Extending Datasets to Support Time Intelligence

Agenda

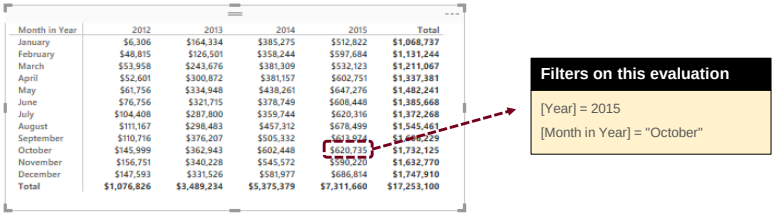
- Understanding Evaluation Contexts
- Creating a Time Dimension Table
- Understanding DAX Time Intelligence Support
- Working with DAX Time Intelligence Functions

Evaluation Contexts

- DAX expressions evaluated using two context
  - Filter context
  - Row context
- Row Context
  - The set of active rows in a calculation
- Filter Context
  - The current row in a table iteration

Understanding Filter Context

- Visuals apply various filters in evaluation contexts



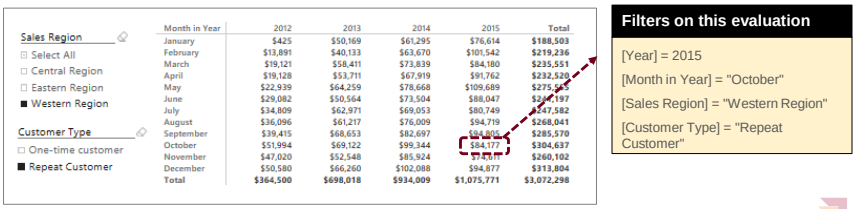
Month in Year	2012	2013	2014	2015	Total
January	$6,306	$164,334	$385,275	$512,822	$1,068,737
February	$48,815	$126,501	$358,244	$597,684	$1,131,244
March	$53,958	$243,676	$381,309	$532,123	$1,211,067
April	$52,601	$300,872	$381,157	$602,751	$1,337,381
May	$61,756	$334,948	$438,261	$647,276	$1,482,241
June	$76,756	$321,715	$378,749	$608,448	$1,385,668
July	$104,408	$287,800	$359,744	$620,316	$1,372,268
August	$111,167	$298,483	$457,312	$678,499	$1,545,461
September	$110,716	$376,207	$505,332	[illegible]	[illegible]
October	$145,999	$362,943	$602,448	$620,735	$1,732,125
November	$156,751	$340,228	$545,572	$590,220	$1,632,770
December	$147,593	$331,526	$581,977	$686,814	$1,747,910
Total	$1,076,826	$3,489,234	$5,375,379	$7,311,660	$17,253,100

Filters on this evaluation

[Year] = 2015

[Month in Year] = "October"

- Filter context also affected by slicers and other filters



Sales Region

Select All

Central Region

Eastern Region

Western Region

Customer Type

One-time customer

Repeat Customer

Month in Year	2012	2013	2014	2015	Total
January	$425	$50,169	$61,295	$76,614	$188,503
February	$13,891	$40,133	$63,670	$101,542	$219,236
March	$19,121	$58,411	$73,839	$84,180	$235,551
April	$19,128	$53,711	$67,919	$91,762	$232,520
May	$22,939	$64,259	$78,668	$109,689	$275,555
June	$29,082	$50,564	$73,504	$88,047	$241,197
July	$34,809	$62,971	$69,053	$80,749	$247,582
August	$36,096	$61,217	$76,009	$94,719	$268,041
September	$39,415	$68,653	$82,697	[illegible]	$285,570
October	$51,994	$69,122	$99,344	$84,177	$304,637
November	$47,020	$52,548	$85,924	[illegible]	$260,102
December	$50,580	$66,260	$102,088	$94,877	$313,804
Total	$364,500	$698,018	$934,009	$1,075,771	$3,072,298

Filters on this evaluation

[Year] = 2015

[Month in Year] = "October"

[Sales Region] = "Western Region"

[Customer Type] = "Repeat Customer"

