Power Bi DAX Part 1

Practice File Link: Excel File
Power Bi File : Power Bi

Understanding DAX Time Intelligence Functions: TOTALYTD, TOTALQTD, and TOTALMTD Introduction

DAX (Data Analysis Expressions) provides powerful time intelligence functions that help in calculating cumulative totals over different time periods. Among them, TOTALYTD, TOTALQTD, and TOTALMTD are essential functions used for Year-To-Date (YTD), Quarter-To-Date (QTD), Month-To-Date (MTD) calculations. These functions are useful in financial reporting, sales analysis, and trend analysis.

1. TOTALYTD (Total Year-To-Date)

TOTALYTD calculates the cumulative total of an expression from the beginning of the year to a specified date.

Syntax: TOTALYTD(<expression>, <dates_column>)
Example: TotalSalesYTD = Sales [Order Date])

Explanation: This formula calculates the cumulative sales from the start of the year to the given date.

2. TOTALQTD (Total Quarter-To-Date)

TOTALQTD calculates the cumulative total from the beginning of the quarter to a specified date.

Syntax: TOTALQTD(<expression>, <dates_column>)
Example: TotalSalesQTD = Sales [OrderDate])

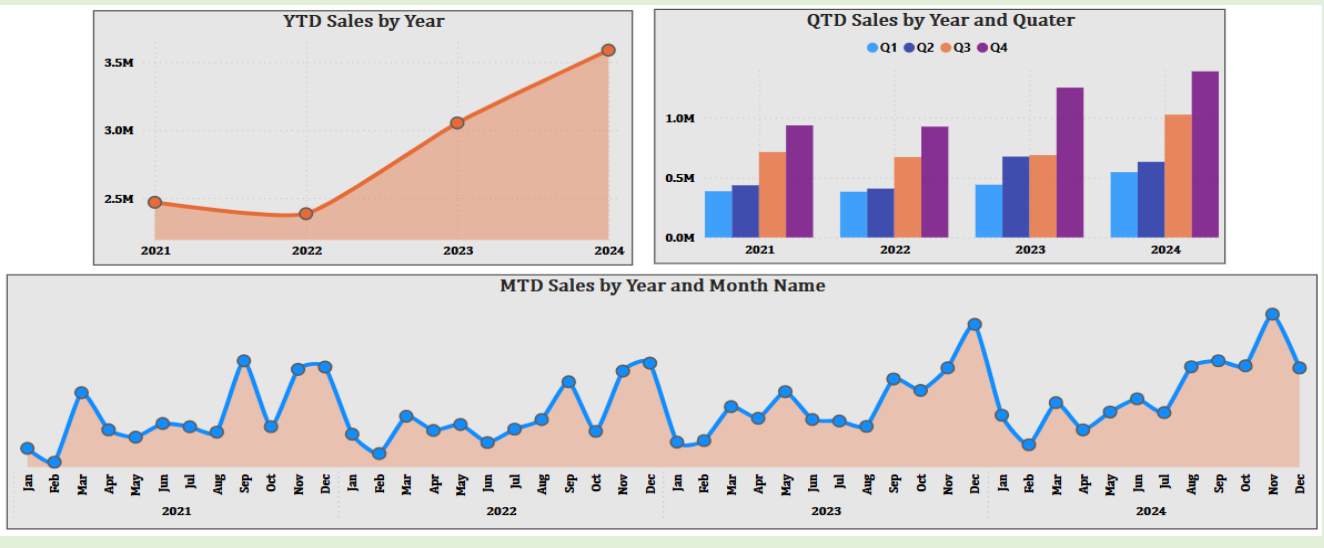
Explanation: This formula calculates the cumulative sales from the start of the current quarter to the given date.

3. TOTALMTD (Total Month-To-Date)

TOTALMTD calculates the cumulative total from the beginning of the month to a specified date.

Syntax: TOTALMTD <(<expression>, <dates_column>)
Example: TotalSalesMTD = Sales [OrderDate])

Year	Month Name	Total_Sales	YTD Sales	QTD Sales	MTD Sales
⊟ 2021	Jan	73,744.13	73,744.13	73,744.13	73,744.13
	Feb	19,603.21	93,347.33	93,347.33	19,603.21
	Mar	292,309.55	385,656.89	385,656.89	292,309.55
	Apr	146,423.00	532,079.88	146,423.00	146,423.00
	May	117,540.56	649,620.44	263,963.56	117,540.56
	Jun	171,186.52	820,806.96	435,150.08	171,186.52
	Jul	158,392.56	979,199.52	158,392.56	158,392.56
	Aug	137,450.20	1,116,649.72	295,842.76	137,450.20
	Sep	417,139.72	1,533,789.45	712,982.48	417,139.72
	Oct	159,024.73	1,692,814.18	159,024.73	159,024.73
	Nov	383,642.56	2,076,456.74	542,667.29	383,642.56
	Dec	392,665.97	2,469,122.71	935,333.26	392,665.97
	Total	2,469,122.71	2,469,122.71	935,333.26	392,665.97



1. How to create the measure to calculate the Previous Month Sales.
PREVIOUSMONTH () is useful when you want to shift exactly one calendar month back.

```
Prev_Month_Sales = CALCULATE([Total_Sales], PREVIOUSMONTH (Calender_Table [Date]))
```

2. How to define the function to calculate % change in sales from Previous month

```
MOM Sales % =
var _Sales = [Total_Sales]
var _Prev = [Prev_Month_Sales]
RETURN
DIVIDE (_Sales-_Prev, Prev, BLANK ())
```

Sales = Current Month Sales (Total Sales measure).
Prev = Sales from the previous month (from your previous measure).
The DIVIDE () function safely handles division by zero.

