

Advanced DAX Functions

DAX SYNTAX

MEASURE NAME

- Measures are always surrounded by brackets (i.e. [Total Quantity]) when referenced in formulas, so spaces are OK

Total Quantity: = SUM(Transactions[quantity])

FUNCTION NAME

'product category'
productCategory - Camel Case
product_category - Snake_case

Referenced TABLE NAME

Referenced COLUMN NAME

COMMON FUNCTION CATEGORIES

MATH & STATS Functions	LOGICAL Functions	TEXT Functions	FILTER Functions	TABLE Functions	DATE & TIME Functions	RELATIONSHIP Functions
<p>Functions used for aggregation or iterative, row-level calculations</p> <p>Common Examples:</p> <ul style="list-style-type: none"> SUM AVERAGE MAX/MIN DIVIDE COUNT/COUNTA COUNTROWS DISTINCTCOUNT <p>Iterator Functions:</p> <ul style="list-style-type: none"> SUMX AVERAGEX MAXX/MINX RANKX COUNTX 	<p>Functions that use conditional expressions (IF/THEN statements)</p> <p>Common Examples:</p> <ul style="list-style-type: none"> IF IFERROR AND OR NOT SWITCH TRUE FALSE 	<p>Functions used to manipulate text strings or value formats</p> <p>Common Examples:</p> <ul style="list-style-type: none"> CONCATENATE COMBINEVALUES FORMAT LEFT/MID/RIGHT UPPER/LOWER LEN SEARCH/FIND REPLACE SUBSTITUTE TRIM 	<p>Functions used to manipulate table and filter contexts</p> <p>Common Examples:</p> <ul style="list-style-type: none"> CALCULATE FILTER ALL ALLEXCEPT ALLSELECTED KEEPFILTERS REMOVEFILTERS SELECTEDVALUE 	<p>Functions that create or manipulate tables and output tables vs. scalar values</p> <p>Common Examples:</p> <ul style="list-style-type: none"> SUMMARIZE ADDCOLUMNS GENERATESERIES DISTINCT VALUES UNION INTERSECT TOPN 	<p>Functions used to manipulate date & time values or handle time intelligence calculations</p> <p>Common Examples:</p> <ul style="list-style-type: none"> DATE DATEDIFF YEARFRAC YEAR/MONTH DAY/HOUR TODAY/NOW WEEKDAY WEEKNUM NETWORKDAYS <p>Time Intelligence:</p> <ul style="list-style-type: none"> DATESYTD DATESMTD DATEADD DATESBETWEEN 	<p>Functions used to manage & modify table relationships</p> <p>Common Examples:</p> <ul style="list-style-type: none"> RELATED RELATEDTABLE CROSSFILTER USERELATIONSHIP

$$\sim(T \& F) = \sim F = T$$

$$\begin{aligned} \text{SetA} &= [1, 2, 3, 4, 5] \\ \text{SetB} &= [3, 4, 5, 6, 7] \end{aligned}$$

$$\begin{aligned} \text{Union} &= [1, 2, 3, 4, 5, 6, 7] \\ \text{Intersection} &= [3, 4, 5] \end{aligned}$$

BASIC MATH & STATS FUNCTIONS

SUM

Evaluates the sum of a column

`=SUM(Column Name)`**AVERAGE**

Returns the average (arithmetic mean) of all the numbers in a column

`=AVERAGE(Column Name)`**MAX**

Returns the largest value in a column or between two scalar expressions

`=MAX(Column Name Or Scalar 1, [Scalar 2])`**MIN**

Returns the smallest value in a column or between two scalar expressions

`=MIN(Column Name Or Scalar 1, [Scalar 2])`**DIVIDE**

Performs division and returns the alternate result (or blank) if DIV/0

`=DIVIDE(Numerator, Denominator, [AlternateResult])`

COUNTING FUNCTIONS

COUNT

Counts the number of non-empty cells in a column(excluding Boolean values)

`=COUNT(Column Name)`**COUNTA**

Counts the number of non-empty cells in a column (including Boolean values)

`=COUNTA(Column Name)`**DISTINCT COUNT**

Counts the number of distinct values in a column

`=DISTINCTCOUNT(Column Name)`**COUNTROWS**

Counts the number of rows in the specified table, or a table defined by an expression

`=COUNTROWS([Table])`

BASIC LOGICAL FUNCTIONS

IF

Checks if a given condition is met and returns one value if the condition is TRUE, and another if the condition is FALSE