Understanding Row Context

- There are 2 cases that create active row context
  - Evaluation of a calculated column

```
=(TODAY()-[BirthDate])/365
```

```
=LEFT([ProductCategory], Find(" >", [ProductCategory]))
```

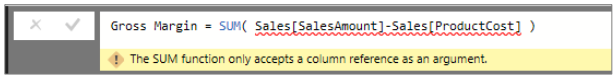
  - Evaluation of a DAX function that iterates over a table

```
=SUMX(RELATEDTABLE(Sales), Sales[SalesAmount])
```

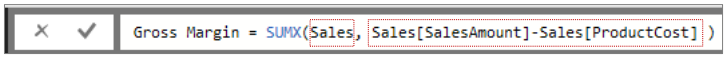
```
=SUMX(Sales,
      Sales[Quantity] * RELATED(Products[UnitCost]))
```

Understanding Iterators Like SUMX

- Standard aggregation functions (e.g. SUM) have no row context
  - You can sum an individual column but you cannot sum an expressions



- Iterator functions (e.g. SUMX) iterate through rows in a target table
  - First argument accepts an expressions evaluates to a table of rows
  - Second argument accepts expression that evaluated for each row in table



- Iterator functions can also be used in calculated columns

ProductId	Product	UnitCost	ListPrice	Category	Subcategory	Product Sales Revenue
1	Batman Action Figure	$6.85	$14.95	Action Figures	Tough Guys	$130,767.65
2	Captain America Action Figure	$7.05	$12.95	Action Figures	Tough Guys	$499,714.6
3	GI Joe Action Figure	$6.10	$14.95	Action Figures	Tough Guys	$168,905.1
4	Green Hulk Action Figure	$2.85	$9.95	Action Figures	Tough Guys	$82,764.1
5	Red Hulk Alter Ego Action Figure	$2.85	$9.95	Action Figures	Tough Guys	$13,691.2

DAX Functions With Tables Parameter

- The following DAX functions create row context
  - AVERAGEX
  - COUNTAX
  - COUNTX
  - MAXX
  - MINX
  - SUMX

DAX Functions that Return a Table

- ALL
- ALLEXCEPT
- CALCULATETABLE
- DISTINCT
- FILTER
- RELATEDTABLE
- VALUES

Using the CALCULATE Function

- CALCULATE provides greatest amount of control
  - You'll see more about Calculate later in this module

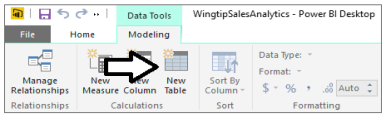
```
Sales Revenue RT =
CALCULATE(
    SUM(Sales[SalesAmount]),
    FILTER(
        ALL('Calendar'),
        'Calendar'[Date] <= MAX('Calendar'[Date])
    )
)
```


Agenda

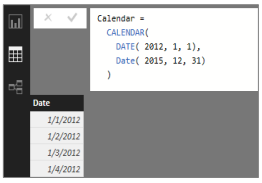
- ✓ Understanding Evaluation Contexts
- ➢ Creating a Time Dimension Table
- Understanding DAX Time Intelligence Support
- Working with DAX Time Intelligence Functions

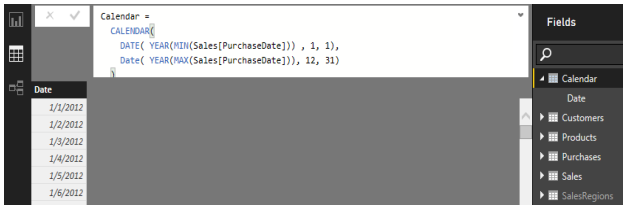
Creating Calendar Table as Calculated Table

- Use **New Table** command in ribbon



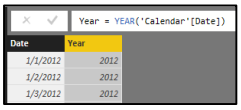
- Create calculate table using CALENDAR function in DAX



