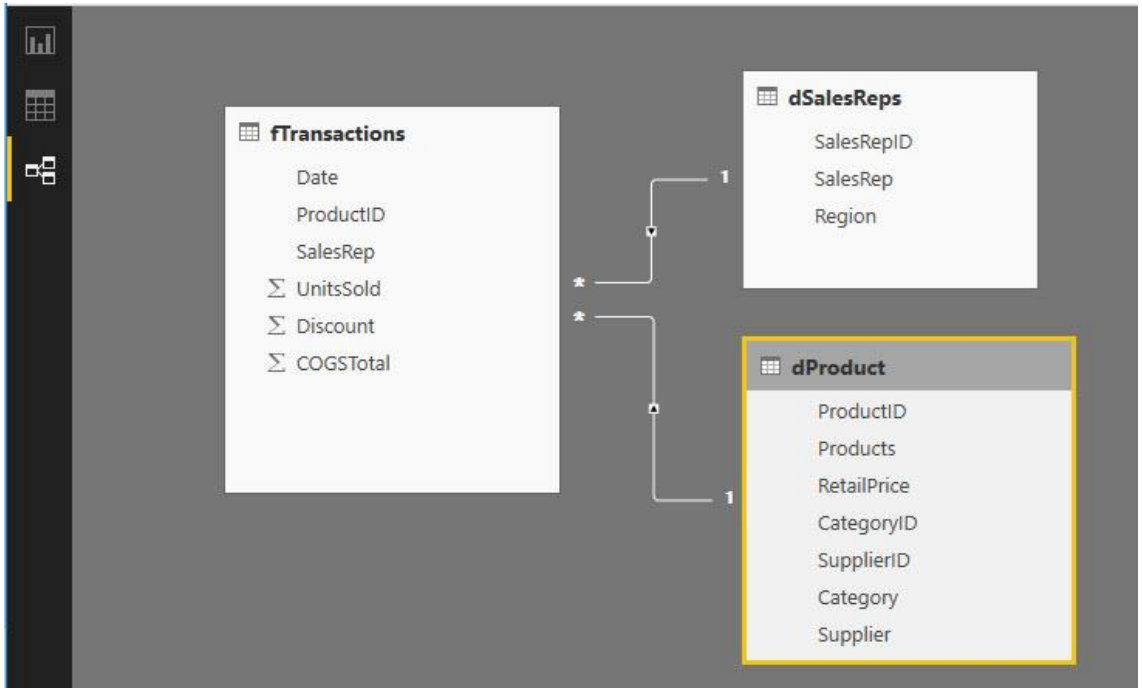


34) Relationships. To create or view Relationships between tables, click the Relationships Icon on the Far-Left.

1] Why Relationships exist in Data Model:

1. Relationships allow us to see Multiple Tables in Report View so that we can:
    i. Drag and drop Fields from Dimension Tables as Filters for the Measures in our Reports and Visualizations.
    ii. Drag and Drop Measures from our Fact tables into Reports and Visualizations.
2. Relationships allow DAX Formulas to access a table on the other side of a Relationship, whether we need to retrieve a single item from the One-Side or many items from the Many-Side.
3. A crucial reason that Relationships exist are so they can Transfer a Filter from a Dimension Table to a Fact Table to help facilitate faster DAX Measure Calculation Time.

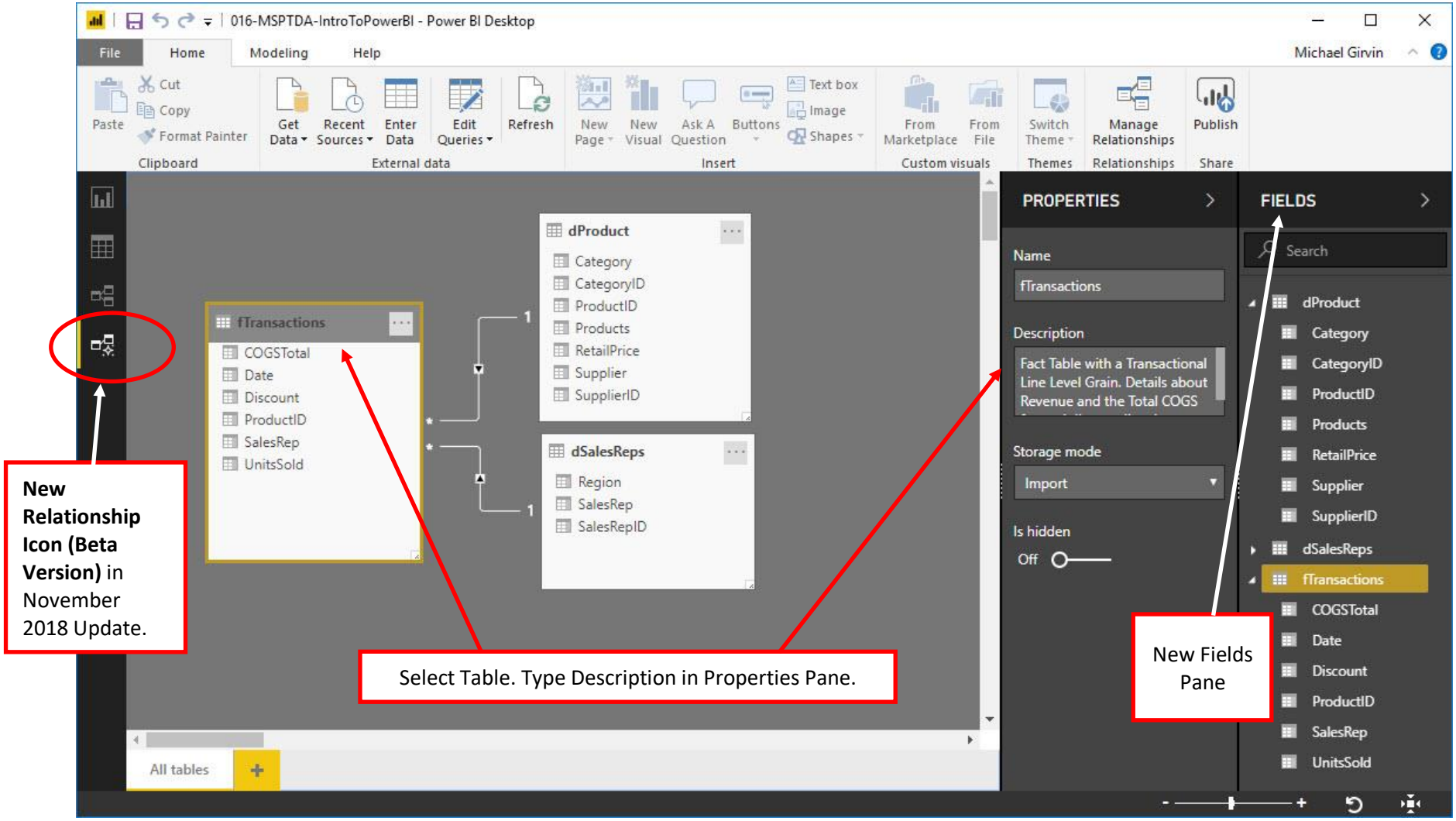
Clicking **Relationship Icon** allows you to view Relationships between tables, or to Drag and Drop Fields to create Relationships.



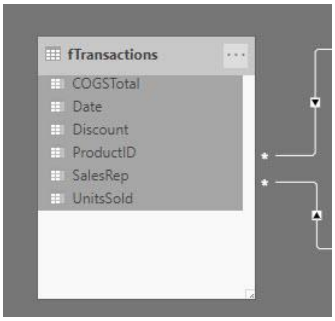
35) **New Relationships View with Properties & Better Selection Capability**. If you have the November 2018 Update (or later version), there are some additional features in Relationship View. Below is a picture of the Beta Version (Nov 2018). The new features and abilities are:

1] Properties Pane where you can type a Description and change other properties. These properties help to refine and document the Data Model.

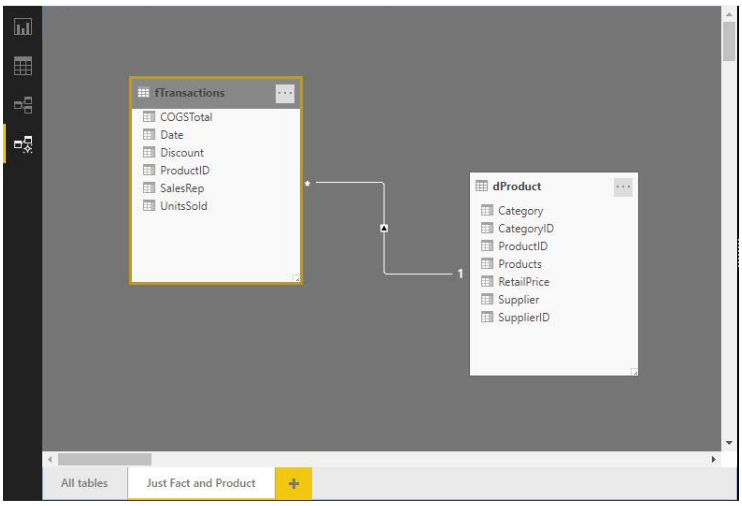
2] Fields Pane which shows list of Table and Field names.



3] As seen in the picture below, this new Relationships View, allows us to easily select multiple Fields and then simultaneously preform actions on the columns. (Selecting multiple columns was not possible in earlier versions of Power BI Desktop). Later we will use this feature to Hide Columns. In the below picture we were able to select all the columns in the fTransactions table.



4] Another new feature in the new Relationships View is the Plus Tab Icon at the bottom, where we can create a view of just part of the Data Model by dragging and dropping tables into the gray area. This can be useful in complicated Data Models when you want to examine a particular section of a Large Data Model.



36) In a Data Model we need a Date Table (Calendar Table) for these reasons :

1. Date Table Provides Date Attributes that we can use as Filters for our Reports.
    i. We want DAX Formulas to work on (iterate over) a unique list in the first column of the Dimension Table, rather than a full column in a Fact Table.
    ii. We need the benefit of having a Dimension Table that can filter the Fact Table (so DAX Formulas can calculate over a smaller set of numbers).
    iii. Time Intelligence Functions (like TOTALYTD, SAMEPERIODLASTYEAR and DATEADD) require a Date Table.
2. Date Dimension Table must have these characteristics:
    i. Unique List in First Column for every possible day for every given year in the Date Column in the Fact Table (so Time Intelligence Functions work correctly, like for DATEADD function).
    ii. Must Mark as Date Table (So Automatic Grouping Feature in Data Model will be disabled).

37) List of DAX Formulas used to create Date Table :

1. DAX Calculated Table:

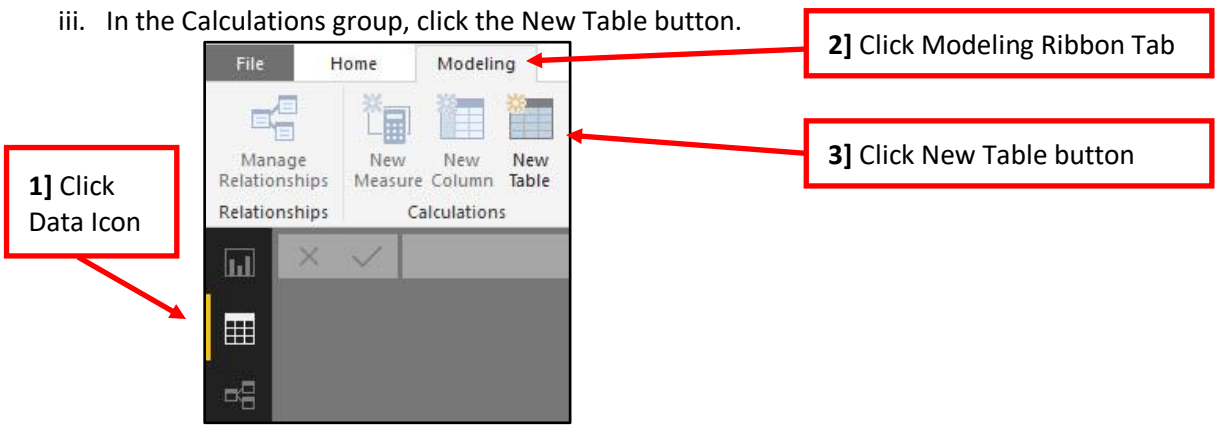
```
dDate =
        CALENDAR (
          DATE ( YEAR ( MIN ( fTransactions[Date] ) ), 1, 1 ),
          DATE ( YEAR ( MAX ( fTransactions[Date] ) ), 12, 31 )
        )
```

2. DAX Calculated Columns:

```
MonthNumber = MONTH(dDate[Date])
or
MonthNumber = dDate[Date].[MonthNo]

Month =
FORMAT(dDate[Date],"mmm")

Year = YEAR(dDate[Date])
or
Year = dDate[Date].[Year]

Quarter = ROUNDUP(dDate[MonthNumber]/3,0)
or
Quarter = dDate[Date].[QuarterNo]

Fiscal Quarter =
        IF(dDate[Quarter]=1,4,dDate[Quarter]-1)

Fiscal Year =
        IF (dDate[Quarter]=1,dDate[Year]-1,dDate[Year])

Fiscal Period =
        "Q" & dDate[Fiscal Quarter]
          & " - "
          & RIGHT ( dDate[Fiscal Year], 2 )

SortFiscalPeriod =
        dDate[Fiscal Year] * 10
          + dDate[Fiscal Quarter]
```

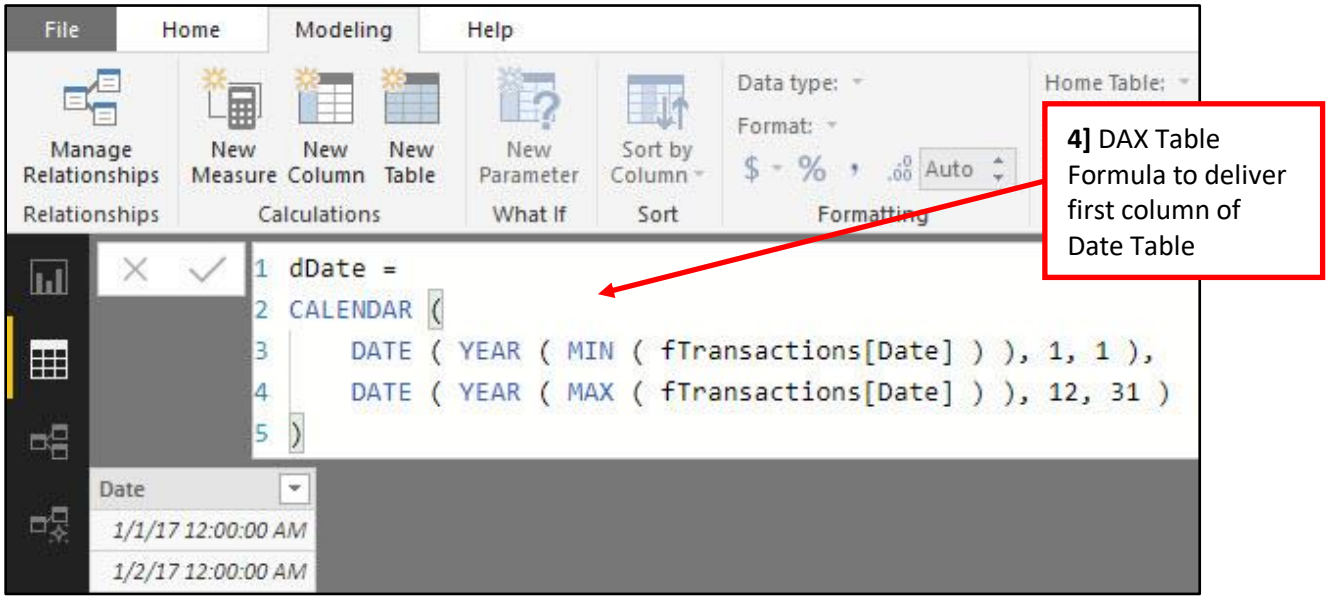
38) Steps to create Date Table using CALENDAR DAX Table Function & Calculated Columns :

i. On Left side of Power BI Desktop Window, click on Data Icon.
ii. Click on the Modeling Ribbon Tab.
iii. In the Calculations group, click the New Table button.

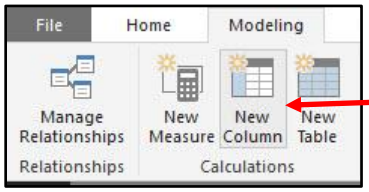


iv. In the Formula Bar we can create a Dynamic Date Table that will automatically create a column of dates for every possible day for every given year in the Fact Table Date Column with the formula shown in the below picture. The explanation of how the formulas works is listed here:

1. Note:
    i. If you want to create the DAX Formula on different lines, use **Shift + Enter** to create a new line and **Tab** to indent the formula line.
    ii. You can zoom in the Formula Bar by using Ctrl and Rolling the Wheel on your Mouse.
2. The DAX CALENDAR function is a Table Function that requires a Start Date and an End Date and will deliver a unique list of all dates between the Start and End Date. This Column of dates will be our Primary Key that we can use to build a One-to-Many Relationship with the Fact Table Date Column.
3. In the first argument of the CALENDAR function we need the first day of the year for the earliest date in the fTransactions Date Column. To achieve this goal, in the first argument of CALENDAR we use the formula element: **DATE(YEAR(MIN(fTransactions[Date])),1,1)**
    i. The MIN Function picks out the smallest date in the Date Column of the fTransactions table.
    ii. The YEAR Function extracts the Year of the smallest date and delivers it to the first argument of the DATE Function.
    iii. The DATE Function uses the year delivered by the YEAR function and then with the Month = 1 and Day = 1, DATE delivers the correct Start Date date to the CALENDAR function.
4. In the second argument of the CALENDAR function we need the last day of the year for the latest date in the fTransactions Date Column. To achieve this goal, in the second argument of CALENDAR we use the formula element: **DATE(YEAR(MAX(fTransactions[Date])),12,31)**
    i. The MAX Function picks out the latest date in the Date Column of the fTransactions table.
    ii. The YEAR Function extracts the Year of the largest date and delivers it to the first argument of the DATE Function.
    iii. The DATE Function uses the year delivered by the YEAR function and then with the Month = 12 and Day = 31, DATE Delivers the correct End Date to the CALENDAR function.
5. The Final Column of Dates is delivered as a Table to the Data Model.

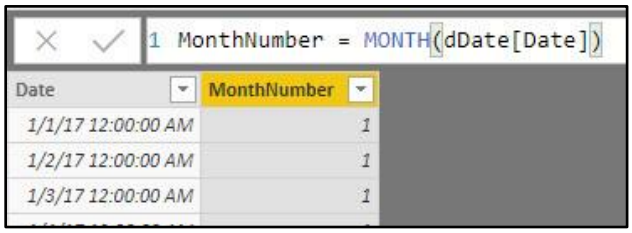


v. To add a Calculated Columns with date attributes to our Date Table, click the New Column button in the Calculations group in the Modeling Ribbon tab.



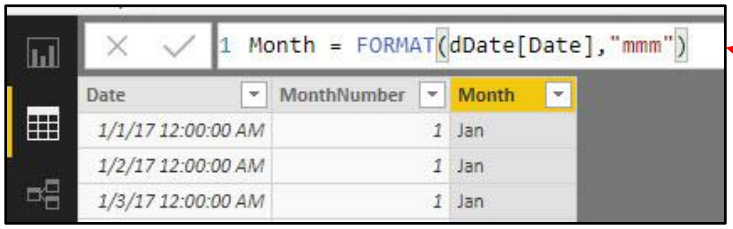
5] New DAX Calculated Column

vi. In the Formula Bar type the formula for Month Number, as seen below. This column is needed so that we can use it as the Sort Column for the Month Name Column and because it will be part of the calculation for Quarters. Later we will hide this column so that it does not appear in the Reporting interface.



6] Calculated Column for Month Number

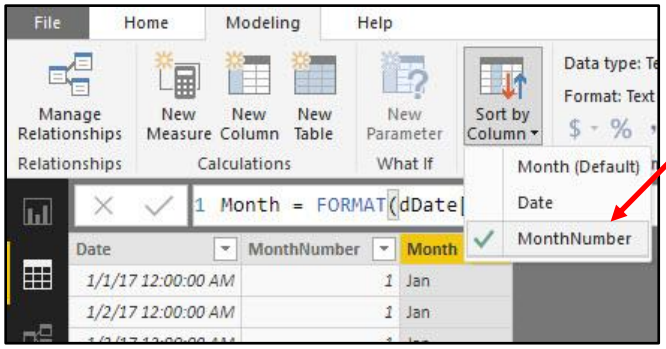
vii. In the Formula Bar type the formula for Month, as seen below. This column uses the FORMAT DAX function. It is similar to the Excel Worksheet function, TEXT. The first argument contains the Date Column and the second argument contains the Custom Number Format for the month name. We used "mmm" to show the three-letter abbreviation for month.



7] Calculated Column for Month Name

viii. Because the Month Three-Letter Abbreviations would NOT sort according to the actual months in a year (Jan, Feb, Mar…), but instead would sort alphabetically if we were to drop this Field into a Report, we need to sort the Month Column by the Month Number Column. To do this, select the Month Column, then in the Modeling Ribbon Tab, Sort group, click the Sort by Column dropdown arrow, then check the MonthNumber Column.

1. The reason a date attribute field like "Month Names in a Year" will not sort correctly in reports is because the default sorting order in the Reporting features (in both Excel Power Pivot and Power BI) is to sort alphabetically. If you are familiar with Excel and PivotTables, you know that Month Names do sort by actual months in a year (Jan, Feb, Mar…); but this is because of a behind-the-scenes Excel Custom List. Because there is no Custom List for Month Names in Power BI, we must manually use "Sort by Column" to get the Months Names to sort by the Month Number Column, which will sort correctly given the default alphabetic sorting order.



8] We MUST Sort Month Column by MonthNumber Column

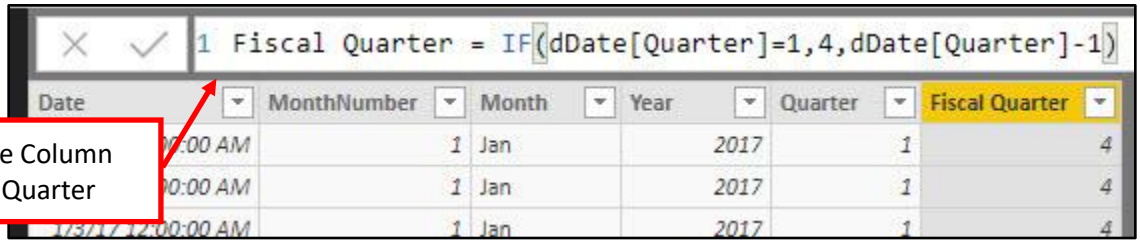
ix. In the Formula Bar type the formula for Year, as seen below.

9] Create Column for Year using formula

x. The company we are modeling for has a Fiscal Year Start of April 1. However, before we calculate the Fiscal Quarter and Year, we need to calculate the standard quarter for a Calendar Year. In the Formula Bar type the formula for Quarter, as seen below. This formula takes the month number and divides by 3. When we roundup to the one's position, this yields the correct quarter number for each row in the Date Table. You can also use an alternative formula as seen below.

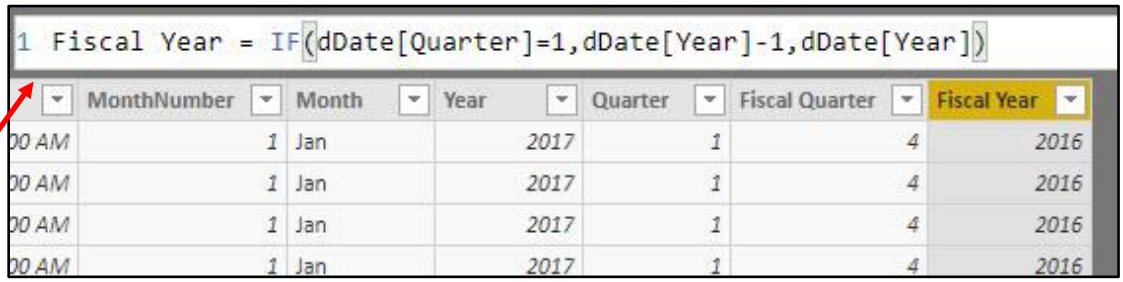
10] Create Column for Standard Quarter using formula

xi. Because our Fiscal Year starts on April 1, we need to convert the Standard Quarter to a Fiscal Quarter. In the Formula Bar type the formula for Fiscal Quarter, as seen below. This formula converts Quarter 1 to Quarter 4, and the remaining Quarters to one less than the Standard Quarter.



