**Case Big Data & Data Analytics Dashboard**

**Pivot Table Based on Multiple Tables in Access DataBase Utilizing the Power Pivot Data Model and Dashboard**

**Create Sophisticated Dashboard:**

This assignment will involve demonstrating:

* Classical DAX analysis measures e.g. sum.
* Classical time intelligence measure e.g. “same period last year”.
* Interactive Dashboard chart that requires hyperlinks, pivot table, slicers.

**Objectives:**

The objectives of this assignment are:

1. Retrieve BigData - Go to UCF Canvas and download the file(s) for Assignment 7. Save the file(s) to a folder named Assignment 7.
2. Use Power Query (Get & Transform) to retrieve the data from the database.
3. Create the Data Model (PowerPivot).
4. Create the measures (analysis, time intelligent) in the data model.
5. Create the Pivot table.
6. Create the Dashboard (has a little VBA code to activate the hyperlinks).

The data is simulated payroll at Disney theme parks. The data covers a period of 3 years. All employees who work between 1st and 15th of the month are paid on the 22nd of the month and those who work between 16th and end of the month are paid on the 7th of the next month. Employees can work two shifts and some even work three shifts.

The database is an Access database and has 8 tables

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Number of Records** | **Source** |
| Employees | 65,000 | Access |
| EmployeeTerminateDate | 13+- thousand | Access |
| Payroll | 4+- million | Access |
| NationalOrigin | 26 | Access |
| State | 51 (includes Washington DC) | Access |
| JobCategory | 6 | Access |
| Park | 6 | Access |
| Position | 25 | Access |

**VBA in Excel:**

Visual Basic is a computer programming language that allows the creation of user-defined functions and the automation of specific computer processes and calculations. Users do not have to buy a copy of Visual Basic professional because Visual Basic for Applications is a standard feature of Microsoft Office products.

<https://msdn.microsoft.com/en-us/library/office/ee814735(v=office.14).aspx>

Extensions to User Interaction

There are times when you want to encourage or compel users to interact with the Office application or document in a particular way that is not part of the standard application. For example, you might want to prompt users to take some particular action when they open, save, or print a document.

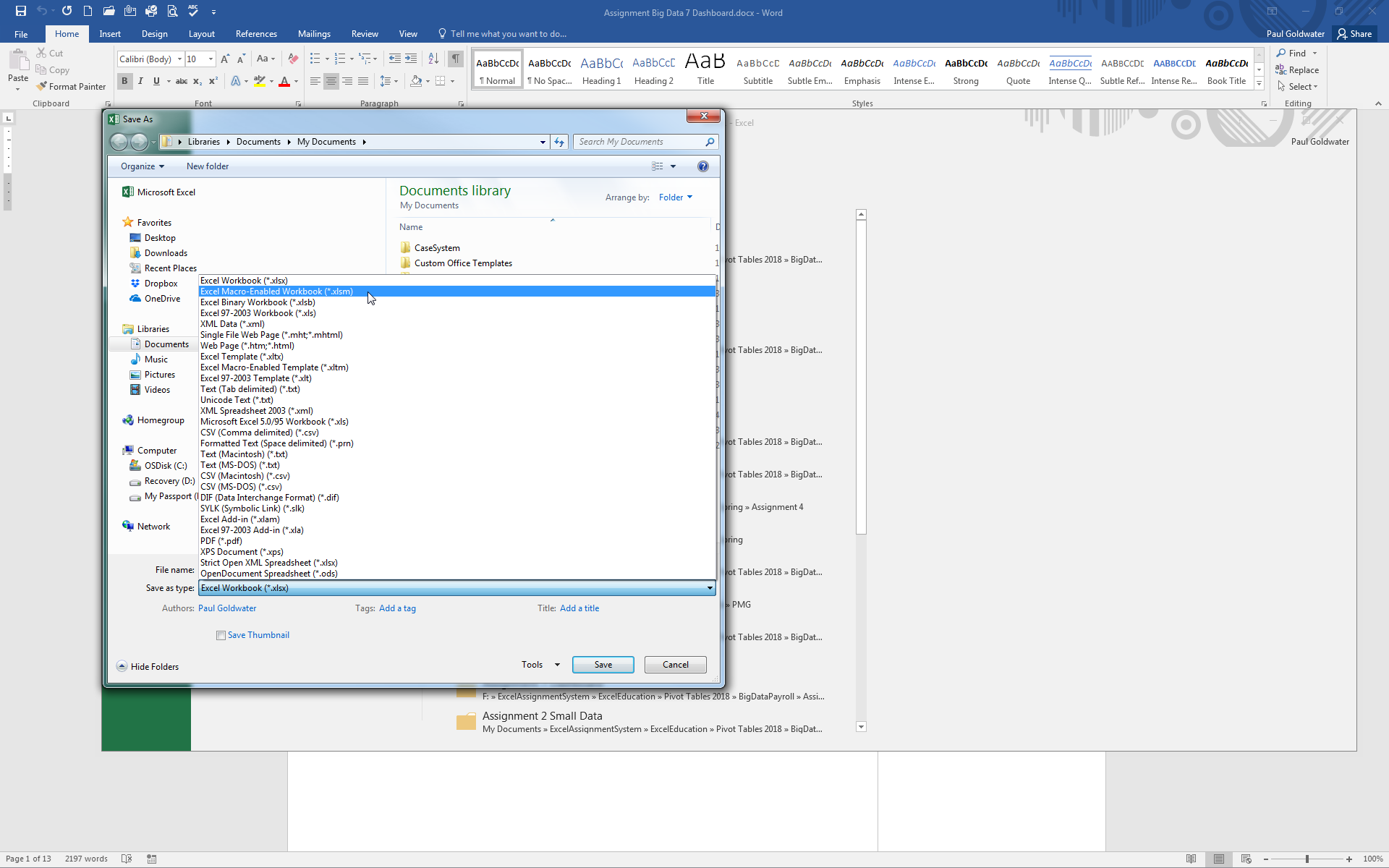
The dashboard has a hyperlink to dynamically change the chart and the only way to activate the hyperlink is with VBA code.

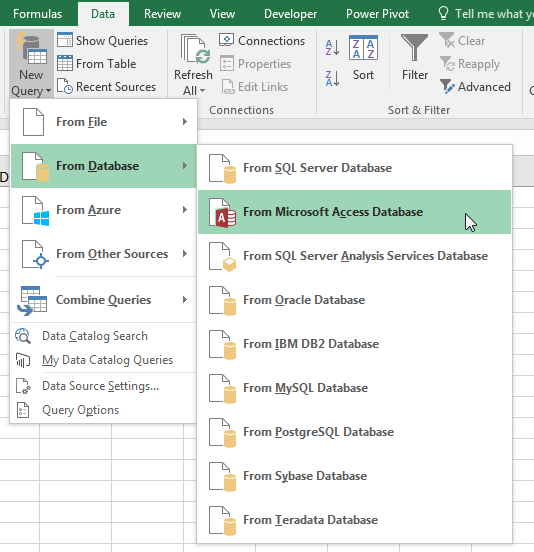
1. Open new Excel workbook, save the workbook Assignment7*LucaPacioli*.xlsm
   1. Substitute your name for LucaPacioli.
   2. Create two worksheets:
      1. PT
      2. DashBoard
   3. Save workbook to the same folder as your data.
      1. Notice the file format is XLSM – Excel Macro-Enabled Workbook (\*.xlsm)
2. Set Formulas -> Calculation Options -> Automatic

