

Extending a Data Model with Time Intelligence

Lab Time: 60 minutes

Lab Folder: C:\Student\Modules\TimeIntelligence\Lab

Lab Overview: In this lab, you will continue to work on the Power BI Desktop project you have been working with over the last two previous labs. At this point, you have already spent time designing the data model for Wingtip sales data that has been imported from a SQL Azure database. In this lab you will continue to extend the data model by adding a time dimension table named **Calendar** and by configuring this table so it can be used for time-based financial analysis. This will allow you to create measures using the DAX time intelligence functions.

Important: This lab assumes you have completed the previous lab titled **Using the Power Pivot Features of Power BI Desktop** in which you created a Power BI Desktop project named **WingtipSalesAnalytics.pbix**. If you would like to begin work on this lab without completing the earlier lab, copy the lab solution file at **C:\Student\Modules\PowerPivot\LabSolution\WingtipSalesAnalytics.pbix** into the folder at **C:\Student\Projects** using the Windows Explorer.

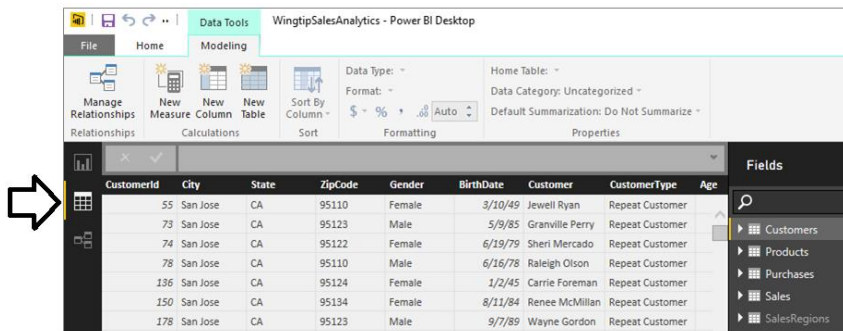
Exercise 1: Adding a Calendar Dimension Table

In this exercise you will create a calculated table named **Calendar** which will play the role of a time dimension table in the data model. The motivation for adding a time dimension table to your data model is that it will allow you to take advantage of the time intelligence support that is built into Power Pivot and DAX.

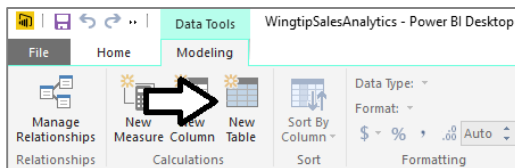
1. Launch Power BI Desktop to start a new project.
2. Open the Power BI Desktop project named **WingtipSalesAnalytics.pbix** from the previous lab located at the following path.

C:\Student\Projects\wingtipSalesAnalytics.pbix

3. When the project opens, click the table icon in the middle of the sidebar to enter data view mode.



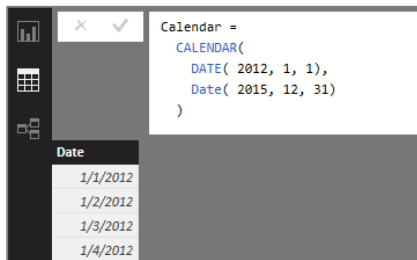
4. Create a new calculated table named **Calendar**.
 - a) Click the **New Table** button in the ribbon.



- b) Type the following DAX formula into the formula bar and press the **Enter** key to create the **Calendar** table.

```
Calendar =  
    CALENDAR(  
        DATE( 2012, 1, 1),  
        DATE( 2015, 12, 31)  
    )
```

- c) You should be able to verify that the **Calendar** table has been created with a single column named **Date**. You should also be able to verify that there is one row in the **Calendar** table for each day in the calendar years of 2012, 2013, 2014 and 2015.



- d) Modify the DAX expression for this calculated table so you do not have to hardcode literal values for the starting year or the ending year. First, replace the literal value for the starting year 2012 with the following DAX expression.

```
YEAR( MIN(Sales[PurchaseDate]) )
```

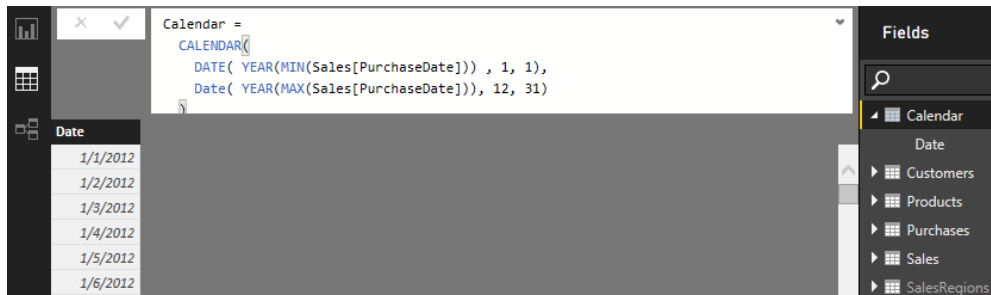
- e) Next, replace the literal value for the ending year 2015 with the following DAX expressions.

```
YEAR( MAX(Sales[PurchaseDate]) )
```

- f) At this point, your DAX expression to create the **Calendar** table should now match the following code listing.

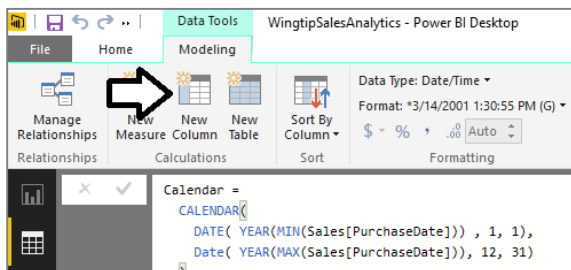
```
Calendar =
CALENDAR(
    DATE( YEAR( MIN(Sales[PurchaseDate]) ), 1, 1),
    Date( YEAR( MAX(Sales[PurchaseDate]) ), 12, 31)
)
```

- g) The **Calendar** table should continue to look as it did before starting at **1/1/20012** and running to **12/31/2015**.



The modification to the DAX expression to calculate the start date and end date dynamically provides flexibility. The new expression will now automatically add new rows for complete years if the **Sales** table is ever updated with purchases that occurred either before the start of 2012 or after the end of 2015.

5. Add a new column to the **Calendar** table named **Year** which display the financial year.
- a) Create a new calculated column by clicking the **New Column** button in the ribbon.



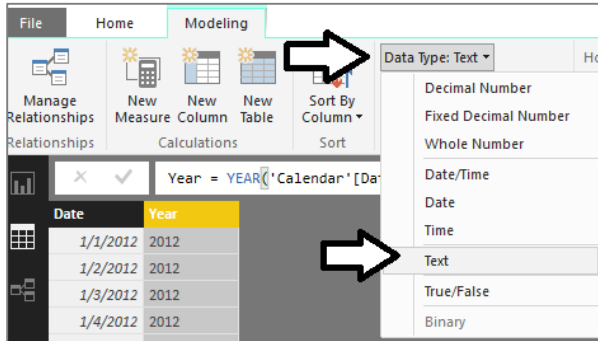
- b) Type in the following DAX expression and press the Enter key.

```
Year = YEAR('Calendar'[Date])
```

- c) You should see that the **Calendar** table now contains a **Year** column displaying the year.

Year = YEAR('Calendar'[Date])	
Date	Year
1/1/2012	2012
1/2/2012	2012
1/3/2012	2012

- d) Configure the **Data Type** of the **Year** column to be **Text**.



6. Add a new column to the **Calendar** table named **Quarter** which display the financial year and quarter.

- a) Create a new calculated column by clicking the **New Column** button in the ribbon.
b) Type in the following DAX expression and press the Enter key.

```
Quarter = QUOTIENT( (MONTH('Calendar'[Date])-1), 3) + 1
```

- c) The **Quarter** column now returns an integer value between 1 and 4 depending on what quarter the **Date** column value lies in.

Quarter = QUOTIENT((MONTH('Calendar'[Date])-1), 3) + 1		
Date	Year	Quarter
3/28/2012	2012	1
3/29/2012	2012	1
3/30/2012	2012	1
3/31/2012	2012	1
4/1/2012	2012	2
4/2/2012	2012	2
4/3/2012	2012	2

- d) Modify the DAX expression for the **Quarter** column to include the year and a bit more text formatting by typing in the following DAX expression and pressing the **Enter** key.

```
Quarter = YEAR('Calendar'[Date]) &
"-Q" &
QUOTIENT( (MONTH('Calendar'[Date])-1), 3) + 1
```

- e) Now the **Quarter** column should return a value for each financial quarter in the format of **2012-Q1**.