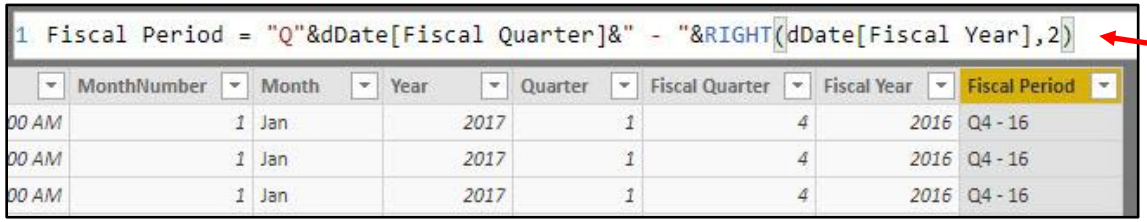
11] Create Column for Fiscal Quarter

xii. Because our Fiscal Year starts on April 1, we need to convert the Standard Year to a Fiscal Year. In the Formula Bar type the formula for Fiscal Quarter, as seen below. This formula converts the Calendar Year for the first three months (one standard quarter) to the correct Fiscal Year (Last Year) and leaves the remaining 9 months (last three quarters) as a Calendar Year.



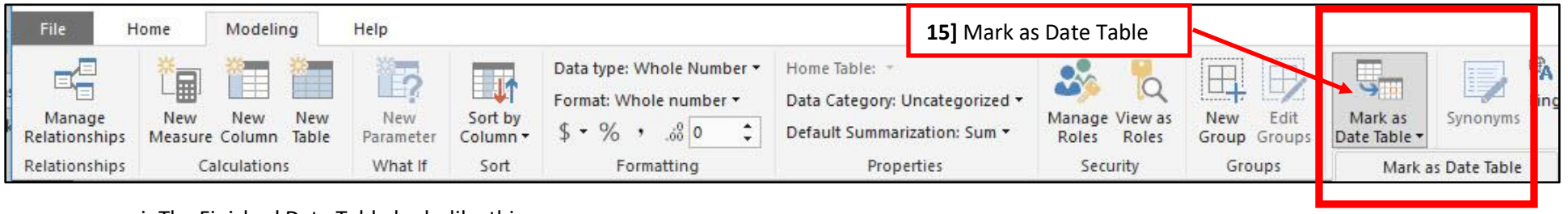
12] Create Column for Fiscal Year

xiii. We would like a short label for our reports. To accomplish this, we create a Text Formula for Fiscal Period. In the Formula Bar type the formula for Fiscal Period, as seen below.



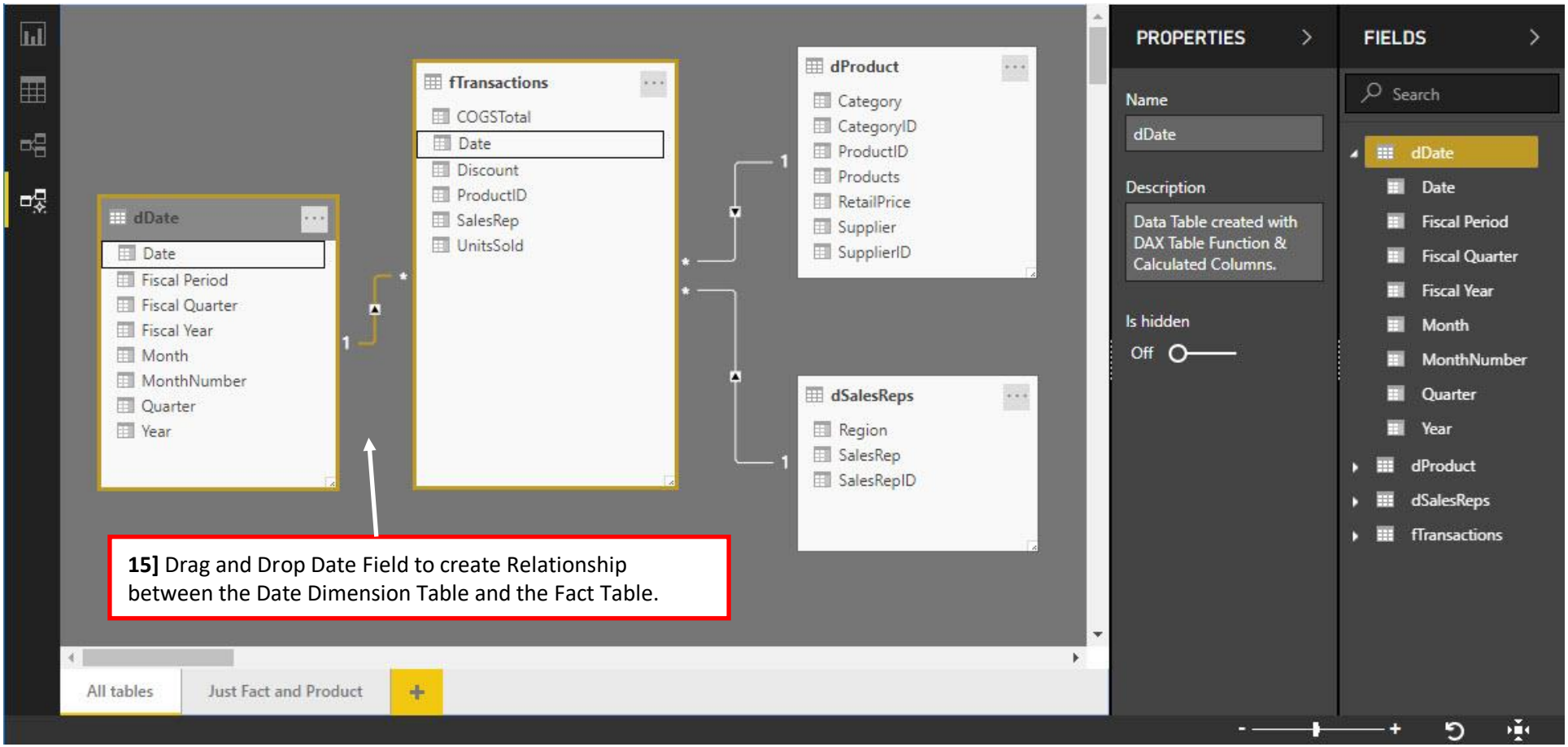
xiv. To get our Fiscal Period Field to sort correctly in our reports, we have to create a helper column that we can use to sort the Fiscal Period Field. To accomplish this, we create a Sorting Formula. In the Formula Bar type the formula for "SortFiscalPeriod", as seen below. After we create this column, we must go back and using the "Sort by Column" feature (as explained in the previous step for the Month Field) to sort Fiscal Period Field by the "SortFiscalPeriod" Field

xv. To mark the Date Table as a Date Table so that Automatic Date Tables are NOT created and so all the Time Intelligence Functions know that the table is a Date Table, in the Modeling Ribbon Tab, in the Calendar group, click the "Mark as Date Table" button.



xvi. The Finished Date Table looks like this:

xvii. In the Relationships area, we can drag and drop the Data Field from the dDate Table to the fTransactions Table to create a One-To-Many Relationship.



39) List of DAX Measures we need for our Visualizations & Tables in our Dashboards :

```
Total Units =
        SUM ( fTransactions[UnitsSold] )

Total Revenue =
        SUMX (
          fTransactions,
          ROUND ( fTransactions[UnitsSold] * ( 1 - fTransactions[Discount] ) * RELATED ( dProduct[RetailPrice] ), 2 )
        )

Total COGS =
        SUM ( fTransactions[COGSTotal] )

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

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

Ave Daily Gross Profit =
        AVERAGEX(dDate,[Total Gross Profit])

Ave Daily Rev =
        AVERAGEX (dDate,[Total Revenue])

Ave Transaction Revenue =
        AVERAGEX (
          fTransactions,
          ROUND ( fTransactions[UnitsSold] * RELATED ( dProduct[RetailPrice] ) * ( 1 - fTransactions[Discount] ), 2 )
        )

Rolling 12 Month Average Transactional Revenue =
        CALCULATE (
          [Ave Transaction Revenue],
          DATESINPERIOD ( dDate[Date], LASTDATE ( dDate[Date] ), -1, YEAR )
        )
```

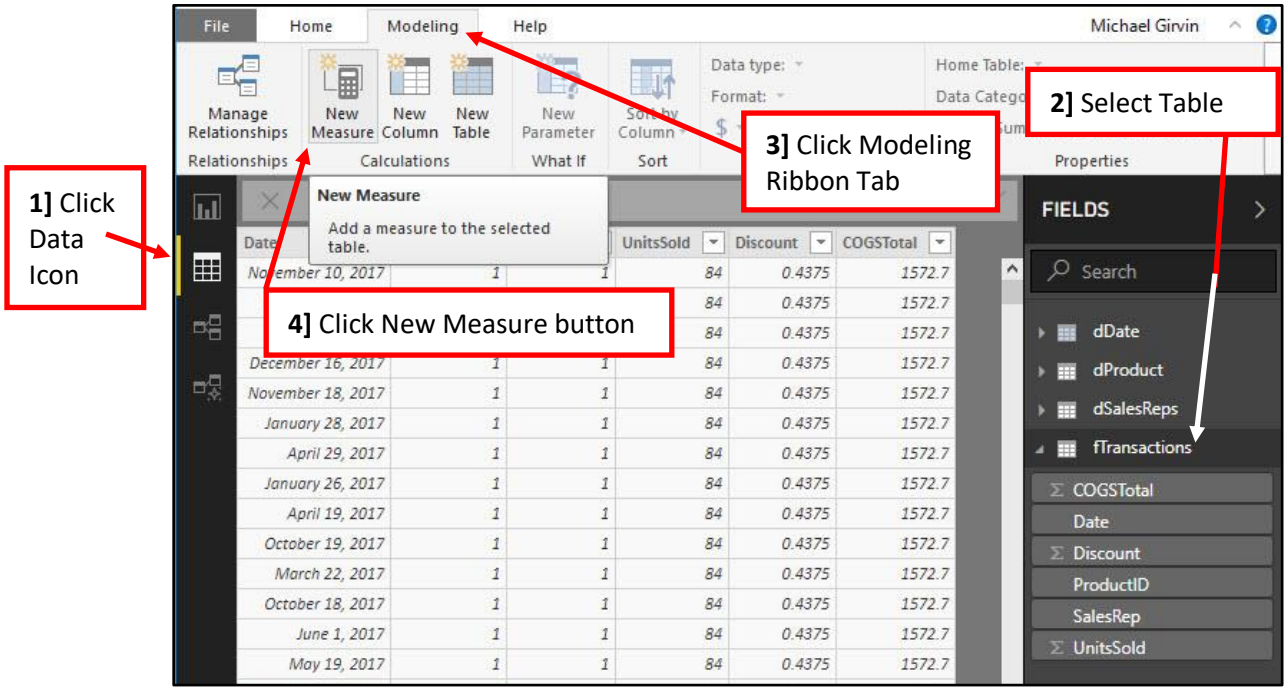
40) Steps to create DAX Measures :

1] On left side of Power BI Desktop Window, click on Data Icon.

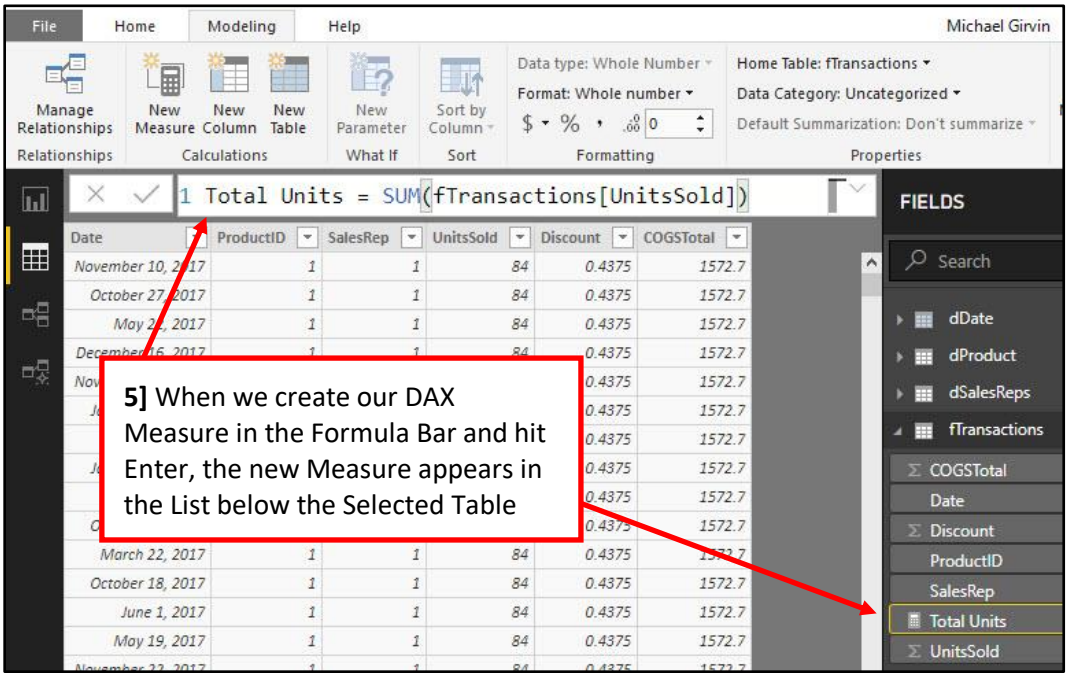
2] On the right side of Power BI Desktop Window, select the table that you want the Measure to appear under.

3] Click on the Modeling Ribbon Tab.

4] In the Calculations group, click the New Measure button.



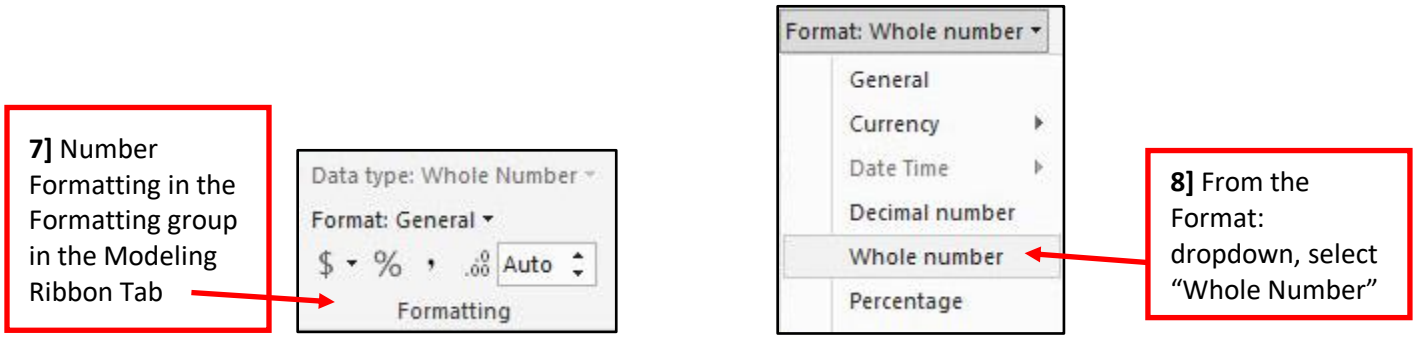
5] Our first Measure is for **Total Units**. Click in the Formula Bar and create the formula as seen below, then hit Enter.



6] Notice that the Icon for a Measure is a Hand-Held-Calculator.

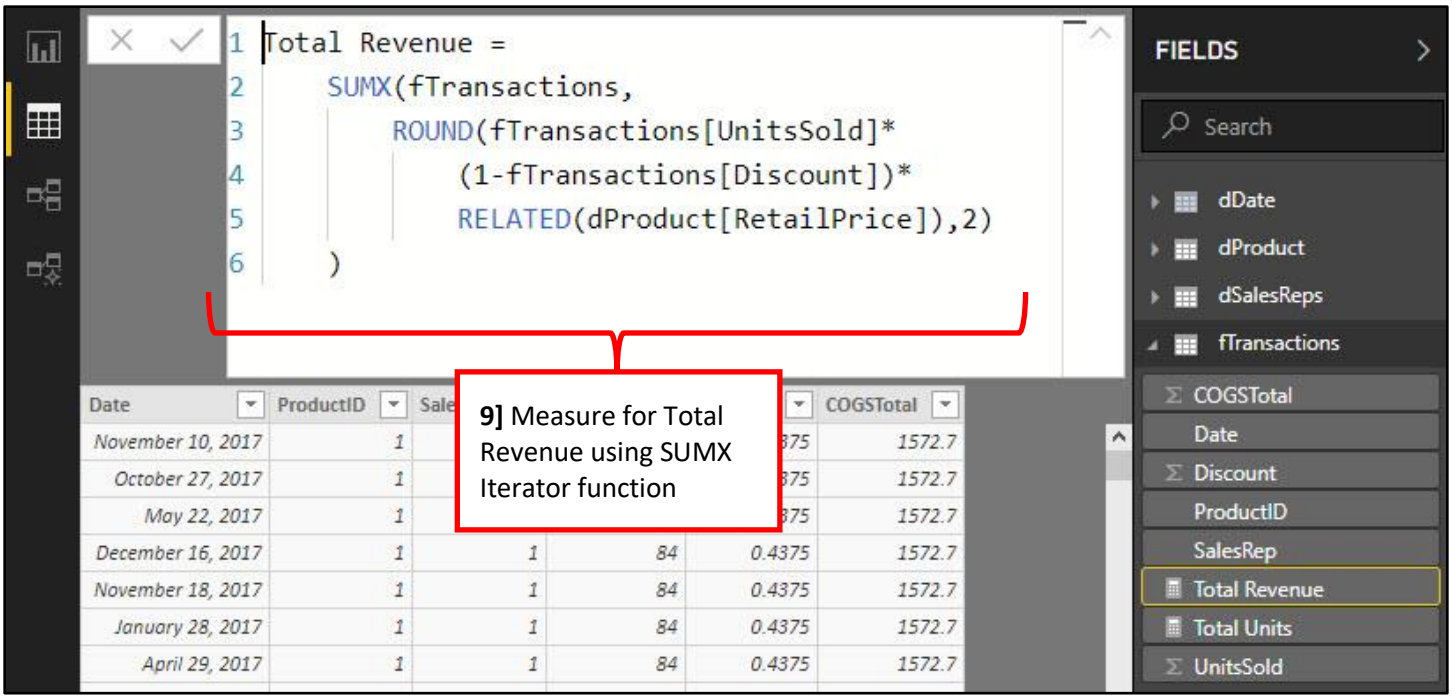
7] Once you create your Measure, select the Measure, and add the desired Number Formatting from the Formatting group in the Modeling Ribbon Tab.

8] From the Format: dropdown Arrow, select "Whole Number" Number Formatting



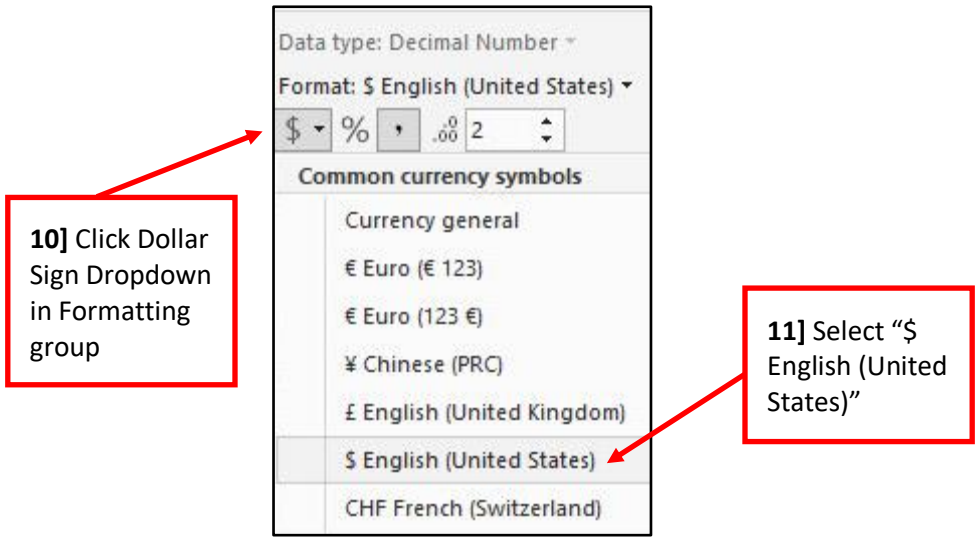
9] The next Measure is for **Total Revenue**. Click in the Formula Bar and create the formula as seen at the top of the next page.

1. SUMX is an Iterator Function that can simulate a Calculated Column inside a Measure. In the first argument of SUMX is the table we will iterate over (the fTransactions table).
2. In the second argument of SUMX is the formula we need to calculate the Transactional Line Item Revenue for each row in the fTransactions table.
    i. This formula takes the Units Sold for each row times one minus the Revenue Discount times the Retail Price.
    ii. For the Retail Price, the RELATED DAX Function uses the Relationship between the Fact Table and the Product Table to lookup the correct price for each product for each row in the Fact Table. The RELATED Function is working on the Many Side of the Relationship to retrieve the price from the One Side of the Relationship.
    iii. The ROUND function rounds the Line Item Revenue for each row in the Fact Table.
    iv. The second argument in SUMX generates an internal array of Line Item Revenue amounts, just as if there were a Calculated Column.
3. The SUM part of the SUMX function then sums the internal array of numbers to get a single aggregated sum.
4. Reminders about creating Measures in Formula Bar:
    i. If you want to create the DAX Formula on different lines, use **Shift + Enter** to create a new line and **Tab** to indent the formula line.
    ii. You can zoom in the Formula Bar by using Ctrl and Rolling the Wheel on your Mouse.

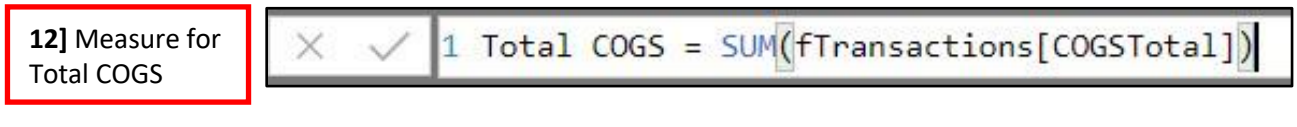


10] After creating the Measure for Total Revenue, click Dollar Sign Dropdown in Formatting group

11] select the Measure and then add the Number Formatting "$ English (United States)".



12] The next Measure is for **Total COGS**. Click in the Formula Bar and create the formula as seen below, then hit Enter. Add the "$ English (United States)" Number Formatting.



13] The next Measure is for **Total Gross Profit**. Click in the Formula Bar and create the formula as seen below, then hit Enter. Add the "$ English (United States)" Number Formatting. This formula uses two Measure that we already created: [Total Revenue] and [Total COGS].

13] Measure for Total Gross Profit

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

14] The next Measure is for **% Gross Profit**. Click in the Formula Bar and create the formula as seen below, then hit Enter. Add the Percentage Number Formatting. This formula compares Total Gross Profit in the numerator to Total Revenue in the Denominator using the DIVIDE DAX function. The advantage of using the DIVIDE function rather than straight division is that if the division results in a Divide-By-Zero error, the DIVIDE function will display a blank.

14] Measure for % Gross Profit

```
1 % Gross Profit = DIVIDE([Total Gross Profit],[Total Revenue])
```

15] The next Measure is for **Average Daily Gross Profit**. Click in the Formula Bar and create the formula as seen below, then hit Enter. Add the "$ English (United States)" Number Formatting.

1. AVERAGEX is an Iterator Function that will allow us to calculate an array of daily gross product amounts and then use the array of values to calculate the average, resulting in an average of daily gross profit amounts.
2. The key to this formula is that we put the Date Dimension Table with the correct day level granularity into the first argument of AVERAGEX. This allows calculations at the day level for the Total Gross Profit BEFORE we average the resulting values.
3. In the second argument of AVERAGEX we use the Total Gross Profit Measure. The goal is to have the Total Gross Profit Measure iterate the dDate table and calculate the Daily Gross Profit, and then use the resulting amounts to calculate the average.
4. Because we are using a Measure that is iterating the dDate Table, the hidden CALCULATE function (that accompanies all Measures) performs Context Transition which takes the row level day-date filter from each row in the dDate Table and converts it into Filter Context allowing the single day-date filter to flow across the Relationship and filter the Fact Table down to just the transactions for the given date and as a result calculate the correct Total Gross Profit for each day.
5. After the array of daily gross profit amounts are created in the second argument of AVERAGEX, then the AVERAGE part of AVERAGEX calculates the average of daily gross profit amounts.