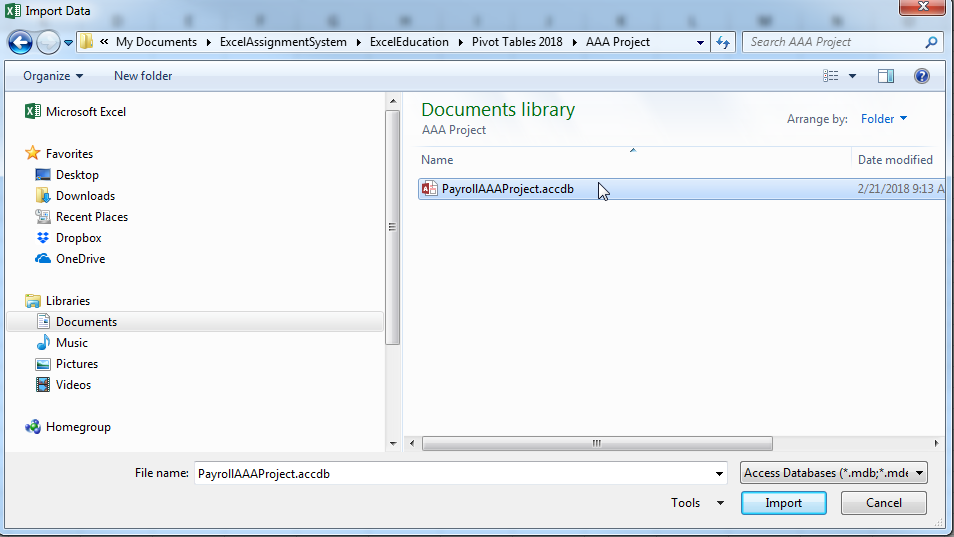
**Get & Transform Data**

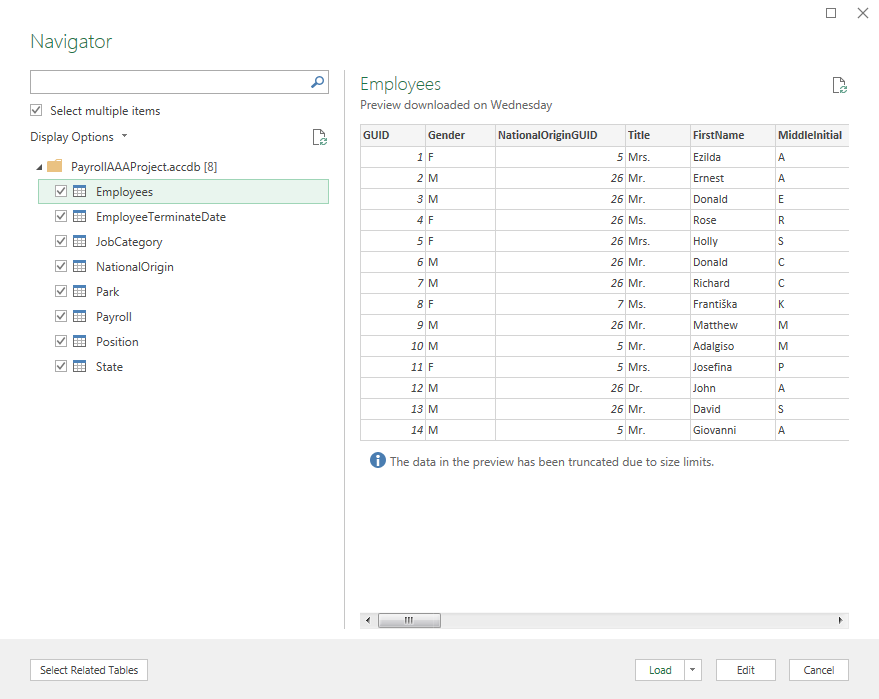
1. From Excel, select *Data* -> *Get &Transform* -> *New Query*
   1. Select *From Database* -> *From Microsoft Access Database*
   2. Navigate to the folder where you have located the Access database you downloaded PayrollAAAProject.accdb
   3. Select PayrollAAAProject.accdb file.
   4. Select *Import*
   5. Select *Select multiple items*
   6. Select all eight tables.
   7. Select the *Load* dropdown
      1. Complete the form as per the image.
      2. Select *Load*.
   8. Patience, the data is being loaded to the Excel data model.
   9. The query pane will appear.

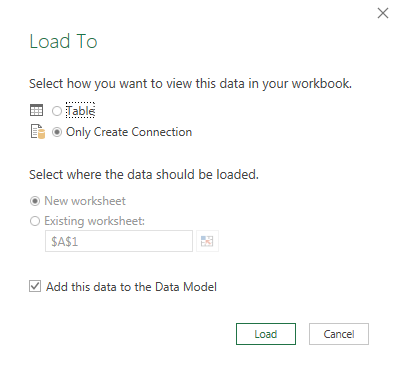












1. From Excel, select *Data* -> *Get &Transform* -> *Get Data*
   1. Select *From Other Sources* -> *Blank Query*
   2. Select *View* -> *Advanced Editor*
   3. Delete existing commands in the window.
   4. Copy this text to the window.

let

Calendar = let

Source = {Number.From(List.Min(Payroll[DepositDate]))..Number.From(List.Max(Payroll[DepositDate]))}

in

Source,

#"Converted to Table" = Table.FromList(Calendar, Splitter.SplitByNothing(), null, null, ExtraValues.Error),

#"Changed Type" = Table.TransformColumnTypes(#"Converted to Table",{{"Column1", type date}}),

#"Renamed Columns" = Table.RenameColumns(#"Changed Type",{{"Column1", "Date"}}),

#"Added Custom" = Table.AddColumn(#"Renamed Columns", "Year", each Date.Year([Date])),

#"Changed Type1" = Table.TransformColumnTypes(#"Added Custom",{{"Year", Int64.Type}}),

#"Added Custom1" = Table.AddColumn(#"Changed Type1", "Month", each Date.Month([Date])),

#"Changed Type2" = Table.TransformColumnTypes(#"Added Custom1",{{"Month", Int64.Type}}),

#"Added Custom2" = Table.AddColumn(#"Changed Type2", "MonthName", each Date.MonthName([Date])),

#"Changed Type3" = Table.TransformColumnTypes(#"Added Custom2",{{"MonthName", type text}}),

#"Added Custom3" = Table.AddColumn(#"Changed Type3", "YYYY-MM", each Date.ToText([Date],"yyyy-MM")),

#"Changed Type4" = Table.TransformColumnTypes(#"Added Custom3",{{"YYYY-MM", type text}}),

#"Added Custom4" = Table.AddColumn(#"Changed Type4", "DayOfWeek", each Date.DayOfWeek([Date])),

#"Changed Type5" = Table.TransformColumnTypes(#"Added Custom4",{{"DayOfWeek", Int64.Type}}),