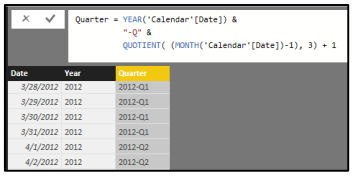


Adding Columns to Calendar Table

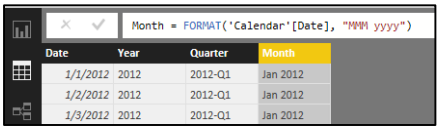
- Year



- Quarter



- Month

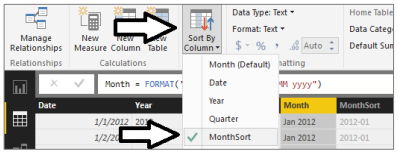


Configuring Sort Columns

- Some columns do not sort in desired fashion by default
  - For example, April will sort before January, February and March
  - Column must be configured with sort column for desired sorting

```
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



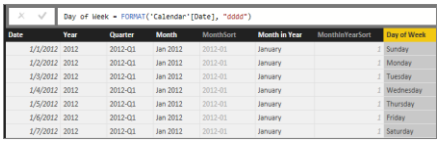
Columns for Month in Year and Day in week

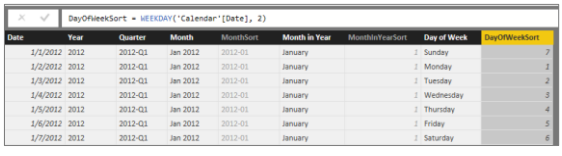
```
Month in Year = FORMAT('Calendar'[Date], "MMMM")
```

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

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

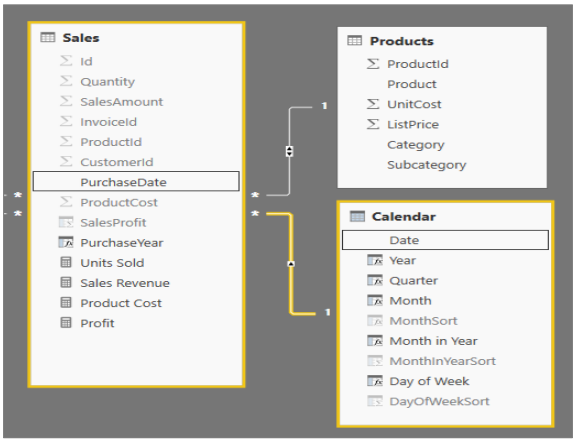
Date	Year	Quarter	Month	MonthSort	Month in Year	MonthInYearSort
1/1/2012	2012	2012-Q1	Jan 2012	2012-01	January	1
1/2/2012	2012	2012-Q1	Jan 2012	2012-01	January	1
1/3/2012	2012	2012-Q1	Jan 2012	2012-01	January	1





Integrating Calendar Table into Data Model

- Calendar table needs relationship to one or more tables



Creating Visuals with a Calendar Table

- **Year** for row labels and **Month in Year** as column labels

Year	January	February	March	April	May	June	July	August	September	October	November	December	Total
2012	$6,306	$48,815	$53,958	$52,601	$61,756	$76,756	$104,408	$111,167	$110,716	$145,999	$156,751	$147,593	$1,076,826
2013	$164,334	$126,501	$243,676	$300,872	$334,948	$321,715	$287,800	$298,483	$376,207	$362,943	$340,228	$331,526	$3,489,234
2014	$385,275	$358,244	$381,309	$381,157	$438,261	$378,749	$359,744	$457,312	$505,332	$602,448	$545,572	$581,977	$5,375,379
2015	$532,822	$597,684	$532,123	$602,751	$647,276	$608,448	$620,316	$678,499	$613,974	$620,735	$590,220	$686,814	$7,311,660
Total	$1,068,737	$1,131,244	$1,211,067	$1,337,381	$1,482,241	$1,385,668	$1,372,268	$1,545,461	$1,606,229	$1,732,125	$1,632,770	$1,747,910	$17,253,100

- **Month in Year** for row labels and **Year** as column labels

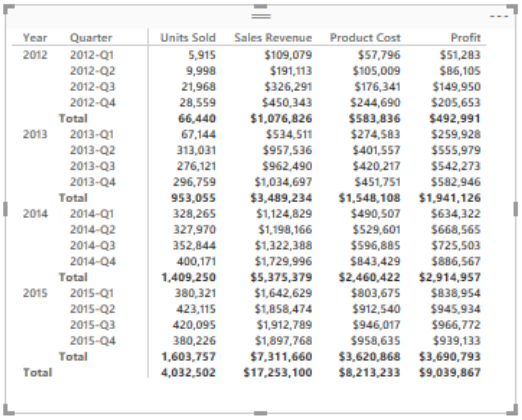
Month in Year	2012	2013	2014	2015	Total
January	$6,306	$164,334	$385,275	$532,822	$1,068,737
February	$48,815	$126,501	$358,244	$597,684	$1,131,244
March	$53,958	$243,676	$381,309	$532,123	$1,211,067
April	$52,601	$300,872	$381,157	$602,751	$1,337,381
May	$61,756	$334,948	$438,261	$647,276	$1,482,241
June	$76,756	$321,715	$378,749	$608,448	$1,385,668
July	$104,408	$287,800	$359,744	$620,316	$1,372,268
August	$111,167	$298,483	$457,312	$678,499	$1,545,461
