IT Real-Time training that work for your career.

PROVIDED TRAINING FOR THOUSANDS OF STUDENTS

SUBJECT, MATERIAL & VIDEOS

VIDEOS PROVIDED

POWER BI

DATE FUNCTIONS

TIME INTELLIGENCE FUNCTIONS

MSBI
IS, AS, RS & MDS
POWER BI
SERVER, DESKTOP & DAX

Trainings:

CLASS ROOM

ONLINE

FAST TRACK

ONE ON ONE

PROJECT TRAINING

Address:

Flat No: 506/B
Nilgiri Block
Aditya Enclave
Mytrivanam Area
Hyderabad.

Website & Blog

www.vinaytechhouse.com

www.msbivinay.blogspot.in

Contact Information

+91 9573168449

040 66638869

Sri VINAY TechHouse

WE'VE WORKED WITH A DIVERSE CUSTOMER BASE. HOW CAN WE HELP YOU?

IT Training, Support and Consulting.

DATE FUNCTIONS [What exactly Date Functions perform?]

Real-time usage of DATE functions

To work on year, month, day, hour, minute, second values individually, to work on date differences, and to generate calendars these functions are helpful.

DATE FUNCTIONS	DESC
CALENDAR	Generate calendar from the given dates
CALENDARAUTO	Automatically generate calendar based on your data model dates availability
DAY	Day from the date value
MONTH	Month from the date value
YEAR	Year from the date value
YEARFRAC	Fraction of year based on the result, rounded to the upper integer.
DATE	Consider three values as date
TIME	Consider three values as time
DATEVALUE	Consider string as date
TIMEVALUE	Consider string as time
EDATE	End of date. Add months to the date. Ex: End date of a policy based on period
EOMONTH	End of month (After adding months gives you the last day in the month)
WEEKDAY	Weekday of the given date { Value between 1-7, 1-Sun…7-Sat}
WEEKNUM	Current week month, by considering 52 weeks in a year
TODAY	Current date with time 12AM
UTCTODAY	Universal Time Coordinator date and time (12AM) London Time
NOW	Current date with current time
UTCNOW	Universal Time Coordinator date and current time London Time
DATEDIFF	Differences between two specified date in the form of interval

Calendar

Returns a table with a single column named “Date” that contains a contiguous set of dates. The range of dates is from the specified start date to the specified end date, inclusive of those two dates.

Syntax:

CALENDAR(<start_date>, <end_date>)

Examples

The following formula returns a table with dates between January 1st, 2005 and December 31st, 2015.

=CALENDAR (DATE (2005, 1, 1), DATE (2015, 12, 31))

Real-time: If we want a calendar on dashboard with business start day of month and end day of month.

For a data model which includes actual sales data and future sales forecasts. The following expression returns the date table covering the range of dates in these two table

=CALENDAR (MINX (Sales, [Date]), MAXX (Forecast, [Date]))

=CALENDAR (MINX (FactPayments, [Date]), MAXX (FactPayments, [Date]))

CalendarAuto

Returns a table with a single column named “Date” that contains a contiguous set of dates. The range of dates is calculated automatically based on data in the model.

Difference between Calendar and CanlendarAuto?

Syntax:

CALENDARAUTO([fiscal_year_end_month])

Example

In this example, the MinDate and MaxDate in the data model are July 1, 2010 and June 30, 2011.

CALENDARAUTO()will return all dates between January 1, 2010 and December 31, 2011.

CALENDARAUTO(3) will return all dates between March 1, 2010 and February 28, 2012.

Date

Returns the specified date in **datetime** format.

Syntax:

DATE(<year>, <month>, <day>)

Example:

Returning a Simple Date

Description: The following formula returns the date July 8, 2009:

Code

=DATE(2009,7,8)

Example:

Years after 1899

Code

=DATE(08,1,2)

Example:

Years before 1899

Code

=DATE(1800,1,2)

Example:

Years after 1899

Code

=DATE(2008,1,2)

Example: *Working with Months*

Code

=DATE(2008,14,2)

=DATE(2008,-3,2)

Example:

Working with Days

Code

=DATE(2008,1,35)

=DATE(2008,1,-15)

DateDiff

Returns the count of interval boundaries crossed between two dates.

Syntax:

DATEDIFF(<start_date>, <end_date>, <interval>)

Example

= DATEDIFF(date(2015,1,1),date(2016,1,1), year)

The following all return 1:

DATEDIFF(min(DimDate[Date]),max(DimDate[Date]), second))
DATEDIFF(min(DimDate[Date]),max(DimDate[Date]), minute))
DATEDIFF(MIN(Calendar[Date]), MAX(Calendar[Date], hour))

DATEDIFF(MIN(Calendar[Date]), MAX(Calendar[Date], day))

DATEDIFF(MIN(Calendar[Date]), MAX(Calendar[Date], week))

DATEDIFF(MIN(Calendar[Date]), MAX(Calendar[Date], month))

DATEDIFF(MIN(Calendar[Date]), MAX(Calendar[Date], quarter))

DATEDIFF(MIN(Calendar[Date]), MAX(Calendar[Date], year))

DateValue

Converts a date in the form of text to a date in datetime format.

Syntax

DATEVALUE(date_text)

Example

The following example returns a different **datetime** value depending on your computer's locale and

=DATEVALUE("8/1/2009")

Day

Returns the day of the month, a number from 1 to 31.

Syntax:

Day (<date>)

Example:

Getting the Day from a Date Column

Description

The following formula returns the day from the date in the column, [Birthdate].