#"Added Custom5" = Table.AddColumn(#"Changed Type5", "DayOfWeekName", each Date.DayOfWeekName([Date])),

#"Added Custom6" = Table.AddColumn(#"Added Custom5", "DayOfYear", each Date.DayOfYear([Date])),

#"Changed Type6" = Table.TransformColumnTypes(#"Added Custom6",{{"DayOfYear", Int64.Type}}),

#"Added Custom7" = Table.AddColumn(#"Changed Type6", "QuarterOfYear", each Date.QuarterOfYear([Date])),

#"Changed Type7" = Table.TransformColumnTypes(#"Added Custom7",{{"QuarterOfYear", Int64.Type}}),

#"Added Custom8" = Table.AddColumn(#"Changed Type7", "WeekOfMonth", each Date.WeekOfMonth([Date])),

#"Changed Type8" = Table.TransformColumnTypes(#"Added Custom8",{{"WeekOfMonth", Int64.Type}}),

#"Added Custom9" = Table.AddColumn(#"Changed Type8", "WeekOfYear", each Date.WeekOfYear([Date])),

#"Changed Type9" = Table.TransformColumnTypes(#"Added Custom9",{{"WeekOfYear", Int64.Type}}),

#"Added Custom10" = Table.AddColumn(#"Changed Type9", "PayPeriod", each if Date.Day([Date]) <= 15 then #datetime(Date.Year([Date]), Date.Month([Date]), 22, 0, 0, 0) else try #datetime(Date.Year([Date]), Date.Month([Date]) + 1, 7, 0, 0, 0) otherwise null),

#"Filtered Rows" = Table.SelectRows(#"Added Custom10", each true),

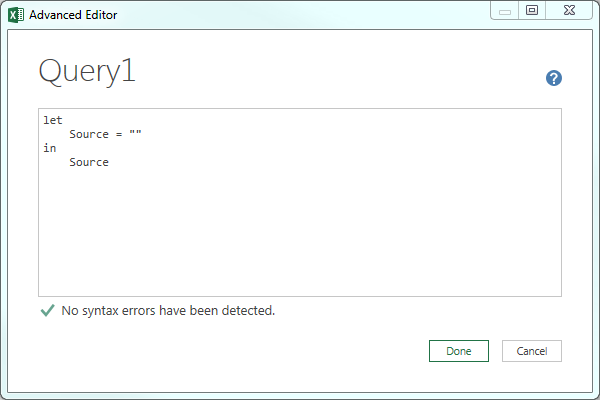
#"Changed Type12" = Table.TransformColumnTypes(#"Filtered Rows",{{"PayPeriod", type date}}),

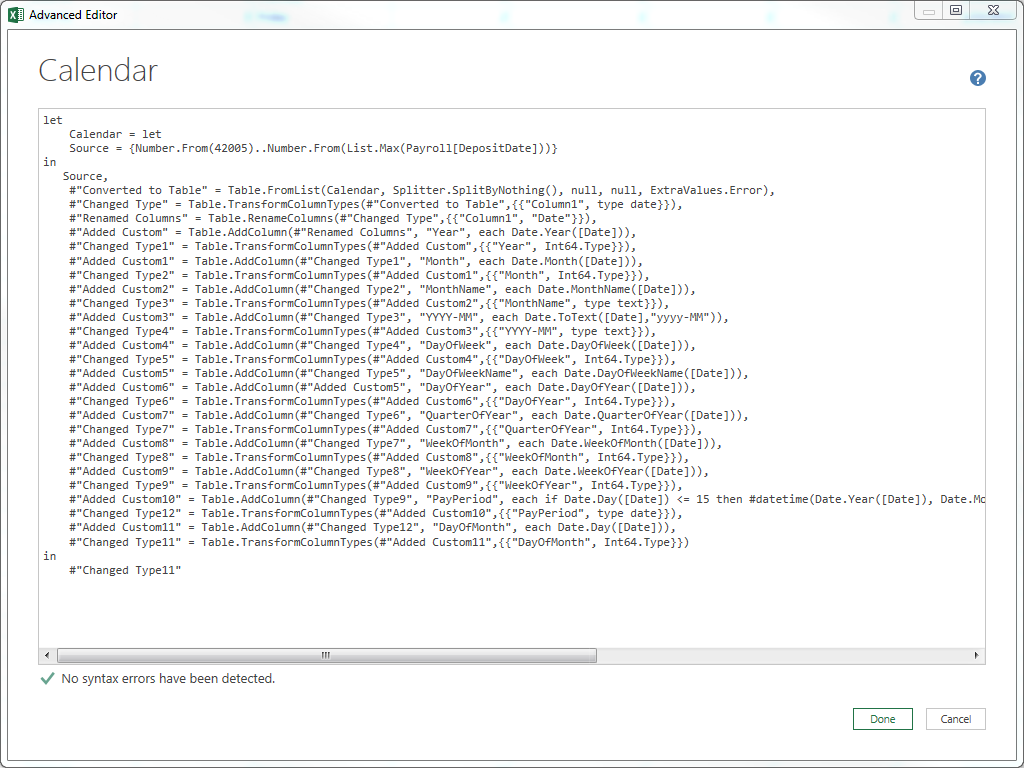
#"Added Custom11" = Table.AddColumn(#"Changed Type12", "DayOfMonth", each Date.Day([Date])),

#"Changed Type11" = Table.TransformColumnTypes(#"Added Custom11",{{"DayOfMonth", Int64.Type}})

in

#"Changed Type11"





* 1. Click *Done*, be patient and a calendar will appear. Notice the calendar is organized as follows:

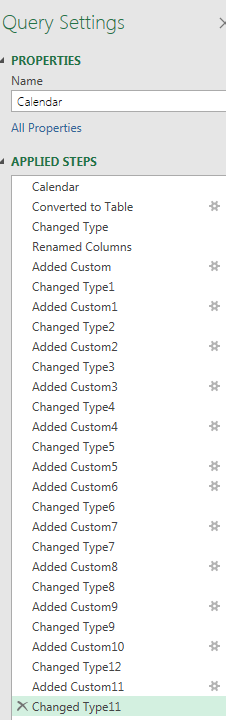
First date is the first day, first month, first year of the Payroll data.

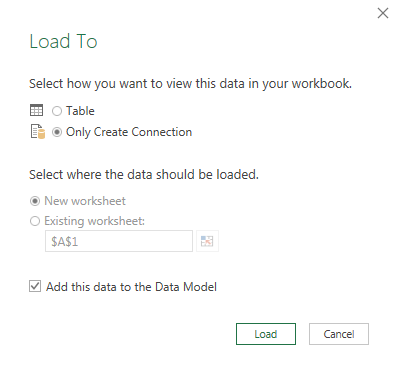
Last date is the last payroll deposit of the Payroll data.

Examine the columns and you will notice each column is a date formula – look in the formula bar. Notice the similarity to the Excel spreadsheet formula bar and Excel date/time functions.

* 1. Rename the Name: as *Calendar*.

1. Select *Home* -> *Close & Load To*…(from dropdown)
   1. Complete the form as per the image.
   2. Press Load.





**Data Model**

1. Select *PowerPivot* -> *Manage* and the PowerPivot window appears.
2. Notice the nine tables have been loaded into the Data Model.
3. Select *Diagram View*

You will notice all the table relationships have been automatically detected by Power Query (Get & Transform)

You may rearrange the tables for visual convenience.