=IF(LogicalTest, ResultIfTrue, [ResultIfFalse])

IFERROR

Evaluates an expression and returns a specified value if it returns an error, otherwise returns the expression itself

=IFERROR(Value, ValueIfError)

SWITCH

Evaluates an expression against a list of values and returns one of multiple possible expressions

=SWITCH(Expression, Value1, Result1, ..., [Else])

AND

Checks whether both arguments are TRUE to return TRUE, otherwise returns FALSE

=AND(Logical1, Logical2)

OR

Checks whether any argument is TRUE to return TRUE, otherwise returns FALSE

=OR(Logical1, Logical2)

Note: Use the `&&` and `||` operators to include more than two conditions

TEXT FUNCTIONS

LEN

Returns the number of characters in a string

=LEN(Text)

CONCATENATE

Joins two text strings into one

=CONCATENATE(Text1, Text2)

UPPER / LOWER

Converts a string to upper or lower case

=UPPER/LOWER (Text)

LEFT/RIGHT/MID

Returns a number of characters from the start/middle/end of a text string

=LEFT/RIGHT(Text, [NumChars])
=MID(Text, StartPosition, NumChars)

SUBSTITUTE

Replaces an instance of existing text with new text in a string

=SUBSTITUTE(Text, OldText, NewText, [InstanceNumber])

SEARCH

Returns the position where a specified string or character is found, reading left to right

=SEARCH(FindText, WithinText, [StartPosition], [NotFoundValue])

BASIC DATE & TIME FUNCTIONS

TODAY/NOW

Returns the current date or exact time

=TODAY/NOW()

DAY/MONTH /YEAR

Returns the day of the month (1-31), month of the year (1-12), or year of a given date

=DAY/MONTH/YEAR(Date)

HOUR/MINUTE /SECOND

Returns the hour (0-23), minute (0-59), or second (0-59) of a given datetime value

=HOUR/MINUTE/SECOND(Datetime)

WEEKDAY/ WEEKNUM

Returns a weekday number from 1 (Sunday) to 7 (Saturday), or the week # of the year

=WEEKDAY/WEEKNUM(Date, [ReturnType])

EOMONTH

Returns the date of the last day of the month, +/- a specified number of months

=EOMONTH(StartDate, Months)

DATEDIFF

Returns the difference between two dates, based on a given interval (day, hour, year, etc.)

=DATEDIFF(Date1, Date2, Interval)

Total Orders Vs Quantity Sold

Invoice Number

Unique Invoice will be the correct Total orders.

ORDERNUMBER	QUANTITYORDERED
10203	20
10153	20
10104	34
10153	42
10212	39
10104	41
10246	46
10412	54
10203	47
10212	33
10212	29
10205	36
10244	40
10212	38

Total Orders =

```
DISTINCTCOUNT(
    'Vehicle Orders'[ORDERNUMBER])
```

307

Total Orders

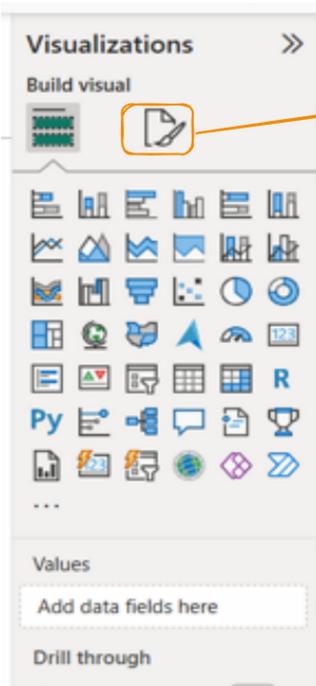
Quantity Sold =

```
SUM(
    'Vehicle Orders'[QUANTITYORDERED]
)
```

99K

Quantity Sold

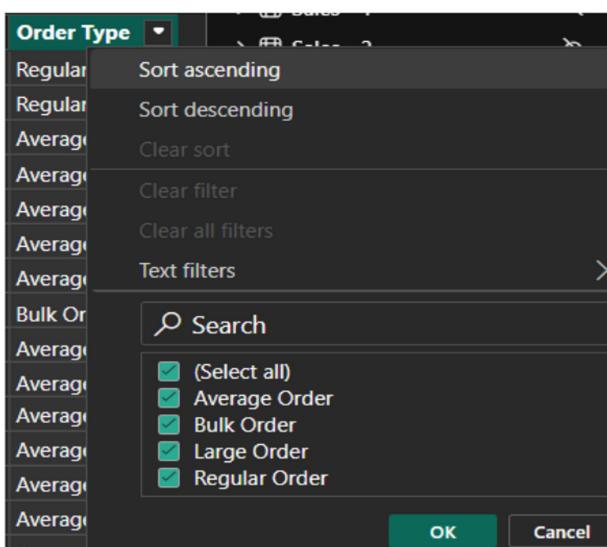
Table: Vehicle Orders (2,823 rows) Column: ORDERNUMBER (307 distinct values)