Code

=DAY(DimDate[Date])

Example:

Getting the Day from a String Date

Description

The following formulas return the day, 4, using dates that have been supplied as strings in an accepted text format.

Code

=DAY("3-4-1007")

=DAY("March 4 2007")

Example: *Using a Day Value as a Condition*

Description

The following expression returns the day that each sales order was placed, and flags the row as a promotional sale item if the order was placed on the 10th of the month.

Code

=IF(DAY([SalesDate])=10,"promotion","")

```
Next_Day=NEXTDAY(DimDate[Date])
-Nextday always requires column [New column and test]
```

EDate

Returns the date that is the indicated number of months before or after the start date. Use **EDATE to calculate maturity dates or due dates** that fall on the same day of the month as the date of issue.

You took a policy premium for 15 years, then what is the end date?

Syntax:

EDATE(<start_date>, <months>)

Example:

The following example returns the date three months after the order date, which is stored in the column [TransactionDate].

=EDATE([TransactionDate],3)

EOmonth

Returns the date in **datetime** format of the last day of the month, before or after a specified number of months.

Use EOMONTH to calculate maturity dates or due dates that fall on **the last day of the month.**

Syntax

EOMONTH(<start_date>, <months>)

Example:***The following expression returns May 31, 2008, because the months argument is rounded to 2.***

=EOMONTH("March 3, 2008",1.5)

Hour

Returns the hour as a number from 0 (12:00 A.M.) to 23 (11:00 P.M.).

Syntax

HOUR(<datetime>)

Example:

The following example returns the hour from the TransactionTime column of a table named Orders.

=HOUR('Orders'[TransactionTime])

Example

The following example returns 15, meaning the hour corresponding to 3 PM in a 24-hour clock. The text value is automatically parsed and converted to a date/time value.

=HOUR("March 3, 2008 3:00 PM")

Minute

Returns the minute as a number from 0 to 59, given a date and time value.

Syntax

MINUTE(<datetime>)

Example

The following example returns the minute from the value stored in the TransactionTime column of the Orders table.

=MINUTE(Orders[TransactionTime])

Example

The following example returns 45, which is the number of minutes in the time 1:45 PM.

=MINUTE("March 23, 2008 1:45 PM")

Month

Returns the month as a number from 1 (January) to 12 (December).

Syntax

MONTH(<datetime>)

Example

The following expression returns 3, which is the integer corresponding to March, the month in the date argument.

=MONTH("March 3, 2008 3:45 PM")

Example

The following expression returns the month from the date in the TransactionDate column of the Orders table.

=MONTH(Orders[TransactionDate])

Now

Differneces between NOW and TODAY()

Returns the current date and time in **datetime** format.

The NOW function is useful when you need to display the current date and time on a worksheet or calculate a value based on the current date and time, and have that value updated each time you open the worksheet.

Syntax

NOW()

Example

The following example returns the current date and time plus 3.5 days:

=NOW()+3.5

Second

Returns the seconds of a time value, as a number from 0 to 59.

Syntax:

_SECOND(<time>)

Example

The following formula returns the number of seconds in the time contained in the TransactionTime column of a table named Orders.

=SECOND('Orders'[TransactionTime])

The following formula returns 3, which is the number of seconds in the time represented by the value, March 3, 2008 12:00:03.

=SECOND("March 3, 2008 12:00:03")

Time

Converts hours, minutes, and seconds given as numbers to a time in **datetime** format.

Syntax:

TIME(hour, minute, second)

Differneces between DATE and TIME?

Example:

The following examples both return the time, 3:00 AM:

=TIME(27,0,0)

=TIME(3,0,0)

Example:

The following examples both return the time, 12:30 PM:

=TIME(0,750,0)

=TIME(12,30,0)

Example:

The following example creates a time based on the values in the columns, intHours , intMinutes , intSeconds :

=TIME([intHours],[intMinutes],[intSeconds])

Difference TIME and TimeValue?

TimeValue

Converts a time in text format to a time in datetime format.

Syntax:

TIMEVALUE(time_text)

Example: =TIMEVALUE("20:45:30")

Difference between Today() and Now() or UTCToday() and UTCNow()?

ToDay

Returns the current date.

Syntax:

TODAY()

Example

If you know that someone was born in 1963, you might use the following formula to find that person's age as of this year's birthday:

=YEAR(TODAY())-1963

UTCNow

Returns the current UTC date and time

Syntax:

UTCNOW()

Example

The following:

EVALUATE { FORMAT(UTCNOW(), "General Date") }

Returns:

[VALUE]
2/2/2018 4:48:08 AM

UTCToday

Returns the current UTC date.

Syntax:

UTCTODAY()

Example

The following:

EVALUATE { FORMAT(UTCTODAY(), "General Date") }

Returns:

[VALUE]2/2/2018

WeekDay

Returns a number from 1 to 7 identifying the day of the week of a date. By default the day ranges from 1 (Sunday) to 7 (Saturday).

Syntax:

WEEKDAY(<date>, <return_type>)

Example

=WEEKDAY([HireDate]+1)

WeekNum

Returns the week number for the given date and year according to the **return_type** value. The week number indicates where the week falls numerically within a year.

Syntax:

WEEKNUM(<date>, <return_type>)

Example

The following example returns the week number of the date February 14, 2010.

=WEEKNUM("Feb 14, 2010", 2)

Example*:*

The following example returns the week number of the date stored in the column, HireDate, from the table, Employees.