1. From Diagram View mark the Calendar as *Mark as Date Table -> Mark as Date Table*
   1. Set *Date* column as the unique identifier.
2. Drag and drop the following table relationship.

*Payroll: DepositDate* to *Calendar: Date*

1. Examine the Employees to EmployeesTerminateDate relationship to ensure Excel correctly detected the Many to 1 relationship. It is actually a 1 to 1 relationship but needs to be expressed as Many Employees for 1 EmployeesTerminateDate. If need be, delete the current relationship and reenter the relationship from Employees: GUID table to the EmployeesTerminate: EmployeeGUID table.

**Measures**

1. Select *Data View -> Payroll* table
2. Create the following four measures. These 4 measures may be located anywhere within the measurement area.

mSumGrossAmount:=SUM(Payroll[GrossAmount])

mAverageGrossAmount:=AVERAGE(Payroll[GrossAmount])

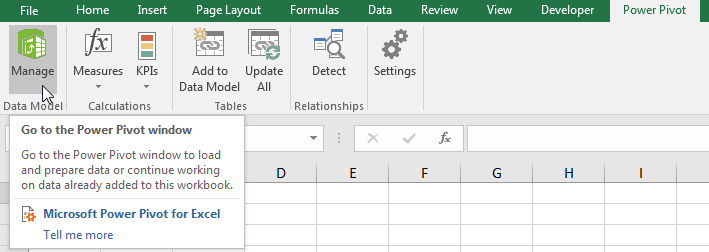
mSumGrossAmountSAMEPERIODLASTYEAR:=CALCULATE([mSumGrossAmount], SAMEPERIODLASTYEAR(Calendar[Date]))

mAverageGrossAmountSAMEPERIODLASTYEAR:=CALCULATE([mAverageGrossAmount], SAMEPERIODLASTYEAR(Calendar[Date]))

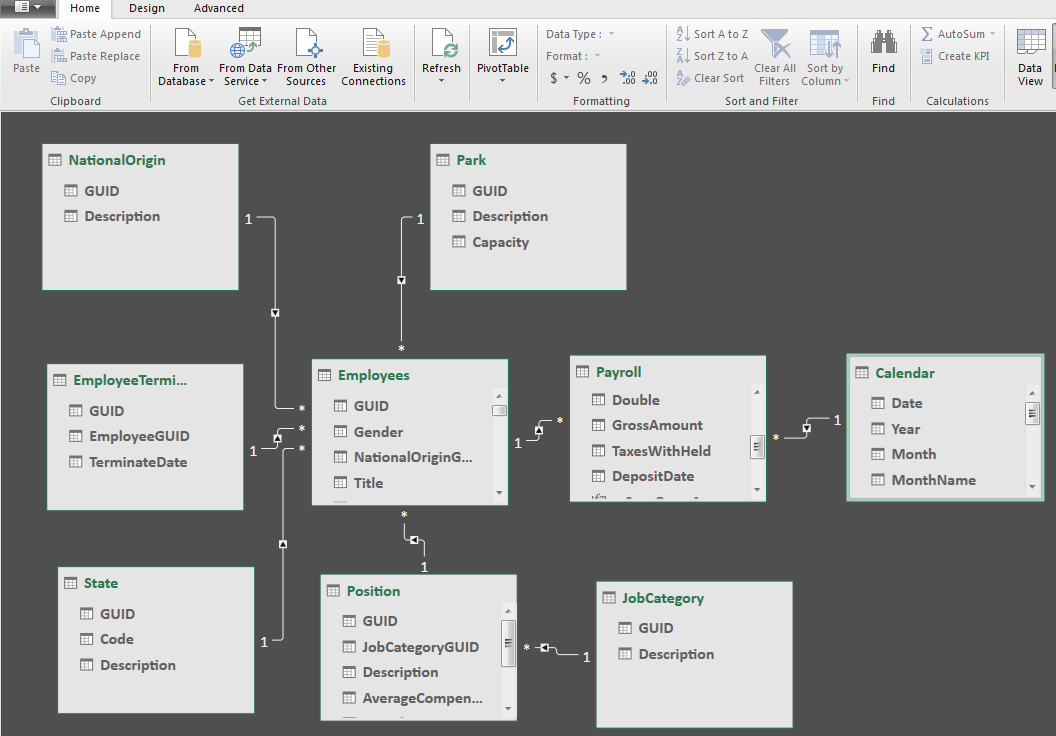
1. Right click each measure and select *Format…* and change the format according to the following table.

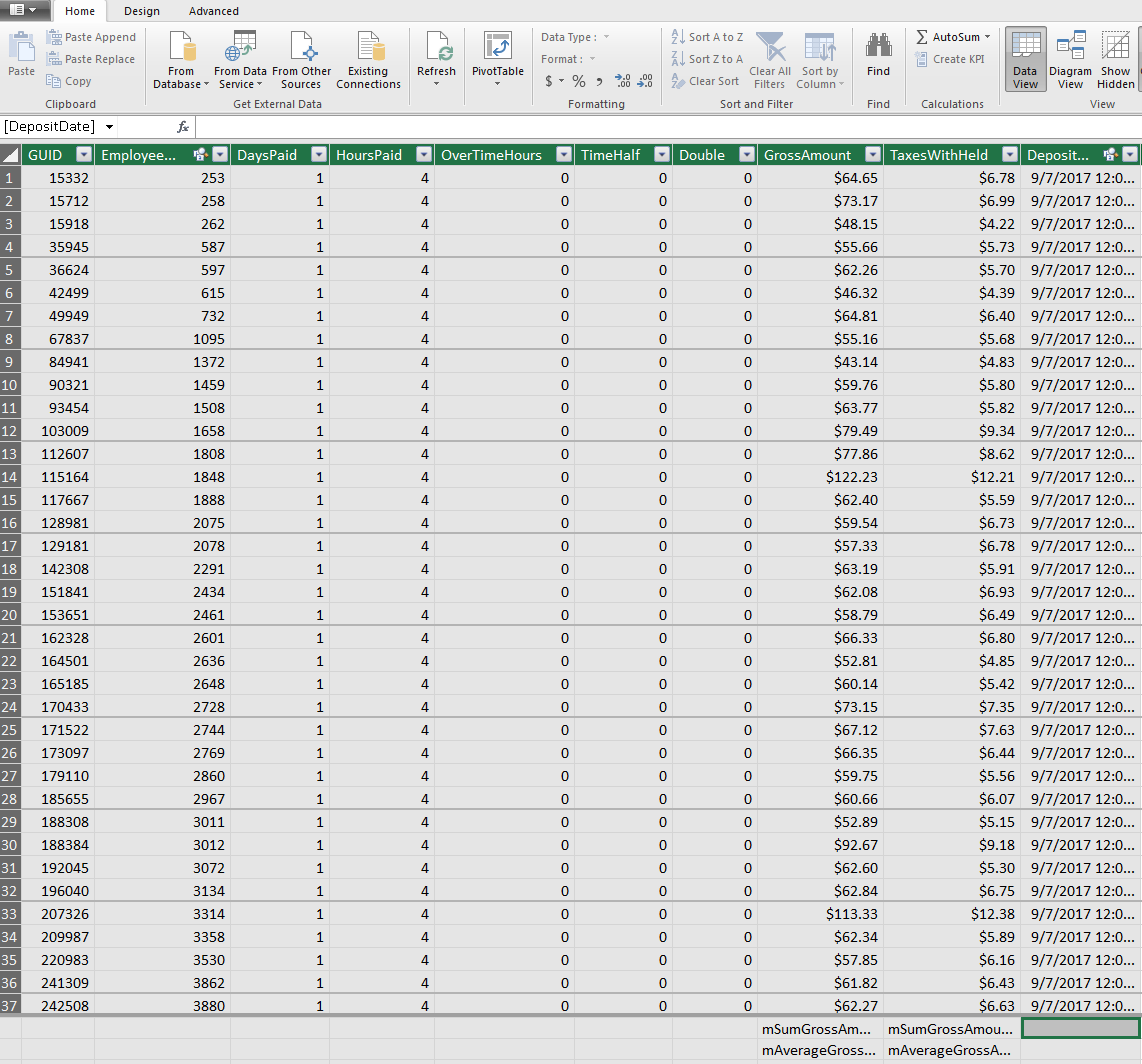
|  |  |
| --- | --- |
| Measurement | Format |
| mSumGrossAmount | *Currency, Symbol: $, Decimal places: 0* |
| mAverageGrossAmount | *Currency, Symbol: $, Decimal places: 0* |
| mSumGrossAmountSAMEPERIODLASTYEAR | *Currency, Symbol: $, Decimal places: 0* |
| mAverageGrossAmountSAMEPERIODLASTYEAR | *Currency, Symbol: $, Decimal places: 0* |