Year	Month Name	Total_Sales	Prev_Month_Sales	MOM Sales %
⊟ 2021	Jan	73,744.13		
	Feb	19,603.21	73,744.13	-73.42%
	Mar	292,309.55	19,603.21	1391.13%
	Apr	146,423.00	292,309.55	-49.91%
	May	117,540.56	146,423.00	-19.73%
	Jun	171,186.52	117,540.56	45.64%
	Jul	158,392.56	171,186.52	-7.47%
	Aug	137,450.20	158,392.56	-13.22%
	Sep	417,139.72	137,450.20	203.48%
	Oct	159,024.73	417,139.72	-61.88%
	Nov	383,642.56	159,024.73	141.25%
	Dec	392,665.97	383,642.56	2.35%
	Total	2,469,122.71		
⊞ 2022		2,385,508.23	392,665.97	507.52%
⊞ 2023		3,052,227.64	407,890.04	648.30%
⊞ 2024		3,587,222.01	560,027.02	540.54%
Total		11,494,080.59		

Year	Total_Sales	Prev_Year_sales	YOY Sales %
2021	2,469,122.71		
2022	2,385,508.23	2,469,122.71	-3.39%
2023	3,052,227.64	2,385,508.23	27.95%
2024	3,587,222.01	3,052,227.64	17.53%
Total	11,494,080.59	7,906,858.58	45.37%

ALL, ALLSELECTED, ALLEXCEPT as Filter Modifiers in CALCULATE

1 ALL: Removes all filters from a table or column.

```
All Sales = CALCULATE([Total_Sales], ALL ('Sales Data'))
```

2 ALLSELECTED: Removes filters applied outside the visual but keeps filters applied within the visual context.

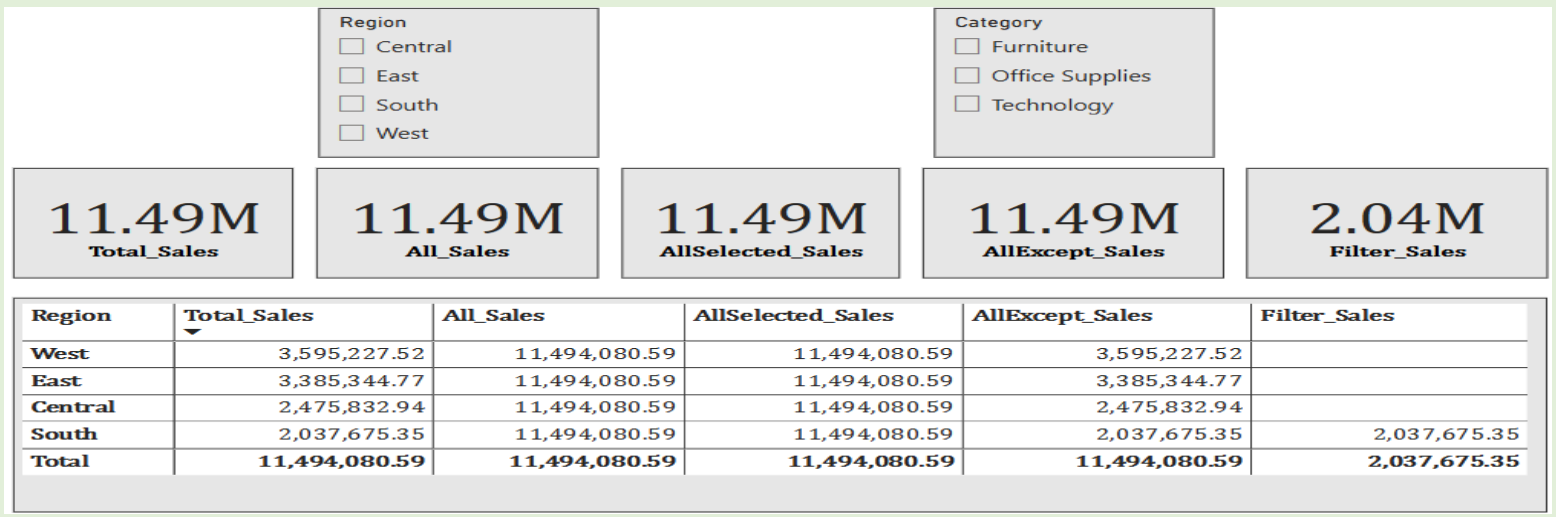
```
AllSelected_Sales = CALCULATE([Total_Sales], ALLSELECTED ('Sales Data'[Region]))
```

3 ALLEXCEPT: Removes all filters except those specified for certain columns.

```
AllExcept_Sales = CALCULATE([Total_Sales], ALLEXCEPT ('Sales Data','Sales Data'[Region]))
```

4 FILTER: Returns a table filtered by a given Boolean condition.

```
Filter Sales = CALCULATE([Total_Sales], FILTER ('Sales Data','Sales Data'[Region] = "South"))
```



Region	Total_Sales	All_Sales	AllSelected_Sales	AllExcept_Sales	Filter_Sales
West	3,595,227.52	11,494,080.59	11,494,080.59	3,595,227.52	
East	3,385,344.77	11,494,080.59	11,494,080.59	3,385,344.77	
Central	2,475,832.94	11,494,080.59	11,494,080.59	2,475,832.94	
South	2,037,675.35	11,494,080.59	11,494,080.59	2,037,675.35	2,037,675.35
Total	11,494,080.59	11,494,080.59	11,494,080.59	11,494,080.59	2,037,675.35