15] Measure for Average Daily Gross Profit

```
1 Ave Daily Gross Profit = AVERAGEX(dDate,[Total Gross Profit])
```

16] The next Measure is for **Average Daily Revenue**. Click in the Formula Bar and create the formula as seen below, then hit Enter. Add the "$ English (United States)" Number Formatting. This formula calculates similarly to the previous formula, except that it calculates the Average of Daily Revenue amounts.

16] Measure for Average Daily Revenue

```
1 Ave Daily Revenue = AVERAGEX(dDate,[Total Revenue])
```

17] The next Measure is for **Average Transactional Revenue**. Click in the Formula Bar and create the formula as seen below, then hit Enter. Add the "$ English (United States)" Number Formatting.

17] Measure for Total Gross Profit

```
1 Ave Transaction Revenue =
2     AVERAGEX(fTransactions,
3         ROUND(fTransactions[UnitsSold]*
4             RELATED(dProduct[RetailPrice])*
5             (1-fTransactions[Discount]),2)
6     )
```

18] The next Measure is for **Rolling 12 Month Average Transactional Revenue**. Click in the Formula Bar and create the formula as seen below, then hit Enter. Add the "$ English (United States)" Number Formatting.

1. Because we are using the previous formula, Ave Transactional Revenue, but we need to change the filter context from the Month Filter that will be listed in the visualization to the Last 12 Months, we need to use the function that can change the Filter Context: the DAX CALCULATE function.
2. In the first argument of CALCULATE we place our Measure.
3. In the second argument of CALCULATE we use the DATESINPERIOD DAX function to generate a list of valid dates for the Measure. In our case, we need all the dates from the last day in the current Filter Context extending back one year.
    i. This function requires:
        1. First argument = the first column from the Date Table.
        2. Second argument = start date. We use the LASTDATE DAX function to pick out the last date in the current Filter Context (like last day in month for a Month Date Period).
        3. Third argument = how many time intervals. We need to jump back one year and so we use minus one.
        4. Fourth argument = the time interval. Our time interval is Year.
4. With the valid list of dates extending back one year in the second argument of the CALCULATE functions, the Fact Table is then filtered down to the correct dates and the transactional amounts are used in the [Ave Transactional Revenue] Measure.

18] Measure for Total Gross Profit

```
1 Rolling 12 Month Ave Transactional Revenue =
2     CALCULATE(
3         [Ave Transaction Revenue],
4         DATESINPERIOD ( dDate[Date], LASTDATE ( dDate[Date] ), -1, YEAR )
5     )
```

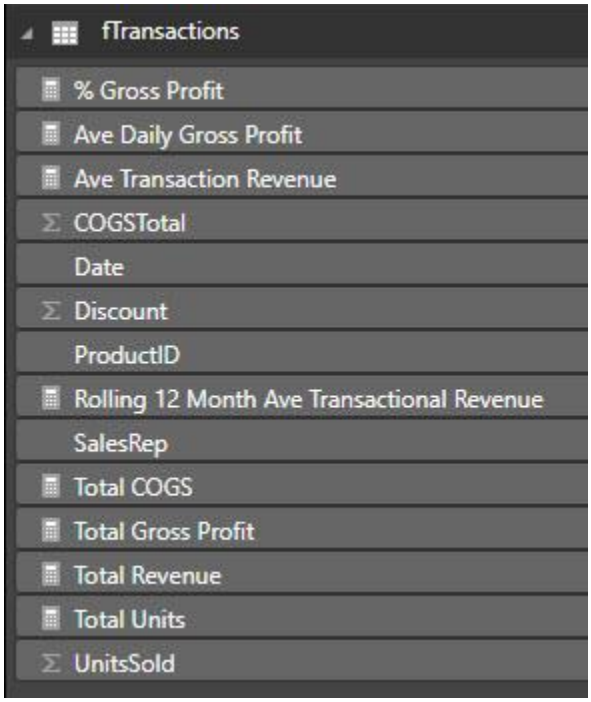
First Column From The Date Table

Start Date

How Many Time Intervals

Time Interval

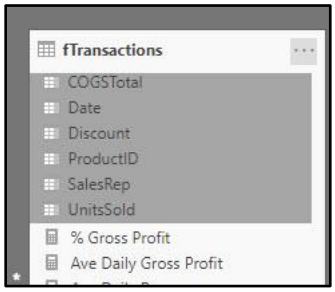
19] The final list of Columns and Measures below the fTransactions Table looks like this:



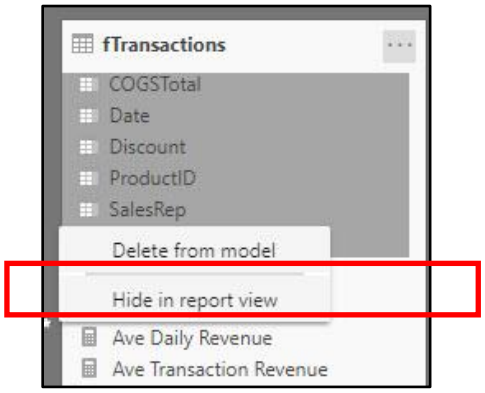
41) Hide Items from Report View :

1] To make the Reporting environment easy to use, we need to hide any Columns, Tables or Measures that we will not need as part of Data Modeling Steps, but not needed in Reporting View.

2] In the Relationship area you can select multiple columns as seen here:

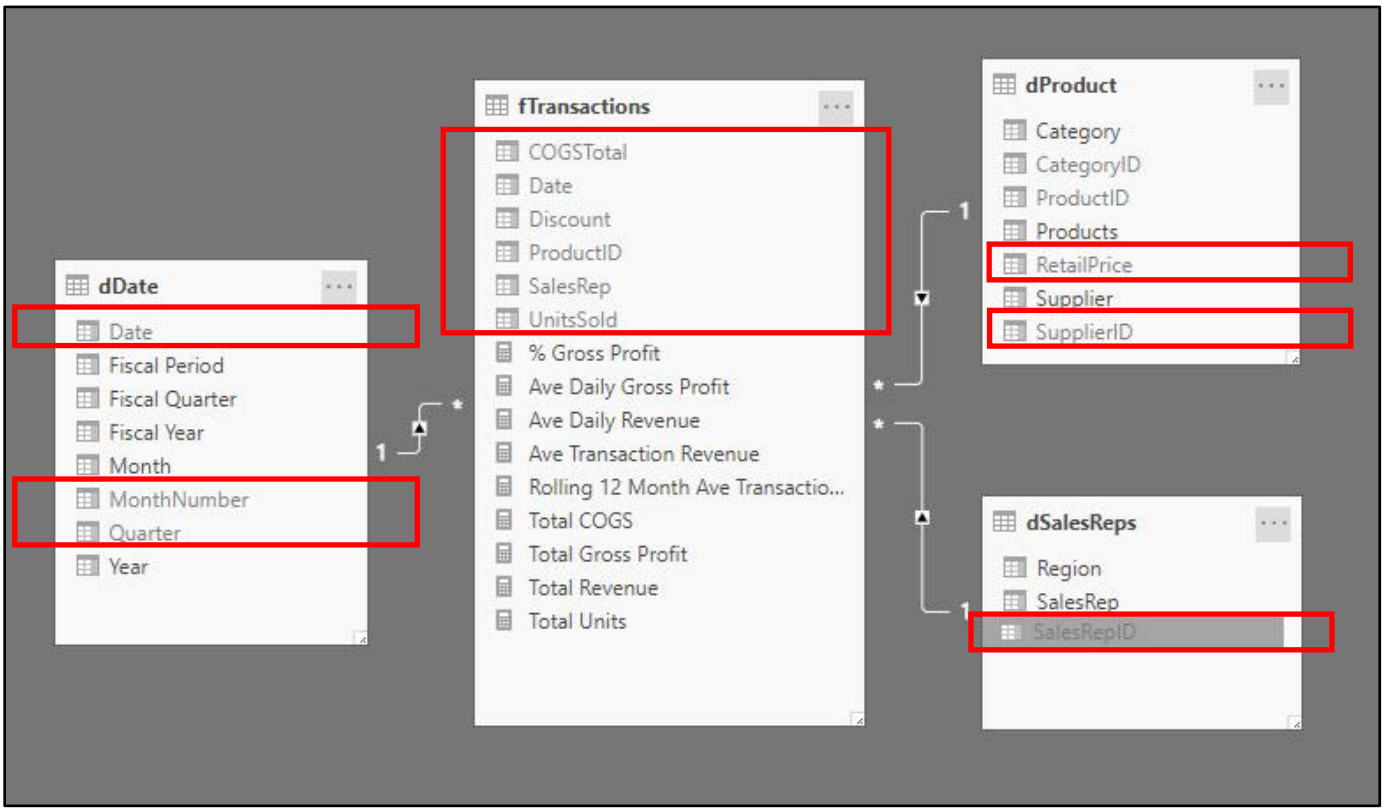


3] Right click and click on “Hide in report view”



4] In our Data Model we want to hide these Fields:

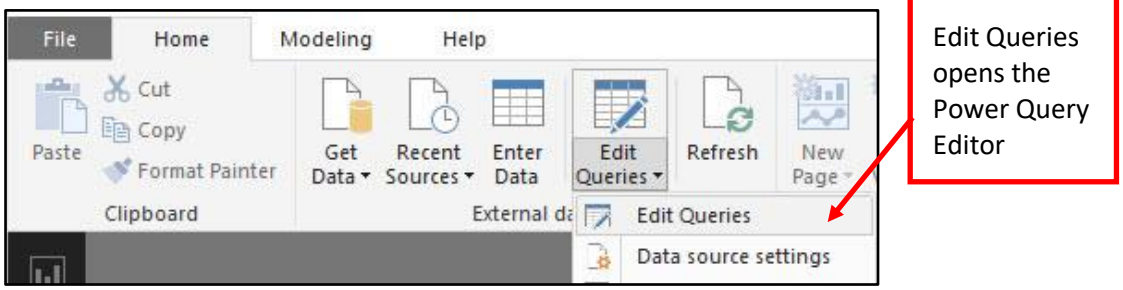
1. dDate Table
    i. Date
    ii. MonthNumber
    iii. Quarter
2. fTransactions Table:
    i. COGSTotal
    ii. Date
    iii. Discount
    iv. ProductID
    v. SalesRep
    vi. UnitsSold
3. dProduct Table:
    i. RetailPrice
    ii. SupplierID
4. dSalesRep Table:
    i. SalesRepID



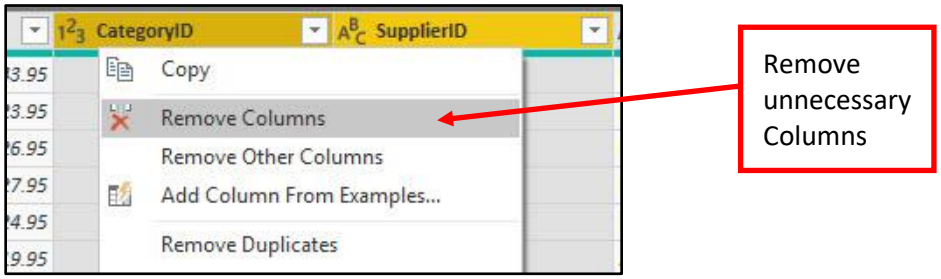
42) Refine Data Model in Power Query by Removing Columns in dProduct Table :

1] We are allowed to go back and edit our Power Query queries at any time if we need to refine the Data Model.

2] In the Home Ribbon Tab, in the "External data" group, click the Edit Queries dropdown arrow and then click on Edit Queries.

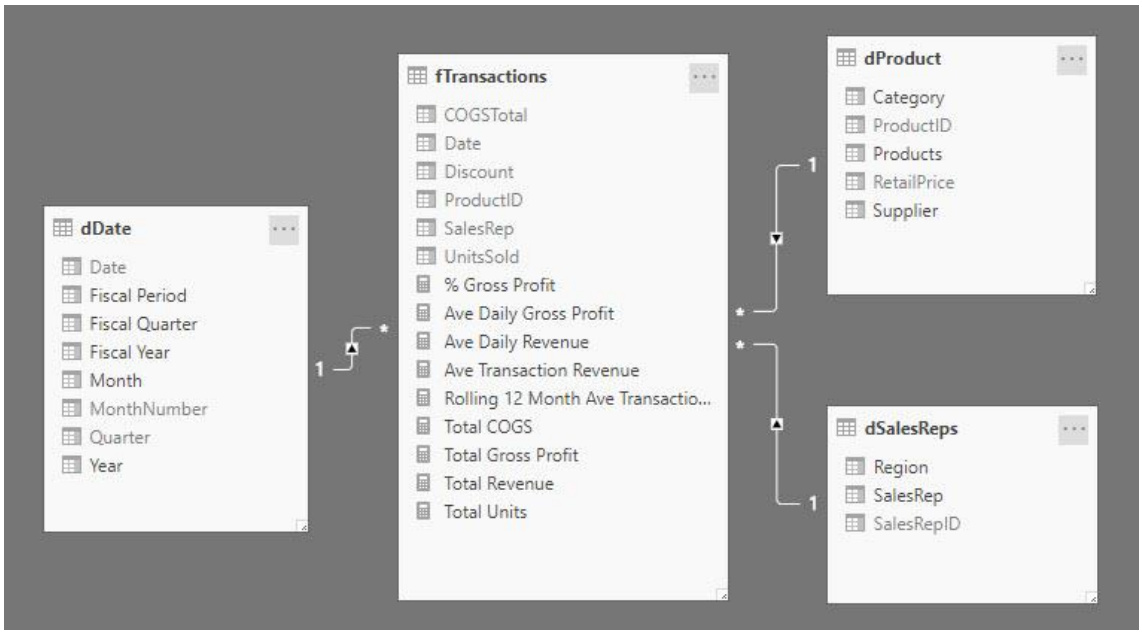


3] In the Power Query Editor, select the dProduct Table Query, then select the CategoryID and SupplierID Columns, right-click the selected columns and then click on Remove Columns.



4] When you click the Close and Apply button.

5] The Refined data Model looks like this:



43) Create “Ave Daily GP” Tab or Page :

Visual Goal: Look at Average Daily Gross Profit across the dimensions: Product, SalesRep and Fiscal Quarter:

Steps:

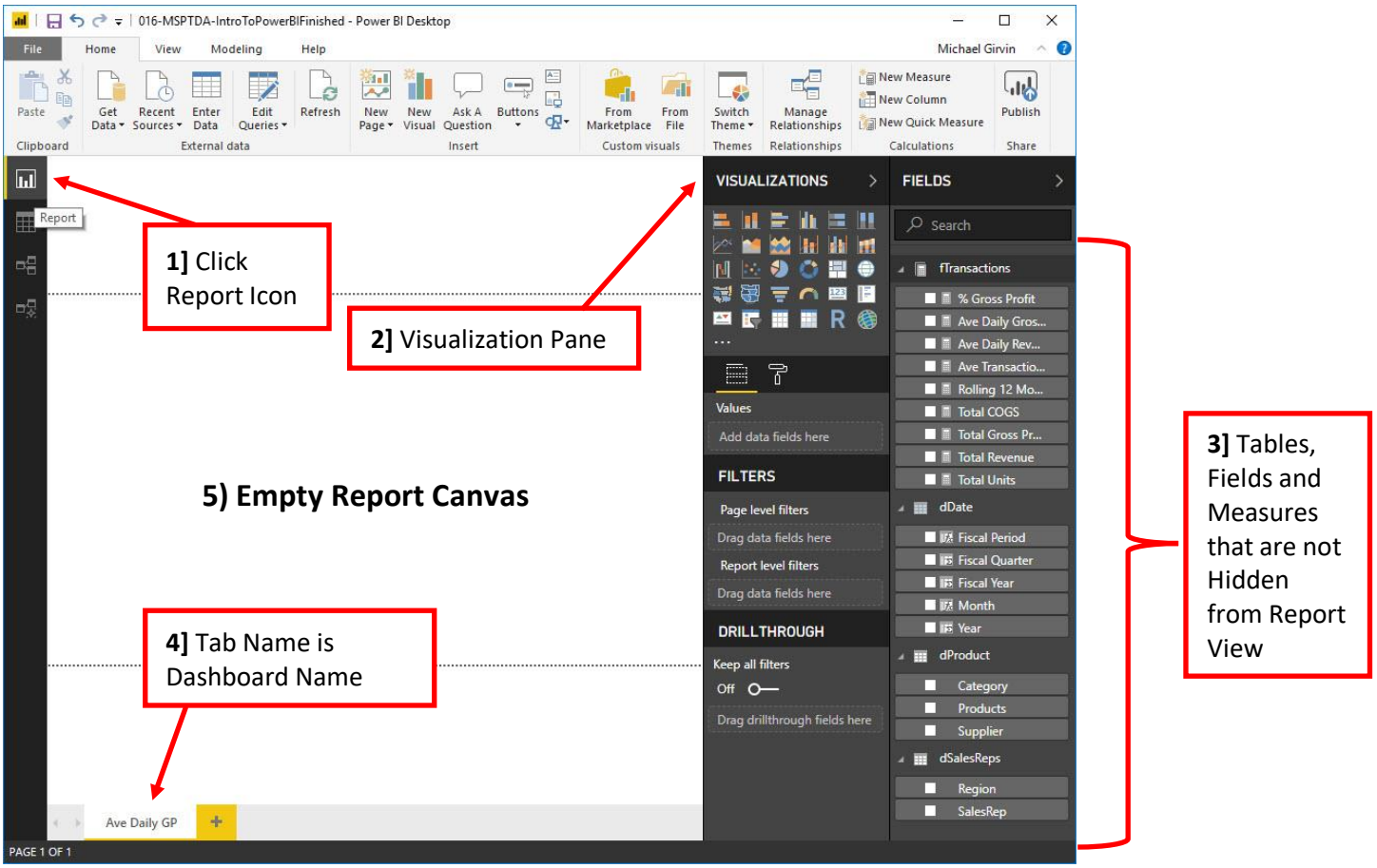
1] On the Far-Left in the Power BI Desktop window, click on the Report Icon.

2] On the Right we can see the Visualization Pane.

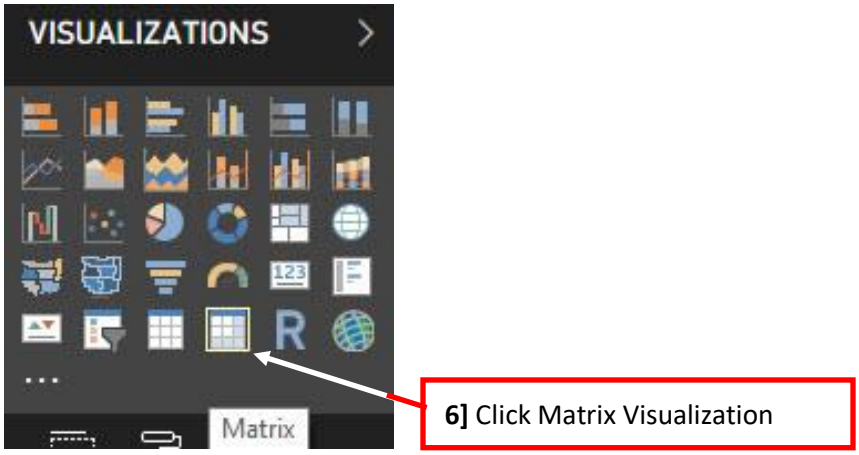
3] In the Fields Pane we can see all the Tables, Fields and Measures that are not Hidden from Report View.

4] At the bottom we double click the Tab and name the Tab “Ave Daily GP”.

5] The white area is the Empty Report Canvas where we can add visualizations to our Tab.

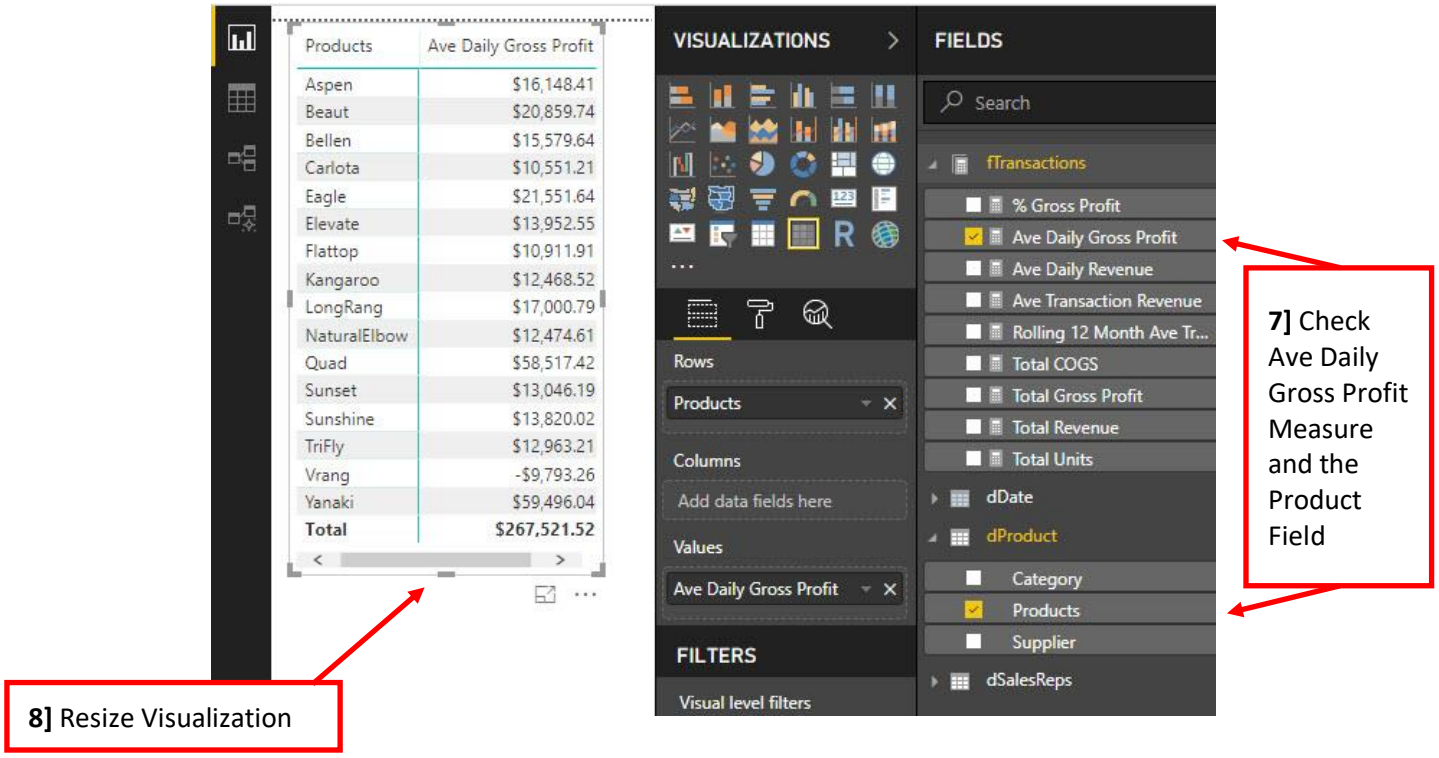


6] From the Visualization Pane, we click on Matrix option.



7] From the Fields Pane check the Ave Daily Gross Profit Measure under the fTransactions table and the Product Field from the dProduct table.

8] Resize Visualization as necessary.