=WEEKNUM('Employees'[HireDate])

Year

Returns the year of a date as a four digit integer in the range 1900-9999.

Syntax:

YEAR(<date>)

Example:

The following example returns 2007.

=YEAR("March 2007")

Example:

Date as Result of Expression

Description

The following example returns the year for today's date.

Code

=YEAR(TODAY())

Oracle system has date format “DD/MM/YYYY”, then how do we convert into “YYYY-MM-DD”?

Using format function

```
= FORMAT(DATEVALUE("04/07/2019"),"YYYY-MM-DD")
```

YearFrac

Calculates the fraction of the year represented by the number of whole days between two dates. Use the YEARFRAC worksheet function to identify the proportion of a whole year's benefits or obligations to assign to a specific term.

Syntax:

YEARFRAC(<start_date>, <end_date>, <basis>)

Example

The following example returns the fraction of a year represented by the difference between the dates in the two columns, TransactionDate and ShippingDate :

=YEARFRAC(Orders[TransactionDate],Orders[ShippingDate])

Example

The following example returns the fraction of a year represented by the difference between the dates, January 1 and March 1:

```
=YEARFRAC("Jan 1 2007","Mar 1 2007")
```

Create Date Table Manually [Trainer constructured using various functions]

```
DimDate=
ADDCOLUMNS (
CALENDAR (DATE(2000,1,1), DATE(2025,12,31)),
"DateAsInteger", FORMAT ( [Date], "YYYYMMDD" ),
"Year", YEAR ( [Date] ),
"Monthnumber", FORMAT ( [Date], "MM" ),
"YearMonthnumber", FORMAT ( [Date], "YYYY/MM" ),
"YearMonthShort", FORMAT ( [Date], "YYYY/mmm" ),
"MonthNameShort", FORMAT ( [Date], "mmm" ),
"MonthNameLong", FORMAT ( [Date], "mmmm" ),
"DayOfWeekNumber", WEEKDAY ( [Date] ),
"DayOfWeek", FORMAT ( [Date], "dddd" ),
"DayOfWeekShort", FORMAT ( [Date], "ddd" ),
"Quarter", "Q" & FORMAT ( [Date], "Q" ),
"YearQuarter", FORMAT ( [Date], "YYYY" ) & "/Q" & FORMAT ( [Date], "Q" )
```

Time-intelligence functions

Date and Time Functions	**Time Intelligence Functions**
Date and time conversion, consideration, datepart operations etc…	Running dates of current and past involved here.

DATESBETWEEN	Return dates between the starting and end dates [2 dates required]
DATESINPERIOD	Return dates between the starting date and the interval [1 date +1 interval required]
ClosingBalanceofmonth	In the current content [year] closing balance of the last month or a specific month closing balance
ClosingBalanceOfQuarter	In the current content [year] closing balance of the last quarter or specific quarter closing balance
ClosingBalanceOfYear	
OpeningBalanceOfMonth	In the current content [year] opening balance of the first month
OpeningBalanceOfQuarter	In the current context [year] opening balance of the first month
OpeningBalanceOfyear	
DATESYTD	Year to date date values , Dates are returned
DATESQTD	Quarter to date date values,Dates are returned
DATESMTD	Month to date date values, Dates are returned
TOTALQTD	Total value for the quarter to date, Total value returned
TOTALMTD	Total value for the month to date, Total value returned
TOTALYTD	Total value for the year to date, Total value returned
FIRSTDATE	First date in the current content
FIRSTNONBLANKDATE	First non blank date in the current content
LASTDATE	Last date in the current content
LASTNONBLANKDATE	Last non blank date in the current content
NEXTDAY	Next day in the current context
NEXTMONTH	Next month in the current context
NEXTQUARTER	Next quarter in the current context
NEXTYEAR	Next year in the current context
PARALLEL PERIOD	Previous years, quarters, months displayed based on the date and interval. Close to previous functions.
SAMEPERIODLASTYEAR	Last year same period in the current context
PREVIOUSDAY	Previous day in the current context
PREVIOUSMONTH	Previous month in the current context [Complete month values]
PREVIOUSQUARTER	Previous quarter in the current context [Complete Quarter values]
PREVIOUSYEAR	Previous year in the current context [Complete year values]
STARTOFMONTH	Month starting in the current context
STARTOFQUARTER	Quarter starting in the current context
STARTOFYEAR	Start year of the business

When do we go for these functions?

Why date table reference mandatory for Time Intelligence

To perform time calculations (not based on dates) which are based on running times or executed times, these calculations are helpful.

They work on years, quarters, months, days etc...for the preceding (previous) or following (next) dates or time information.

Overview: Periodical calucations [year / month / day/ week current or previous or parallel dates etc…calculations and ratios]

ClosingBalanceMonth

Evaluates the **expression** at the last date of the month in the current context.

Syntax:

CLOSINGBALANCEMONTH(<expression>,<dates>[,<filter>])

Example

The following sample formula creates a measure that calculates the 'Month End Inventory Value' of the product inventory.

o see how this works, create a PivotTable and add the fields, CalendarYear, MonthNumberOfYear and DayNumberOfMonth, to the **Row Labels** area of the PivotTable. Then add a measure, named **Month EndInventory Value**, using the formula defined in the code section, to the **Values** area of the PivotTable.