1. Return to Excel









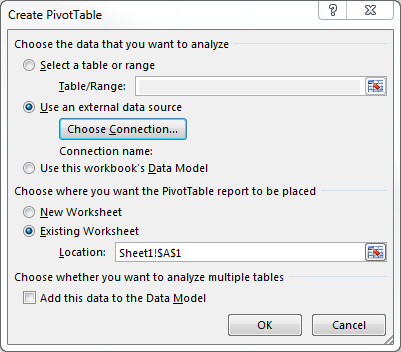
**Worksheet & Pivot Table**

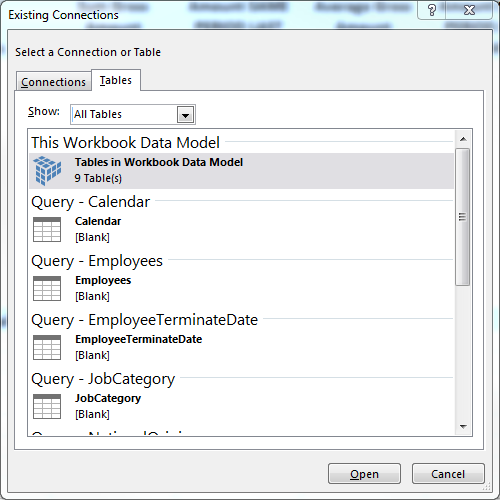
**PT Worksheet**

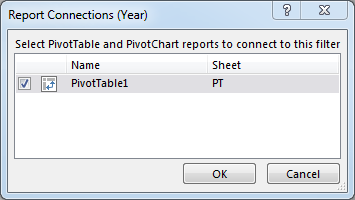
1. Select worksheet *PT*
2. Select cell A1
3. Insert PivotTable
   1. Select *Use an external data source*
   2. Select *Choose Connection*
   3. Select *Tables* tab
   4. Select *This Workbook Data Model*
   5. Select *Open*
   6. Select *OK*
4. Select the DashBoard worksheet
   1. Select Insert -> Slicer -> Select Calendar -> Year -> OK
      1. Select 2017
      2. Select menu Options -> Report Connections -> select PivotTable1
      3. Select OK

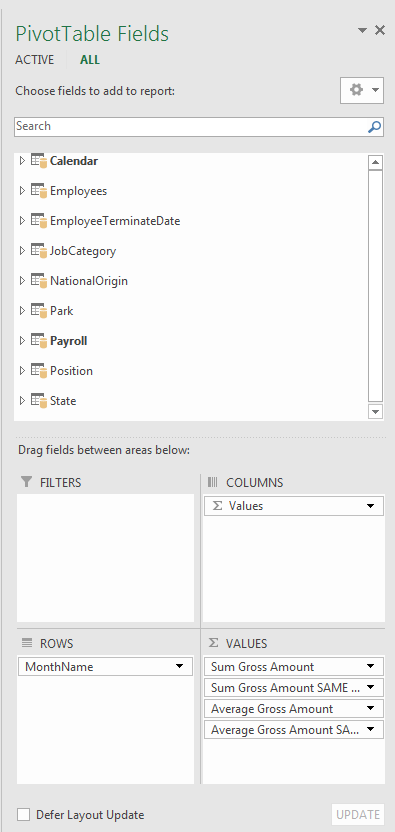
This is necessary because the measures have time intelligent measures (same period last year) – the year filter must be applied before dragging and dropping time intelligent measures onto the pivottable.

1. Select PT worksheet.
2. Refer to the image and drag and drop the fields to the appropriate areas of the pivottable pane.
   1. Drag and drop *Calendar:* *MonthName* to the *ROWS* area.
   2. Drag and drop *Payroll:* mSumGrossAmount to the *VALUES* area.
   3. Drag and drop *Payroll:* mAverageGrossAmount to the *VALUES* area.
   4. Drag and drop *Payroll:* mSumGrossAmountSAMEPERIODLASTYEAR to the *VALUES* area.
   5. Drag and drop *Payroll:* mAverageGrossAmountSAMEPERIODLASTYEAR to the *VALUES* area.









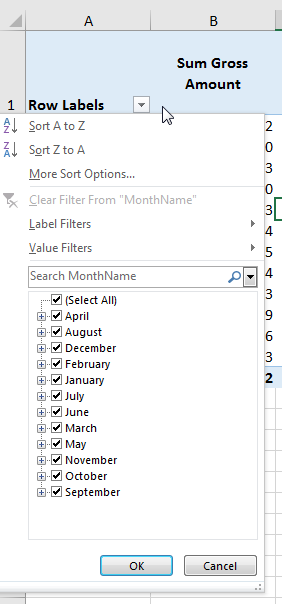
1. Click each ∑ VALUES dropdown and select *Value Fields Settings -> Custom Name:*

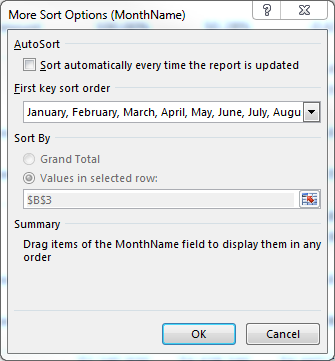
|  |  |
| --- | --- |
| ∑ VALUES | Value Fields Settings -> Custom Name |
| mSumGrossAmount | Sum Gross Amount |
| mSumGrossAmountSAMEPERIODLASTYEAR | Sum Gross Amount SAME PERIOD LAST YEAR |
| mAverageGrossAmount | Average Gross Amount |
| mAverageGrossAmountSAMEPERIODLASTYEAR | Average Gross Amount SAME PERIOD LAST YEAR |

We have changed the measurement name (for display purpose) to a more descriptive term; verbose still, but unequivocal during this learning process.