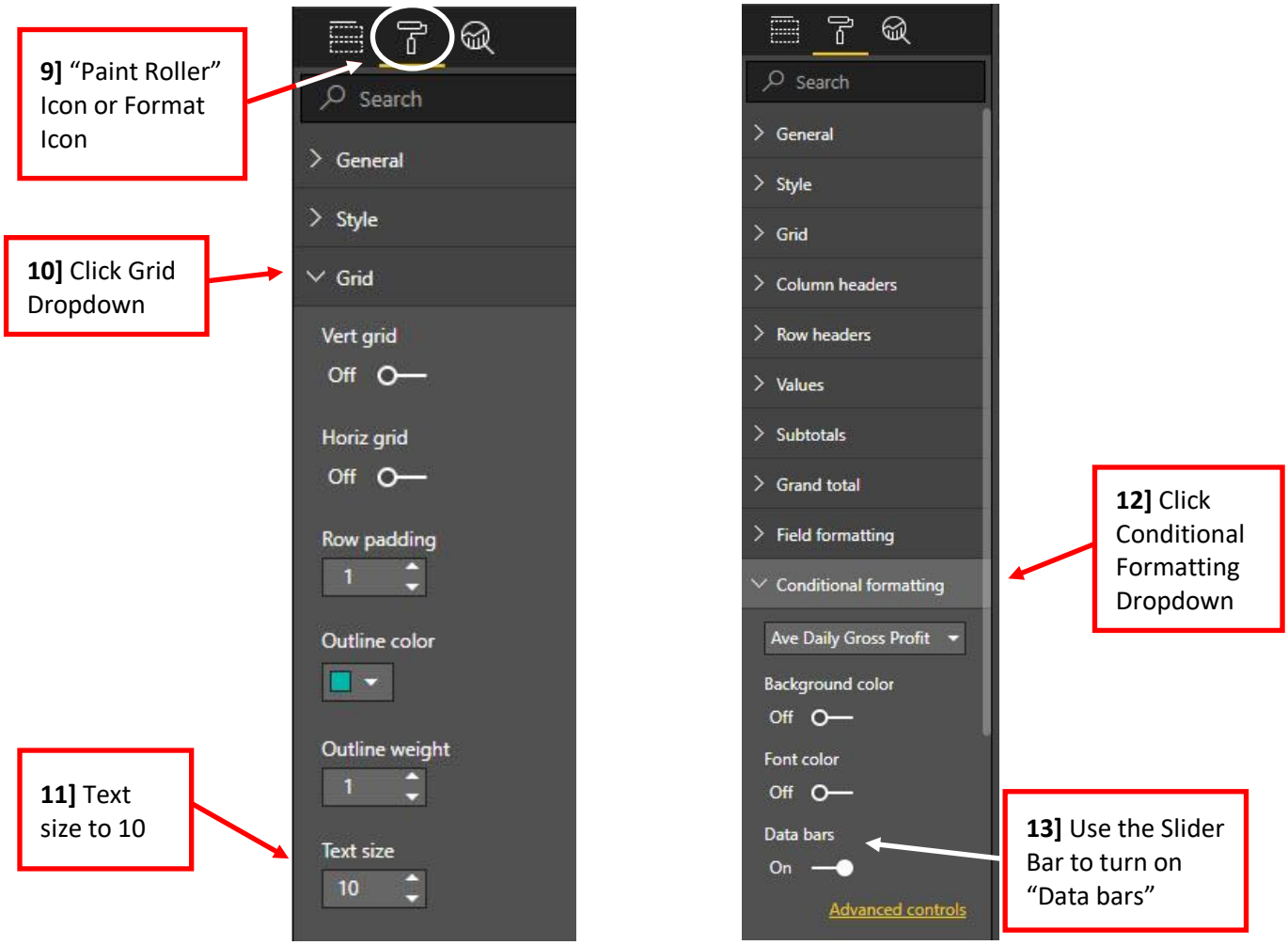
9] With the Matrix selected, over in the Visualization Pane, click on the "Paint Roller" Icon or Format Icon. There are many options for Formatting. Often times we have to click around trying to find what we want.

10] Click Grid Dropdown.

11] Then change the Text size to 10.

12] Click Conditional Formatting Dropdown.

13] Use the Slider Bar to turn on "Data bars".



14] The Result can be seen below. The reasons we chose this visualization are:

1. We used a Matrix because we wanted to see the specific number detail.
2. We used Data Bar Conditional Formatting to get a quick visual impression.

Products	Ave Daily Gross Profit
Aspen	$16,148.41
Beaut	$20,859.74
Bellen	$15,579.64
Carlota	$10,551.21
Eagle	$21,551.64
Elevate	$13,952.55
Flattop	$10,911.91
Kangaroo	$12,468.52
LongRang	$17,000.79
NaturalElbow	$12,474.61
Quad	$58,517.42
Sunset	$13,046.19
Sunshine	$13,820.02
TriFly	$12,963.21
Vrang	-$9,793.26
Yanaki	$59,496.04
Total	**$267,521.52**

14] Matrix Visualization for Ave Daily Gross Profit by Product

15] Hierarchies. Before we create our next visualization, we need to create a Hierarchy. Hierarchies are multiple Fields grouped together, for example Region and Sales Representatives within each Region. The benefits of Hierarchies are:

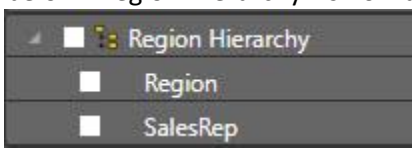
1. Hierarchies allow us to drag a single item to a visualization and have the single item display multiple fields.
2. With Hierarchies we can also Drill Down in Visualizations.

16] We want the two Fields SalesRep and Region to become a Hierarchy.

17] To create a Hierarchy for SaleRep within Region, click and drag the SalesRep Filed so it is over the Region Field, and then drop the SalesRep Field. Instantly you have created a Hierarchy. (This is much different than in Excel Power Pivot where you have to select both Field and use a Right-Click Method).

17] Click & Drag Sales Rep Field ***over*** Region, then ***drop*** to create a Hierarchy

18] As seen below "Region Hierarchy" is now a single Item that can be used in Visualizations.

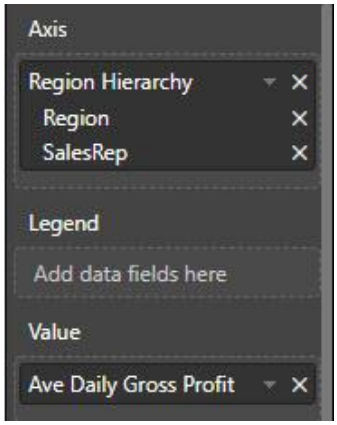


18] "Region Hierarchy" can be used in Visualizations

19] Click back in the white area of the Report Canvas, and then click on the Clustered Column visualization button.

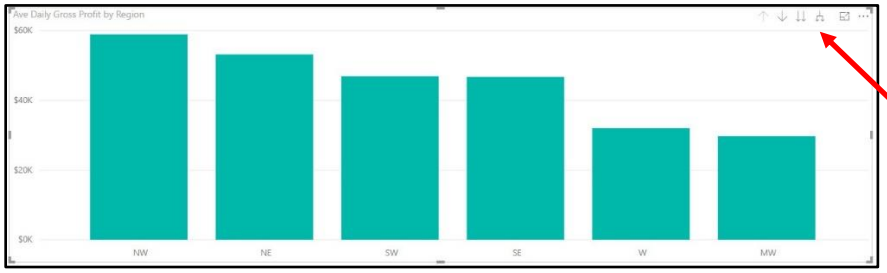
19] Clustered Column Chart

20] from the Fields Pane, drag the Region Hierarchy to the Axis area and the Ave Daily Gross Profit Measure to the Values area.



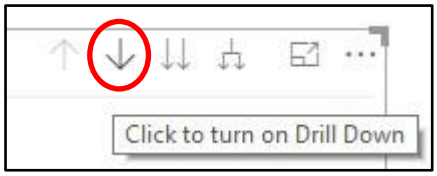
20] Region Hierarchy to Axis.
Ave Daily Gross Profit Measure to Values.

21] Clustered Column Chart is shown below. Notice that only the Region is shown. To Drill Down to the next level, Sales Rep we need to take a closer look at the icon buttons in the upper right corner of the chart.



21] Drill Down Icons in upper right of Clustered Column Chart

22] The First Arrow is greyed-out because the chart is showing the highest level, the Region level. Second Arrow from the left is the "Turn On or Off" Drill Down Icon. This drill down option will allow you to drill down into the level below in the hierarchy.



22] "Turn On or Off" Drill Down Icon

23] After you click "Turn On Drill Down", the button changes to a dark-grey color.

1. If this option is turned **ON**, this will cause a "Click on a Column in The Chart" to Drill Down to the next level. For example, if we click on the SE column, we would see only the columns for the SalesReps in the SE Region.

23.1] Drill Down is "On". If we click SE Column, we see 3 columns for the 3 SalesReps in the SE Region.

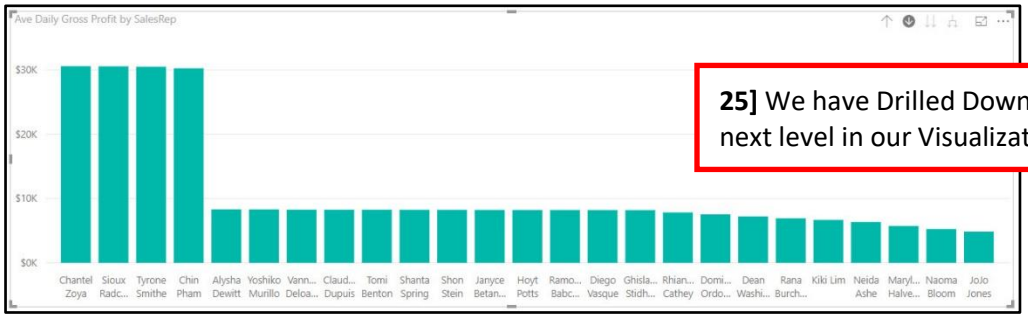
2. If this option is turned **OFF**, we can not drill down into a particular column. With this option off, clicking a column will only filter the other visualizations in the Dashboard.

23.2] Drill Down is "Off"

24] With the "Drill Down" Arrow (second from left) turned off AND with the chart showing only the Region Columns, if we click the Double Downward Arrow (third arrow from the left), we will go to the next level to show all the columns for all the Sale Reps.

24] Go To Next Level button

25] With the Column Chart showing all Regions, after we click the "Go To Next Level" button, the chart shows the SalesRep Amounts.



25] We have Drilled Down to the next level in our Visualization.

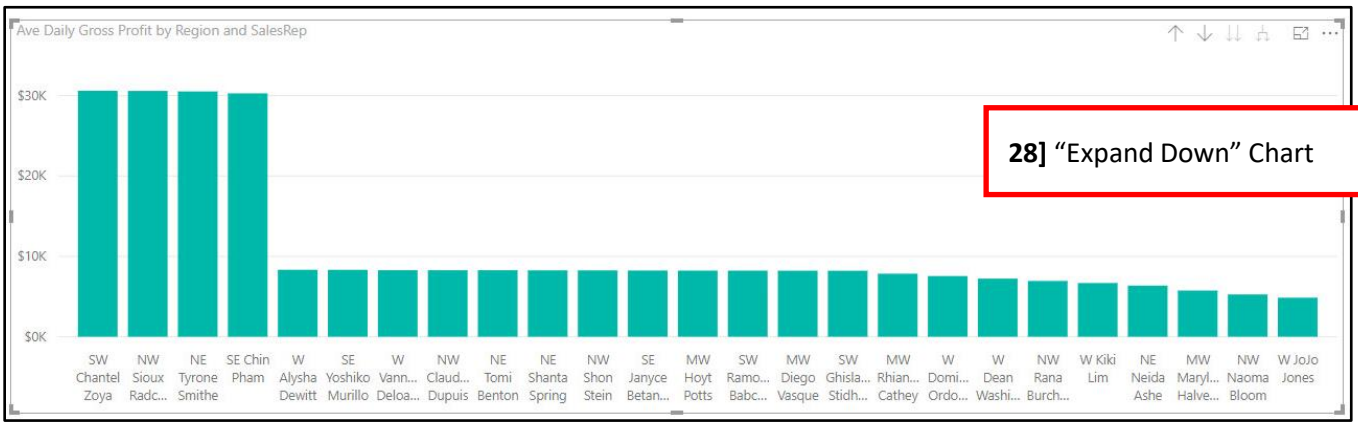
26] After we Drill Down, the first button on the left, the “Drill Up” button, is no longer greyed-out. Click the Drill Up button to move up a level in the Hierarchy.

26] “Drill Up” Icon

27] If we click the “Expand Down” Icon (4th Icon from the left), we will see both labels in the axis.

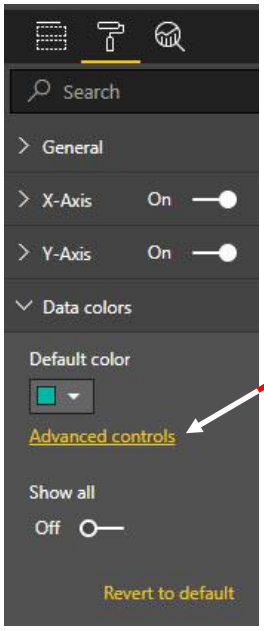
27] “Expand Down” Icon

28] After clicking the “Expand Down” Icon, The Chart shows labels for both Fields in horizontal axis.



28] “Expand Down” Chart

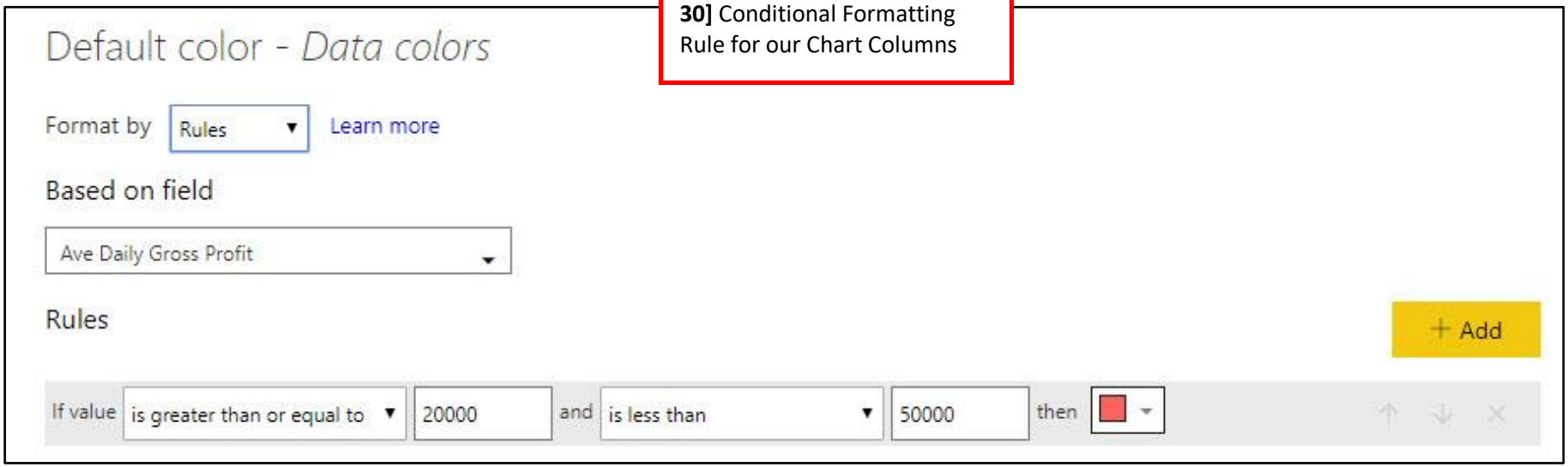
29] Next, we want to add a Conditional Formatting Rule to the Columns. To do this we click the “Paint Roller” Format button, then click the “Data color” dropdown arrow, then click the “Advanced controls”.



29] To access Conditional Formatting, click the “Advanced controls”

30] To add a Fill Color for “Ave Daily Gross Profit” amounts greater than or equal to 20,000 and less than or equal to 50,000, create the rule as seen in the below picture.

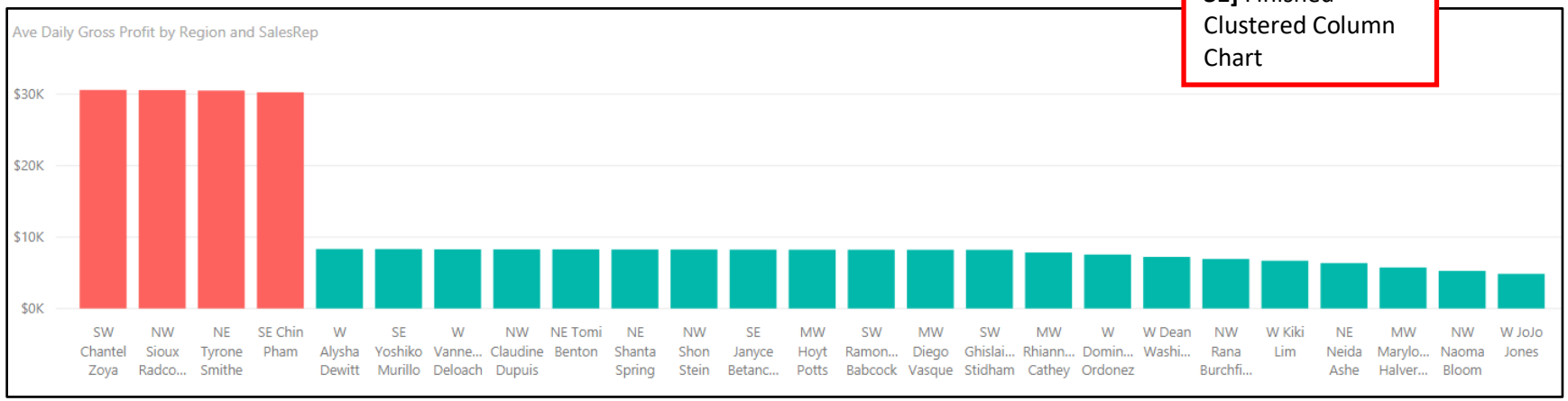
30] Conditional Formatting Rule for our Chart Columns



31] The finished Clustered Column Chart can be seen below. The reasons we chose this visualization are:

1. With the Column Chart we can easily compare amounts across categories.
2. We can also Drill Down or Drill Up to see just sales Rep amounts or Region Amounts.

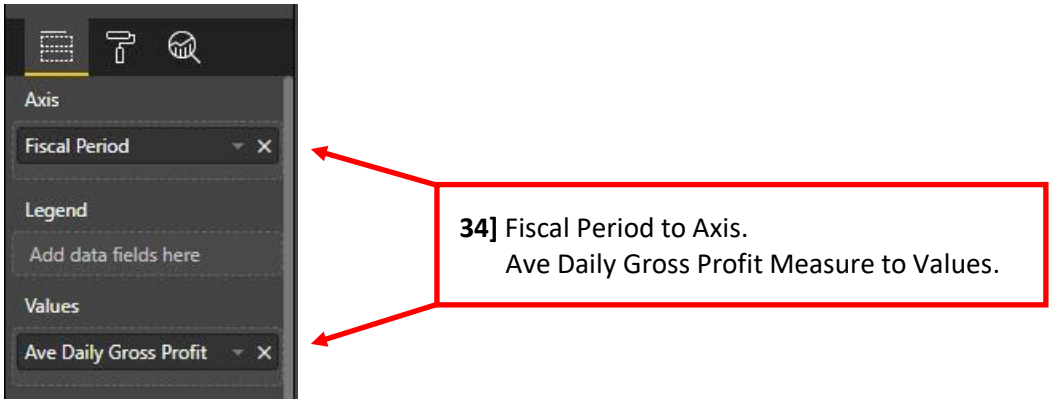
31] Finished Clustered Column Chart



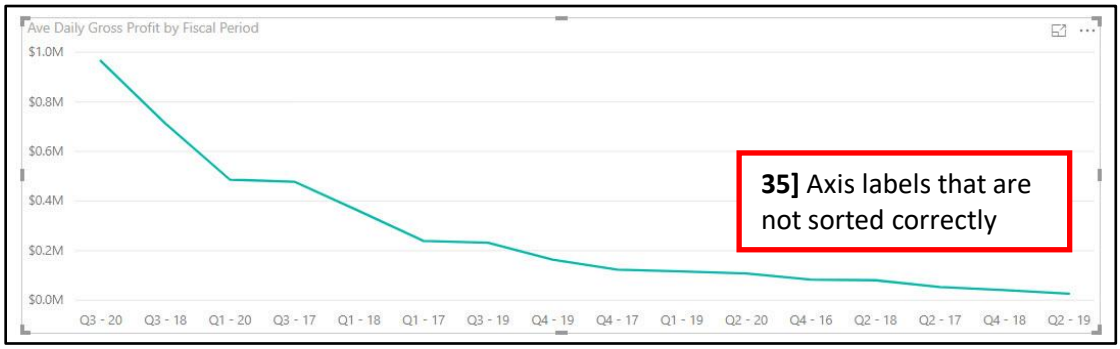
32] Our next goal is to add a line chart that shows the change in “Ave Daily Gross Profit” over the Fiscal periods. However, we will run into a problem that we will have to deal with.

33] Click back in the white area of the Report Canvas, and then click on the “Line chart” visualization button.

34] From the Fields Pane, drag the Fiscal Period Field to the Axis area and the Ave Daily Gross Profit Measure to the Values area.

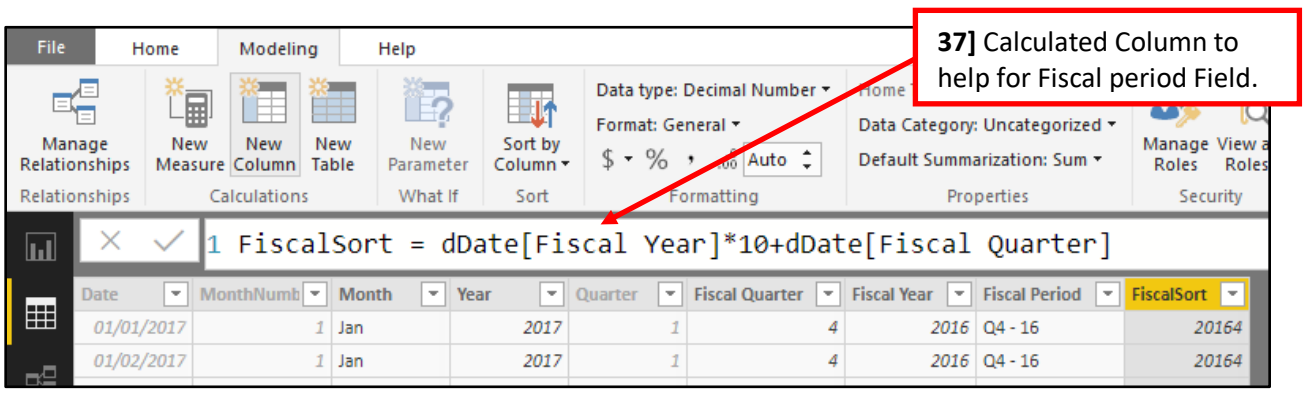


35] The resulting Line Chart contains axis labels that are not sorted correctly. We must go back to the Date Dimension Table and fix this sorting problem.

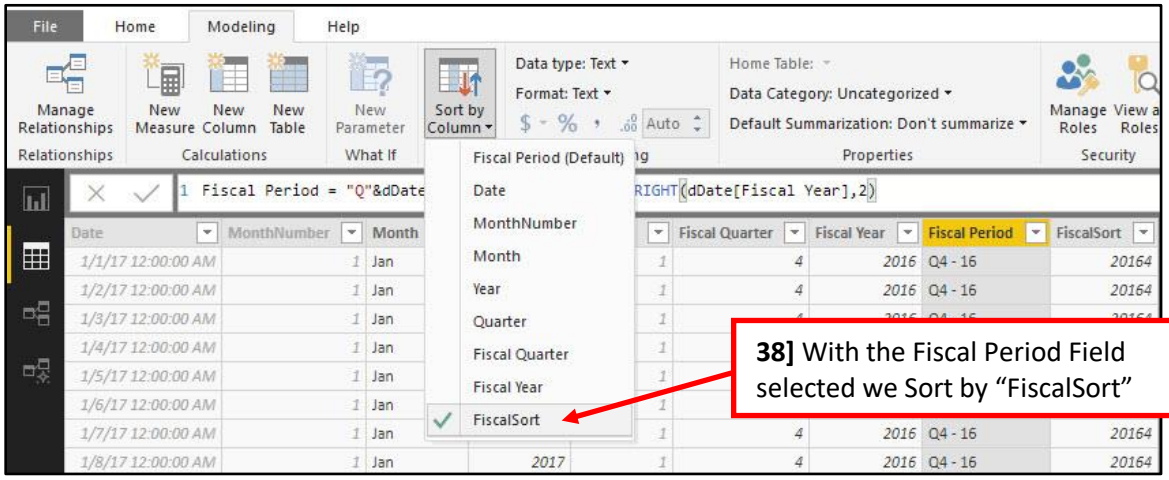


36] Edit DAX Date Table. To fix the sorting problem we need to add a new column in the Date Dimension Table that contains a number that we can use to Sort the Fiscal Period Field by using the “Sort by Column” feature.