Code

```
CLOSINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime [DateKey])
```

Sc: Closing balances of Online courses

```
ClosingBalanceOfAMonth = CLOSINGBALANCEMONTH( sum(FactPayments[Discount_Fee]),DimDate[Date],FILTER(DimCourseMode,DimCourseMode[ModeID]="Online"))
```

ClosingBalanceQuarter

Evaluates the **expression** at the last date of the quarter in the current context.

Syntax:

CLOSINGBALANCEQUARTER(<expression>,<dates>[,<filter>])

Example

The following sample formula creates a measure that calculates the 'Quarter End Inventory Value' of the productinventory.

To see how this works, create a PivotTable and add the fields, CalendarYear, CalendarQuarter and MonthNumberOfYear, to the **Row Labels** area of the PivotTable. Then add a measure, named **Quarter EndInventory Value**, using the formula defined in the code section, to the **Values** area of the PivotTable.

Code

```
=CLOSINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])
```

```
ClosingBalanceOfAQuarter = CLOSINGBALANCEQUARTER(
sum(FactPayments[Discount_Fee]),DimDate[Date],FILTER(DimCourseMode,DimCourseMode[ModeID]="Online"))
```

ClosingBalanceYear

Evaluates the **expression** at the last date of the year in the current context.

Syntax:

CLOSINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])

The following sample formula creates a measure that calculates the 'Month End Inventory Value' of the product

inventory. To see how this works, create a PivotTable and add the fields, CalendarYear, MonthNumberOfYear and DayNumberOfMonth, to the Row Labels area of the PivotTable. Then add a measure, named Month End

Inventory Value, using the formula defined in the code section, to the Values area of the PivotTable.

Code

```
=CLOSINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])
```

Example: Closing DiscountFee of year for the Online mode

```
ClosingBalanceOfAYear = CLOSINGBALANCEYEAR(
sum(FactPayments[Discount_Fee]),DimDate[Date],FILTER(DimCourseMode,DimCourseMode[ModeID]="Online"))
```

DATEADD

Returns a table that contains a column of dates, shifted either forward or backward in time by the specified number of intervals from the dates in the current context.

Syntax:

DATEADD(<dates>,<number_of_intervals>,<interval>)

Example:

Shifting a Set of Dates

Description

The following formula calculates dates that are one year before the dates in the current context.

Code [New table]

```
Newtable = ADDCOLUMNS(DimDate,"Dateadd",DATEADD(DimDate[Date],-1,year))
```

DatesBetween

Returns a table that contains a column of dates that begins with the **start_date** and continues until the **end_date**.

Syntax:

DATESBETWEEN(<dates>,<start_date>,<end_date>)

Example: Create table between the specified table

```
Partial_Calendar =
DATESBETWEEN(DimDate[Date],date(2019,01,01),date(2019,03,31) )
```

Example :Finding the Total Discount Fee for Jan and Feb months for the required courses.

a)Take Slicer with CourseID

b)Create a new measure like below and use inside a card

```
DiscountFee_Jan_Feb =
calculate(sum(FactPayments[Discount_Fee]),DATESBETWEEN(DimDate[Date],date(2019,01,01),date(2019,02,28) ))
```

c)Choose CourseID and see the result

DatesInPeriod

Q:Difference between Datesbetween and DatesInPeriod?

Returns a table that contains a column of dates that begins with the **start_date** and continues for the specified **number_of_intervals**.

Syntax

DATESINPERIOD(<dates>,<start_date>,<number_of_intervals>,<interval>)

Example 1: Generate a calendar with the given date period

Partial_Calendar = DATESINPERIOD(DimDate[Date],date(2019,01,01),-20,DAY)

Example

Finding the Total Discount Fee for the last 20 days from the specified date for courses.

a)Take Slicer with CourseID

b)Create a new measure like below and use inside a card

DiscountFee_Specifc_Days = calculate(sum(FactPayments[Discount_Fee]),
DATESINPERIOD(DimDate[Date],date(2019,01,01),-58,DAY)))

c)Choose CourseID and see the result

TABLE UNDERSTANDING:

Year	Quarter	Month	Day	Sales
2019	1	1	1	1000
2019	1	1	3	2000
2019	1	2	8	2000
2019	1	2	12	1000
2019	2	4	27	2000
2019	2	4	14	1000

Assume today is 15thApril2019

Q: What is Year To Date sales?

Starting day of the year to till date in this year

1000+2000+2000+1000+2000+1000=9000

Q: What is Quarter To Date sales?

Starting date of the quarter to till date in this quarter

2000+1000=3000

Q: What is Month To Date sales?

Starting date of the month to till date in this month

2000+1000=3000

Scenario: Assume today is 15th May 2019, then what about TotalYTD, TotalQTD, and TotalMTD?

TOTALMTD: 1st May- 15th May

TOTALQTD: 1st April – 15th May

TOTALYTD: 1st Jan- 15th May

Scenario: Last year 20000 sales, this year 30000 sales, then what is the year over year sales?

Year over year sales= current-previous =30000-20000=10000

Year over year sales growth=(current-previous)/current =(30000-20000)/30000=0.3

Additional: Compulsory for practice [Ensure DimDate marked as Date Table]

Identifying total internet sales amount

sum of sales:=sum(FactInternetSales[SalesAmount])

Identifying current quarter sales till today.

Current quarter sales:=**TOTALQTD**(FactInternetSales[sum of Sales],dimdate[FullDateAlternateKey])

Note: Use maxdate as current date to identify till this quarter data

Create a max date measure like below:

Maxdate= DATE(year(today()), month(today()), day(today())

Totalqtd(FactInternetSales[sum of Sales],dimdate[FullDateAlternateKey], dimdate[Date]<=maxdate)

Identifying current month sales till today.