Quarter = YEAR('Calendar'[Date]) & "-Q" & QUOTIENT((MONTH('Calendar'[Date])-1), 3) + 1		
Date	Year	Quarter
3/28/2012	2012	2012-Q1
3/29/2012	2012	2012-Q1
3/30/2012	2012	2012-Q1
3/31/2012	2012	2012-Q1
4/1/2012	2012	2012-Q2
4/2/2012	2012	2012-Q2

7. Add a new column to the **Calendar** table named **Month** which display the financial year and month.

- Create a new calculated column by clicking the **New Column** button in the ribbon.
- Type in the following DAX expression and press the Enter key.

```
Month = FORMAT('Calendar'[Date], "MMM yyyy")
```

- The column should now display

Month = FORMAT('Calendar'[Date], "MMM yyyy")				
Date	Year	Quarter	Month	
1/1/2012	2012	2012-Q1	Jan 2012	
1/2/2012	2012	2012-Q1	Jan 2012	
1/3/2012	2012	2012-Q1	Jan 2012	

The **Month** column is a good example of a column whose value will not automatically be sorted in the chronological sort order. The default sort order of a text column like **Month** is to sort month names alphabetically so that April will sort before February, and February will sort before January. Therefore, you will now create an addition column named **MonthSort** whose sole purpose will be to provide assistance to the **Month** column to sort its values chronologically.

8. Add a new sort column to the **Calendar** table named **MonthSort** to control the sort order of the **Month** column.

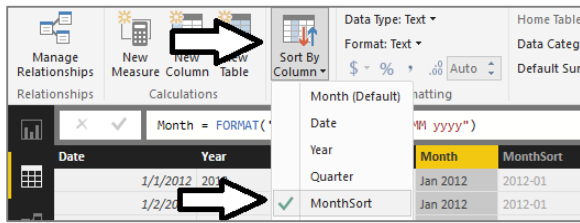
- Create a new calculated column by clicking the **New Column** button in the ribbon.
- Type in the following DAX expression and press the Enter key.

```
MonthSort = FORMAT('Calendar'[Date], "yyyy-MM")
```

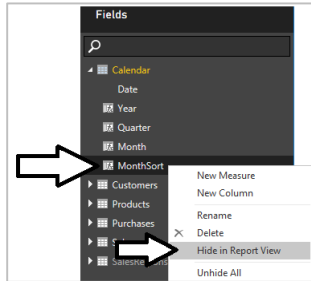
- You should be able to see that the **MonthSort** column produces a text value for each date in the format of **2012-01**. The key aspect of this format is that **MonthSort** values are sorted chronologically when they are sorted alphabetically.

MonthSort = FORMAT('Calendar'[Date], "yyyy-MM")				
Date	Year	Quarter	Month	MonthSort
1/1/2012	2012	2012-Q1	Jan 2012	2012-01
1/2/2012	2012	2012-Q1	Jan 2012	2012-01
1/3/2012	2012	2012-Q1	Jan 2012	2012-01

- Configure the **Month** column to use the **MonthSort** column as its sort column. Accomplish this by clicking the column header of the **Month** column to select it and then by dropping down the **Sort By Column** menu button in the ribbon and selecting the **MonthSort** column.



- e) Hide the **MonthSort** column by right-clicking it on the **Fields** list and selecting the **Hide in Report View** menu command.



9. Add a new column to the **Calendar** table named **Month in Year** to display the financial month without the year.
- Create a new calculated column by clicking the **New Column** button in the ribbon.
 - Type in the following DAX expression and press the Enter key.

```
Month in Year = FORMAT('Calendar'[Date], "MMM")
```

- c) The **Month in Year** column should now display the month name for each date.

Month in Year = FORMAT('Calendar'[Date], "MMM")					
Date	Year	Quarter	Month	MonthSort	Month in Year
1/1/2012	2012	2012-Q1	Jan 2012	2012-01	January
1/2/2012	2012	2012-Q1	Jan 2012	2012-01	January
1/3/2012	2012	2012-Q1	Jan 2012	2012-01	January

Just like the **Month** column, the **Month in Year** column will need the assistance of a sort column.

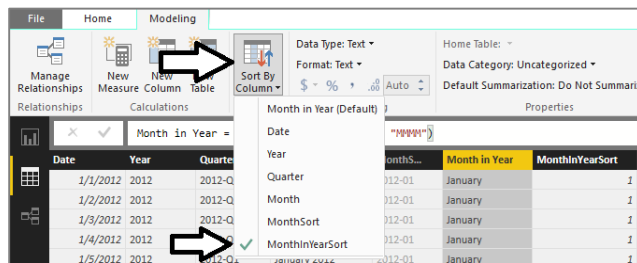
10. Add a new sort column to the **Calendar** table named **MonthInYearSort** to control the sort order of the **Month in Year** column.
- Create a new calculated column by clicking the **New Column** button in the ribbon.
 - Type in the following DAX expression and press the Enter key.

```
MonthInYearSort = MONTH('Calendar'[Date])
```

- c) As you can see, the **MonthInYearSort** column displays an integer value between 1 and 12 to indicate the month.

MonthInYearSort = MONTH('Calendar'[Date])						
Date	Year	Quarter	Month	MonthSort	Month in Year	MonthInYearSort
1/1/2012	2012	2012-Q1	Jan 2012	2012-01	January	1
1/2/2012	2012	2012-Q1	Jan 2012	2012-01	January	1
1/3/2012	2012	2012-Q1	Jan 2012	2012-01	January	1

- d) Configure the **Month in Year** column to use the **MonthInYearSort** column as its sort column. Accomplish this by clicking the column header of the **Month in Year** column to select it and then by dropping down the **Sort By Column** menu button in the ribbon and selecting the **MonthInYearSort** column.



- e) Hide the **MonthInYearSort** column by right-clicking it in the **Fields** list and selecting the **Hide in Report View** command.

11. Add a new column to the **Calendar** table named **Day of Week** which display the day of the week.

- Create a new calculated column by clicking the **New Column** button in the ribbon.
- Type in the following DAX expression and press the Enter key.

```
Day of Week = FORMAT('Calendar'[Date], "dddd")
```

- c) The **Day of Week** column should now display the name of the day (e.g. Monday) for each date.

Day of Week = FORMAT('Calendar'[Date], "dddd")							
Date	Year	Quarter	Month	MonthSort	Month in Year	MonthInYearSort	Day of Week
1/1/2012	2012	2012-Q1	Jan 2012	2012-01	January	1	Sunday
1/2/2012	2012	2012-Q1	Jan 2012	2012-01	January	1	Monday
1/3/2012	2012	2012-Q1	Jan 2012	2012-01	January	1	Tuesday
1/4/2012	2012	2012-Q1	Jan 2012	2012-01	January	1	Wednesday
1/5/2012	2012	2012-Q1	Jan 2012	2012-01	January	1	Thursday
1/6/2012	2012	2012-Q1	Jan 2012	2012-01	January	1	Friday
1/7/2012	2012	2012-Q1	Jan 2012	2012-01	January	1	Saturday

12. Add a new sort column to the **Calendar** table named **DayOfWeekSort** to control the sort order of the **Day of Week** column.

- Create a new calculated column by clicking the **New Column** button in the ribbon.
- Type in the following DAX expression and press the Enter key.

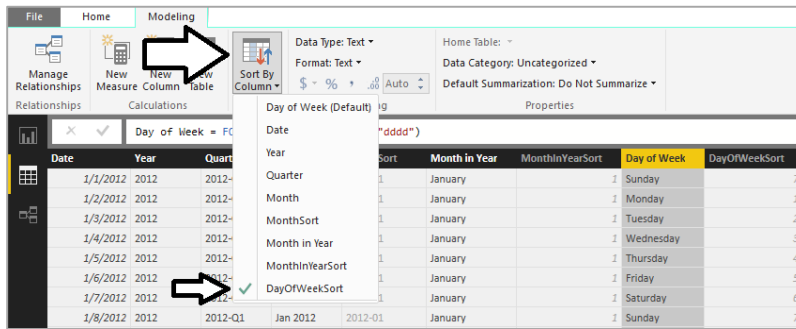
```
DayOfWeekSort = WEEKDAY('Calendar'[Date], 2)
```

The second argument passes to **WEEKDAY** function determines the starting day for the week. If you pass a value of 1, the starting day for the week will be Sunday. In this case you have passed a value of 2 which will make Monday the first day of the week.