September	$110,716	$376,207	$505,332	$613,974	$1,606,229
October	$145,999	$362,943	$602,448	$620,735	$1,732,125
November	$156,751	$340,228	$545,572	$590,220	$1,632,770
December	$147,593	$331,526	$581,977	$686,814	$1,747,910
Total	$1,076,826	$3,489,234	$5,375,379	$7,311,660	$17,253,100

- **Month in Year** for row labels and **Year** as column labels

Day of Week	2012	2013	2014	2015	Total
Monday	$163,919	$486,428	$802,943	$1,090,519	$2,543,808
Tuesday	$154,244	$462,695	$772,438	$1,163,588	$2,552,965
Wednesday	$162,811	$425,869	$752,274	$1,029,697	$2,370,651
Thursday	$169,687	$513,948	$837,905	$1,066,526	$2,588,067
Friday	$146,575	$582,823	$736,297	$1,059,080	$2,524,774
Saturday	$162,219	$546,922	$694,322	$909,334	$2,312,797
Sunday	$117,372	$470,550	$779,200	$992,915	$2,360,038
Total	$1,076,826	$3,489,234	$5,375,379	$7,311,660	$17,253,100

Hierarchical Row Labels in a Matrix

- Dimensional hierarchy can be represented with matrix
  - No support as of November 2016 to collapse and expand children



Year	Quarter	Units Sold	Sales Revenue	Product Cost	Profit
2012	2012-Q1	5,915	$109,079	$57,796	$51,283
	2012-Q2	9,998	$191,113	$105,009	$86,105
	2012-Q3	21,968	$326,291	$176,341	$149,950
	2012-Q4	28,559	$450,343	$244,690	$205,653
	Total	**66,440**	**$1,076,826**	**$583,836**	**$492,991**
2013	2013-Q1	67,144	$534,511	$274,583	$259,928
	2013-Q2	313,031	$957,536	$401,557	$555,979
	2013-Q3	276,121	$962,490	$420,217	$542,273
	2013-Q4	296,759	$1,034,697	$451,751	$582,946
	Total	**953,055**	**$3,489,234**	**$1,548,108**	**$1,941,126**
2014	2014-Q1	328,265	$1,124,829	$490,507	$634,322
	2014-Q2	327,970	$1,198,166	$529,601	$668,565
	2014-Q3	352,844	$1,322,388	$596,885	$725,503
	2014-Q4	400,171	$1,729,996	$843,429	$886,567
	Total	**1,409,250**	**$5,375,379**	**$2,460,422**	**$2,914,957**
2015	2015-Q1	380,321	$1,642,629	$803,675	$838,954
	2015-Q2	423,115	$1,858,474	$912,540	$945,934
	2015-Q3	420,095	$1,912,789	$946,017	$966,772
	2015-Q4	380,226	$1,897,768	$958,635	$939,133
	Total	**1,603,757**	**$7,311,660**	**$3,620,868**	**$3,690,793**
Total		**4,032,502**	**$17,253,100**	**$8,213,233**	**$9,039,867**

Agenda

- ✓ Understanding Evaluation Contexts
- ✓ Creating a Time Dimension Table
- ➢ Understanding DAX Time Intelligence Support
- Working with DAX Time Intelligence Functions

Functions That Return a Single Date

- FIRSTDATE / LASTDATE
- FIRSTNONBLANK / LASTNONBLANK
- STARTOFMONTH / ENDOFMONTH
- STARTOFQUARTER / ENDOFQUARTER
- STARTOFYEAR / ENDOFYEAR

To Date Functions