Current month sales:=**TOTALMTD**(FactInternetSales[sum of sales],dimdate[FullDateAlternateKey])

Note: Use maxdate as current date to identify till this month data

Totalmtd(FactInternetSales[sum of Sales],dimdate[FullDateAlternateKey], dimdate[Date]<=maxdate)

Identifying previous quarter sales.

Previous quarter sales:=Calculate([sum of sales],**PreviousQuarter**(dimdate[FullDateAlternateKey]))

Identifying previous month sales.

Previous month sales:=Calculate([sum of sales],**Previousmonth**(dimdate[FullDateAlternateKey]))

Identifying day over day growth percentage

Legacy Method: Require three measures

Measure 1: Fulltotal

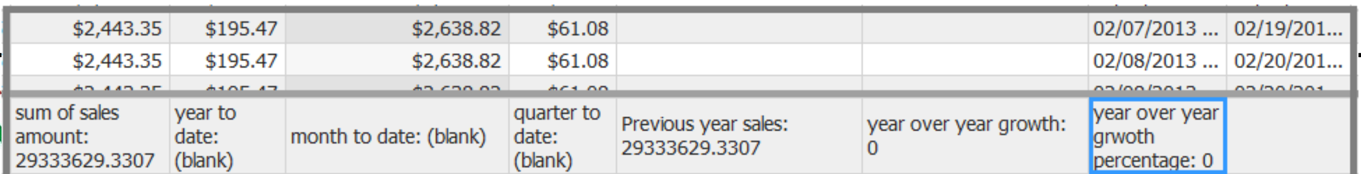
Sum of fee = sum([sum of sales])

Measure 2: Lastyear total

Sum of Previous Year = CALCULATE([sum of sales],SAMEPERIODLASTYEAR(dimdate[FullDateAlternateKey]))

Measure3: Year over year growth

Sum of Amount YoY% = IF([sum of sales] ,DIVIDE(([sum of sales]-[Sumof Previous Year]), [sum of sales]))

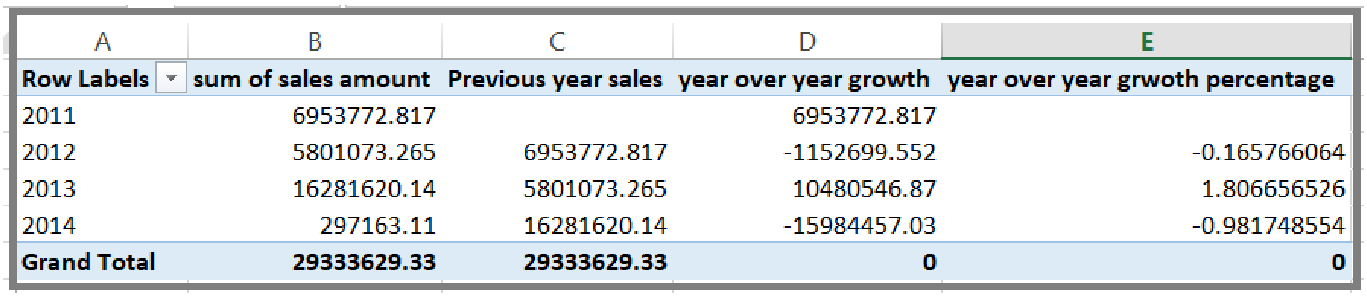


$2,443.35	$195.47	$2,638.82	$61.08			02/07/2013 ...	02/19/201...
$2,443.35	$195.47	$2,638.82	$61.08			02/08/2013 ...	02/20/201...
sum of sales amount: 29333629.3307	year to date: (blank)	month to date: (blank)	quarter to date: (blank)	Previous year sales: 29333629.3307	year over year growth: 0	year over year grwoth percentage: 0	

Observation:

a) Publish or Deploy after all measures creation

b) Model Menu--> Analyze in Excel, and specify like below [Drag and drop]



Row Labels	sum of sales amount	Previous year sales	year over year growth	year over year grwoth percentage
2011	6953772.817		6953772.817	
2012	5801073.265	6953772.817	-1152699.552	-0.165766064
2013	16281620.14	5801073.265	10480546.87	1.806656526
2014	297163.11	16281620.14	-15984457.03	-0.981748554
Grand Total	**29333629.33**	**29333629.33**	**0**	**0**

Step1:

Row Section: Drag and drop year from DimDate table

Column Section: Sum of Sales

Observe

Step2:

Take

Column Section: Sum of Sales, Previous Year Sales

Observe

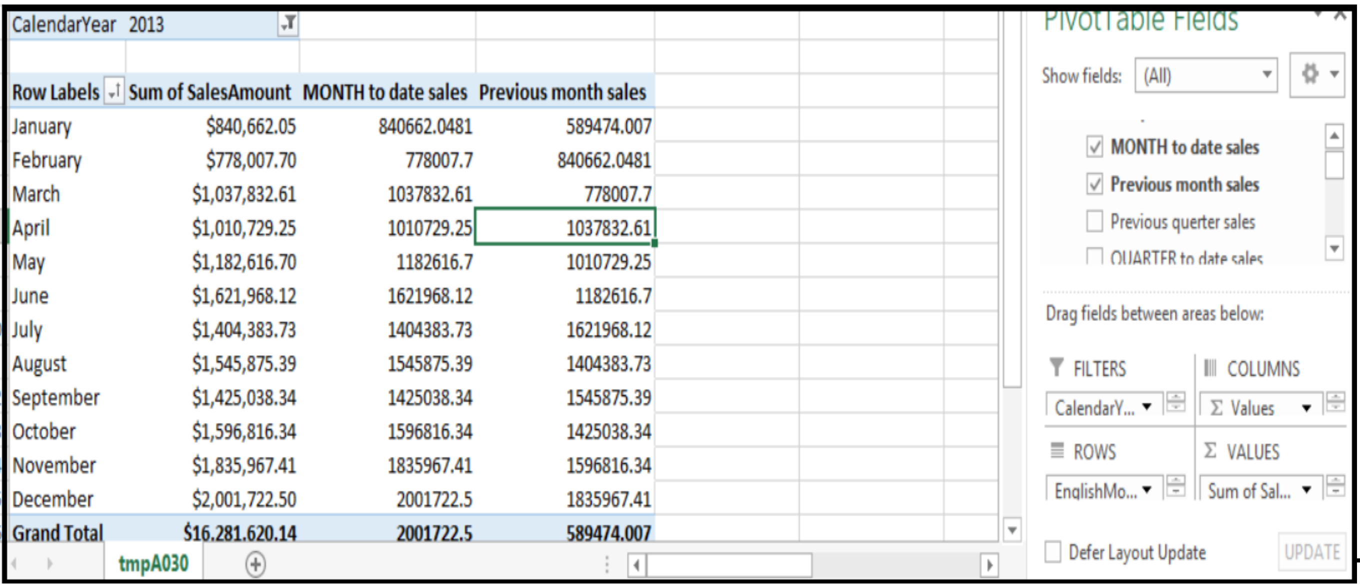
Step3:

Take

Column Section: Sum of Sales, Previopus Year Sales, Year over Year measure

Observe

Step4: For months data validation, do like below



CalendarYear 2013

Row Labels	Sum of SalesAmount	MONTH to date sales	Previous month sales
January	$840,662.05	840662.0481	589474.007
February	$778,007.70	778007.7	840662.0481
March	$1,037,832.61	1037832.61	778007.7
April	$1,010,729.25	1010729.25	1037832.61
May	$1,182,616.70	1182616.7	1010729.25
June	$1,621,968.12	1621968.12	1182616.7
July	$1,404,383.73	1404383.73	1621968.12
August	$1,545,875.39	1545875.39	1404383.73
September	$1,425,038.34	1425038.34	1545875.39
October	$1,596,816.34	1596816.34	1425038.34
November	$1,835,967.41	1835967.41	1596816.34
December	$2,001,722.50	2001722.5	1835967.41
Grand Total	$16,281,620.14	2001722.5	589474.007

PivotTable Fields

Show fields: (All)

☑ MONTH to date sales
☑ Previous month sales
☐ Previous querter sales
☐ QUARTER to date sales

Drag fields between areas below:

FILTERS: CalendarY...
COLUMNS: Σ Values
ROWS: EnglishMo...
VALUES: Sum of Sal...

☐ Defer Layout Update

Returns a table that contains a column of the dates for the month to date, in the current context.

Syntax

DATESMTD(<dates>)

Example

Disply month to date Discount Fee

Q: How it operates at filters and current data?

Q: Incase MTD, YTD and QTD having discrepancies, how do you identify and resolve?

```
DiscountFee MTD = CALCULATE(SUM(FactPayments[Discount_Fee]),
DATESMTD(FactPayments[Date]))
DiscountFee MTD = TOTALMTD(SUM(FactPayments[Discount_Fee]), DimDate[Date]))
```

```
To practice DatesMTD/QTD/YTD or TotalMTD/QTD/YTD
a)Take Year Slicer
b)Take Month Slicer
c)Take Quarter Slicer
d) Take Card and use the measure created above
```

DatesQTD

Returns a table that contains a column of the dates for the quarter to date, in the current context.

Syntax

DATESQTD(<dates>)

```
DiscountFee QTD = CALCULATE(SUM(FactPayments[Discount_Fee]),
DATESQTD(FactPayments[Date]))
DiscountFee QTD = TOTALQTD(SUM(FactPayments[Discount_Fee]), DimDate[Date]))
```

DatesYTD

Returns a table that contains a Column of the dates for the year to date, in the current context.

Syntax

DATESYTD(<dates> [,<year_end_date>])

```
DiscountFee YTD = CALCULATE(SUM(FactPayments[Discount_Fee]),
DATESYTD(FactPayments[Date]))
```

EndOfMonth

Returns the last date of the month in the current context for the specified column of dates.

Syntax :ENDOFMONTH(<dates>)

Example:

```
Last_Month_Of_Year = ENDOFMONTH(DimDate[Date])
```

Q:How these END functions are helpful?

Last_Month_Of_Year=ENDOFMONTH(DateTime[DateKey])

EndOfQuarter

Returns the last date of the quarter in the current context for the specified column of dates.

Syntax

ENDOFQUARTER(<dates>)

Example:

```
Last_Quarter_Of_Year = ENDOFQUARTER(DimDate[Date])
Last_Quarter_Of_Year=ENDOFQUARTER(DateTime[DateKey])
```

EndOfYear

Returns the last date of the year in the current context for the specified column of dates.