- a) Now the **DayOfWeekSort** column should return an integer value for each date indicating the day of the week. As you can see, each date that is a Monday has a value of 1.

DayOfWeekSort = WEEKDAY('Calendar'[Date], 2)								
Date	Year	Quarter	Month	MonthSort	Month in Year	MonthInYearSort	Day of Week	DayOfWeekSort
1/1/2012	2012	2012-Q1	Jan 2012	2012-01	January		1 Sunday	7
1/2/2012	2012	2012-Q1	Jan 2012	2012-01	January		1 Monday	1
1/3/2012	2012	2012-Q1	Jan 2012	2012-01	January		1 Tuesday	2
1/4/2012	2012	2012-Q1	Jan 2012	2012-01	January		1 Wednesday	3
1/5/2012	2012	2012-Q1	Jan 2012	2012-01	January		1 Thursday	4
1/6/2012	2012	2012-Q1	Jan 2012	2012-01	January		1 Friday	5
1/7/2012	2012	2012-Q1	Jan 2012	2012-01	January		1 Saturday	6

- b) Configure the **Day of Week** column to use the **DayInWeekSort** column as its sort column. Accomplish this by clicking the column header of the **Day of Week** column to select it and then by dropping down the **Sort By Column** menu button in the ribbon and selecting the **DayInWeekSort** column.

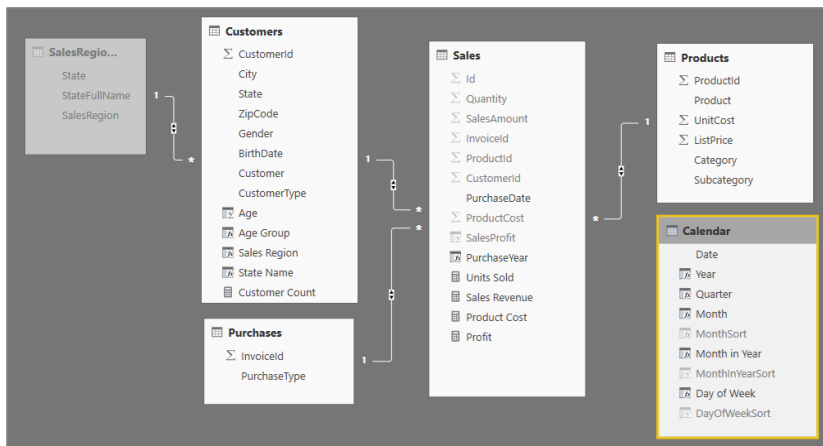


- c) Hide the **DayInWeekSort** column by right-clicking it in the **Fields** list and selecting the **Hide in Report View** command.

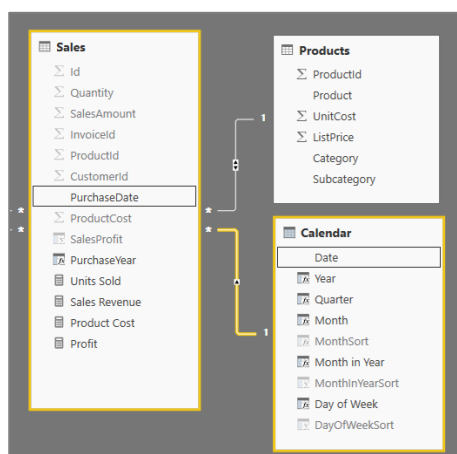
At this point, you have created the **Calendar** table and added all the columns that it requires. The next step to integrate the **Calendar** table into the data model will be to create a relationship between the **Calendar** table and the **Sales** table.

13. Create a relationship between the **Calendar** table and the **Sales** table.

- Navigate to relationship view. You should be able to see that the **Calendar** table is present. You should also be able to verify that the **Calendar** table does not have any existing relationships.
- Using the mouse, rearrange the tables in relationship view to match the following screenshot where the **Calendar** table is positioned directly below the **Products** table and to the immediate right of the **Sales** table.



- c) Create a new relationship by performing a drag and drop operation with the mouse to drag the **PurchaseDate** column from the **Sales** table and dropping it on the **Date** column of the **Calendar** table.



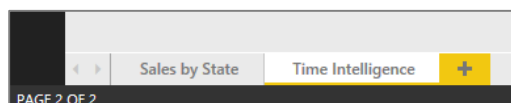
You have now completed the work of adding the **Calendar** table to the data model.

14. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

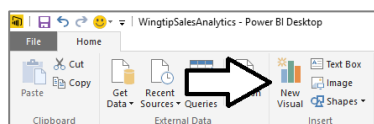
Exercise 2: Creating Visuals using the Calendar Table

In this exercise, you will leverage the **Calendar** table that you created in the previous exercise by creating a few new visuals to display sales revenue totals aggregated over various time intervals.

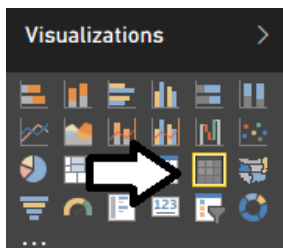
1. Create a new page in the project's report.
 - a) Navigate to report view.
 - b) Add a new page by clicking the (+) button to the right of the current page title.
 - c) Once the new page has been created, modify its title to **Time Intelligence**.



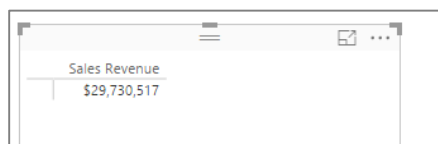
- d) Now that you have created a new page, you can now add a few new visuals.
2. Create a new matrix visual to show sales revenue for specific time periods.
 - a) Click the **New Visual** button on the ribbon to add a new visual to the page.



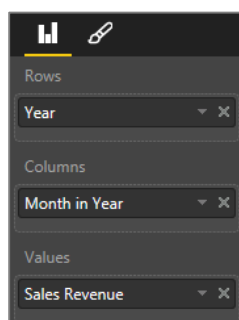
- b) Change the visual to a matrix by clicking the **Matrix** button in the **Visualizations** list.



- c) Select the checkbox next to the **Sales Revenue** measure in the **Fields** list. When you select the **Sales Revenue** measure, the report designer will add it to the **Values** well and the visual will show a single value for total sales revenue across the entire **Sales** table.



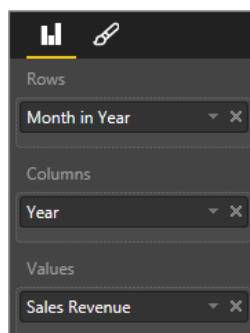
- d) Now it's time to extend the matrix by adding row labels and column labels. First, drag and drop the **Year** column from the **Calendar** table in the **Fields** list into the **Rows** well in the **Visualizations** pane.
- e) Now drag and drop the **Month in Year** column from the **Calendar** table in the **Fields** list into the **Columns** well in the **Visualizations** pane.



- f) The matrix now has a column for each month and a row for each year.
- g) Use your mouse to resize the matrix visual so you can see all the columns.

Year	January	February	March	April	May	June	July	August	September	October	November	December	Total
2012	\$3,063	\$33,218	\$49,213	\$40,434	\$83,840	\$136,670	\$144,244	\$197,952	\$215,097	\$239,513	\$376,503	\$424,240	\$1,943,986
2013	\$307,182	\$291,942	\$346,186	\$380,869	\$377,376	\$353,586	\$391,202	\$476,884	\$504,532	\$577,439	\$579,507	\$769,473	\$5,356,177
2014	\$629,969	\$609,637	\$628,618	\$661,588	\$748,193	\$814,333	\$788,469	\$869,143	\$890,958	\$988,789	\$999,574	\$1,644,980	\$10,274,251
2015	\$959,863	\$969,330	\$675,533	\$722,456	\$698,311	\$785,793	\$921,994	\$1,064,189	\$1,088,863	\$1,211,810	\$1,305,029	\$1,732,932	\$12,156,103
Total	\$1,900,077	\$1,904,126	\$1,699,551	\$1,805,347	\$1,907,720	\$2,090,382	\$2,245,908	\$2,628,168	\$2,699,449	\$3,017,551	\$3,260,613	\$4,571,625	\$29,730,517

- h) Now experiment by pivoting the matrix visual to display the exact same data using a different layout. Accomplish this by moving the **Month in Year** field into the **Rows** well and then moving the **Year** field into the **Columns** well. In effect, the **Month in Year** field and the **Year** field have just switched places.