1. Select *Row Labels* dropdown in the pivottable.
2. Select *More Sort Options…* -> *More Options* complete the form as per the image.
3. Select *OK*. If the months do not sort correctly, reselect the *Column Labels* -> *MonthName* -> *Sort A to Z*
4. Select any cell in the pivottable.
5. Select *PivotTable Tools -> Design*
   1. *-> Subtotals -> Do Not Show Subtotals*
   2. *-> Grand Totals -> On for Columns Only*
6. Right click, select *PivotTable options…* and uncheck *Autofit column widths on update*
7. Select columns A to Z and right click, set *Column Width…* to 15
8. Select range B1:E1 and set:
   1. *Wrap Text* to True
   2. *Middle Align* (vertical alignment) to center
   3. *Center* (horizontal alignment) to center









1. Create algebra for DashBoard chart.
2. Select cell G1 and use the Name Box to name the cell *SelectedItem*
3. Select range G2:G13 and enter the array formula =INDEX($B$2:$E$13,,$K$2) and press Chtrl+Shift+Enter
4. Select range G2:G13 -> select Home -> Format -> Format Cells -> Currency, Symbol $, Decimal places 0
5. Select cell H1 and enter the formula ="Monthly Breakup of "&SelectedItem

This cell will be used as the heading for the DashBoard chart.

1. Select cell J2 and enter the text “Chart selection”
2. Select cell J3 and enter the text “Charting data offset values”
3. Select range J2:J3 and right align text.
4. Select cell K2 and enter the formula =MATCH(SelectedItem,B1:E1,0)

The formula will generate #N/A – ignore this, when you complete the DashBoard the algebra will evaluate correctly.

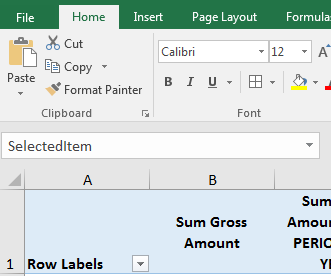
1. Select cell K3 and enter the formula =COUNTIF(B1:B14,">0")-1
2. Select Formulas -> Name Manager -> New
   1. Complete the form as per the image.
   2. Press OK, Close
3. Select Formulas -> Name Manager -> New
   1. Complete the form as per the image.
   2. Press OK, Close

We have created two dynamic range names. This was necessary because the year may have less than 12 months and the chart (DashBoard) needs to correctly reflect the actual months in the year chosen in the Year slicer.

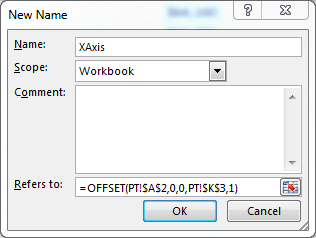
If you wish to read more about dynamic range names:

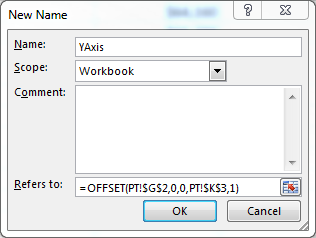
<https://support.microsoft.com/en-us/help/830287/how-to-create-a-dynamic-defined-range-in-an-excel-worksheet>











**Create Visual Basic Code**

1. Press Alt F11
2. Right click VBAProject (*your workbook name*.xlsm)
   1. Select Insert -> Module
   2. Select the Module1
3. Enter this code snippet.

Option Explicit

Public Function DataToChart(SeriesName As Excel.Range)

With Sheet1

'Move data for chart

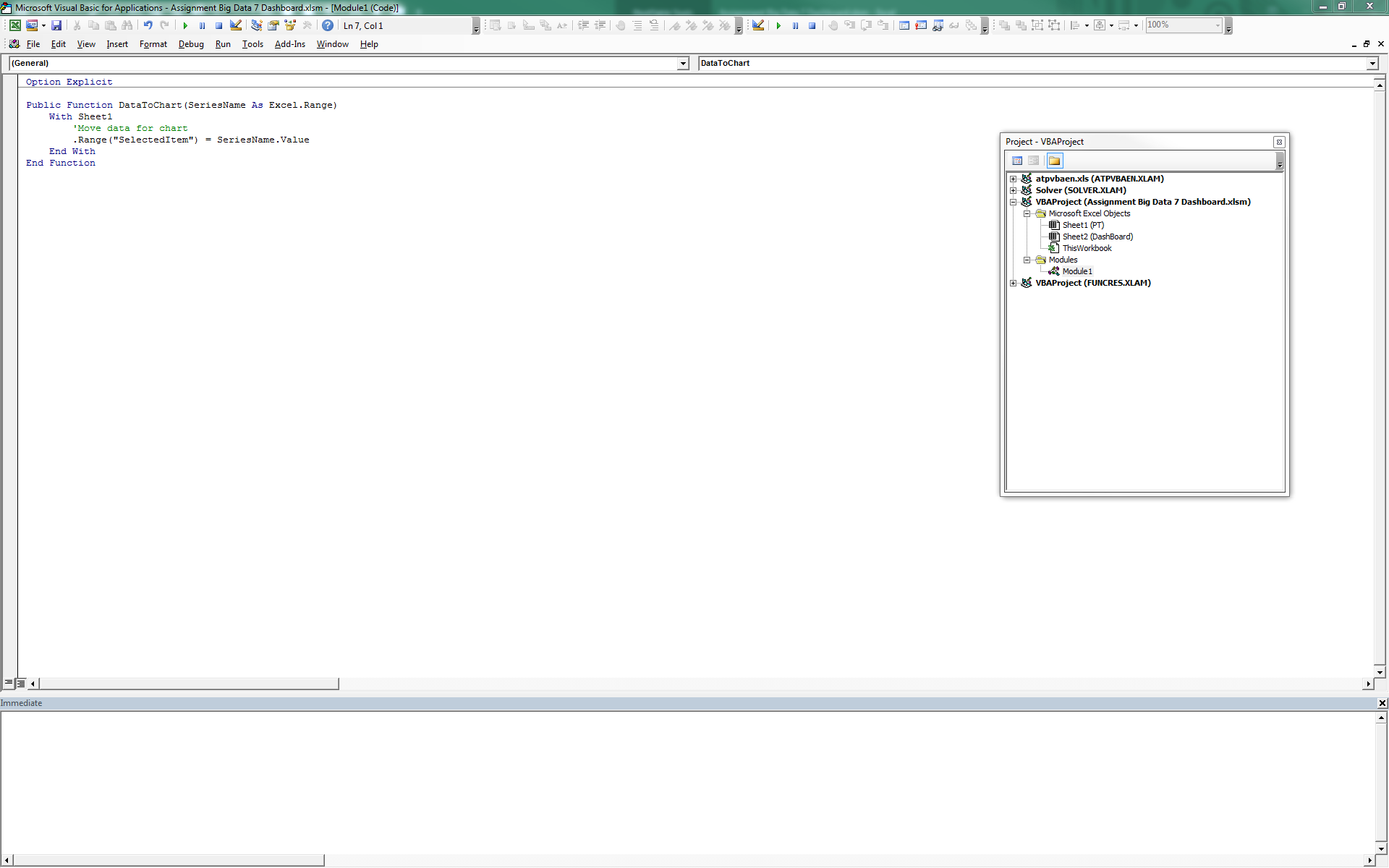
.Range("SelectedItem") = SeriesName.Value

