

## Time Intelligence

What is Time Intelligence?

Time Intelligence means doing calculations over periods of time or dates. All the Time Intelligence Functions will need a **date column** to perform the calculations, and this date column should contain **unique**, **no null** and **contiguous** date values to get the accurate results.

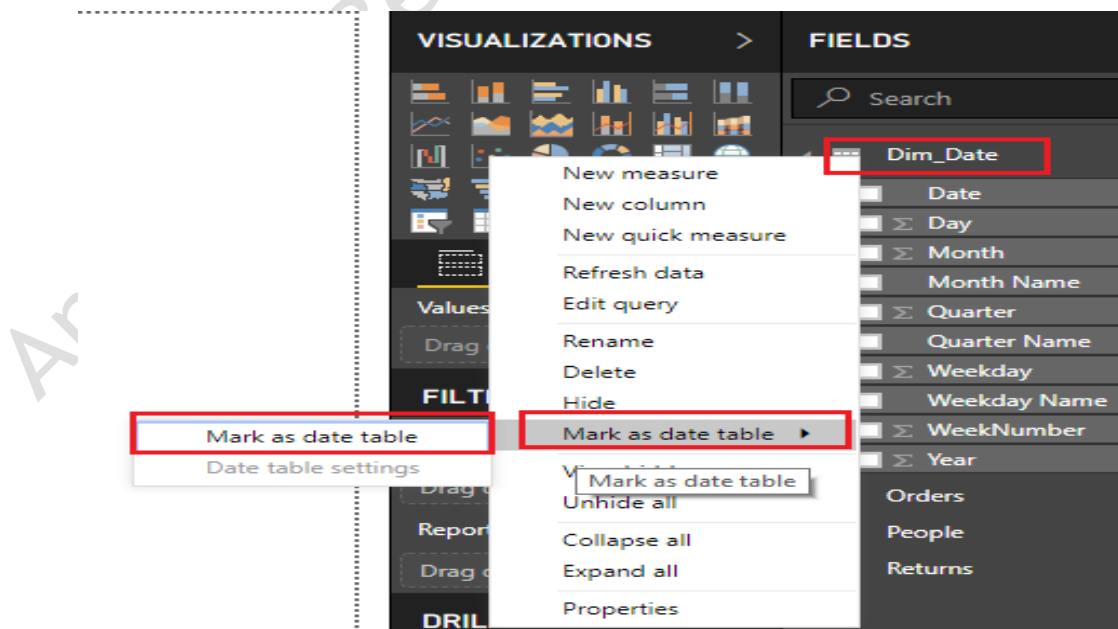
Time Intelligence simply means doing BI calculations over periods of time, or over dates. For example, a common request in many reports is to show a value, such as Sales, aggregated month-by-month, from the beginning of each year, so that you know, in this case, what were the total sales, in that year, up to the given month. DAX, Data Analysis Expressions, the language of Power BI, includes many predefined time intelligence functions, including **TOTALYTD**, which is exactly the function you need to use in this particular scenario.

Many data analysts prefer to create their own date tables, which is fine. In Power BI Desktop, you can specify the table you want your model to use as its date table, and subsequently create date-related visuals, tables, measures, and so on, using that table's date data. When you specify your own date table, you control the date hierarchies created in your model, and use them in measures and other operations that use your model's date table.