- i) The matrix should now display a row for each month and a column for each year.

Month in Year	2012	2013	2014	2015	Total
January	\$3,063	\$307,182	\$629,969	\$959,863	\$1,900,077
February	\$33,218	\$291,942	\$609,637	\$969,330	\$1,904,126
March	\$49,213	\$346,186	\$628,618	\$675,533	\$1,699,551
April	\$40,434	\$380,869	\$661,588	\$722,456	\$1,805,347
May	\$83,840	\$377,376	\$748,193	\$698,311	\$1,907,720
June	\$136,670	\$353,586	\$814,333	\$785,793	\$2,090,382
July	\$144,244	\$391,202	\$788,469	\$921,994	\$2,245,908
August	\$197,952	\$476,884	\$869,143	\$1,084,189	\$2,628,168
September	\$215,097	\$504,532	\$890,958	\$1,088,863	\$2,699,449
October	\$239,513	\$577,439	\$988,789	\$1,211,810	\$3,017,551
November	\$376,503	\$579,507	\$999,574	\$1,305,029	\$3,260,613
December	\$424,240	\$769,473	\$1,644,980	\$1,732,932	\$4,571,625
Total	\$1,943,986	\$5,356,177	\$10,274,251	\$12,156,103	\$29,730,517

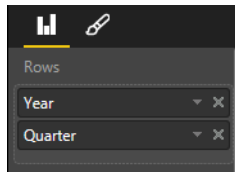
3. Create a new matrix visual to show sales revenue for specific time periods.
 - a) Click the **New Visual** button on the ribbon to add a new visual to the page.
 - b) Make sure this new visual is positioned directly below the first visual that you created.
 - c) Change the visual to a matrix by clicking the Matrix button in the **Visualizations** list.
 - d) Drag and drop the **Day of Week** column from the **Calendar** table in the **Fields** list into the **Rows** well.
 - e) Drag and drop the **Year** column from the **Calendar** table into the **Columns** well.
 - f) Drag and drop the **Sales Revenue** measure from the **Calendar** table in the **Fields** list into the **Values** well.

Rows	Day of Week	X
Columns	Year	X
Values	Sales Revenue	X

- g) The matrix should now display a row for each day of the week and a column for each year.

Day of Week	2012	2013	2014	2015	Total
Monday	\$314,471	\$801,337	\$1,460,373	\$1,682,345	\$4,258,527
Tuesday	\$262,321	\$791,863	\$1,553,063	\$1,726,955	\$4,334,202
Wednesday	\$269,499	\$671,754	\$1,525,827	\$1,786,688	\$4,253,768
Thursday	\$246,499	\$777,814	\$1,427,989	\$1,749,475	\$4,201,776
Friday	\$329,852	\$803,028	\$1,445,129	\$1,790,611	\$4,368,620
Saturday	\$289,566	\$747,619	\$1,447,230	\$1,736,439	\$4,220,853
Sunday	\$231,779	\$762,762	\$1,414,640	\$1,683,591	\$4,092,772
Total	\$1,943,986	\$5,356,177	\$10,274,251	\$12,156,103	\$29,730,517

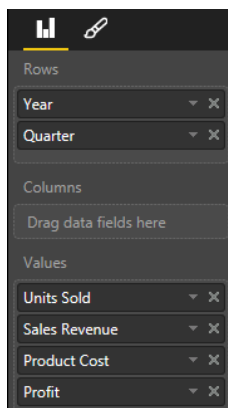
4. Add a third matrix visual to analyze sales data calculated at both the yearly level and the quarterly level.
 - a) Click the **New Visual** button on the ribbon to add a new visual to the page.
 - b) Change the visual to a matrix by clicking the Matrix button in the Visualizations list.
 - c) Drag and drop the **Year** column from the **Calendar** table into the **Rows** well.
 - d) Drag and drop the **Quarter** column from the **Calendar** table into the **Rows** well.



- e) At this point, the visual has row labels for the **Year** column and the **Quarter** column.

Year	Quarter
2012	2012-Q1
	2012-Q2
	2012-Q3
	2012-Q4
2013	2013-Q1
	2013-Q2
	2013-Q3
	2013-Q4
2014	2014-Q1
	2014-Q2
	2014-Q3
	2014-Q4
2015	2015-Q1
	2015-Q2
	2015-Q3
	2015-Q4

- f) Drag and drop the **Units Sold** measure from the **Sales** table into the **Values** well.
g) Drag and drop the **Sales Revenue** measure from the **Sales** table into the **Values** well.
h) Drag and drop the **Product Cost** measure from the **Sales** table into the **Values** well.
i) Drag and drop the **Profit** measure from the **Sales** table into the **Values** well.



- j) The matrix should now display values for each measure calculated at the quarterly level as well as at the yearly level.

Year	Quarter	Sales Revenue	Product Cost	Profit
2012	2012-Q1	\$85,494	\$40,088	\$45,406
	2012-Q2	\$260,944	\$130,287	\$130,657
	2012-Q3	\$557,293	\$269,314	\$287,979
	2012-Q4	\$1,040,256	\$540,222	\$500,034
	Total	\$1,943,986	\$979,909	\$964,077
2013	2013-Q1	\$945,310	\$517,474	\$427,836
	2013-Q2	\$1,111,831	\$557,730	\$554,101
	2013-Q3	\$1,372,617	\$571,187	\$801,430
	2013-Q4	\$1,926,420	\$864,530	\$1,061,889
	Total	\$5,356,177	\$2,510,921	\$2,845,256
2014	2014-Q1	\$1,868,225	\$892,244	\$975,981
	2014-Q2	\$2,224,114	\$1,081,051	\$1,143,063
	2014-Q3	\$2,548,569	\$1,332,729	\$1,215,840
	2014-Q4	\$3,633,343	\$1,877,978	\$1,755,365
	Total	\$10,274,251	\$5,184,002	\$5,090,249
2015	2015-Q1	\$2,604,726	\$1,364,369	\$1,240,357
	2015-Q2	\$2,206,560	\$1,219,892	\$986,669
	2015-Q3	\$3,095,046	\$1,724,893	\$1,370,153
	2015-Q4	\$4,249,771	\$2,358,468	\$1,891,304
	Total	\$12,156,103	\$6,667,621	\$5,488,482
Total		\$29,730,517	\$15,342,453	\$14,388,064

5. Using your mouse, arrange the new matrix visual on the right side of the page at the top to match the following screenshot.