End With

End Function

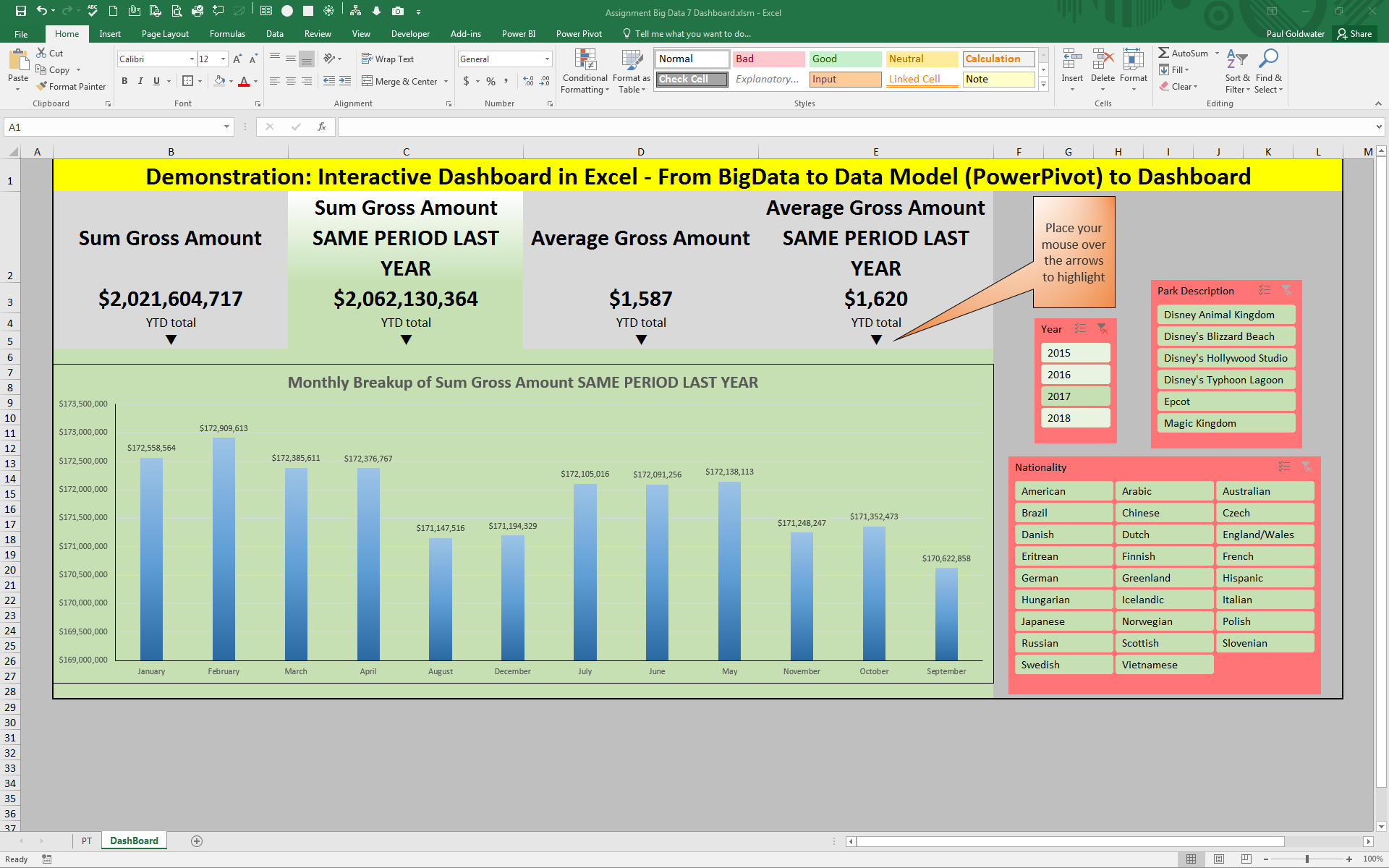
We need this code to dynamically change the DashBoard chart as the manager moves the mouse over the options.

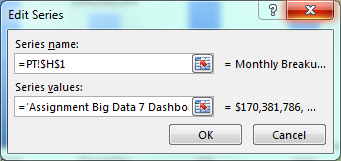
1. Close the VBA window.

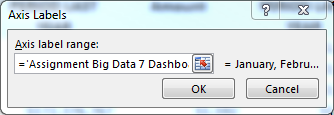


**DashBoard Worksheet**

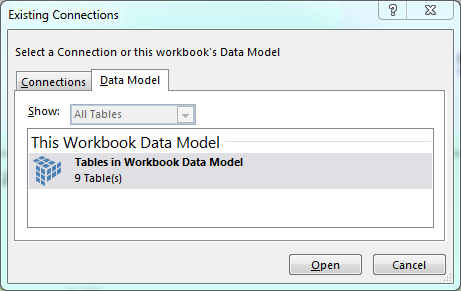
1. Select the DashBoard worksheet.
2. Select entire sheet cells and set the cell fill color to: Red 191 Green 191 Blue 191
3. Select range B2:E5 and set cell fill color to: Red 217 Green 217 Blue 217
4. Select range B6:E28 and set cell fill color to: Red 198 Green 224 Blue 180
5. Set column width:
   1. Column A 5
   2. Columns B to E 40
   3. Column F to Z 8
6. Select range B1:L1 and enter “Demonstration: Interactive Dashboard in Excel - From BigData to Data Model (PowerPivot) to Dashboard”
   1. Merge and center the range.
   2. Set Font:
      1. Size 24
   3. Select background color to Red 255 Green 255 Blue 0
7. Select B2 and enter the formula =PT!B1
   1. Drag B2 formula to cell E2
   2. Set WrapText to true.
   3. Set Font:
      1. Size 24
      2. Align center
8. Select cell B3:
   1. Enter the formula, = then select sheet PT and cell B14
      1. This action will produce =GETPIVOTDATA("[Measures].[mSumGrossAmount]",PT!$A$1)
   2. Set Font:
      1. Size 24
      2. Align center
   3. Repeat for cells C3 to E3 with the corresponding pivottable cell.
9. Enter YTD in range B4:E4
   1. Set Font:
      1. Size 14
      2. Align center
10. Select cell B5 and enter the formula =IFERROR(HYPERLINK(DataToChart(B2)),"q")
    1. Drag B5 formula to cell E5
    2. Set Font:
       1. Size 14
       2. Align center
       3. Font Name: Wingdings 3
11. Insert -> Clustered Column
    1. Right click chart and select *Select data…*
       1. Select *Add*
          1. Series name: -> select cell PT!$H$1
          2. Series values: =*your workbook name*.xlsm'!YAxis
          3. Select OK
          4. Select Edit
          5. Axis label range: = *your workbook name*.xlsm'!XAxis
          6. Select OK
          7. Select OK
12. Drag and resize chart to range B7:E27
13. Right click chart title and set font to bold and size 18
14. Right click chart area and set Fill to *No fill*
15. Select vertical axis and set Line to Solid line Color black
16. Select horizontal axis and set Line to Solid line Color black