Syntax

ENDOFYEAR(<dates> [,<year_end_date>])

=ENDOFYEAR(DateTime[DateKey],"06/30/2004")

Example:

```
Last_Month_Of_Year = ENDOFYEAR(DimDate[Date])

Last_Month_Of_Year=ENDOFYEAR(DateTime[DateKey])
```

FirstDate

Q:How these FIRST functions are helpful?

Returns the first date in the current context for the specified column of dates.

Syntax

FIRSTDATE(<dates>)

Q:Diff between FIRST and FIRSTNONBLANK?

```
FD = FIRSTDATE(DimDate[Date])
EOM = FIRSTDATE(FactPayments[Date]) -First Business Date
```

FirstNonBlank

Returns the first value in the column, **column**, filtered by the current context, where the expression is not blank.

Syntax

FIRSTNONBLANK(<column>,<expression>)

```
   EOM =
FIRSTNONBLANK(FactPayments[Discount_Fee],DimDate[DateDiff2019_2017])
```

LastDate

Returns the last date in the current context for the specified column of dates.

Syntax

LASTDATE(<dates>)

```
LD = LASTDATE(DimDate[Date])
EOM = LASTDATE(FactPayments[Date])
```

LastNonBlank

Returns the last value in the column, **column**, filtered by the current context, where the expression is not blank.

Syntax

LASTNONBLANK(<column>,<expression>)

```
EOM =
LASTNONBLANK(FactPayments[Discount_Fee],DimDate[DateDiff2019_2017])
```

NextDay

Returns a table that contains a column of all dates from the next day, based on the first date specified in the **dates** column in the current context.

Q: When do we go for Next functions?

Syntax

NEXTDAY(<dates>)

```
For a given day, it will find the next day total dicountfee.
```

```
EOM = CALCULATE(SUM(FactPayments[Discount_Fee]),
NEXTDAY(DimDate[Date]))
```

NextMonth

Returns a table that contains a column of all dates from the next month, based on the first date in the **dates** column in the current context.

Syntax

NEXTMONTH(<dates>)

```
For a given month, it will find the next month total dicountfee.
```

```
EOM = CALCULATE(SUM(FactPayments[Discount_Fee]),
NEXTMONTH(DimDate[Date]))
```

NextQuarter

Returns a table that contains a column of all dates in the next quarter, based on the first date specified in the **dates** column, in the current context.

Syntax

NEXTQUARTER(<dates>)

```
For a given quarter, it will find the next quarter total dicountfee.
```

```
EOM = CALCULATE(SUM(FactPayments[Discount_Fee]),
NEXTQUARTER(DimDate[Date]))
```

NextYear

Returns a table that contains a column of all dates in the next year, based on the first date in the **dates** column, in the current context.

Syntax

NEXTYEAR(<dates>[,<year_end_date>])

For a given year, it will find the next year total dicountfee.

```
DiscountFee_NextYear = CALCULATE(SUM(FactPayments[Discount_Fee]),
NEXTYEAR(DimDate[Date]))
```

OpeningBalanceMonth

Evaluates the **expression** at the first date of the month in the current context.

Syntax : OPENINGBALANCEMONTH(<expression>,<dates>[,<filter>])

```
OpeningBalanceOfAMonth = OPENINGBALANCEMONTH(
sum(FactPayments[Discount_Fee]),DimDate[Date],FILTER(DimCourseMode,DimCourse
Mode[ModeID]="Online"))
```

OpeningBalanceQuarter

Evaluates the **expression** at the first date of the quarter, in the current context.

Syntax

Q: When do we go for Opening functions?

OPENINGBALANCEQUARTER(<expression>,<dates>[,<filter>])

```
OpeningBalanceOfQUARTER = OPENINGBALANCEQUARTER(
sum(FactPayments[Discount_Fee]),DimDate[Date],FILTER(DimCourseMode,DimCourse
Mode[ModeID]="Online"))
```

OpeningBalanceYear

Evaluates the **expression** at the first date of the year in the current context.

Syntax

OPENINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])

```
OpeningBalanceOfAyear = OPENINGBALANCEYEAR(
sum(FactPayments[Discount_Fee]),DimDate[Date],FILTER(DimCourseMode,Dim
CourseMode[ModeID]="Online"))
```

=OPENINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])

Q: What is superset and subset functions in this case?

ParallelPeriod

Returns a table that contains a column of dates that represents a period parallel to the dates in the specified **dates** column, in the current context, with the dates shifted a number of intervals either forward in time or back in time.

Q: Similarities between Previous periods and ParallelPeriod?

Syntax

PARALLELPERIOD(<dates>,<number_of_intervals>,<interval>)

The PARALLELPERIOD function is similar to the DATEADD function except that PARALLELPERIOD always returns full periods at the given granularity level instead of the partial periods that DATEADD returns.

Example

The following sample formula creates a measure that calculates the previous year sales for Internet sales.