37] In the Data area, select the dDate table, then in the Modeling Ribbon Tab, in the Calculations group, click the New Column button and then create the Calculated Column as seen below:



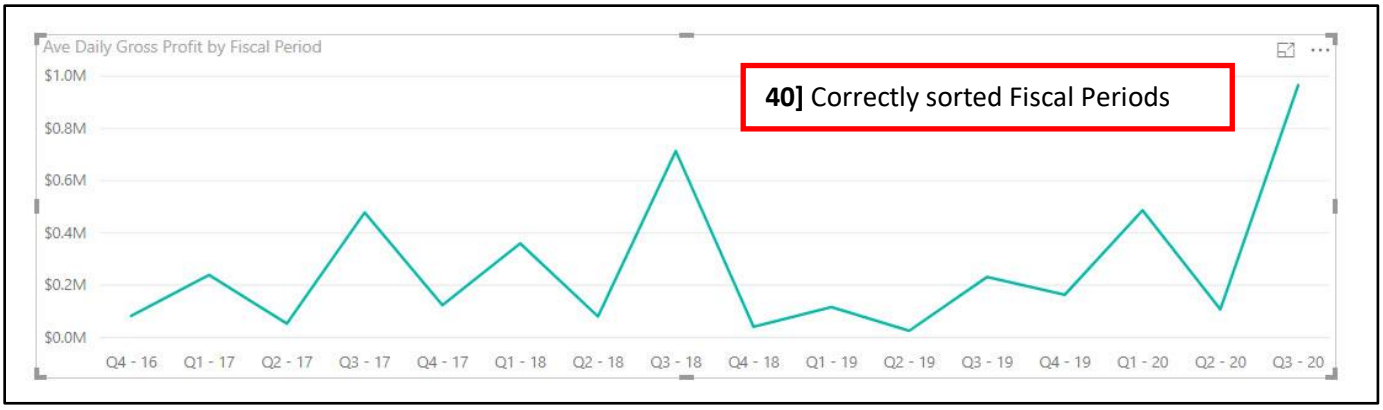
38] Making sure the “Fiscal Period” Column is selected, In the Modeling Ribbon Tab, in the Sort group we use the “Sort by Column” feature to sort by the “FiscalSort” column.



39] Because the “FiscalSort” Column is a Helper Column that we use in the Data Model but do not need in the Report View, right-click the “FiscalSort” Column Header and click on “Hide in Report View”.

40] Now, the Line Chart shows the correctly sorted Fiscal Periods along the horizontal axis, as seen below. The reasons we chose this visualization is:

1. We wanted to see the trend or patter over time.



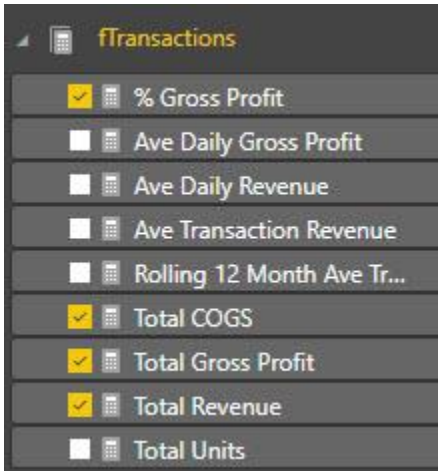
41] Our final element in our Dashboard is a Card that will show the values of a number of Measures.

42] Click back in the white area of the Report Canvas, and then click on the "Multi-row card" visualization button.

1. 

42] click on the "Multi-row card" visualization

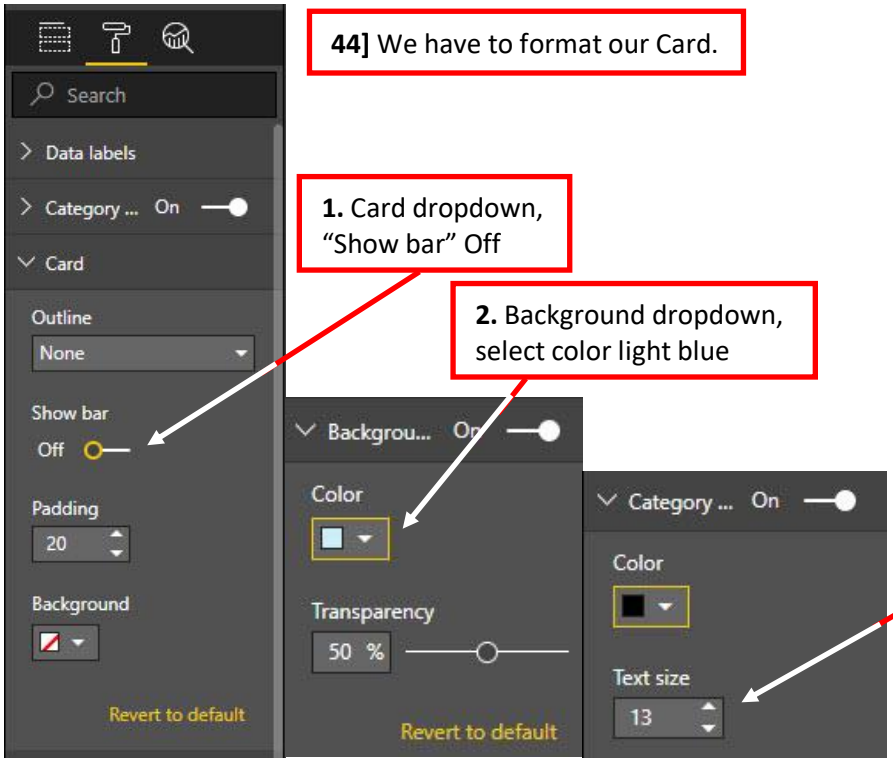
43] With the Card Selected, check the Measures as seen in the picture below.

1. 

43] Check the Measures:
1. Total Revenue
2. Total COGS
3. Total Gross Profit
4. % Gross Profit

44] With the Card Selected, Format the Card in the following ways:

1. Card dropdown, "Show bar" Off.
2. Background dropdown, select color light blue.
3. Category dropdown, Text size = 13.

4. 

44] We have to format our Card.

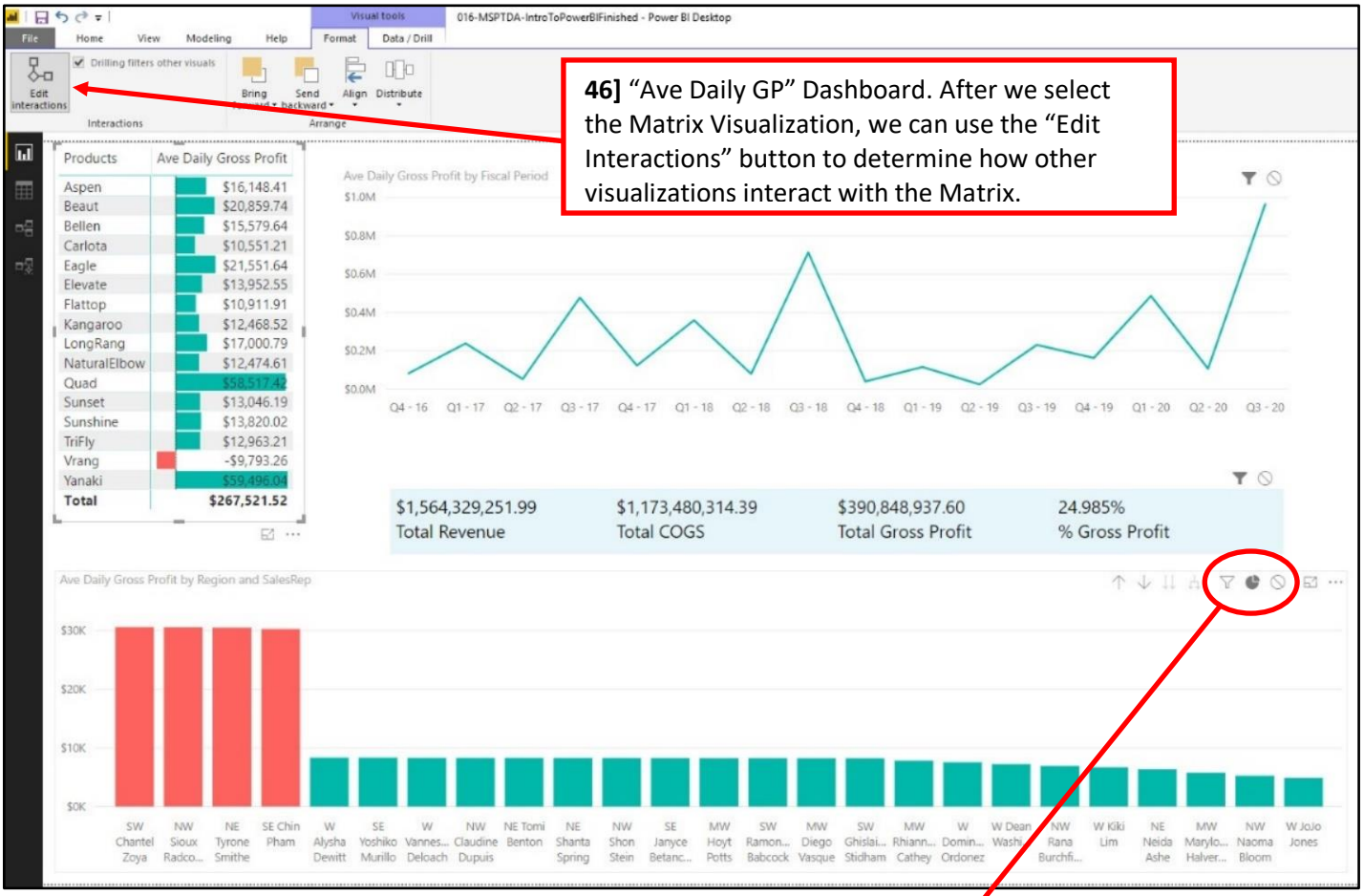
1. Card dropdown, "Show bar" Off

2. Background dropdown, select color light blue

3. Category dropdown, Text size = 13

45] After you resize and re-position the Card, our “Ave Daily GP” Dashboard is almost complete, as seen in the picture below.

46] The final task for this Dashboard is to edit the filtering interactions between the different visualizations. With the Matrix selected, click on the Format Ribbon Tab, then click the “Edit Interactions” button. This will allow us to edit the other visualizations (not the Matrix) and determine how they interact with the Matrix when we click on an element in the Matrix.



46] “Ave Daily GP” Dashboard. After we select the Matrix Visualization, we can use the “Edit Interactions” button to determine how other visualizations interact with the Matrix.

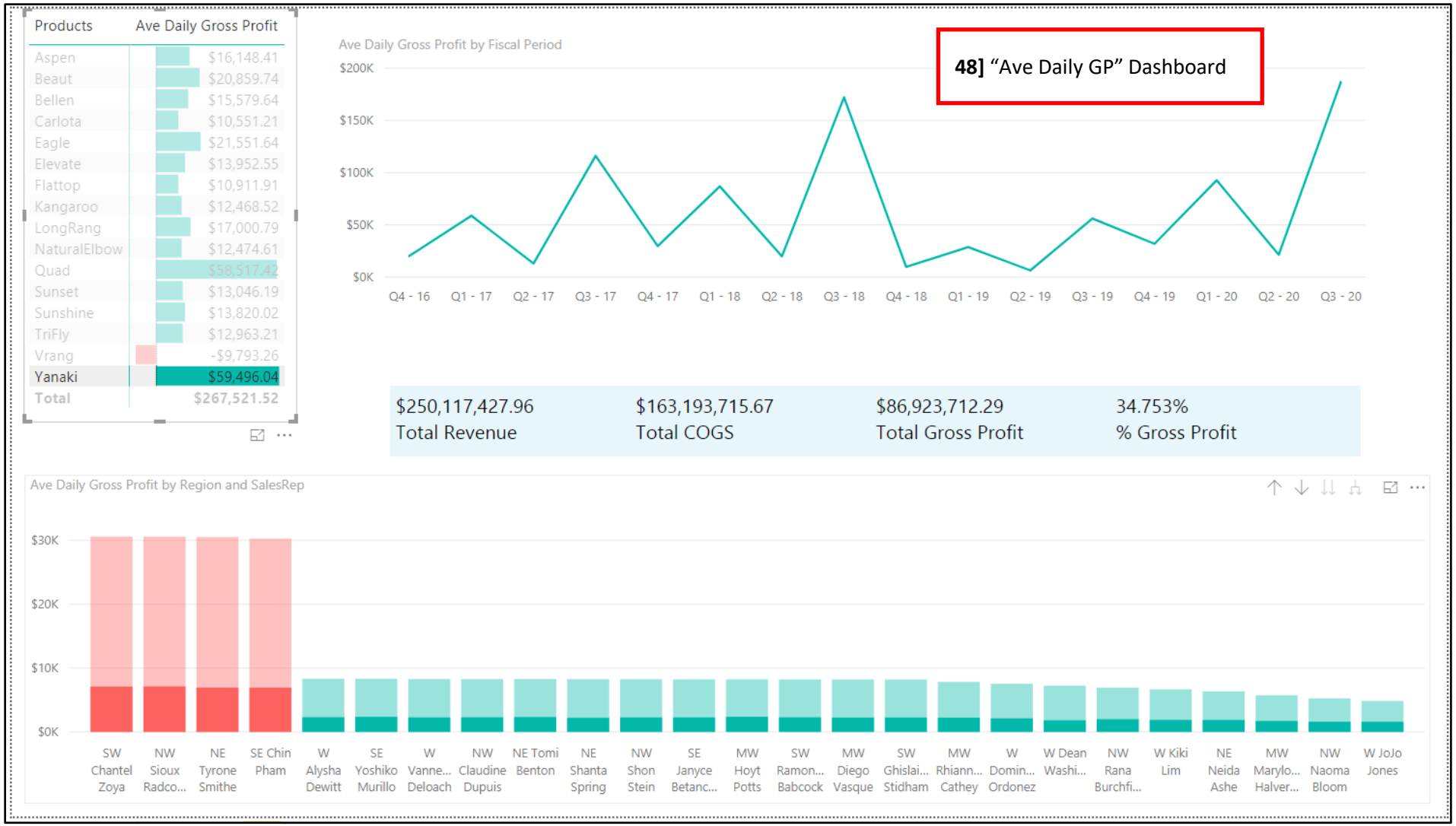
47] A close up of the Filtering Options that determine how the Clustered Column Chart will interact with the Matrix Visualization when we click on an element in the Matrix Visualization. Click to select option. When you click to select, the icon become grey.

Filter = Whole Column Value (Height) will reflect Filter From Matrix

Piece of Pie = Partially Formatted Column will reflect Filter From Matrix

NO Filter = Filter From Matrix will have NO effect on Column

48] Final “Ave Daily GP” Dashboard as seen below with the “Yanaki” Product selected in the Matrix Visualization:



For the Remaining Three Tabs (Pages), this handout will not list all the steps. Instead there will be a picture of the final Tab and a list of the basic elements for each Tab.

44) Create “Fiscal Report” Tab (Page) :

1] Add a “Multi-row card” at the top with the Measures for Total Units, Total Revenue, Total COGS and Total Gross Profit.

2] Add a Clustered Bar Chart for Total Gross Profit by Fiscal Period. And for using the “Paint Roller” to format, add Data Labels and for Data Colors click Advanced Controls and choose Color Scale for the Total Gross Profit Measure.

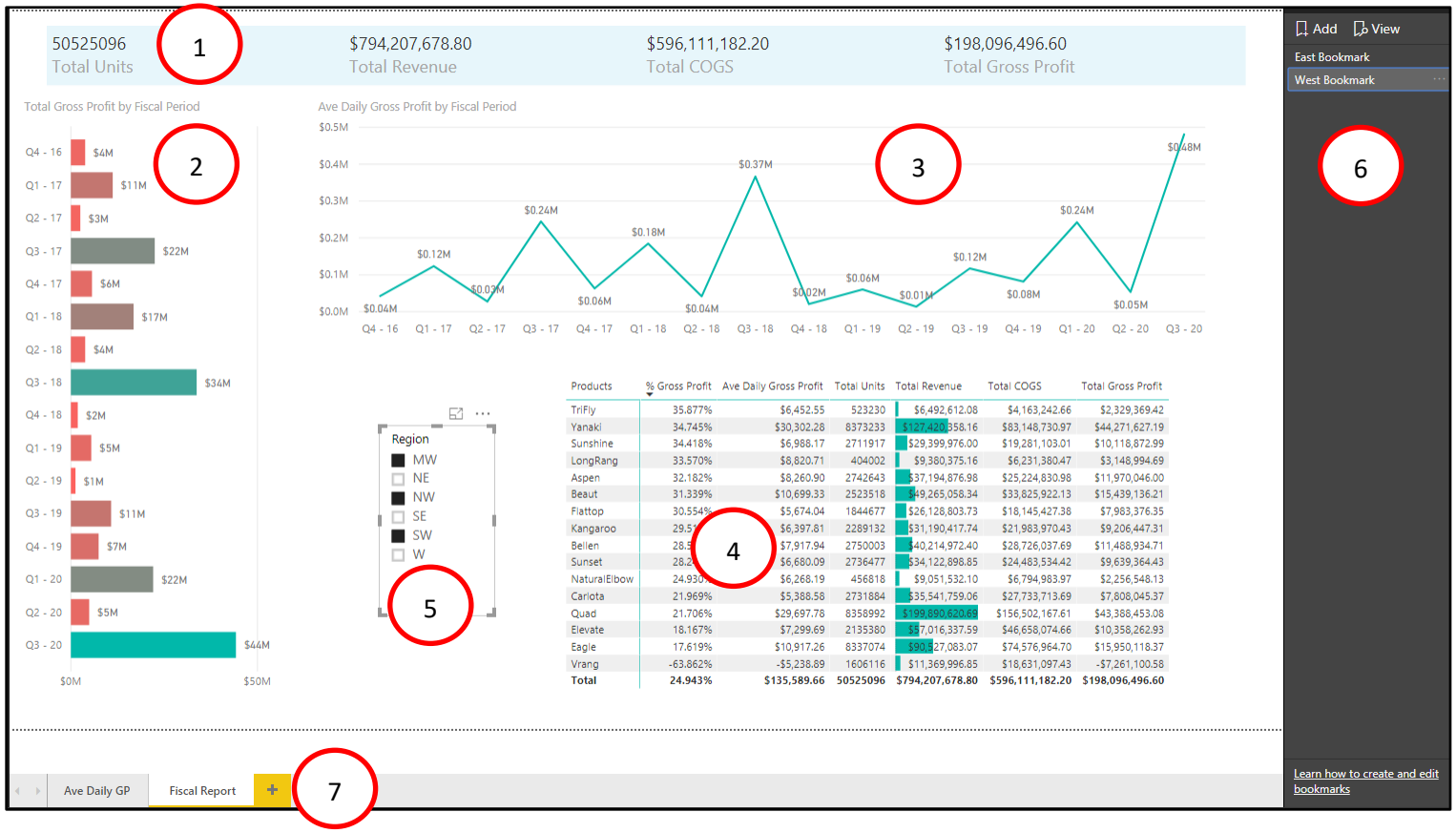
3] Add a Line Chart that shows “Ave Daily Gross Profit” by Fiscal period and add Data Labels.

4] Add a Matrix with Products listed in the Row Area and that lists the Measures % Gross Profit, Ave Daily Gross Profit, Total Units, Total Revenue, Total COGS and Total Gross Profit. Then add Conditional Formatting-Data Bars to the Total Revenue Measure. Finally sort the % Gross Profit Column Biggest to Smallest.

5] Add a Slicer (from Visualization Pane)

6] Bookmark . From the View Ribbon Tab in the Power BI Desktop Window, check the Box for Bookmark, then add two bookmarks, one for NE and SE Region, and then a second bookmark for the remaining Regions.

7] Name the Tab “Fiscal Report”.



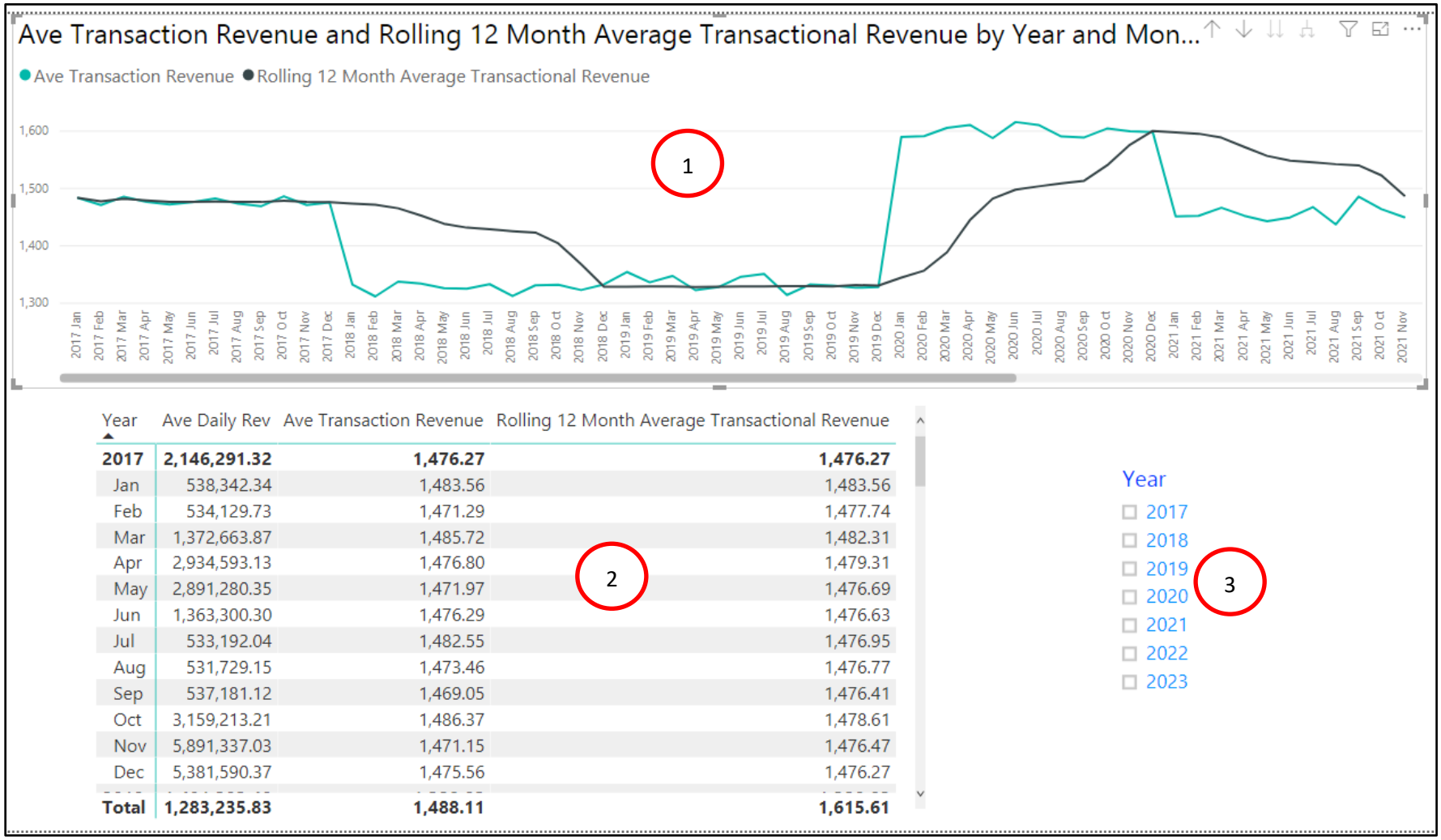
Products	% Gross Profit	Ave Daily Gross Profit	Total Units	Total Revenue	Total COGS	Total Gross Profit
TriFly	35.877%	$6,452.55	523230	$6,492,612.08	$4,163,242.66	$2,329,369.42
Yanaki	34.745%	$30,302.28	8373233	$127,420,358.16	$83,148,730.97	$44,271,627.19
Sunshine	34.418%	$6,988.17	2711917	$29,399,976.00	$19,281,103.01	$10,118,872.99
LongRang	33.570%	$8,820.71	404002	$9,380,375.16	$6,231,380.47	$3,148,994.69
Aspen	32.182%	$8,260.90	2742643	$37,194,876.98	$25,224,830.98	$11,970,046.00
Beaut	31.339%	$10,699.33	2523518	$49,265,058.34	$33,825,922.13	$15,439,136.21
Flattop	30.554%	$5,674.04	1844677	$26,128,803.73	$18,145,427.38	$7,983,376.35
Kangaroo	29.5%	$6,397.81	2289132	$31,190,417.74	$21,983,970.43	$9,206,447.31
Bellen	28.5%	$7,917.94	2750003	$40,214,972.40	$28,726,037.69	$11,488,934.71
Sunset	28.2%	$6,680.09	2736477	$34,122,898.85	$24,483,534.42	$9,639,364.43
NaturalElbow	24.930%	$6,268.19	456818	$9,051,532.10	$6,794,983.97	$2,256,548.13
Carlota	21.969%	$5,388.58	2731884	$35,541,759.06	$27,733,713.69	$7,808,045.37
Quad	21.706%	$29,697.78	8358992	$199,890,620.66	$156,502,167.61	$43,388,453.08
Elevate	18.167%	$7,299.69	2135380	$17,016,337.59	$46,658,074.66	$10,358,262.93
Eagle	17.619%	$10,917.26	8337074	$90,127,083.07	$74,576,964.70	$15,950,118.37
Vrang	-63.862%	-$5,238.89	1606116	$11,369,996.85	$18,631,097.43	-$7,261,100.58
Total	**24.943%**	**$135,589.66**	**50525096**	**$794,207,678.80**	**$596,111,182.20**	**$198,096,496.60**

45) Create “Ave Last 12 Months” Tab (Page) :

1] Add a Line Chart with the two Fields Year and Month in the Axis, and the Measures Ave Transaction Revenue and Rolling 12 Month Average Transactional Revenue in the Values area. For both the Title and Legend, format them to have a Font size of 12 and Bold.