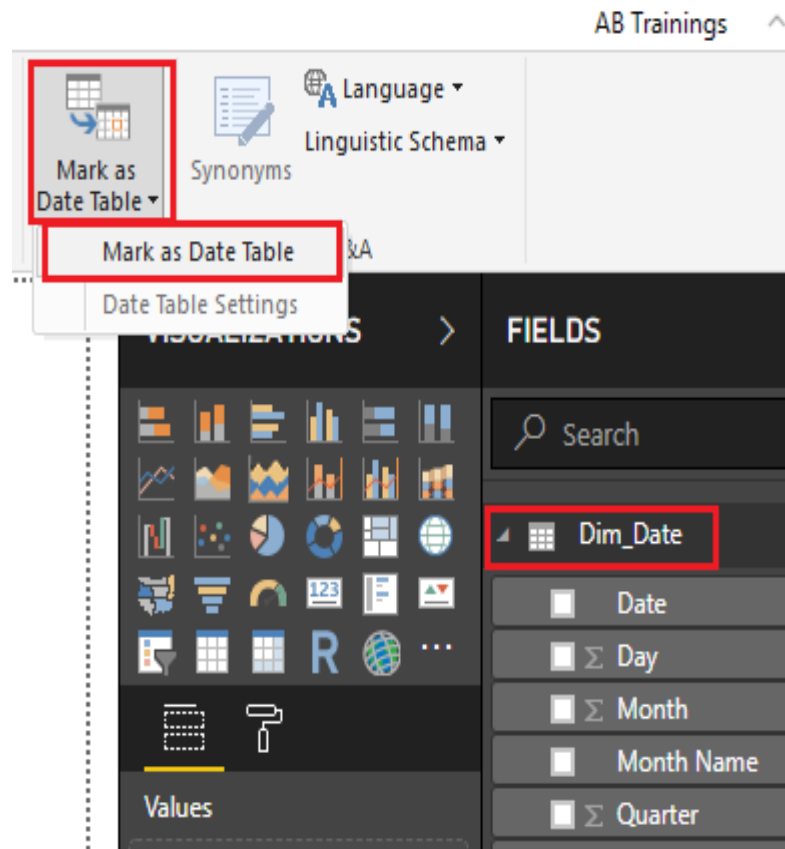
To make DAX Time Intelligence functions work properly set date dimension table as “Mark as date table”.

### Setting your own date table

To set a date table select the table you want to use as a date table in the Fields pane, then right-click the table and select Mark as date table → Mark as date table in the menu that appears, as shown in the following image.



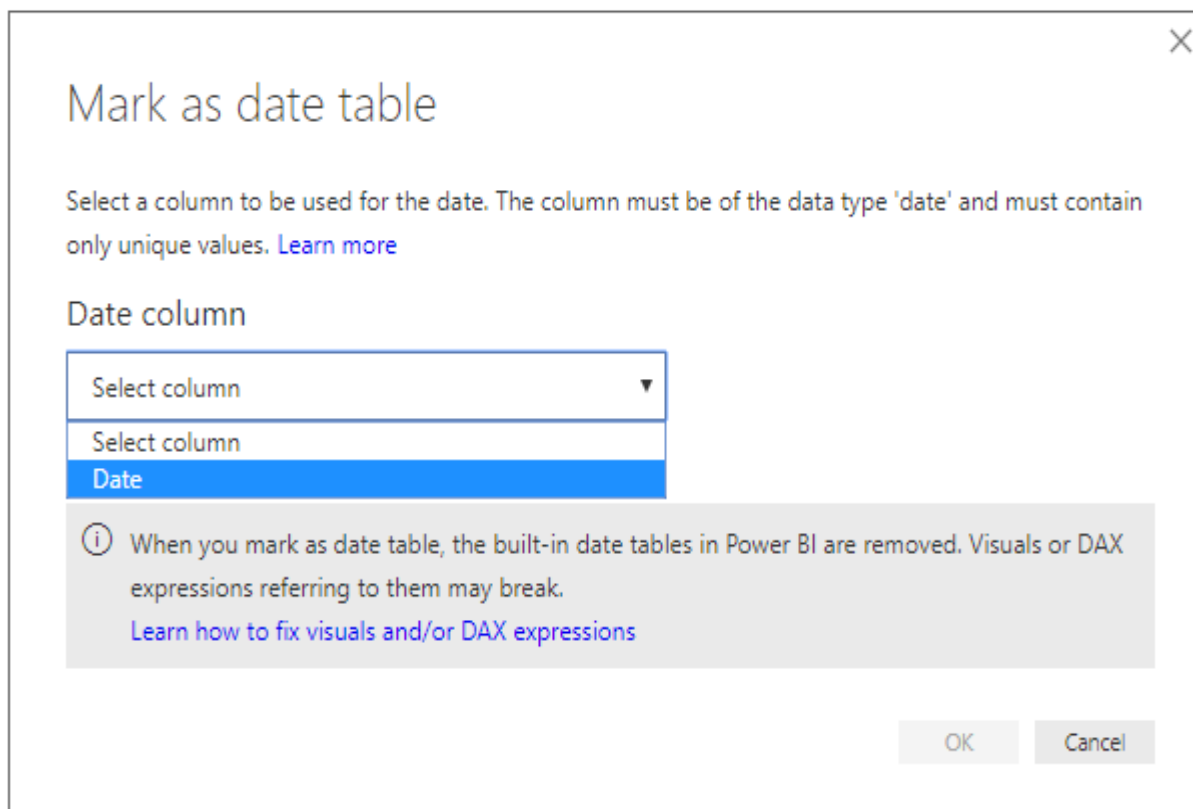
You can also select the table and then select **Mark as Date Table** from the **Modeling ribbon**, shown here.