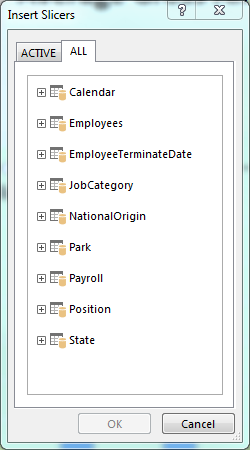




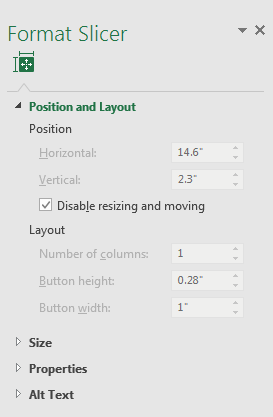


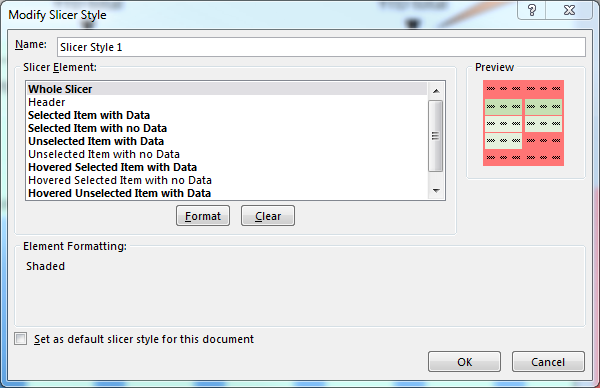
1. Right click a chart column select Add Data Labels -> Add Data Labels
2. Select Insert -> Slicer -> Select Park -> Description -> OK
3. Select Insert -> Slicer -> Select NationalOrigin -> Description -> OK
4. Select each slicer and locate and size as per the image.
5. Select each slicer, right click and select *Size and Properties…* check *Disable resizing and moving*
6. Select each slicer, right click and select *Size and Properties* -> Properties. Ensure *Locked* is unchecked.
7. Select all three Slicers by holding down the control key and selecting each slicer.
   1. Modify the Slicer Style
   2. You may adopt your own colors.





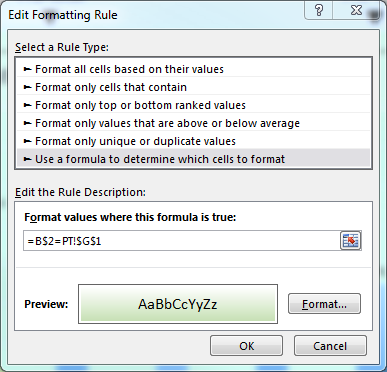
|  |  |  |
| --- | --- | --- |
|  |  |  |

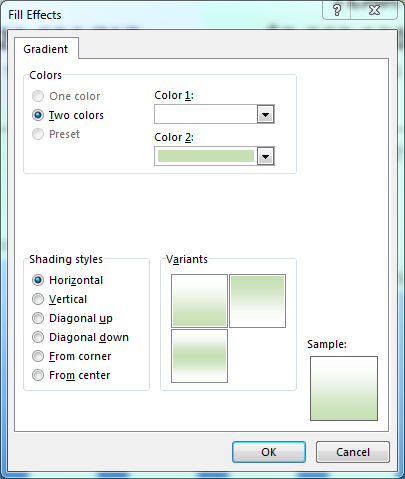


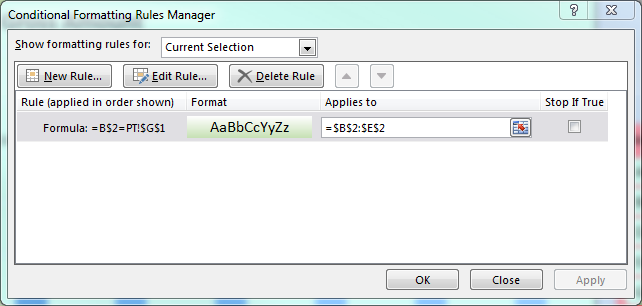


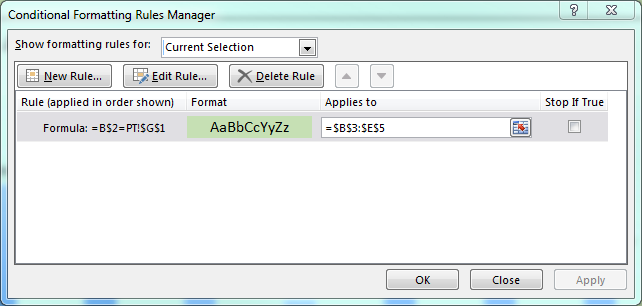
Conditional Formatting Rules

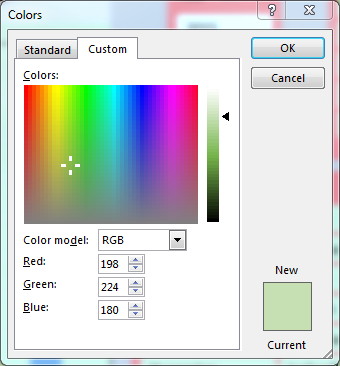
1. Select range B2:E2
   1. Select Conditional Formatting -> New Rule -> Use a formula to determine which cells to format
   2. Enter the formula =B$2=PT!$G$1
   3. Select Format… -> Fill -> Fill Effects -> complete as per the image
      1. Select Color 2 -> More Colors -> Custom and set the color to: Red 198 Green 224 Blue 180
      2. Press OK multiple times to return to Excel
2. Select range B3:E5
   1. Select Conditional Formatting -> New Rule -> Use a formula to determine which cells to format
   2. Enter the formula =B$2=PT!$G$1
   3. Select Format… -> Fill -> More Colors -> Custom complete as per the image
      1. Press OK multiple times to return to Excel







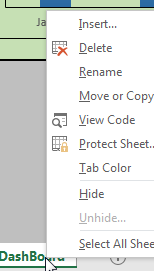


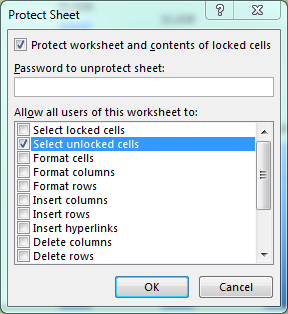


Protect Worksheet

1. Right click the tab DashBoard
   1. Select Protect Sheet…
   2. Only check *Select unlocked cells*
   3. Do not enter a password – this is merely to prevent your manager(s) inadvertently “messing up” the DashBoard.
   4. Select OK
2. Right click tab PT and select *Hide* – the managers do not need to see the pivottable and our algebra.







It is time to play!!!! Select slicer options and move your mouse over row 5.

Enjoy!!! Job Well done!!!!