2] Add a Matrix with Year and Month in the Row area and the Measures Ave Daily Rev, Ave Transaction Revenue and Rolling 12 Month Average Transactional Revenue in the Values area. Use the “Expand Down” Icon to show both Year and Month in the Matrix.

3] Add a Slicer for Year. In the upper left corner of the Slicer, use the dropdown to make the Slicer show the Years as a list.



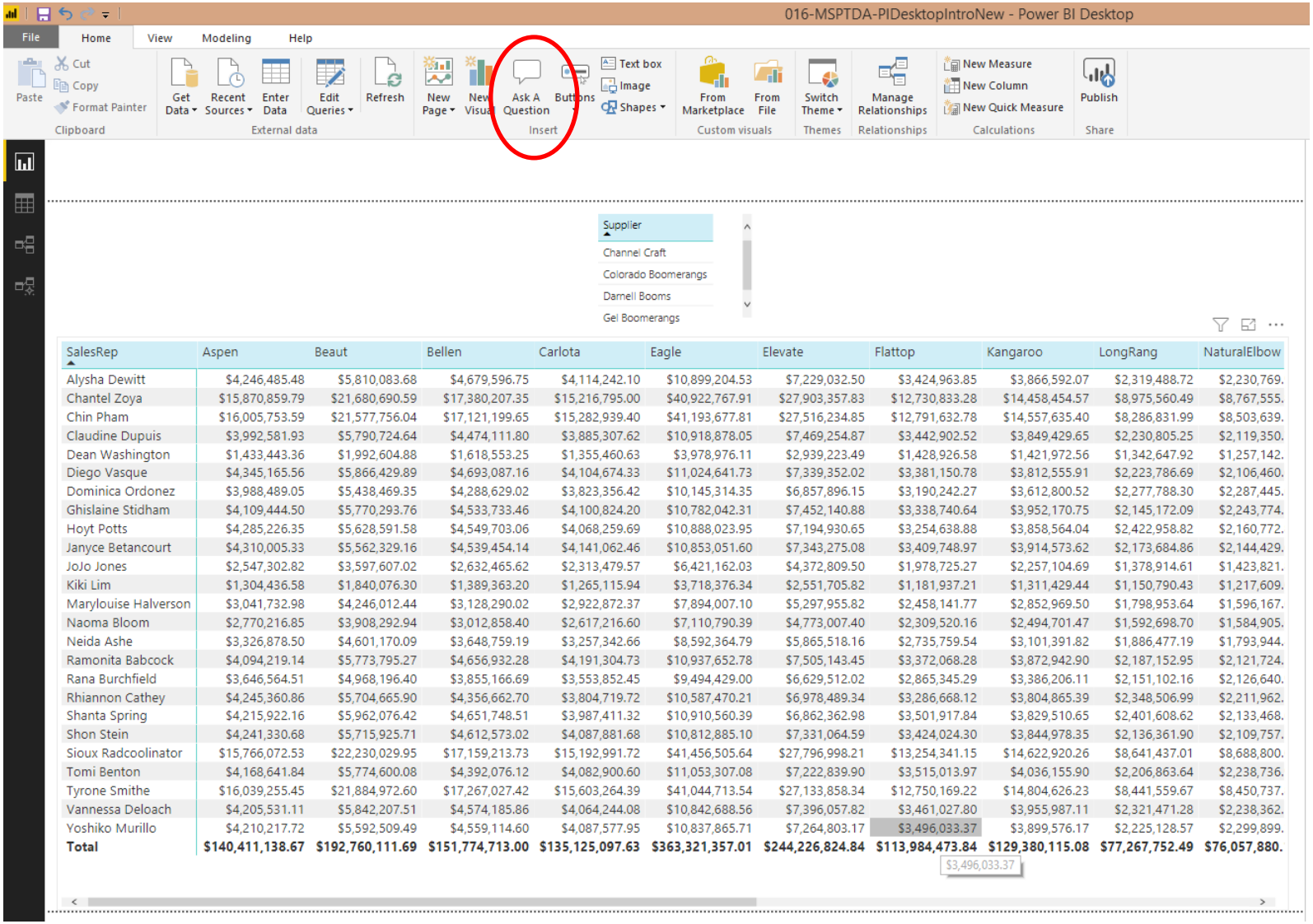
Year	Ave Daily Rev	Ave Transaction Revenue	Rolling 12 Month Average Transactional Revenue
2017	**2,146,291.32**	**1,476.27**	**1,476.27**
Jan	538,342.34	1,483.56	1,483.56
Feb	534,129.73	1,471.29	1,477.74
Mar	1,372,663.87	1,485.72	1,482.31
Apr	2,934,593.13	1,476.80	1,479.31
May	2,891,280.35	1,471.97	1,476.69
Jun	1,363,300.30	1,476.29	1,476.63
Jul	533,192.04	1,482.55	1,476.95
Aug	531,729.15	1,473.46	1,476.77
Sep	537,181.12	1,469.05	1,476.41
Oct	3,159,213.21	1,486.37	1,478.61
Nov	5,891,337.03	1,471.15	1,476.47
Dec	5,381,590.37	1,475.56	1,476.27
Total	**1,283,235.83**	**1,488.11**	**1,615.61**

46) Create “Question” Tab (Page) :

1] In this example, we wanted to learn how to use the “Ask A Question” option in the Insert group in the Home Ribbon Tab in the Power BI Desktop Window. We typed out these two questions:

1. “Matrix Total Revenue SalesRep by Product”
2. “Supplier Slicer”.

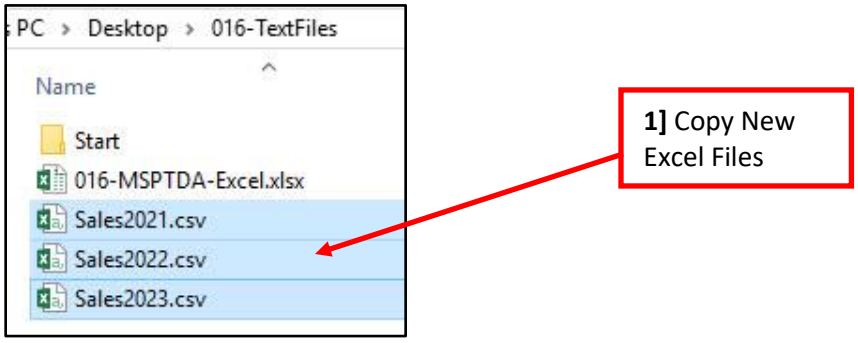
2] Then we added some formatting



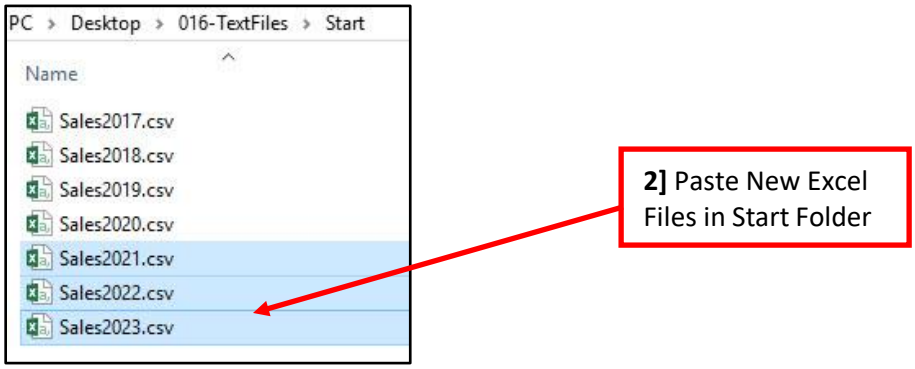
SalesRep	Aspen	Beaut	Bellen	Carlota	Eagle	Elevate	Flattop	Kangaroo	LongRang	NaturalElbow
Alysha Dewitt	$4,246,485.48	$5,810,083.68	$4,679,596.75	$4,114,242.10	$10,899,204.53	$7,229,032.50	$3,424,963.85	$3,866,592.07	$2,319,488.72	$2,230,769.
Chantel Zoya	$15,870,859.79	$21,680,690.59	$17,380,207.35	$15,216,795.00	$40,922,767.91	$27,903,357.83	$12,730,833.28	$14,458,454.57	$8,975,560.49	$8,767,555.
Chin Pham	$16,005,753.59	$21,577,756.04	$17,121,199.65	$15,282,939.40	$41,193,677.81	$27,516,234.85	$12,791,632.78	$14,557,635.40	$8,286,831.99	$8,503,639.
Claudine Dupuis	$3,992,581.93	$5,790,724.64	$4,474,111.80	$3,885,307.62	$10,918,878.05	$7,469,254.87	$3,442,902.52	$3,849,429.65	$2,230,805.25	$2,119,350.
Dean Washington	$1,433,443.36	$1,992,604.88	$1,618,553.25	$1,355,460.63	$3,978,976.11	$2,939,223.49	$1,428,926.58	$1,421,972.56	$1,342,647.92	$1,257,142.
Diego Vasque	$4,345,165.56	$5,866,429.89	$4,693,087.16	$4,104,674.33	$11,024,641.73	$7,339,352.02	$3,381,150.78	$3,812,555.91	$2,223,786.69	$2,106,460.
Dominica Ordonez	$3,988,489.05	$5,438,469.35	$4,288,629.02	$3,823,356.42	$10,145,314.35	$6,857,896.15	$3,190,242.27	$3,612,800.52	$2,277,788.30	$2,287,445.
Ghislaine Stidham	$4,109,444.50	$5,770,293.76	$4,533,733.46	$4,100,824.20	$10,782,042.31	$7,452,140.88	$3,338,740.64	$3,952,170.75	$2,145,172.09	$2,243,774.
Hoyt Potts	$4,285,226.35	$5,628,591.58	$4,549,703.06	$4,068,259.69	$10,888,023.95	$7,194,930.65	$3,254,638.88	$3,858,564.04	$2,422,958.82	$2,160,772.
Janyce Betancourt	$4,310,005.33	$5,562,329.16	$4,539,454.14	$4,141,062.46	$10,853,051.60	$7,343,275.08	$3,409,748.97	$3,914,573.62	$2,173,684.86	$2,144,429.
JoJo Jones	$2,547,302.82	$3,597,607.02	$2,632,465.62	$2,313,479.57	$6,421,162.03	$4,372,809.50	$1,978,725.27	$2,257,104.69	$1,378,914.61	$1,423,821.
Kiki Lim	$1,304,436.58	$1,840,076.30	$1,389,363.20	$1,265,115.94	$3,718,376.34	$2,551,705.82	$1,181,937.21	$1,311,429.44	$1,150,790.43	$1,217,609.
Marylouise Halverson	$3,041,732.98	$4,246,012.44	$3,128,290.02	$2,922,872.37	$7,894,007.10	$5,297,955.82	$2,458,141.77	$2,852,969.50	$1,798,953.64	$1,596,167.
Naoma Bloom	$2,770,216.85	$3,908,292.94	$3,012,858.40	$2,617,216.60	$7,110,790.39	$4,773,007.40	$2,309,520.16	$2,494,701.47	$1,592,698.70	$1,584,905.
Neida Ashe	$3,326,878.50	$4,601,170.09	$3,648,759.19	$3,257,342.66	$8,592,364.79	$5,865,518.16	$2,735,759.54	$3,101,391.82	$1,886,477.19	$1,793,944.
Ramonita Babcock	$4,094,219.14	$5,773,795.27	$4,656,932.28	$4,191,304.73	$10,937,652.78	$7,505,143.45	$3,372,068.28	$3,872,942.90	$2,187,152.95	$2,121,724.
Rana Burchfield	$3,646,564.51	$4,968,196.40	$3,855,166.69	$3,553,852.45	$9,494,429.00	$6,629,512.02	$2,865,345.29	$3,386,206.11	$2,151,102.16	$2,126,640.
Rhiannon Cathey	$4,245,360.86	$5,704,665.90	$4,356,662.70	$3,804,719.72	$10,587,470.21	$6,978,489.34	$3,286,668.12	$3,804,865.39	$2,348,506.99	$2,211,962.
Shanta Spring	$4,215,922.16	$5,962,076.42	$4,651,748.51	$3,987,411.32	$10,910,560.39	$6,862,362.98	$3,501,917.84	$3,829,510.65	$2,401,608.62	$2,133,468.
Shon Stein	$4,241,330.68	$5,715,925.71	$4,612,573.02	$4,087,881.68	$10,812,885.10	$7,331,064.59	$3,424,024.30	$3,844,978.35	$2,136,361.90	$2,109,757.
Sioux Radcoolinator	$15,766,072.53	$22,230,029.95	$17,159,213.73	$15,192,991.72	$41,456,505.64	$27,796,998.21	$13,254,341.15	$14,622,920.26	$8,641,437.01	$8,688,800.
Tomi Benton	$4,168,641.84	$5,774,600.08	$4,392,076.12	$4,082,900.60	$11,053,307.08	$7,222,839.90	$3,515,013.97	$4,036,155.90	$2,206,863.64	$2,238,736.
Tyrone Smithe	$16,039,255.45	$21,884,972.60	$17,267,027.42	$15,603,264.39	$41,044,713.54	$27,133,858.34	$12,750,169.22	$14,804,626.23	$8,441,559.67	$8,450,737.
Vannessa Deloach	$4,205,531.11	$5,842,207.51	$4,574,185.86	$4,064,244.08	$10,842,688.56	$7,396,057.82	$3,461,027.80	$3,955,987.11	$2,321,471.28	$2,238,362.
Yoshiko Murillo	$4,210,217.72	$5,592,509.49	$4,559,114.60	$4,087,577.95	$10,837,865.71	$7,264,803.17	$3,496,033.37	$3,899,576.17	$2,225,128.57	$2,299,899.
Total	**$140,411,138.67**	**$192,760,111.69**	**$151,774,713.00**	**$135,125,097.63**	**$363,321,357.01**	**$244,226,824.84**	**$113,984,473.84**	**$129,380,115.08**	**$77,267,752.49**	**$76,057,880.**

47) Source Data Changes and Refresh :

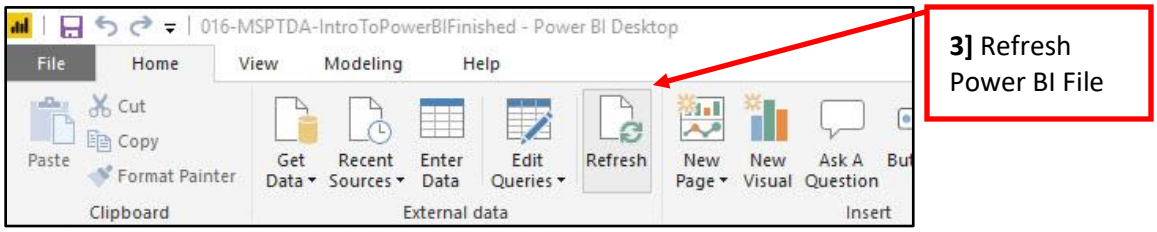
1] Copy the Excel files for the years 2021 to 2023.



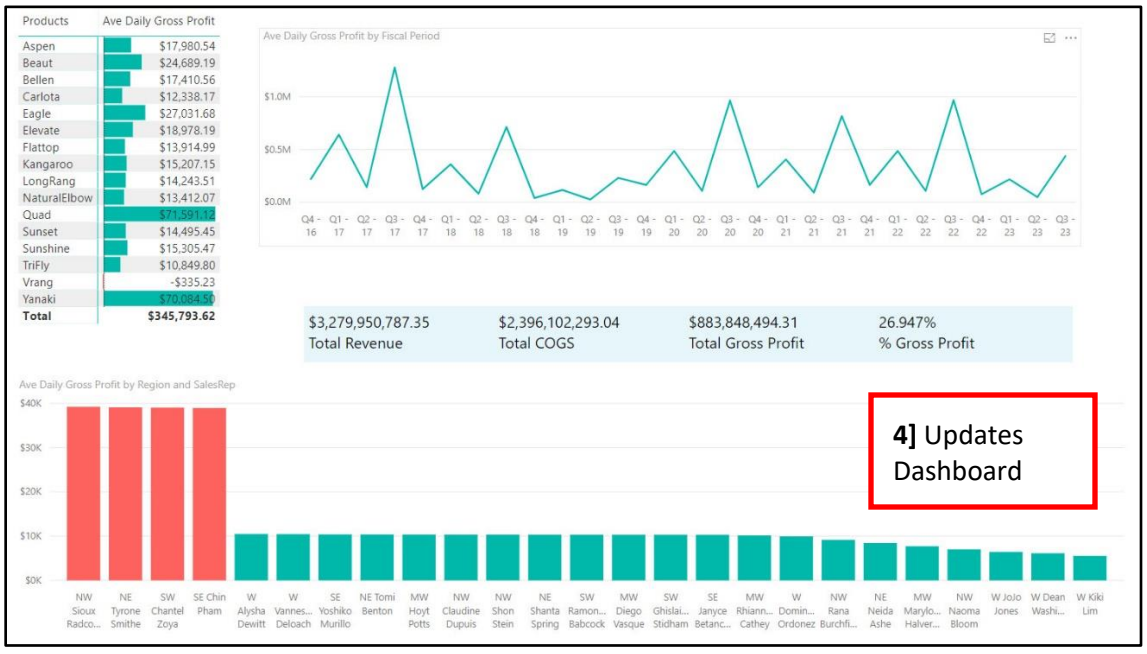
2] Paste the Excel files for the years 2021 to 2023 into the Start Folder.



3] In the Home Ribbon Tab, click the Refresh button.



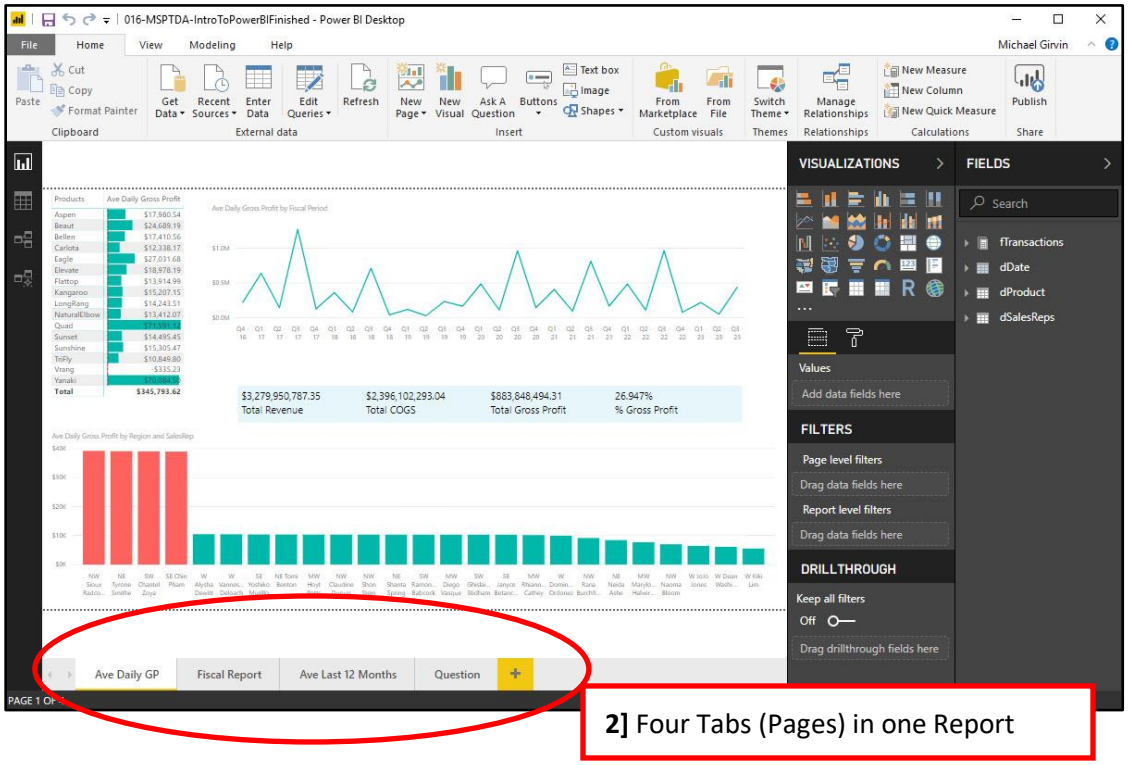
4] After refreshing, the Dashboards are updated with the latest data



48) Publish Report so Tabs (Pages) can be consumed on any Device :

1] After we save, we now have a Power BI Desktop file with the extension ".pbix".

2] All of the Tabs together are called the report, as seen here:



3] The report can be published to powerbi.com so that the Report and the Tabs (Pages) can be consumed on any device.

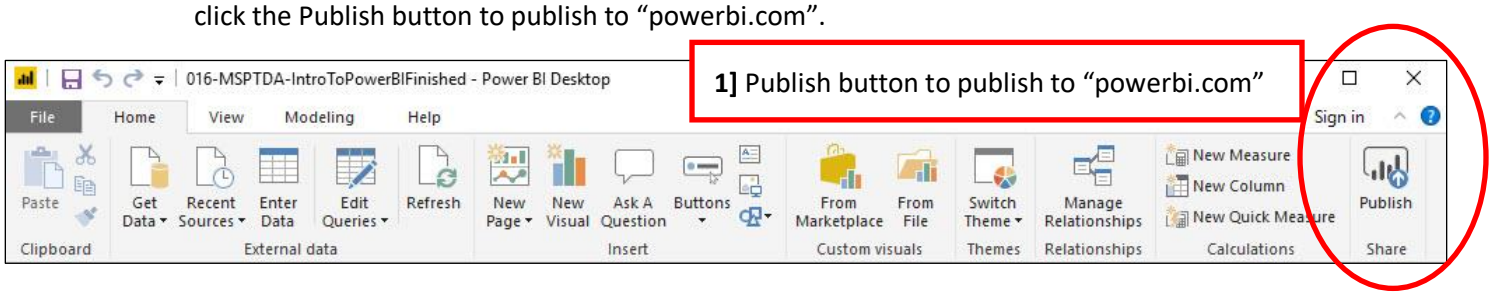
4] In order to share:

1. You need a Power BI Pro license, https://docs.microsoft.com/en-us/power-bi/service-features-license-type
   or
2. We can Publish to Web and report will be available to public.

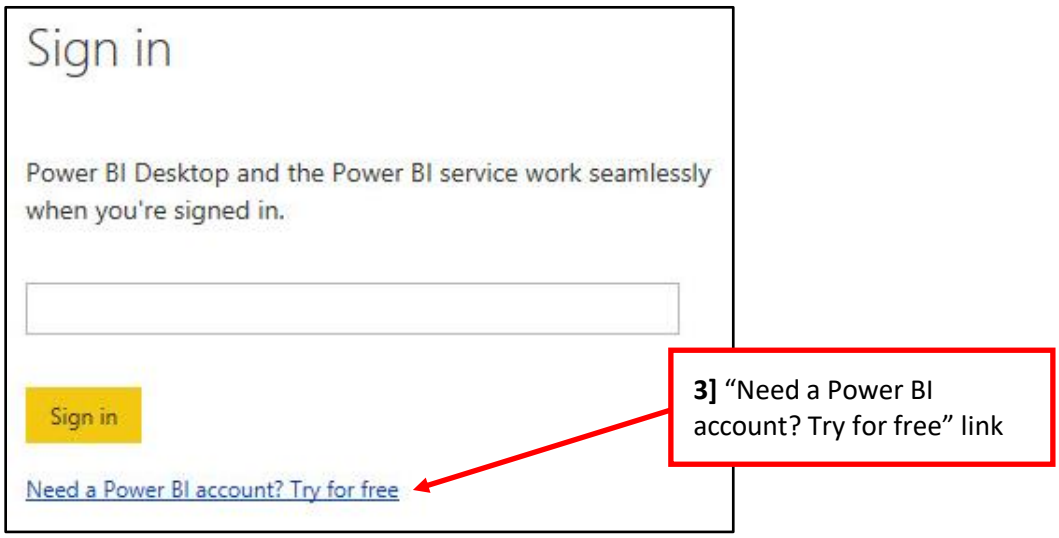
49) Publish and Share with Power BI Pro Account :

1] To Publish you will need an account, either an individual account or an account through your employer. If you don't have an account, you can create an account after you click the Publish button.