When you specify your own date table, Power BI Desktop performs the following validations of that column and its data, to ensure that the data

- contains unique values
- contains no null values
- contains contiguous date values (from beginning to end)
- if it is a Date/Time data type, it has the same timestamp across each value

Once you specify a date table, you can select which column in that table is the date column. You can specify which column to use by selecting the table in the Fields pane, then right-click the table and select **Mark as date table** → **Date table settings**. The following window appears, where you can select the column to use as the date table from the drop-down box.



### DAX Time Intelligence Functions

All the Time Intelligence Functions will need a **date column** to perform the calculations, and this date column should contain **unique**, **no null** and **contiguous** date values to get the accurate results.

#### TOTALMTD

Evaluates the value of the expression for the month to date, in the current context. TOTALMTD will give Running Totals or cumulative sum for each Month.

#### Example

Total MTD = TOTALMTD(SUM(Orders[Sales]), Dim\_Date[Date])

#### TOTALQTD

Evaluates the value of the expression for the dates in the quarter to date, in the current context. TOTALQTD will give Running Totals or cumulative sum for each Quarter.

#### Example

Total QTD = TOTALQTD(SUM(Orders[Sales]), Dim\_Date[Date])

**TOTALYTD**

Evaluates the year-to-date value of the expression in the current context. TOTALYTD will give Running Totals or cumulative sum for each year.

**Example**

Total YTD = TOTALYTD(SUM(Orders[Sales]), Dim\_Date[Date])

Year	Quarter Name	Month Name	Day	Sales	Total YTD Ex	Total QTD Ex	Total MTD
2014	Qtr1	January	3	16.45	16.45	16.45	16.45
2014	Qtr1	January	4	288.06	304.51	304.51	304.51
2014	Qtr1	January	5	19.54	324.04	324.04	324.04
2014	Qtr1	January	6	4,407.10	4,731.14	4,731.14	4,731.14
2014	Qtr1	January	7	87.16	4,818.30	4,818.30	4,818.30
2014	Qtr1	January	8		4,818.30	4,818.30	4,818.30
2014	Qtr1	January	9	40.54	4,858.85	4,858.85	4,858.85
2014	Qtr1	January	10	54.83	4,913.68	4,913.68	4,913.68
2014	Qtr1	January	11	9.94	4,923.62	4,923.62	4,923.62
2014	Qtr1	January	12		4,923.62	4,923.62	4,923.62
2014	Qtr1	January	13	3,553.80	8,477.41	8,477.41	8,477.41
2014	Qtr1	January	14	61.96	8,539.37	8,539.37	8,539.37
2014	Qtr1	January	15	149.95	8,689.32	8,689.32	8,689.32
2014	Qtr1	January	16	299.96	8,989.29	8,989.29	8,989.29
2014	Qtr1	January	17		8,989.29	8,989.29	8,989.29
2014	Qtr1	January	18	64.86	9,054.15	9,054.15	9,054.15
2014	Qtr1	January	19	378.59	9,432.74	9,432.74	9,432.74
2014	Qtr1	January	20	2,673.87	12,106.61	12,106.61	12,106.61
2014	Qtr1	January	21	25.25	12,131.86	12,131.86	12,131.86
2014	Qtr1	January	22		12,131.86	12,131.86	12,131.86
2014	Qtr1	January	23	46.02	12,177.88	12,177.88	12,177.88
<b>Total</b>				<b>2,297,200.86</b>			

**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.

**Example**

Previous Day = CALCULATE(SUM(Orders[Sales]), PREVIOUSDAY(Dim\_Date[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.

#### Example

Previous Month = CALCULATE(SUM(Orders[Sales]), PREVIOUSMONTH(Dim\_Date[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.

#### Example

Previous Quarter = CALCULATE(SUM(Orders[Sales]), PREVIOUSQUARTER(Dim\_Date[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.

#### Example

Previous Year = CALCULATE(SUM(Orders[Sales]), PREVIOUSYEAR(Dim\_Date[Date]))

### 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.

#### Example

Next Day = CALCULATE ([Sum of Sales], NEXTDAY(Dim\_Date[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.