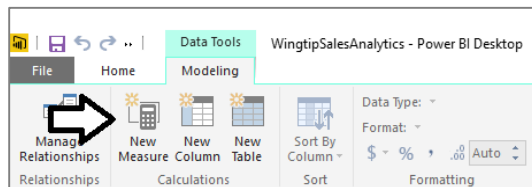
Month in Year	2012	2013	2014	2015	Total	Year	Quarter	Sales Revenue	Product Cost	Profit
January	\$3,063	\$307,182	\$629,969	\$959,863	\$1,900,077	2012	2012-Q1	\$85,494	\$40,088	\$45,406
February	\$33,218	\$291,942	\$609,637	\$969,330	\$1,904,126		2012-Q2	\$260,944	\$130,287	\$130,657
March	\$49,213	\$346,186	\$626,618	\$675,533	\$1,699,551		2012-Q3	\$557,293	\$269,314	\$287,979
April	\$40,434	\$380,869	\$661,588	\$722,456	\$1,805,347		2012-Q4	\$1,040,256	\$540,222	\$500,034
May	\$83,840	\$377,376	\$748,193	\$698,311	\$1,907,720		Total	\$1,943,986	\$979,909	\$964,077
June	\$136,670	\$353,586	\$814,333	\$785,793	\$2,090,382	2013	2013-Q1	\$945,310	\$517,474	\$427,836
July	\$144,244	\$391,202	\$788,469	\$921,994	\$2,245,908		2013-Q2	\$1,111,831	\$557,730	\$554,101
August	\$197,952	\$476,884	\$869,143	\$1,084,189	\$2,628,168		2013-Q3	\$1,372,617	\$571,187	\$801,430
September	\$215,097	\$504,532	\$890,958	\$1,088,863	\$2,699,449		2013-Q4	\$1,926,420	\$864,530	\$1,061,889
October	\$239,513	\$577,439	\$988,789	\$1,211,810	\$3,017,551		Total	\$5,356,177	\$2,510,921	\$2,845,256
November	\$376,503	\$579,507	\$999,574	\$1,305,029	\$3,260,613	2014	2014-Q1	\$1,868,225	\$892,244	\$975,981
December	\$424,240	\$769,473	\$1,644,980	\$1,732,932	\$4,571,625		2014-Q2	\$2,224,114	\$1,081,051	\$1,143,063
Total	\$1,943,986	\$5,356,177	\$10,274,251	\$12,156,103	\$29,730,517		2014-Q3	\$2,548,569	\$1,332,729	\$1,215,840
							2014-Q4	\$3,633,343	\$1,877,978	\$1,755,365
							Total	\$10,274,251	\$5,184,002	\$5,090,249
Day of Week	2012	2013	2014	2015	Total	2015	2015-Q1	\$2,604,726	\$1,364,369	\$1,240,357
Monday	\$314,471	\$801,337	\$1,460,373	\$1,682,345	\$4,258,527		2015-Q2	\$2,206,560	\$1,219,892	\$986,669
Tuesday	\$262,321	\$791,863	\$1,553,063	\$1,726,955	\$4,334,202		2015-Q3	\$3,095,046	\$1,724,893	\$1,370,153
Wednesday	\$269,499	\$671,754	\$1,525,627	\$1,786,688	\$4,253,768		2015-Q4	\$4,249,771	\$2,356,468	\$1,893,304
Thursday	\$246,499	\$777,814	\$1,427,989	\$1,749,475	\$4,201,776		Total	\$12,156,103	\$6,667,621	\$5,488,482
Friday	\$329,852	\$803,028	\$1,445,129	\$1,790,611	\$4,368,620	Total		\$29,730,517	\$15,342,453	\$14,388,064
Saturday	\$289,566	\$747,619	\$1,447,230	\$1,736,439	\$4,220,853					
Sunday	\$231,779	\$762,762	\$1,414,640	\$1,683,591	\$4,092,772					
Total	\$1,943,986	\$5,356,177	\$10,274,251	\$12,156,103	\$29,730,517					

6. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

Exercise 3: Creating Measures using DAX Time Intelligence Functions

In this exercise, you will leverage various the Time Intelligence functions in DAX to analyze sales revenue using quarter to date (QTD) totals and year to date (YTD) totals. You will also use DAX to write an expression which calculate a running total of sales revenue through the entire 4 years of sales activity.

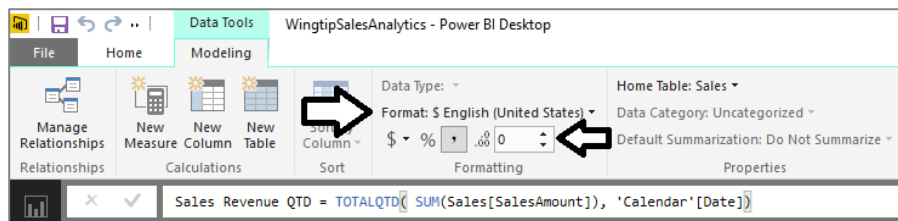
- Create a measure named **Sales Revenue QTD** that calculates a quarter-to-date aggregate sum on the **SalesAmount** column of the **Sales** table.
 - Navigate to data view.
 - Select the **Sales** table from the **Fields** list.
 - Create a new measure by clicking the **New Measure** button in the ribbon.



- Enter to following DAX expression into the formula bar to create the new measure named **Sales Revenue QTD**.

Sales Revenue QTD = TOTALQTD(SUM(Sales[SalesAmount]), 'Calendar'[Date])

- Press the **ENTER** key to add the measure to data model.
- Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Currency > English (United States)**. Also use the spinner control below the format menu to set the number of decimal places shown to zero.

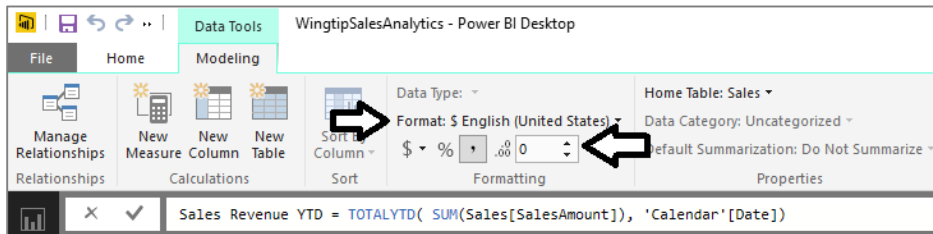


- Create a measure named **Sales Revenue YTD** that calculates a year-to-date aggregate sum on the **SalesAmount** column of the **Sales** table.
 - Navigate to data view.

- b) Select the **Sales** table from the **Fields** list.
- c) Create a new measure by clicking the **New Measure** button in the ribbon.
- d) Enter to following DAX expression into the formula bar to create the new measure named **Sales Revenue YTD**.

```
Sales Revenue YTD = TOTALYTD( SUM(Sales[SalesAmount]), 'Calendar'[Date])
```

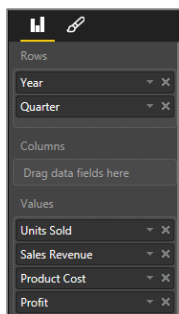
- e) Press the **ENTER** key to add the measure to data model.
- f) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Currency > English (United States)**. Also use the spinner control below the format menu to set the number of decimal places shown to zero.



- 3. Create a measure named **Sales Revenue RT** that calculates a running total aggregate sum on the **SalesAmount** column of the **Sales** table.
 - a) Navigate to data view.
 - b) Select the **Sales** table from the **Fields** list.
 - c) Create a new measure by clicking the **New Measure** button in the ribbon.
 - d) Enter to following DAX expression into the formula bar to create the new measure named **Sales Revenue RT**.

```
Sales Revenue RT =  
CALCULATE(  
  SUM(Sales[SalesAmount]),  
  FILTER(  
    ALL('Calendar'),  
    'Calendar'[Date] <= MAX('Calendar'[Date])  
  )  
)
```

- e) Press the **ENTER** key to add the measure to data model.
 - f) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Currency > English (United States)**. Also use the spinner control below the format menu to set the number of decimal places shown to zero.
- 4. Use the three new measures in a matrix visual.
 - a) Navigate to report mode.
 - b) Select the matrix visual you created last that is currently positioned on the right-hand side of the page.
 - c) Examine the bottom of the **Visualizations** pane. Currently, the **Rows** well contains the **Year** column and the **Quarter** column. There are also four other measures in the **Values** well.



- d) Remove all the measures from the **Values** well except for the **Sales Revenue** measure. At this point, your visual should match the one shown in the following screenshot where sales revenue totals as shown at the quarterly level and at the yearly level.

Year	Quarter	Sales Revenue
2012	2012-Q1	\$85,494
	2012-Q2	\$260,944
	2012-Q3	\$557,293
	2012-Q4	\$1,040,256
	Total	\$1,943,986
2013	2013-Q1	\$945,310
	2013-Q2	\$1,111,831
	2013-Q3	\$1,372,617
	2013-Q4	\$1,926,420
	Total	\$5,356,177
2014	2014-Q1	\$1,868,225
	2014-Q2	\$2,224,114
	2014-Q3	\$2,548,569
	2014-Q4	\$3,633,343
	Total	\$10,274,251
2015	2015-Q1	\$2,604,726
	2015-Q2	\$2,206,560
	2015-Q3	\$3,095,046
	2015-Q4	\$4,249,771
	Total	\$12,156,103
Total		\$29,730,517

- e) Drag and drop the **Month** column from the **Calendar** table into the **Rows** well below the two other columns.

Rows	
Year	✕
Quarter	✕
Month	✕

- f) You should be able to see that now the visual now has a deeper level of granularity because it is show sales revenue broken out into a separate aggregate value for each month.

Year	Quarter	Month	Sales Revenue
2012	2012-Q1	Jan 2012	\$3,063
		Feb 2012	\$33,218
		Mar 2012	\$49,213
		Total	\$85,494
	2012-Q2	Apr 2012	\$40,434
		May 2012	\$83,840
		Jun 2012	\$136,670
		Total	\$260,944
	2012-Q3	Jul 2012	\$144,244
		Aug 2012	\$197,952
		Sep 2012	\$215,097

- g) Set a filter on the matrix visual so that is only displays sales revenue for the calendar years of 2012 and 2013. Accomplish this by setting a filter where the Year column is less then or equal to **2013**. After setting the filter as shown in the follow screenshot, click the **Apply Filter** link below to apply your filter to the data shown in the visual.