NOTE

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
PARALLELPERIOD(DateTime[DateKey],-1,year))
```

Total discount for the previous year

```
DiscountFee total = CALCULATE(SUM(FactPayments[Discount_Fee]),
PARALLELPERIOD(DimDate[Date],-1,year))
```

Total discount for the previous month

```
DiscountFee total = CALCULATE(SUM(FactPayments[Discount_Fee]),
PARALLELPERIOD(DimDate[Date],-1,MONTH))
```

Total discount for the previous quarter

```
DiscountFee total = CALCULATE(SUM(FactPayments[Discount_Fee]),
PARALLELPERIOD(DimDate[Date],-1,QUARTER))
```

PreviousDay

Returns a table that contains a column of all dates representing the day that is previous to the first date in the **dates** column, in the current context.

Syntax : PREVIOUSDAY(<dates>)

Example

The following sample formula creates a measure that calculates the 'previous day sales' for the Internet sales.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
PREVIOUSDAY('DateTime'[DateKey]))
```

```
DiscountFee total = CALCULATE(SUM(FactPaymments[DiscountFee]),
PREVIOUSDAY(DimDate[Date]))
```

PreviousMonth

Returns a table that contains a column of all dates from the previous month, based on the first date in the **dates** column, in the current context.

Syntax

PREVIOUSMONTH(<dates>)

Example

The following sample formula creates a measure that calculates the 'previous month sales' for the Internet sales.

=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSMONTH('DateTime'[DateKey]))

DiscountFee total = CALCULATE(SUM(FactPaymments[DiscountFee]), PREVIOUSDMONTH(DimDate[Date]))

PreviousQuarter

Returns a table that contains a column of all dates from the previous quarter, based on the first date in the **dates** column, in the current context.

Syntax

PREVIOUSQUARTER(<dates>)

Example

The following sample formula creates a measure that calculates the 'previous quarter sales' for Internet sales. **=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSQUARTER('DateTime'[DateKey]))**

DiscountFee total = CALCULATE(SUM(FactPaymments[DiscountFee]), PREVIOUSQUARTER(DimDate[Date]))

Previousyear

Returns a table that contains a column of all dates from the previous year, given the last date in the **dates** column, in the current context.

Syntax

PREVIOUSYEAR(<dates>[,<year_end_date>])

How many ways we findout previous year data?

3 ways 1. 2. 3.

The following sample formula creates a measure that calculates the previous year sales for the Internet sales.

=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSYEAR('DateTime'[DateKey]))

DiscountFee total = CALCULATE(SUM(FactPaymments[DiscountFee]), PREVIOUSYEAR(DimDate[Date]))

SamePeriodLastYear

Returns a table that contains a column of dates shifted one year back in time from the dates in the specified **dates** column, in the current context.

Q: Difference between PreviousYear and SamePeriodLastYear?

Previous—Last year entire values

SamePeriodLastYear—Based on date selection.

How do we identify same period two years data?

Syntax

SAMEPERIODLASTYEAR(<dates>)

Example

To see how this works, create a PivotTable and add the fields, CalendarYear to the **Row Labels** area of the PivotTable. Then add a measure, named **Previous Year Sales**, using the formula defined in the code section, to the **Values** area of the PivotTable.

=CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]), SAMEPERIODLASTYEAR(DateTime[DateKey]))

DiscountFee LastYear = CALCULATE(SUM(FactPaymments[DiscountFee]), SAMEPERIODLASTYEAR(DimDate[Date]))

StartOfMonth

Returns the first date of the month in the current context for the specified column of dates.

Syntax

STARTOFMONTH(<dates>)

Example

The following sample formula creates a measure that returns the start of the month, for the current context.

=STARTOFMONTH(DateTime[DateKey])

StartOfQuarter

Returns the first date of the quarter in the current context for the specified column of dates.

Syntax

STARTOFQUARTER(<dates>)

Example

=STARTOFQUARTER(DateTime[DateKey])

StartOfYear

Returns the first date of the year in the current context for the specified column of dates.

Syntax

STARTOFYEAR(<dates>)

Example

=STARTOFYEAR(DateTime[DateKey])

TotalMTD

Evaluates the value of the **expression** for the month to date, in the current context.

Syntax

TOTALMTD(<expression>,<dates>[,<filter>])

Q: Differences between DatesMTD and TotalMTD?

TotalMTD= Sum + DatesMTD

Example

To see how this works, create a PivotTable and add the fields, CalendarYear, MonthNumberOfYear and DayNumberOfMonth, to the **Row Labels** area of the PivotTable. Then add a measure, named **Month-to-dateTotal**, using the formula defined in the code section, to the **Values** area of the PivotTable.

=TOTALMTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
DiscountFee Monthtodate =
TOTALMTD(SUM(FactPayments[DiscountFee]),DimDate[Date])on

TotalQTD

Evaluates the value of the **expression** for the dates in the quarter to date, in the current context.

Syntax

TOTALQTD(<expression>,<dates>[,<filter>])

Example

To see how this works, create a PivotTable and add the fields, CalendarYear, CalendarQuarter and MonthNumberOfYear, to the **Row Labels** area of the PivotTable. Then add a measure, named **Quarter-to-dateTotal**, using the formula defined in the code section, to the **Values** area of the PivotTable.

=TOTALQTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])

DiscountFee Quartertodate =
TOTALQTD(SUM(FactPayments[DiscountFee]),DimDate[Date])on

TotalYTD

Evaluates the year-to-date value of the **expression** in the current context.

Syntax

TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])

Return value

A scalar value that represents the **expression** evaluated for the current year-to-date **dates**.

=TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey],
ALL('DateTime'), "6/30")

Example

To see how this works, create a PivotTable and add the fields, CalendarYear, CalendarQuarter, and MonthNumberOfYear, to the **Row Labels** area of the PivotTable. Then add a measure, named **Year-to-date Total**, using the formula defined in the code section, to the **Values** area of the PivotTable.

=TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])

DiscountFee Monthtodate = TOTALYTD(SUM(FactPayments[DiscountFee]),DimDate[Date])