#### Example

Next Month = CALCULATE ([Sum of Sales], NEXTMONTH(Dim\_Date[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.

#### Example

Next Quarter = CALCULATE ([Sum of Sales], NEXTQUARTER(Dim\_Date[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.

**Example**

Next Year = CALCULATE ([Sum of Sales], NEXTYEAR(Dim\_Date[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.

**Example**

LastYearSales = CALCULATE(SUM(Orders[Sales]), SAMEPERIODLASTYEAR(Dim\_Date[Date]))

By using **same period last year**, you will be knowing

- Last Year Same Month what is the value
- Last Year Same Quarter what is the value
- Last Year what is the Value

Year	Sales	LastYearSales
2014	484,247.50	
2015	470,532.51	484,247.50
2016	609,205.60	470,532.51
2017	733,215.26	609,205.60
2018		733,215.26
<b>Total</b>	<b>2,297,200.86</b>	<b>2,297,200.86</b>

Year	Quarter Name	SumOfSales	LastYearSales
2014	Qtr1	74,447.80	
2014	Qtr2	86,538.76	
2014	Qtr3	143,633.21	
2014	Qtr4	179,627.73	
2015	Qtr1	68,851.74	74,447.80
2015	Qtr2	89,124.19	86,538.76
2015	Qtr3	130,259.58	143,633.21
2015	Qtr4	182,297.01	179,627.73
2016	Qtr1	93,237.18	68,851.74
2016	Qtr2	136,082.30	89,124.19
2016	Qtr3	143,787.36	130,259.58
2016	Qtr4	236,098.75	182,297.01
2017	Qtr1	123,144.86	93,237.18
2017	Qtr2	133,764.37	136,082.30
2017	Qtr3	196,251.96	143,787.36
2017	Qtr4	280,054.07	236,098.75
2018	Qtr1		123,144.86
2018	Qtr2		133,764.37
2018	Qtr3		196,251.96
2018	Qtr4		280,054.07
<b>Total</b>		<b>2,297,200.86</b>	<b>2,297,200.86</b>

### Year Over Year (YOY Growth)

Current Year Sales = SUM(Orders[Sales])

Last Year Sales = CALCULATE (SUM(Orders[Sales]), SAMEPERIODLASTYEAR(Dim\_Date[Date]))

% YOY = DIVIDE ([Current Year Sales], [Last Year Sales], "NO LY Data")

Year	Sum of Sales	Last Year Sales	%YOY
2014	484,247.50		
2015	470,532.51	484,247.50	97.17%
2016	609,205.60	470,532.51	129.47%
2017	733,215.26	609,205.60	120.36%
2018		733,215.26	
<b>Total</b>	<b>2,297,200.86</b>	<b>2,297,200.86</b>	<b>100.00%</b>

### MOM Growth

Sum of Sales = SUM(Orders[Sales])

Last Month Sales = CALCULATE(SUM(Orders[Sales]), PREVIOUSMONTH(Dim\_Date[Date]))

% MOM = DIVIDE ([Sum of Sales], [Last Month Sales])

Year	Month Name	Sum of Sales	Last Month Sales	% MOM
2014	January	14,236.90		
2014	February	4,519.89	14,236.89	31.75%
2014	March	55,691.01	4,519.89	1232.13%
2014	April	28,295.35	55,691.01	50.81%
2014	May	23,648.29	28,295.35	83.58%
2014	June	34,595.13	23,648.29	146.29%
2014	July	33,946.39	34,595.13	98.12%
2014	August	27,909.47	33,946.39	82.22%
2014	September	81,777.35	27,909.47	293.01%
2014	October	31,453.39	81,777.35	38.46%
2014	November	78,628.72	31,453.39	249.98%
2014	December	69,545.62	78,628.72	88.45%
2015	January	18,174.08	69,545.62	26.13%
2015	February	11,951.41	18,174.08	65.76%
2015	March	38,726.25	11,951.41	324.03%
2015	April	34,195.21	38,726.25	88.30%
2015	May	30,131.69	34,195.21	88.12%
2015	June	24,797.29	30,131.69	82.30%
2015	July	28,765.33	24,797.29	116.00%
2015	August	36,898.33	28,765.33	128.27%
<b>Total</b>		<b>2,297,200.86</b>		