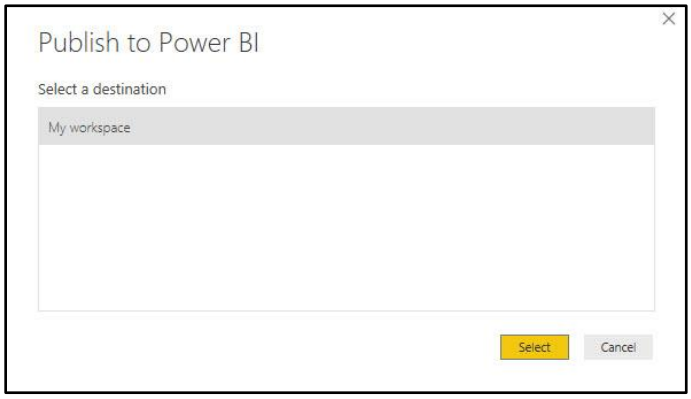
2] To publish the report, go to the Home Ribbon Tab, then all the way on the right side, in the Share group, click the Publish button to publish to "powerbi.com".



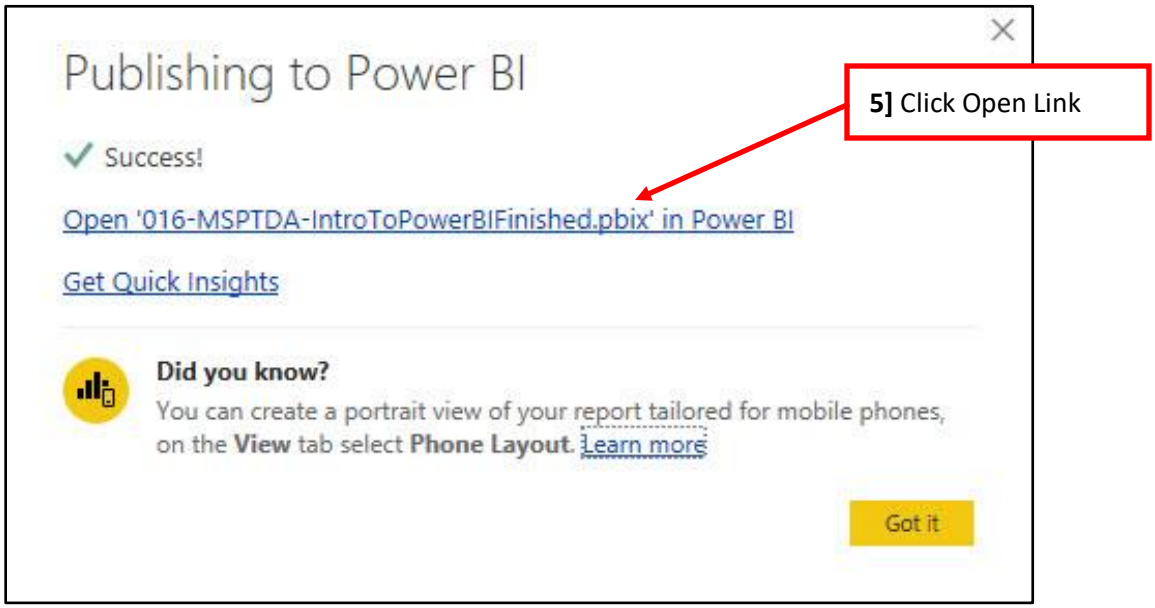
3] Sign in or use the “Need a Power BI account? Try for free” link.



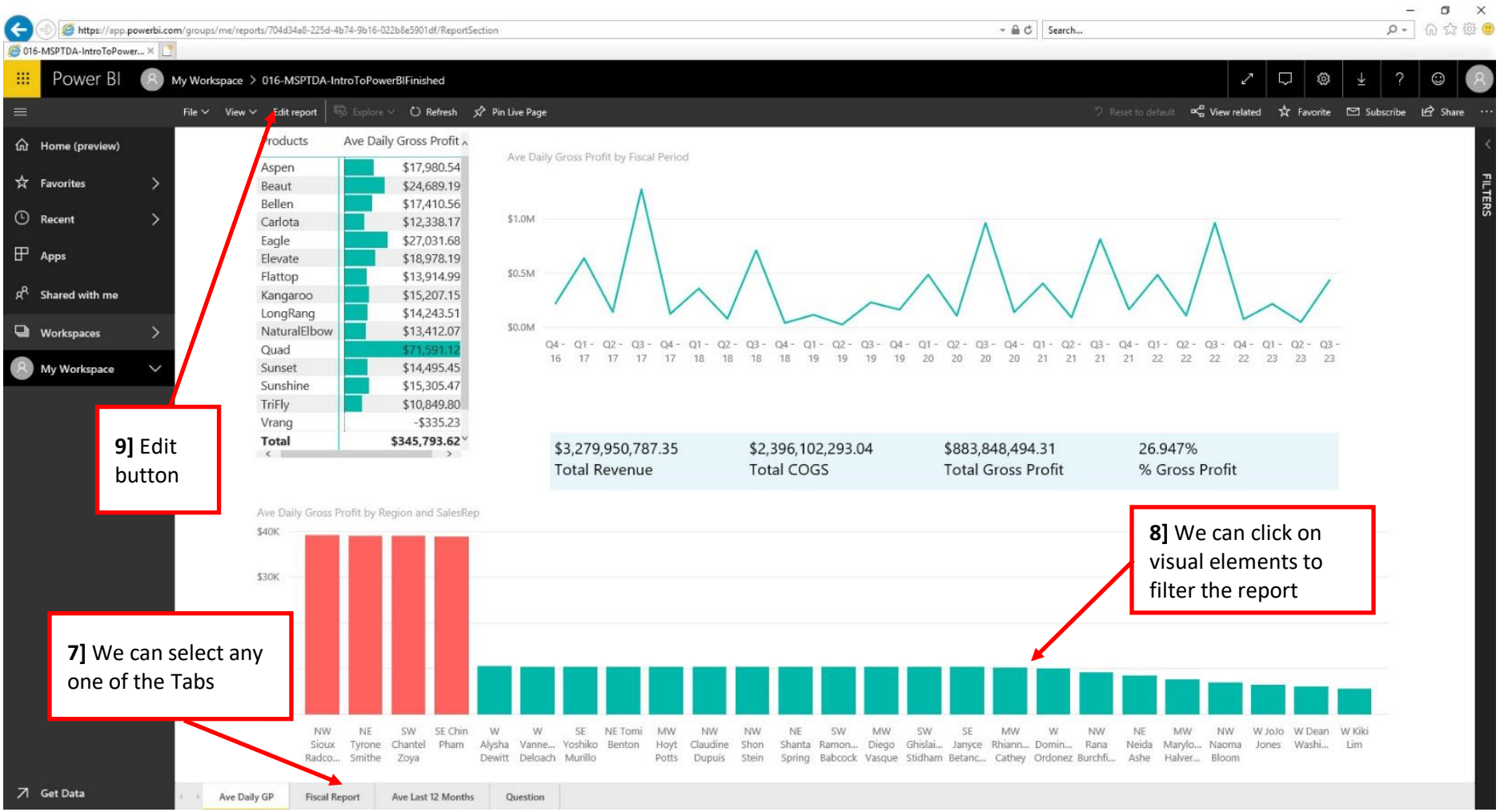
4] In next window, we publish to My Workspace.



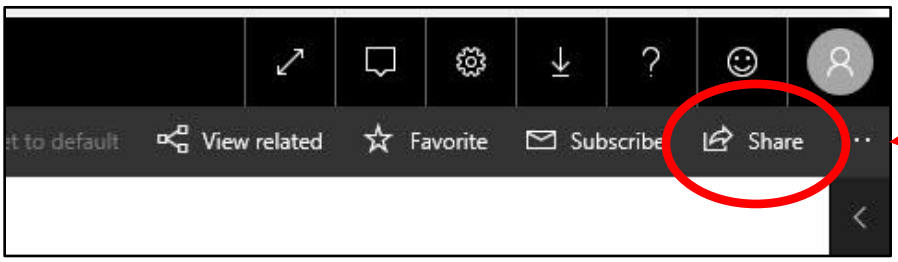
5] Click open link to open report in the powerbi.com web site.



6] We can view the report in powerbi.com
7] We can click on any one of the tabs
8] We can click on any one of the visual elements to filter the report.
9] We can edit the report using the Edit button.



10] If you have Power BI Pro, you can use the Share button:



10] If you have Power BI Pro, you can use the Share button

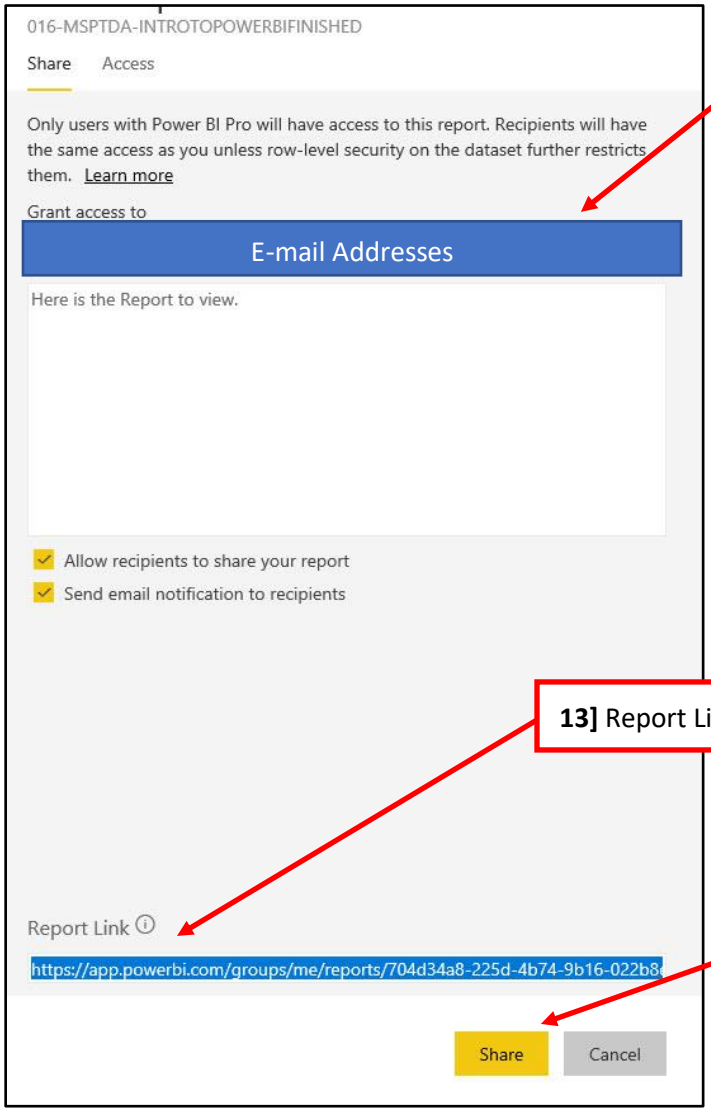
11] After clicking Share, you can try it free for 60 days:

11] Try for 60 days.

12] Now when we click Share, we can add the e-mails of the people who can see the report.

13] Report Link is available.

14] Click Share and e-mails will be sent.



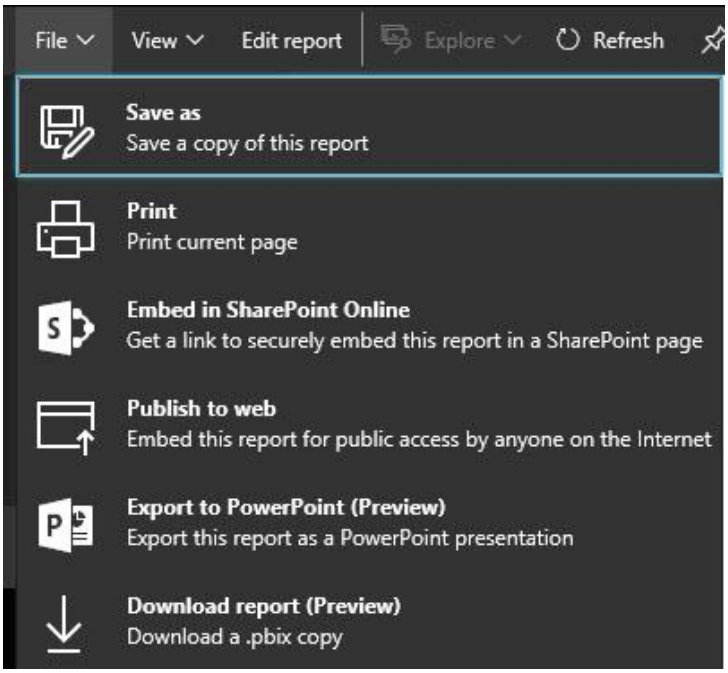
12] List e-mails

13] Report Link

14] Click Share

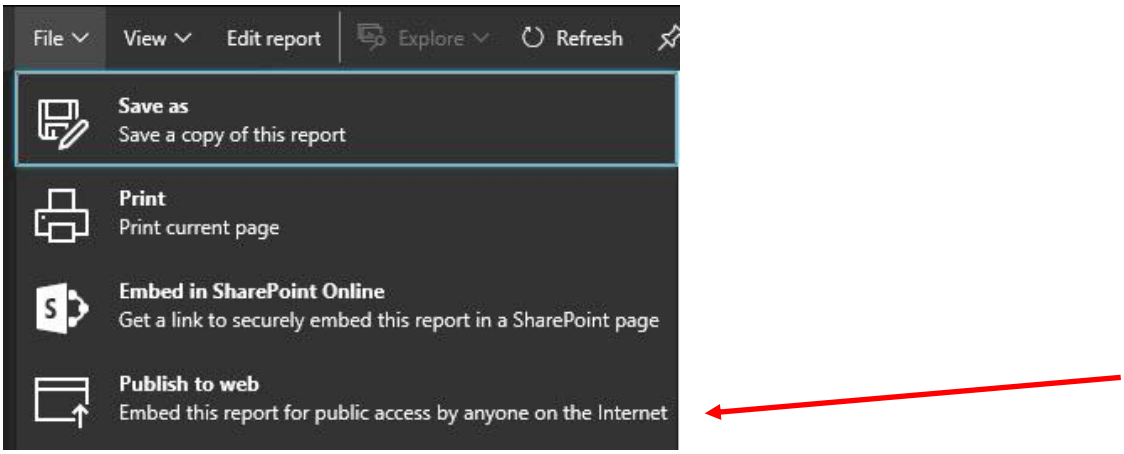
50) From the published report at powerbi.com, we can use the File menu to :

1] Save As.
2] Print Current Page.
3] Embed Code for SharePoint Online.
4] Publish to Web for public consumption (get embed code to use in a web site).
5] Export as a PowerPoint Slide Show.
6] Download ".pbix" file.



51) Publish to Web with Free Power BI Desktop version and allow public to review Report :

1] From File Menu, select "Publish to web", then copy code and paste code into web site editing tool.



52) In Power BI Desktop, if we use Ctrl + P (Print) :

i. In Power BI Desktop, if we use Ctrl + P (Print), all of our Dashboards (each tab in our report) will be published as a pdf file

53) Power BI Web site :

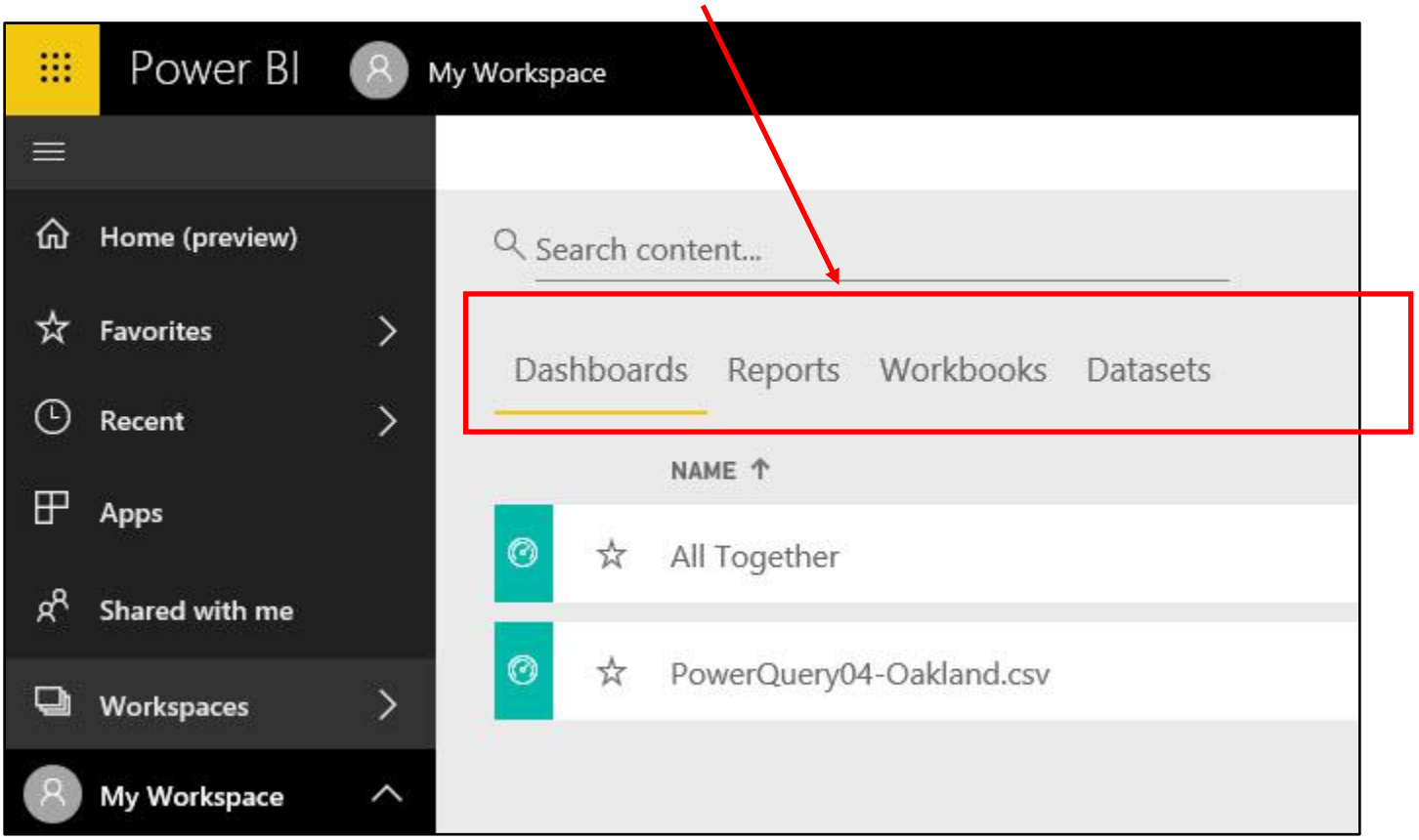
i. To log in to see published reports and accomplish other Power BI Online goals, go to powerbi.com and log in.

54) Dashboards, Reports, Workbooks or Data Sets (Terminology for Power BI Report and Power BI Online) :

i. Power BI File ".pbix":
    1. Report = Power BI file:
        i. Created with Power BI Desktop
        ii. Contains:
            1. The Tabs (Pages)
            2. The Data Model connected to the Columnar Database.
    2. Data Model contains:
        i. Tables that are stored in Columnar Database.
        ii. Relationships between tables.
        iii. DAX Formulas.
        iv. Hierarchies.
    3. Tab (Page) in Power BI File = A Page with a set of visuals.

ii. Power BI Online "powerbi.com":
    1. Workspace = collection of Dashboards, Reports, Workbooks and Data Sets for a specified set of people (e-mail addresses).
    2. Dashboard = a place to pin Pages (Tabs) or individual visuals in Power BI Service website. A Dashboard can contain Pages and visuals from different Reports and Data Sets.
    3. Reports = list of published ".pbix" files.
    4. Workbooks = published Excel Workbooks that we can view in Excel Online.
    5. Data Set in Power BI Report = a single Data Model.
        i. Each Report is connected to One Data Set.
        ii. Dashboards can be connected to Tabs from different Reports and thus different Data Sets.

iii. Picture of the four areas in Power BI Online after you click on a Workspace:

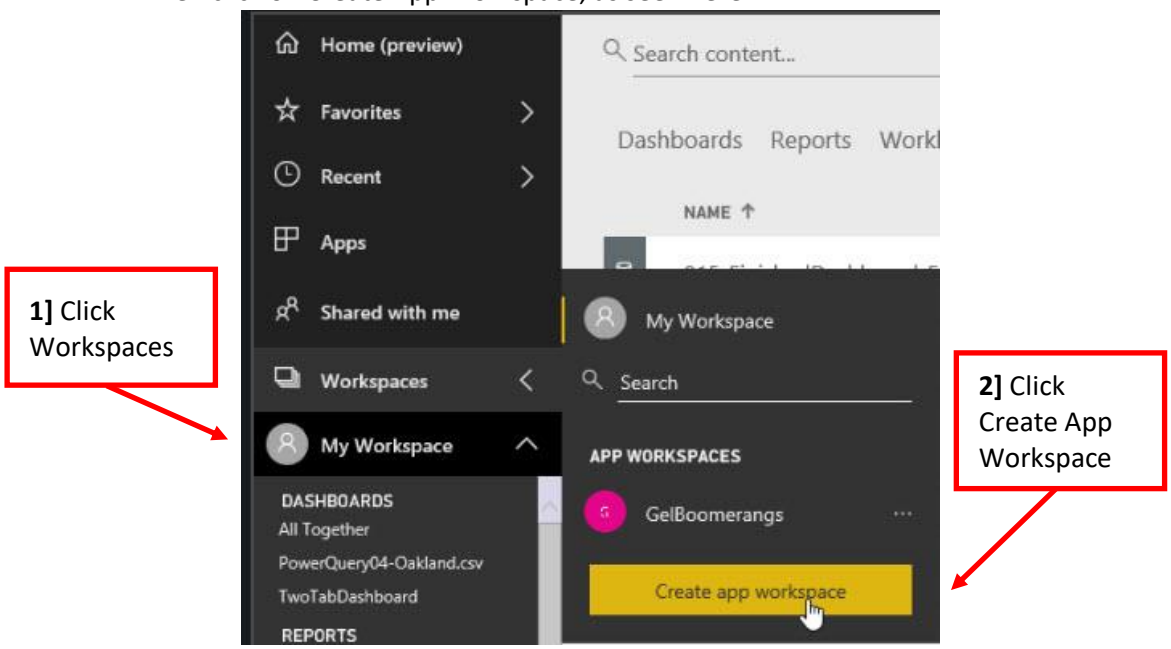


55) Create Workspace :

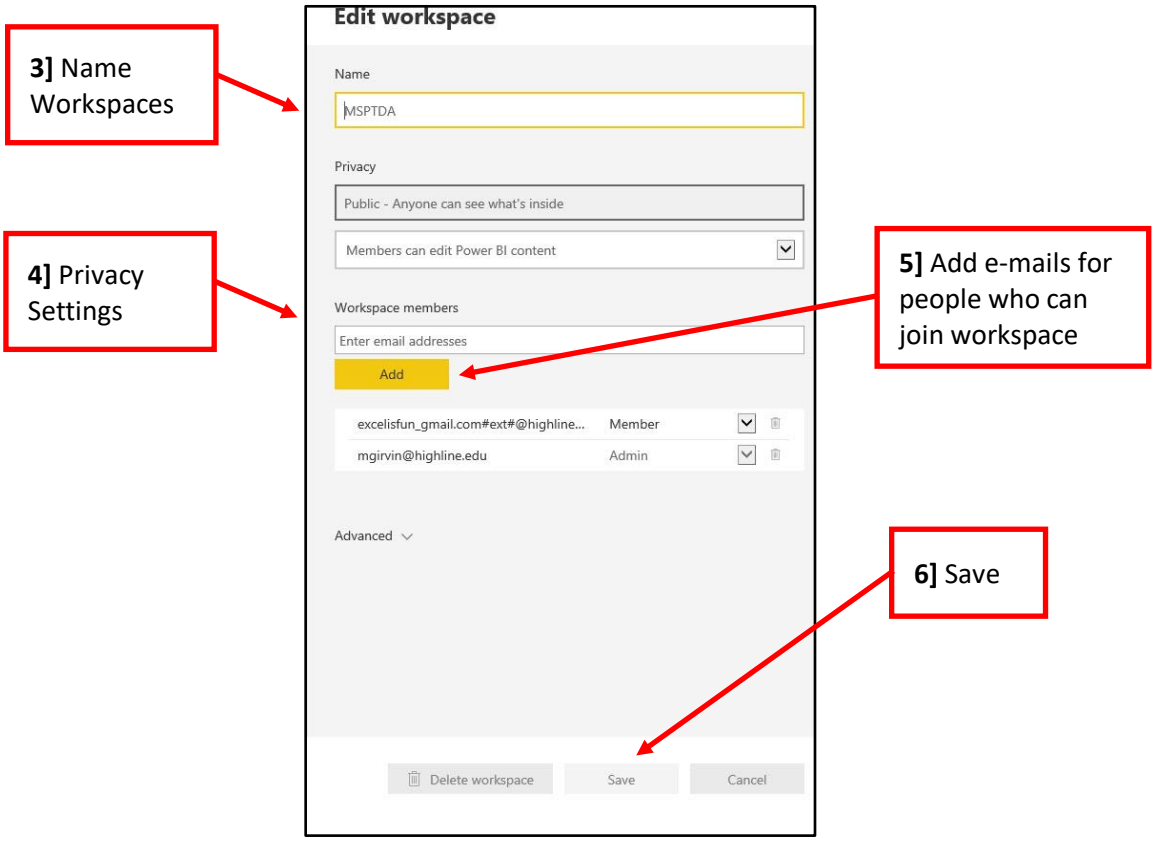
i. Workspace = collection of Dashboards, Reports, Workbooks and Data Sets for a specified set of people (e-mail addresses).