Filters

Visual level filters

Month(All)

Quarter(All)

Sales Revenue(All)

Year
is less than or equal t...

Show items when the value:

is less than or equal t

2013

☒ And ☐ Or

Apply filter

- h) Resize the matrix visual to take up the entire right-hand side of the page.

Month in Year	2012	2013	2014	2015	Total
January	\$3,063	\$307,182	\$629,969	\$959,863	\$1,900,077
February	\$33,218	\$291,942	\$609,637	\$969,330	\$1,904,126
March	\$49,213	\$346,186	\$628,618	\$675,533	\$1,699,551
April	\$40,434	\$380,869	\$661,588	\$722,456	\$1,805,347
May	\$83,840	\$377,376	\$748,193	\$698,311	\$1,907,720
June	\$136,670	\$353,586	\$814,333	\$785,793	\$2,090,382
July	\$144,244	\$391,202	\$780,469	\$921,994	\$2,245,908
August	\$197,952	\$476,884	\$869,143	\$1,084,189	\$2,628,168
September	\$215,097	\$504,532	\$890,958	\$1,088,863	\$2,699,449
October	\$239,513	\$577,439	\$988,789	\$1,211,810	\$3,017,551
November	\$376,503	\$579,507	\$999,574	\$1,305,029	\$3,260,613
December	\$424,240	\$769,473	\$1,644,980	\$1,732,932	\$4,571,625
Total	\$1,943,986	\$5,356,177	\$10,274,251	\$12,156,103	\$29,730,517

Day of Week	2012	2013	2014	2015	Total
Monday	\$314,471	\$801,337	\$1,460,373	\$1,682,345	\$4,258,527
Tuesday	\$262,321	\$791,863	\$1,553,063	\$1,726,955	\$4,334,202
Wednesday	\$269,499	\$671,754	\$1,525,827	\$1,786,688	\$4,253,768
Thursday	\$246,499	\$777,814	\$1,427,989	\$1,749,475	\$4,201,776
Friday	\$329,852	\$603,028	\$1,445,129	\$1,790,611	\$4,368,620
Saturday	\$289,566	\$747,619	\$1,447,230	\$1,736,439	\$4,220,853
Sunday	\$231,779	\$762,762	\$1,414,640	\$1,683,591	\$4,092,772
Total	\$1,943,986	\$5,356,177	\$10,274,251	\$12,156,103	\$29,730,517

Year	Quarter	Month	Sales Revenue
2012	2012-Q1	Jan 2012	\$3,063
		Feb 2012	\$33,218
		Mar 2012	\$49,213
	Total		\$85,494
	2012-Q2	Apr 2012	\$40,434
		May 2012	\$83,840
		Jun 2012	\$136,670
	Total		\$260,944
	2012-Q3	Jul 2012	\$144,244
		Aug 2012	\$197,952
		Sep 2012	\$215,097
	Total		\$557,293
	2012-Q4	Oct 2012	\$239,513
		Nov 2012	\$376,503
		Dec 2012	\$424,240
	Total		\$1,040,256
Total			\$1,943,986
2013	2013-Q1	Jan 2013	\$307,182
		Feb 2013	\$291,942
		Mar 2013	\$346,186
	Total		\$945,310
	2013-Q2	Apr 2013	\$380,869
		May 2013	\$377,376
		Jun 2013	\$353,586
	Total		\$1,111,831
	2013-Q3	Jul 2013	\$391,202
		Aug 2013	\$476,884
		Sep 2013	\$504,532
	Total		\$1,372,617
	2013-Q4	Oct 2013	\$577,439
		Nov 2013	\$579,507
		Dec 2013	\$769,473
	Total		\$1,926,420
Total			\$5,356,177
			\$7,300,163

- i) Drag and drop the **Sales Revenue QTD** measure from the **Sales** table into the **Values** well.
j) Drag and drop the **Sales Revenue YTD** measure from the **Sales** table into the **Values** well.
k) Drag and drop the **Sales Revenue RT** measure from the **Sales** table into the **Values** well.

Values
Sales Revenue
Sales Revenue QTD
Sales Revenue YTD
Sales Revenue RT

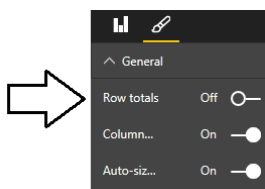
- l) The matrix visual should now display three new columns for the three measure you added to the **Values** well.

Year	Quarter	Month	Sales Revenue	Sales Revenue QTD	Sales Revenue YTD	Sales Revenue RT
2012	2012-Q1	Jan 2012	\$3,063	\$3,063	\$3,063	\$3,063
		Feb 2012	\$33,218	\$36,280	\$36,280	\$36,280
		Mar 2012	\$49,213	\$85,494	\$85,494	\$85,494
		Total	\$85,494	\$85,494	\$85,494	\$85,494
	2012-Q2	Apr 2012	\$40,434	\$40,434	\$125,927	\$125,927
		May 2012	\$83,840	\$124,274	\$209,768	\$209,768
		Jun 2012	\$136,670	\$260,944	\$346,438	\$346,438

- m) Currently the matrix is showing a **Total** row with a quarterly rollup. You should observe that the **Total** row can be confusing with measures like these because the it displays the exact same aggregate value as the row above it.

Year	Quarter	Month	Sales Revenue	Sales Revenue QTD	Sales Revenue YTD	Sales Revenue RT
2012	2012-Q1	Jan 2012	\$3,063	\$3,063	\$3,063	\$3,063
		Feb 2012	\$33,218	\$36,280	\$36,280	\$36,280
		Mar 2012	\$49,213	\$85,494	\$85,494	\$85,494
		Total	\$85,494	\$85,494	\$85,494	\$85,494
	2012-Q2	Apr 2012	\$40,434	\$40,434	\$125,927	\$125,927
		May 2012	\$83,840	\$124,274	\$209,768	\$209,768

- n) Look inside the **General** section for the visual in the **Visualizations** pane and set modify the **Row totals** property to **Off**.



- o) The matrix visual is now better suited to display the three new measures you created in this exercise.

Year	Quarter	Month	Sales Revenue	Sales Revenue QTD	Sales Revenue YTD	Sales Revenue RT
2012	2012-Q1	Jan 2012	\$3,063	\$3,063	\$3,063	\$3,063
		Feb 2012	\$33,218	\$36,280	\$36,280	\$36,280
		Mar 2012	\$49,213	\$85,494	\$85,494	\$85,494
	2012-Q2	Apr 2012	\$40,434	\$40,434	\$125,927	\$125,927
		May 2012	\$83,840	\$124,274	\$209,768	\$209,768
		Jun 2012	\$136,670	\$260,944	\$346,438	\$346,438
	2012-Q3	Jul 2012	\$144,244	\$144,244	\$490,681	\$490,681
		Aug 2012	\$197,952	\$342,196	\$688,634	\$688,634

- p) Now imagine your boss asks you to determine in what month the company reached 1 million dollars in total sales revenue. By looking at down the list of values for the **Sales Revenue RT** measure, you can see that the company finally hit \$1,000,000 in sales revenue in October of 2012.

Year	Quarter	Month	Sales Revenue	Sales Revenue QTD	Sales Revenue YTD	Sales Revenue RT
2012	2012-Q1	Jan 2012	\$3,063	\$3,063	\$3,063	\$3,063
		Feb 2012	\$33,218	\$36,280	\$36,280	\$36,280
		Mar 2012	\$49,213	\$85,494	\$85,494	\$85,494
	2012-Q2	Apr 2012	\$40,434	\$40,434	\$125,927	\$125,927
		May 2012	\$83,840	\$124,274	\$209,768	\$209,768
		Jun 2012	\$136,670	\$260,944	\$346,438	\$346,438
	2012-Q3	Jul 2012	\$144,244	\$144,244	\$490,681	\$490,681
		Aug 2012	\$197,952	\$342,196	\$688,634	\$688,634
		Sep 2012	\$215,097	\$557,293	\$903,731	\$903,731
	2012-Q4	Oct 2012	\$239,513	\$239,513	\$1,143,243	\$1,143,243
		Nov 2012	\$376,503	\$616,016	\$1,519,746	\$1,519,746

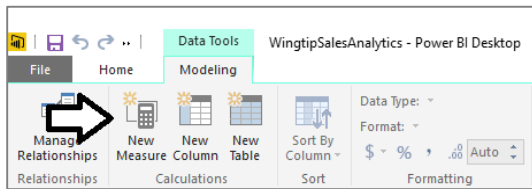
5. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

You have now learned how to use Time Intelligence functions in DAX together with a calendar table. Now you will move on to the final exercise where you will create additional measures to monitor sales growth.