- Convenience To Date functions
  - TOTALMTD
  - TOTALQTD
  - TOTALYTD
- Convenience To Date functions really do this
  - CALCULATE (Expression, DATESMTD(Date) [, SetFilter])
  - TOTALMTD (Expression, Date [, SetFilter])

Function That Return a Table of Dates

- PREVIOUSDAY / NEXTDAY
- PREVIOUSMONTH / NEXTMONTH
- PREVIOUSQUARTER / NEXTQUARTER
- PREVIOUSYEAR / NEXTYEAR
- DATESMTD / DATESQTD / DATESYTD
- SAMEPERIODLASTYEAR
- DATEADD
- DATESBETWEEN
- DATESINPERIOD
- PARALLELPERIOD

Balance Functions

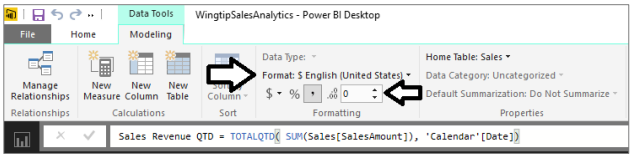
- DAX Functions return value at specific point in time
  - OPENINGBALANCEMONTH
  - OPENINGBALANCEQUARTER
  - OPENINGBALANCEYEAR
  - CLOSINGBALANCEMONTH
  - CLOSINGBALANCEQUARTER
  - CLOSINGBALANCEYEAR

Agenda

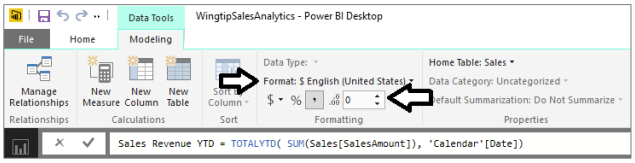
- ✓ Understanding Evaluation Contexts
- ✓ Creating a Time Dimension Table
- ✓ Understanding DAX Time Intelligence Support
- ➢ Working with DAX Time Intelligence Functions

Calculated Fields for QTD and YTD Sales

- TOTALQTD function calculates quarter-to-date totals

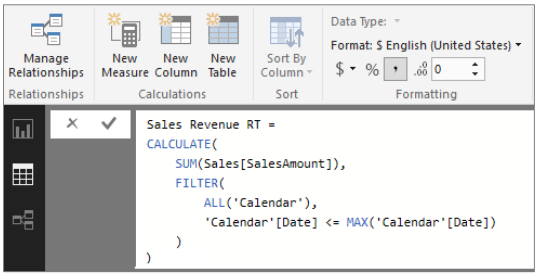


- TOTALYTD function calculates year-to-date totals



Using the CALCULATE Function

- Calculate a running total of sales revenue across years



Matrix Visual with To-Date Running Totals

- Running totals calculated using DAX

Year	Quarter	Month	Sales Revenue	Sales Revenue QTD	Sales Revenue YTD	Sales Revenue RT
2012	2012-Q1	Jan 2012	$6,306	$6,306	$6,306	$6,306
		Feb 2012	$48,815	$55,121	$55,121	$55,121
		Mar 2012	$53,958	$109,079	$109,079	$109,079
	2012-Q2	Apr 2012	$52,601	$52,601	$161,680	$161,680
		May 2012	$61,756	$114,357	$223,436	$223,436
		Jun 2012	$76,756	$191,113	$300,192	$300,192
	2012-Q3	Jul 2012	$104,408	$104,408	$404,600	$404,600
		Aug 2012	$111,167	$215,575	$515,767	$515,767
		Sep 2012	$110,716	$326,291	$626,483	$626,483
	2012-Q4	Oct 2012	$145,999	$145,999	$772,483	$772,483
		Nov 2012	$156,751	$302,750	$929,234	$929,234
		Dec 2012	$147,593	$450,343	$1,076,826	$1,076,826
2013	2013-Q1	Jan 2013	$164,334	$164,334	$164,334	$1,241,161
		Feb 2013	$126,501	$290,835	$290,835	$1,367,661

- Question: when did company reach $1,000,000 in sales

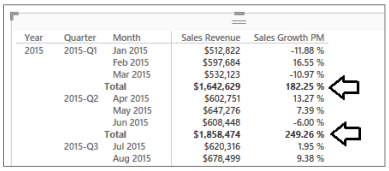
Nov 2012	$156,751	$302,750	$929,234	$929,234
Dec 2012	$147,593	$450,343	$1,076,826	$1,076,826
Jan 2013	$164,334	$164,334	$164,334	$1,241,161
Feb 2013	$126,501	$290,835	$290,835	$1,367,661

Sales Growth PM Measure - First Attempt

- Create a 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])
  )
)
```