ii. To create Workspace:

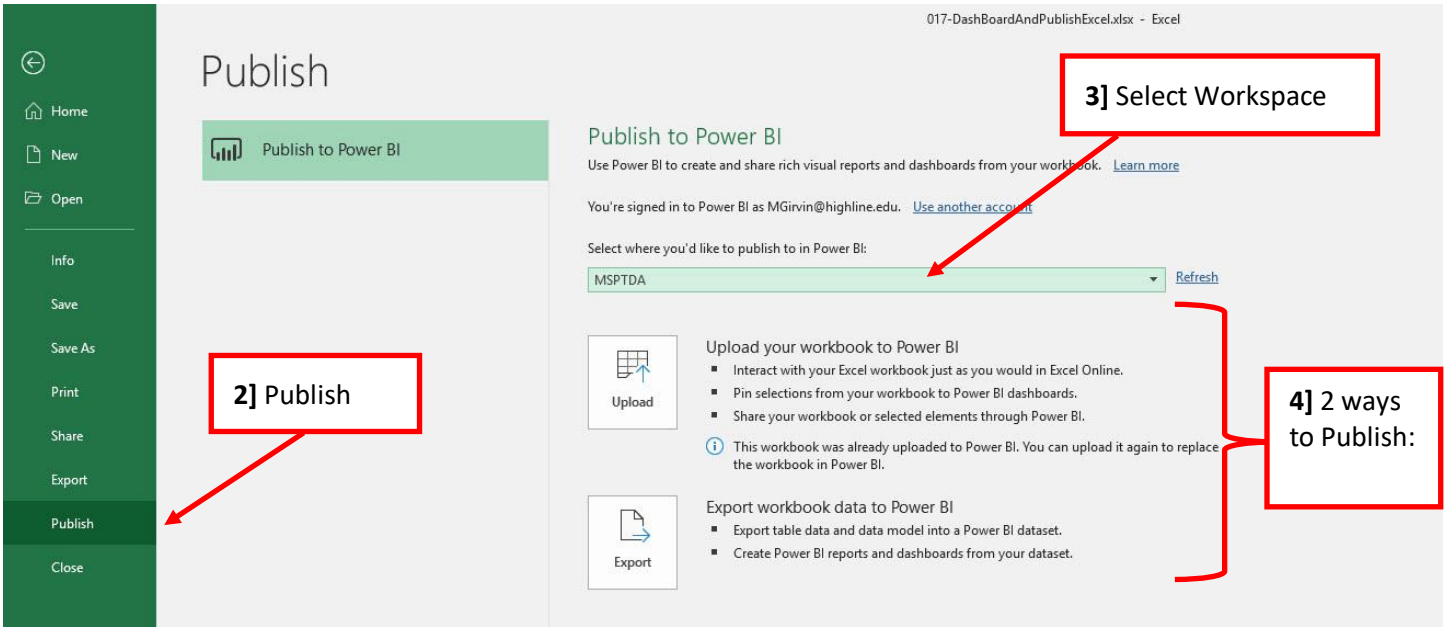
1. On the left click “Workspaces”.
2. Then click on Create App Workspace, as seen here



3. Name the Workspace.
4. Create Privacy Settings.
5. Add e-mails for people who can join workspace.
6. Click Save.



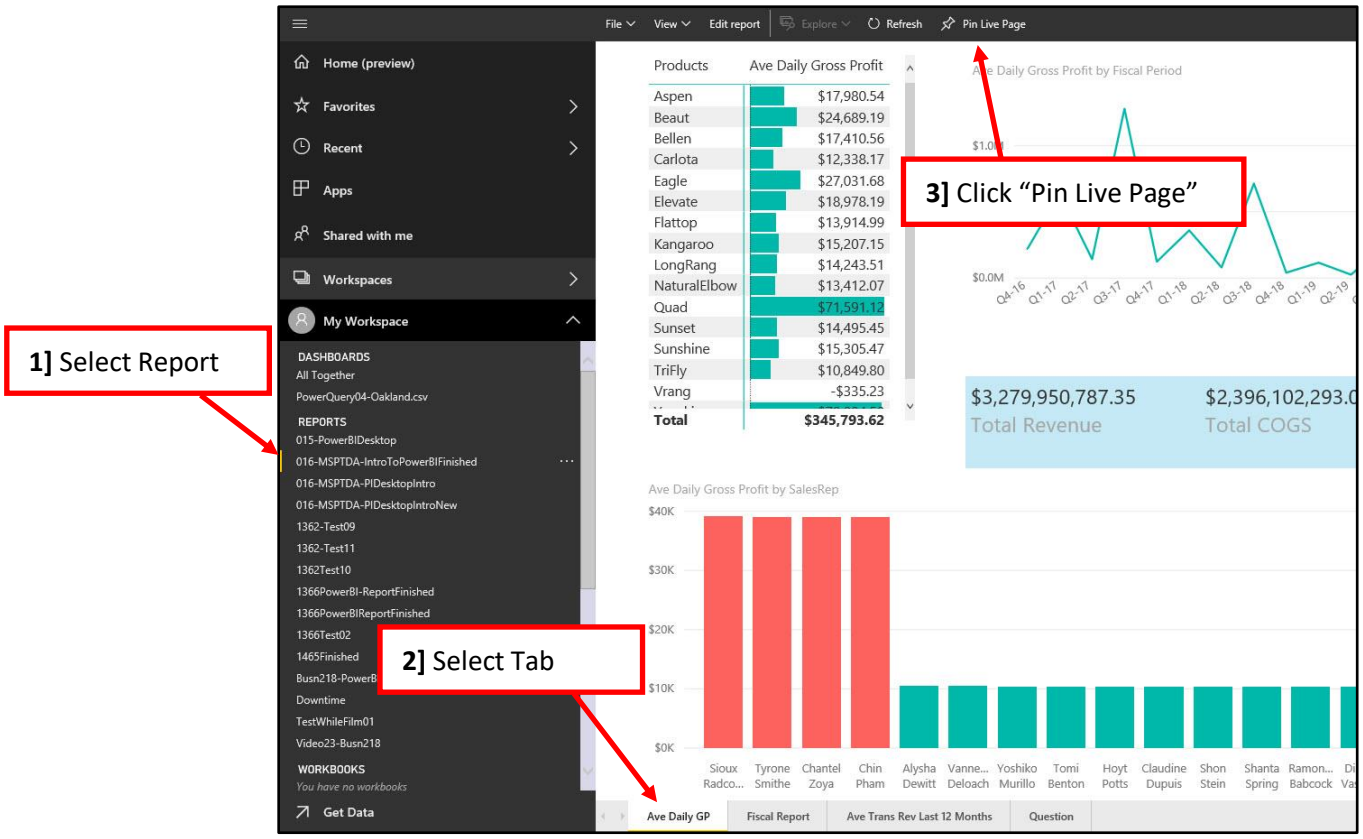
56) Publish Power BI ".pbix" files to Power BI Online (powerbi.com) :
    i. Use the Publish button in the ".pbix" File (as shown earlier in handout).
57) Publish Excel File to Power BI (Upload Excel File to powerbi.com) with Office 365 :
    i. In Excel go to File Menu.
    ii. Then click on Publish.
    iii. Select Workspace.
    iv. In the Publish to Power BI window select one of these options:
        1. Upload your workbook to Power BI
            i. You can't edit your workbook when open in Power BI.
            ii. When you upload, the Data Model is uploaded.
            iii. Your workbook will appear in Reports.
            iv. Choose this option if want to see your worksheets, PivotTables and Charts at Power BI Online (powerbi.com) web site using Excel Online.
        2. Export workbook data to Power BI
            i. The Excel Power Pivot Data Model is exported into a new dataset in Power BI.
            ii. If you have any Power View sheets, those will be re-created in Power BI as reports.
            iii. You can then build a new Power BI Report in the Power BI Online Edit mode.



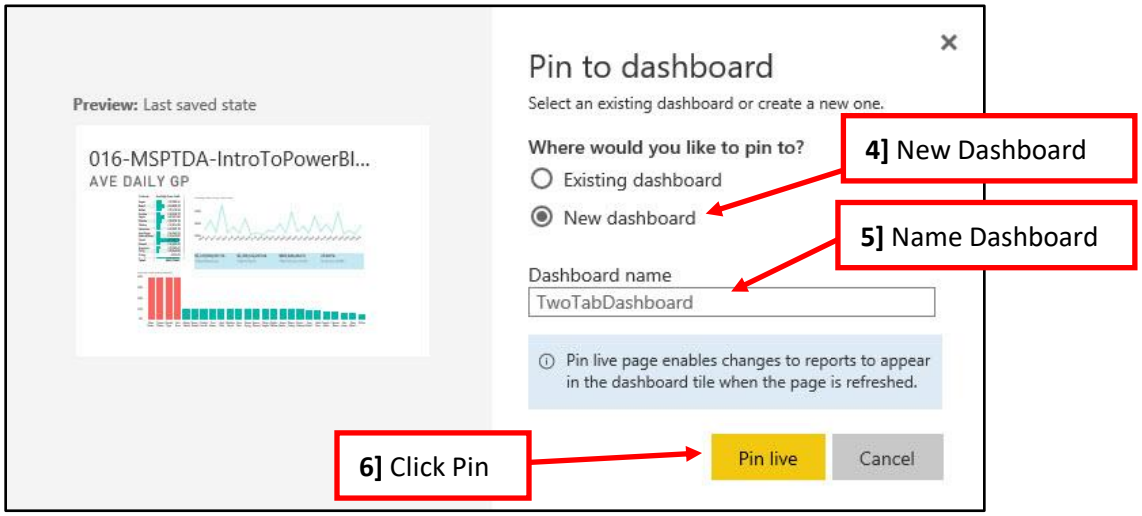
58) Publish Tabs to Power BI Online Dashboard :

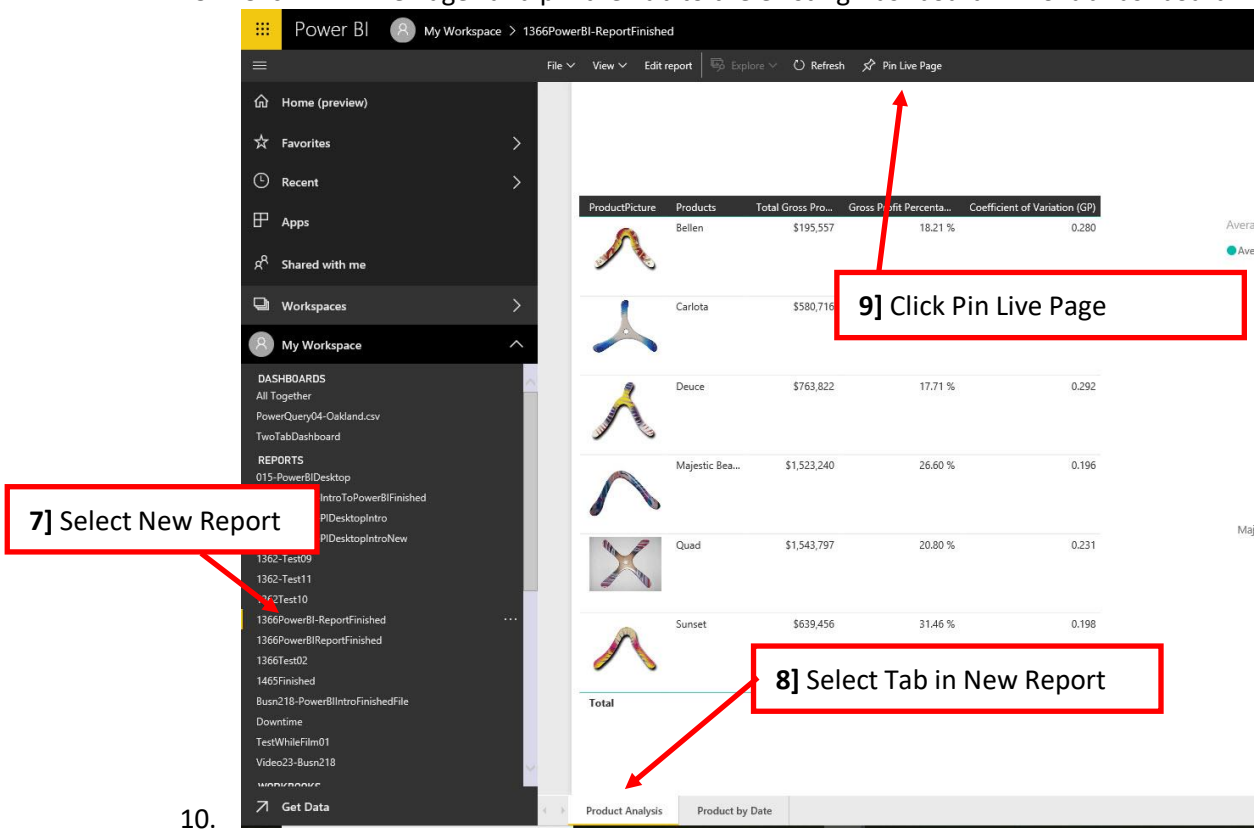
i. Steps to create a Dashboard:

1. Select a Power BI Report.
2. Select a Tab in a Power BI Report.
3. Click “Pin Live Page”.

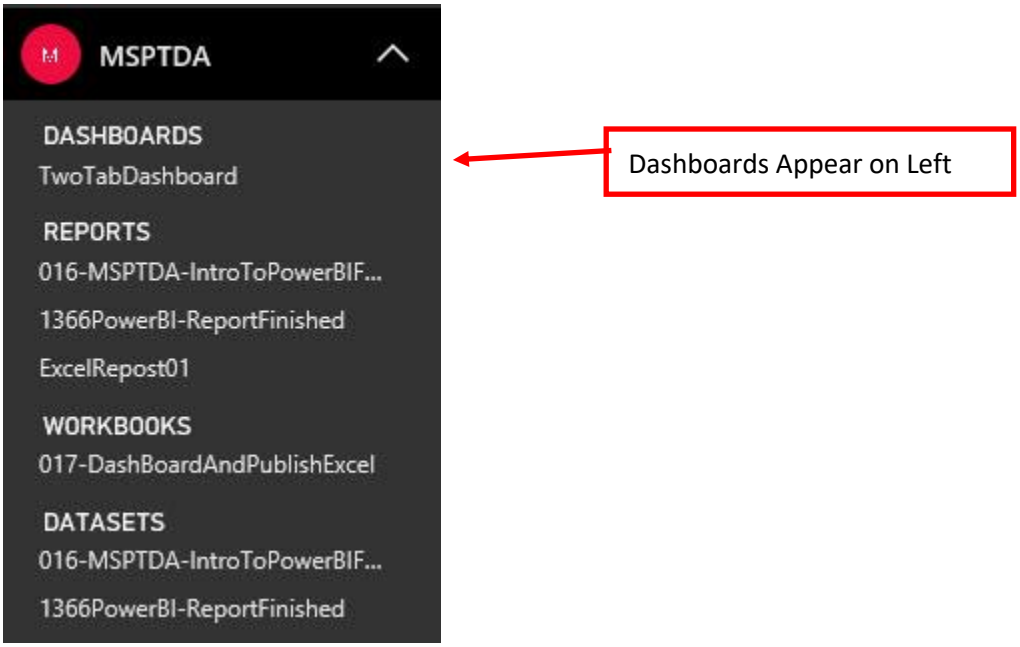


4. Click New Dashboard.
5. Type Dashboard Name: “TwoTabDashboard”.
6. Click “Pin Live”.



7. Select New Report.
8. Select Tab in New Report.
9. Click “Pin Live Page” and pin the Tab to the existing Dashboard “TwoTabDashboard”.
10. 

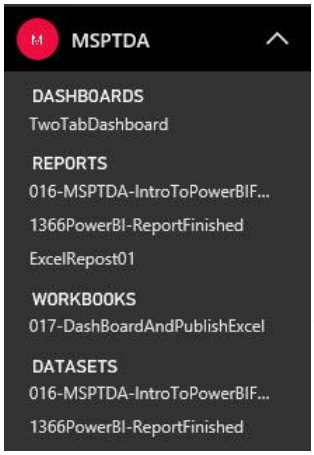
11. All Dashboards will now appear in Dashboard Group in your powerbi.com site:



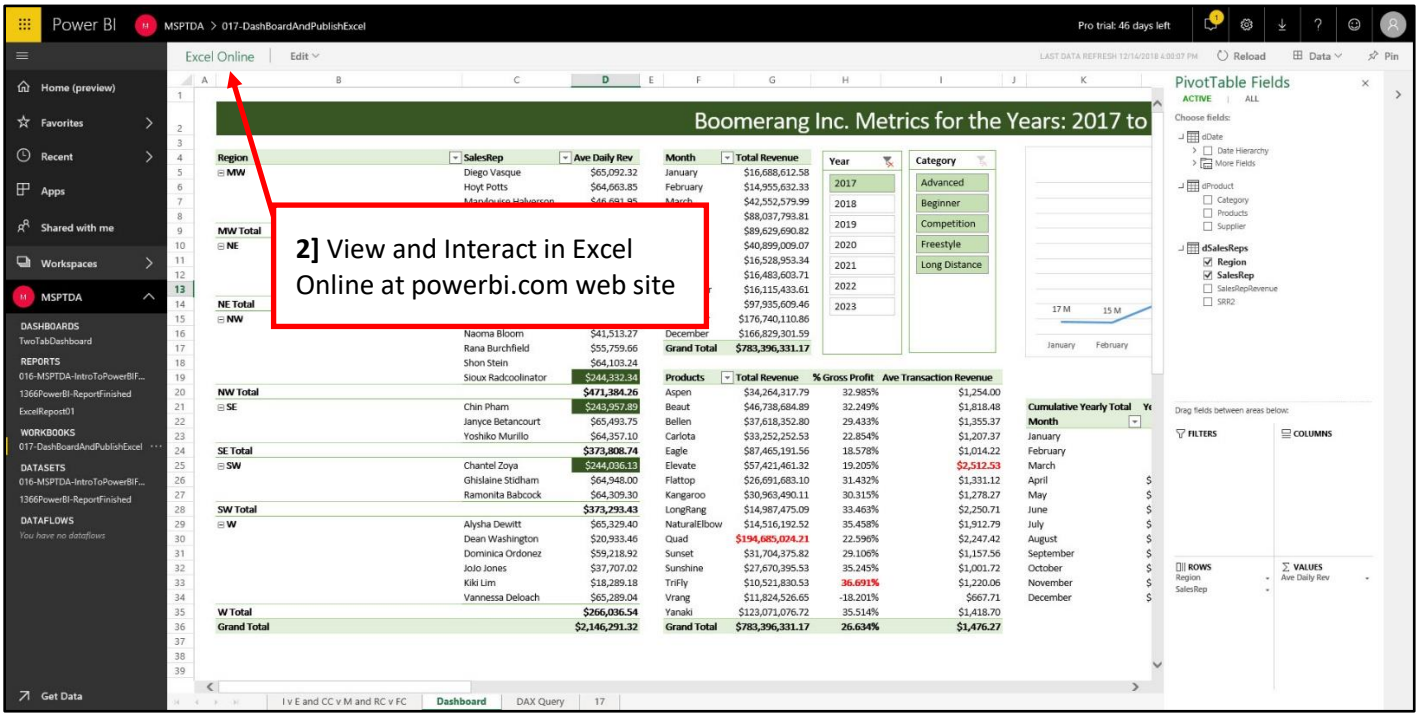
59) View Excel Sheets from Uploaded Excel File :

i. On the Left, click on Workbooks.

1.

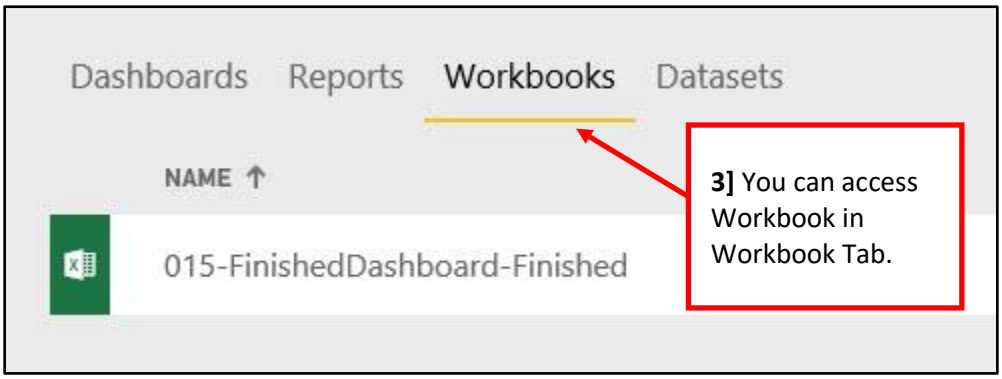


ii. You can view and interact with the Worksheets from the Published Excel Workbook File in Online Excel at the powerbi.com web site:



iii. The Published Workbook File is also listed in the main report menu

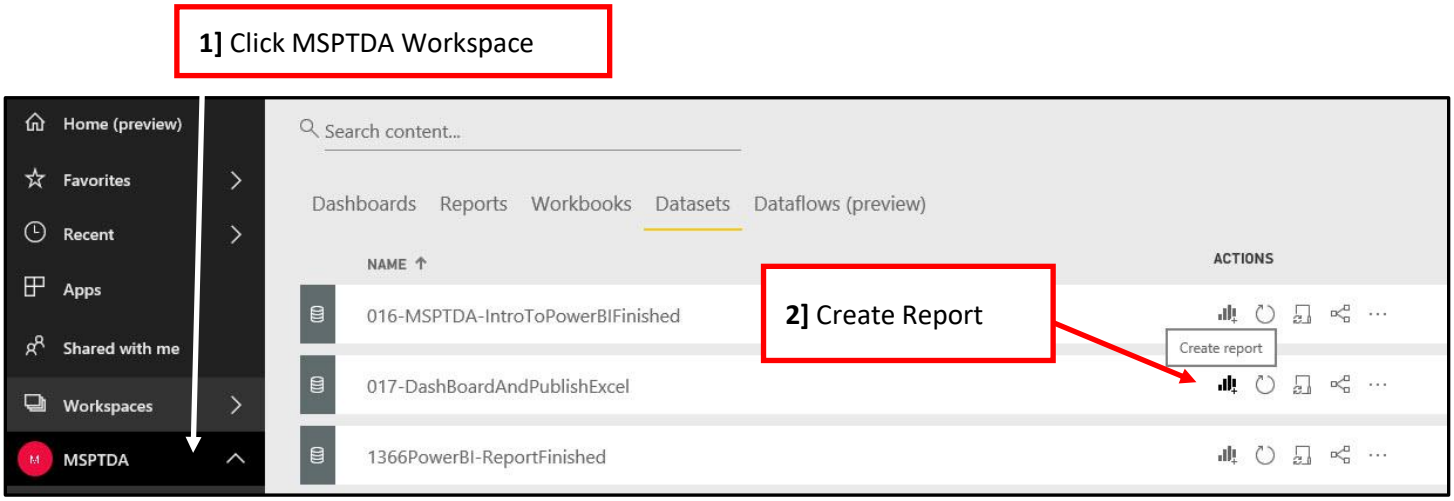
i.



60) Create Power BI Report based on Excel File :

i. On the left click MSPTDA Workspace.

ii. In Data Sets area, you can click on “Create Report” to Create a Power BI Report.



iii. After creating a Power BI Report and saving it, it will appear on the left in the Report area.