Exercise 4: Creating Measures to Monitor Growth in Sales Revenue

In this exercise you will create new measures to calculate the growth of sales revenue on a quarter-by-quarter basis as well as on a month-by-month basis. After that you will create additional measures that will act as KPIs to monitor the health of sales growth and provide visual indications as to how each month has done when compared to the previous month.

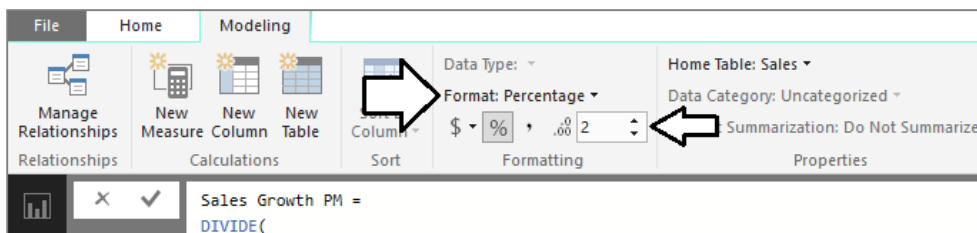
1. Create a measure named **Sales Growth PM** that calculates the percentage increase between sales revenue for the current month and sales revenue for the previous month.
 - a) Navigate to data view.
 - b) Select the **Sales** table from the **Fields** list.
 - c) Create a new measure by clicking the **New Measure** button in the ribbon.



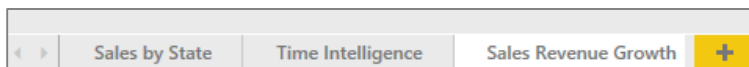
- d) Enter the following DAX expression into the formula bar to create the measure named **Sales Growth PM**.

```
Sales Growth PM =
DIVIDE(
    SUM(Sales[SalesAmount]) -
    CALCULATE(
        SUM(Sales[SalesAmount]),
        PREVIOUSMONTH(Calendar[Date])
    ),
    CALCULATE(
        SUM(Sales[SalesAmount]),
        PREVIOUSMONTH(Calendar[Date])
    )
)
```

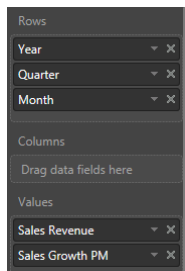
- e) Press the **ENTER** key to add the calculated column to data model.
- f) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Percentage**. Also use the spinner control below the format menu to set the number of decimal places shown to **2**.



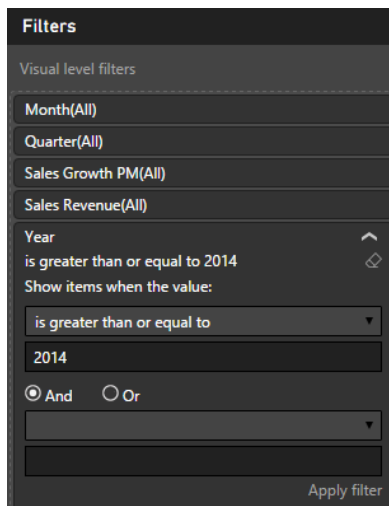
2. Create a new page in the project's report.
- Navigate to report view.
 - Add a new page by clicking the (+) button to the right of the current page title.
 - Once the new page has been created, modify its title to **Sales Revenue Growth**.



3. Create a new matrix visual to show month-to-month sales revenue growth in 2014 and 2015.
- Click the **New Visual** button on the ribbon to add a new visual to the page.
 - Change the visual to a matrix by clicking the Matrix button in the **Visualizations** list.
 - Drag and drop the **Year** column from the **Calendar** table into the **Rows** well.
 - Drag and drop the **Quarter** column from the **Calendar** table into the **Rows** well.
 - Drag and drop the **Month** column from the **Calendar** table into the **Rows** well.
 - Drag and drop the **Sales Revenue** measure from the **Sales** table into the **Value** well.
 - Drag and drop the **Sales Growth PM** measure from the **Sales** table into the **Value** well.
 - The **Rows** well and the **Values** well for the matrix visual should match the following screenshot.



- i) Move down in the **Visualizations** page to the **Filters** section. Set a filter for the **Year** column where value is greater or equal 2014.



- j) Use your mouse to resize the matrix visual so you can see all the rows and columns. Give the matrix visual a width that is almost the same size of the page so you can add additional columns over the next few steps without having to resize the visual again.

Year	Quarter	Month	Sales Revenue	Sales Growth PM
2014	2014-Q1	Jan 2014	\$629,969	-18.13 %
		Feb 2014	\$609,637	-3.23 %
		Mar 2014	\$628,618	3.11 %
	Total		\$1,868,225	142.79 %
	2014-Q2	Apr 2014	\$661,588	5.24 %
		May 2014	\$748,193	13.09 %
		Jun 2014	\$814,333	8.84 %
	Total		\$2,224,114	253.81 %
	2014-Q3	Jul 2014	\$788,469	-3.18 %
		Aug 2014	\$869,143	10.23 %
		Sep 2014	\$890,958	2.51 %
	Total		\$2,548,569	212.96 %
2014-Q4	Oct 2014	\$988,789	10.98 %	
	Nov 2014	\$999,574	1.09 %	
	Dec 2014	\$1,644,980	64.57 %	
Total		\$3,633,343	307.80 %	
Total		\$10,274,251	1235.23 %	
2015	2015-Q1	Jan 2015	\$959,863	-41.65 %
		Feb 2015	\$989,330	0.99 %
		Mar 2015	\$675,533	-30.31 %
	Total		\$2,604,726	58.34 %
	2015-Q2	Apr 2015	\$722,456	6.95 %
		May 2015	\$698,311	-3.34 %
		Jun 2015	\$785,793	12.53 %
	Total		\$2,206,560	226.64 %
	2015-Q3	Jul 2015	\$921,994	17.33 %
		Aug 2015	\$1,094,189	17.59 %
		Sep 2015	\$1,088,663	0.43 %
	Total		\$3,095,046	293.88 %
2015-Q4	Oct 2015	\$1,211,810	11.29 %	
	Nov 2015	\$1,305,029	7.69 %	
	Dec 2015	\$1,732,932	32.79 %	
Total		\$4,249,771	290.29 %	
Total		\$12,156,103	638.98 %	
Total		\$22,430,354	2815.03 %	

- k) Inspect the values produced by the **Sales Growth PM** measure. You can see that a value has been calculated for each month. You should also notice that the matrix currently displays values for the **Sales Growth PM** measure in the **Total** row. The values in the **Total** row are calculated at the quarterly level and not at the month level.

Year	Quarter	Month	Sales Revenue	Sales Growth PM
2014	2014-Q1	Jan 2014	\$629,969	-18.13 %
		Feb 2014	\$609,637	-3.23 %
		Mar 2014	\$628,618	3.11 %
		Total	\$1,868,225	142.79 %
	2014-Q2	Apr 2014	\$661,588	5.24 %
		May 2014	\$748,193	13.09 %
		Jun 2014	\$814,333	8.84 %
		Total	\$2,224,114	253.81 %
	2014-Q3	Jul 2014	\$788,469	-3.18 %
		Aug 2014	\$869,143	10.23 %

The **Sales Growth PM** measure was written to perform calculations on a month-to-month basis. However, there is currently a problem whenever this measure is evaluated in the context based on a larger time interval such as a quarter or a year. More specifically, the **Sales Growth PM** measure is currently producing a large and erroneous value when it is evaluated in the context of a quarter. Now that you have seen the problem, it's time to modify the DAX expression for the **Sales Growth PM** measure to return a blank value whenever the measure is evaluated in a context where the time interval is at a granularity other than at the month level.