- Use measure in matrix evaluating month and quarter
  - Measure returns correct value for Month
  - Measure returns incorrect and erroneous value for Quarter



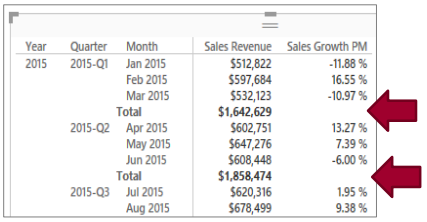
Year	Quarter	Month	Sales Revenue	Sales Growth PM
2015	2015-Q1	Jan 2015	$512,822	-11.88 %
		Feb 2015	$597,684	16.55 %
		Mar 2015	$532,123	-10.97 %
		Total	**$1,642,629**	**182.25 %**
	2015-Q2	Apr 2015	$602,751	13.27 %
		May 2015	$647,276	7.39 %
		Jun 2015	$608,448	-6.00 %
		Total	**$1,858,474**	**249.26 %**
	2015-Q3	Jul 2015	$620,316	1.95 %
		Aug 2015	$678,499	9.38 %

Using the ISFILTERED Function

- ISFILTERED function used to determine when perform evaluation

```
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()
)
```

- Expression returns Blank when evaluation context isn't appropriate



Year	Quarter	Month	Sales Revenue	Sales Growth PM
2015	2015-Q1	Jan 2015	$512,822	-11.88 %
		Feb 2015	$597,684	16.55 %
		Mar 2015	$532,123	-10.97 %
		Total	**$1,642,629**	
	2015-Q2	Apr 2015	$602,751	13.27 %
		May 2015	$647,276	7.39 %
		Jun 2015	$608,448	-6.00 %
		Total	**$1,858,474**	
	2015-Q3	Jul 2015	$620,316	1.95 %
		Aug 2015	$678,499	9.38 %

Simulating KPIs with Power BI Desktop

- KPIs are not yet support
  - But you can create something similar using measures

```
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), "BAD"
  ))
```

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

Year	Quarter	Month	Sales Revenue	Sales Growth PM	Sales Growth PM Eval	Sales Growth PQ	Sales Growth PQ Eval
2014	2014-Q1	Jan 2014	$385,275	16.21 %	GOOD		
		Feb 2014	$358,244	-7.02 %	BAD		
		Mar 2014	$381,309	6.44 %	OK		
		Total	**$1,124,829**			**8.71 %**	**OK**
	2014-Q2	Apr 2014	$381,157	-0.04 %	BAD		
		May 2014	$438,261	14.98 %	GOOD		
		Jun 2014	$378,749	-13.58 %	AWFUL		
		Total	**$1,198,166**			**6.52 %**	**OK**
	2014-Q3	Jul 2014	$359,744	-5.02 %	BAD		
		Aug 2014	$457,312	27.12 %	EXCELLENT		
		Sep 2014	$505,332	10.50 %	GOOD		
		Total	**$1,322,388**			**10.37 %**	**GOOD**
	2014-Q4	Oct 2014	$602,448	19.22 %	GOOD		
		Nov 2014	$545,572	-9.44 %	BAD		
		Dec 2014	$581,977	6.67 %	OK		
		Total	**$1,729,996**			**30.82 %**	**EXCELLENT**

Summary

- ✓ Understanding Evaluation Contexts
- ✓ Creating a Time Dimension Table
- ✓ Understanding DAX Time Intelligence Support
- ✓ Working with DAX Time Intelligence Functions