4. Modify the DAX expression for the **Sales Growth PM** measure.

- Navigate to data view.
- Select the **Sales Growth PM** measure of the **Sales** table from the **Fields** list. When you select the **Sales Growth PM** measure in the **Fields** list, you should then be able to see and modify its DAX expression in the formula bar.
- Before you can modify the DAX expression for the **Sales Growth PM** measure, you must be able to use the **ISFILTERED** function provided by DAX. You can write the following DAX expression to determine whether the current evaluation context is filtering at the month level.

```
ISFILTERED(Calendar[Month])
```

- You can also write the following DAX expression to make sure that the current evaluation context is not filtering at a more granular level such as at the **Date** level.

```
ISFILTERED(Calendar[Date]) = FALSE()
```

- You will need to ensure that both these expressions are true before the **Sales Growth PM** measure evaluates to a value other than a blank value. You can write the following DAX expression using the DAX **&&** operator to return true when both inner conditions are true

```
( ISFILTERED(Calendar[Month]) && ISFILTERED(Calendar[Date]) = FALSE() )
```

- Update the DAX expression for the **Sales Growth PM** measure to match the following code listing.

```
Sales Growth PM =
IF(
    ( ISFILTERED(Calendar[Month]) && ISFILTERED(Calendar[Date]) = FALSE() ),
    DIVIDE(
        SUM(Sales[SalesAmount]) -
        CALCULATE(
            SUM(Sales[SalesAmount]),
            PREVIOUSMONTH(Calendar[Date])
        ),
        CALCULATE(
            SUM(Sales[SalesAmount]),
            PREVIOUSMONTH(Calendar[Date])
        )
    ),
    BLANK()
)
```

- Navigate back to report view and inspect the effects of your changes to the visual on the **Sales Revenue Growth** page.
- You should see that the **Sales Growth PM** measure is now returning blank values in the **Total** row for the quarterly evaluation.

Year	Quarter	Month	Sales Revenue	Sales Growth PM
2014	2014-Q1	Jan 2014	\$629,969	-18.13 %
		Feb 2014	\$609,637	-3.23 %
		Mar 2014	\$628,618	3.11 %
		Total	\$1,868,225	
	2014-Q2	Apr 2014	\$661,588	5.24 %
		May 2014	\$748,193	13.09 %
		Jun 2014	\$814,333	8.84 %
		Total	\$2,224,114	
	2014-Q3	Jul 2014	\$788,469	-3.18 %
		Aug 2014	\$869,143	10.23 %

It is widely-accepted among BI experts and BI novices alike that a blank value is always preferable to a large, erroneous value.

5. Create a measure named **Sales Growth PM Eval** that inspects the value of the **Sales Growth PM** measure and evaluates to a short string value to indicate the health of the sales growth value.
 - a) Navigate to data view.
 - b) Select the **Sales** table from the **Fields** list.
 - c) Create a new measure by clicking the **New Measure** button in the ribbon.
 - d) Enter the following DAX expression into the formula bar to create the measure named **Sales Growth PM Eval**.

```
Sales Growth PM Eval =
IF(
    ISNUMBER([Sales Growth PM]),
    SWITCH(TRUE(),
        ([Sales Growth PM] >= 0.2), "EXCELLENT",
        ([Sales Growth PM] >= 0.1), "GOOD",
        ([Sales Growth PM] >= 0), "OK",
        ([Sales Growth PM] >= -0.1), "BAD",
        ([Sales Growth PM] < -0.1), "AWFUL"
    )
)
```

- e) Navigate to report view and select the matrix visual on the **Sales Revenue Growth** page.
- f) Drag and drop the **Sales Growth PM Eval** measure from the **Sales** table into the **Values** well in the **Visualizations** pane.

Values
Sales Revenue
Sales Growth PM
Sales Growth PM Eval

- g) You should now see values for the **Sales Growth PM Eval** measure which indicate the health of sales revenue growth.

Year	Quarter	Month	Sales Revenue	Sales Growth PM	Sales Growth PM Eval
2014	2014-Q1	Jan 2014	\$629,969	-18.13 %	AWFUL
		Feb 2014	\$609,637	-3.23 %	BAD
		Mar 2014	\$628,618	3.11 %	OK
		Total	\$1,868,225		
	2014-Q2	Apr 2014	\$661,588	5.24 %	OK
		May 2014	\$748,193	13.09 %	GOOD
		Jun 2014	\$814,333	8.84 %	OK
		Total	\$2,224,114		
	2014-Q3	Jul 2014	\$788,469	-3.18 %	BAD
		Aug 2014	\$869,143	10.23 %	GOOD

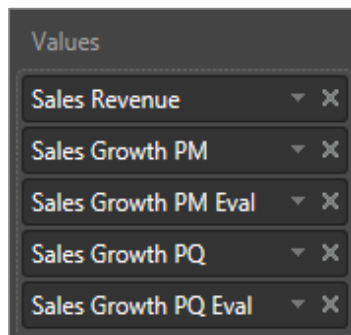
6. Create a measure named **Sales Growth PQ** that calculates the percentage increase between sales revenue for the current quarter and sales revenue for the previous quarter.
 - a) Navigate to data view.
 - b) Select the **Sales** table from the **Fields** list.
 - c) Create a new measure by clicking the **New Measure** button in the ribbon.
 - d) Enter the following DAX expression into the formula bar to create the measure named **Sales Growth PQ**.

```
Sales Growth PQ =
IF(
    ( ISFILTERED('Calendar'[Quarter]) &&
      (ISFILTERED('Calendar'[Month]) = FALSE() && ISFILTERED(Calendar[Date]) = FALSE())
    ),
    DIVIDE(
        SUM(Sales[SalesAmount]) -
        CALCULATE(
            SUM(Sales[SalesAmount]),
            PREVIOUSQUARTER(Calendar[Date])
        ),
        CALCULATE(
            SUM(Sales[SalesAmount]),
            PREVIOUSQUARTER(Calendar[Date])
        )
    ),
    BLANK()
)
```

- e) Press the **ENTER** key to add the calculated column to data model.
 - f) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Percentage**. Also use the spinner control below the format menu to set the number of decimal places shown to **2**.
7. Create a measure named **Sales Growth PQ Eval** that inspects the value of the **Sales Growth PQ** measure and evaluates to a short string value to indicate the health of the sales growth value.
 - a) Navigate to data view.
 - b) Select the **Sales** table from the **Fields** list.
 - c) Create a new measure by clicking the **New Measure** button in the ribbon.
 - d) Enter the following DAX expression into the formula bar to create the measure named **Sales Growth PQ Eval**.

```
Sales Growth PQ Eval =
IF(
    ISNUMBER([Sales Growth PQ]),
    SWITCH(TRUE(),
        ([Sales Growth PQ] >= 0.2), "EXCELLENT",
        ([Sales Growth PQ] >= 0.1), "GOOD",
        ([Sales Growth PQ] >= 0), "OK",
        ([Sales Growth PQ] >= -0.1), "BAD",
        ([Sales Growth PQ] < -0.1), "AWFUL"
    )
)
```

8. Add the **Sales Growth PQ** measure and the **Sales Growth PQ Eval** measure to the matrix visual.
 - a) Navigate to report view and select the matrix visual on the **Sales Revenue Growth** page.
 - b) There should already be three fields in the **Values** well for this visual.
 - c) Drag and drop the **Sales Growth PQ** measure from the **Sales** table into the **Values** well in the **Visualizations** pane.
 - d) Drag and drop the **Sales Growth PQ Eval** measure from the **Sales** table into the **Values** well in the **Visualizations** pane.



- e) You should now see values for the **Sales Growth PQ** measure and the **Sales Growth PQ Eval** measure in the **Total** rows which are evaluated at the quarterly level.

Year	Quarter	Month	Sales Revenue	Sales Growth PM	Sales Growth PM Eval	Sales Growth PQ	Sales Growth PQ Eval
2014	2014-Q1	Jan 2014	\$629,969	-18.13 %	AWFUL		
		Feb 2014	\$609,637	-3.23 %	BAD		
		Mar 2014	\$628,618	3.11 %	OK		
		Total	\$1,868,225			-3.02 %	BAD
	2014-Q2	Apr 2014	\$661,588	5.24 %	OK		
		May 2014	\$748,193	13.09 %	GOOD		
		Jun 2014	\$814,333	8.84 %	OK		
		Total	\$2,224,114			19.05 %	GOOD
	2014-Q3	Jul 2014	\$788,469	-3.18 %	BAD		
		Aug 2014	\$869,143	10.23 %	GOOD		
		Sep 2014	\$890,958	2.51 %	OK		
		Total	\$2,548,569			14.59 %	GOOD

9. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.