6 - 94 [Order Quantity]

0 - 25 [Regular Order]
26 - 50 [Average Order]
51 - 75 [Bulk Order]
> 75 [Large Order]

```
1 Order Type =  
2     IF('Vehicle Orders'[QUANTITYORDERED] <= 25 , "Regular Order",  
3         IF('Vehicle Orders'[QUANTITYORDERED] <=50 , "Average Order",  
4             IF('Vehicle Orders'[QUANTITYORDERED] <= 75 , "Bulk Order" , "Large Order"  
5 )))
```



SWITCH

- Replacing a nested IF, another being used to reduce the filter.

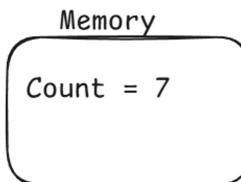
```
SWITCH(Expression, Value1, Result1,  
..., [Else])
```

Returns different results depending on the value of an expression.

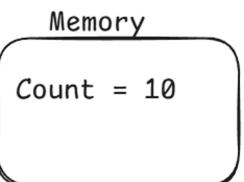
```
Order Type (Switch) =  
SWITCH(  
    TRUE(),  
    'Vehicle Orders'[QUANTITYORDERED] <=25, "Regular Order",  
    'Vehicle Orders'[QUANTITYORDERED] <=50, "Average Order",  
    'Vehicle Orders'[QUANTITYORDERED] <=75, "Bulk Order",  
    "Large Order")
```

```
count = 0;  
while(qty < 50):  
    count++  
  
print("result - " + count)  
  
7 ["Thala for a reason"]
```

QUANTITYORDERED
20
20
34
42
39
41
46
54
47
33
29



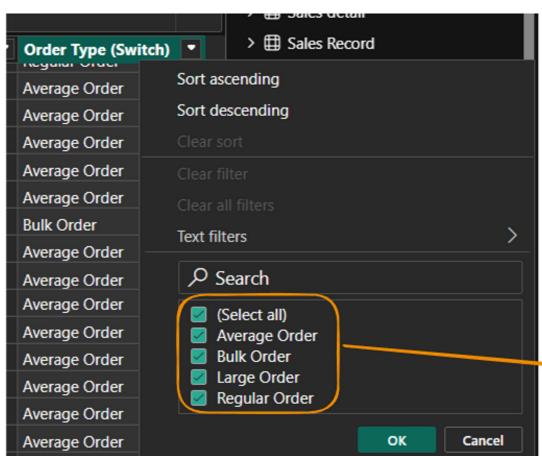
```
count = 0;  
while(TRUE()):  
    if(qty < 50):  
        count++
```



```
print("result - " + count)
```

When will it print then? ↴

When all the records are iterated.



With the help of Switch Statement , I'll make the filter count = 3.

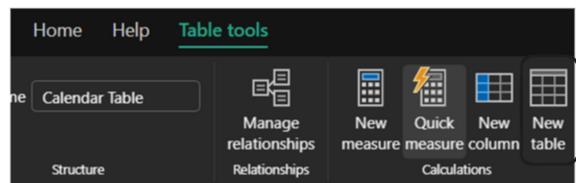
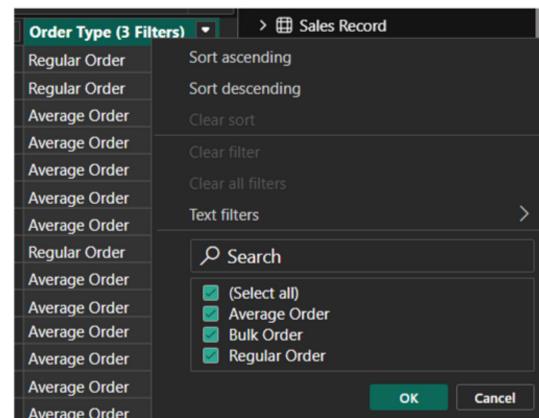
0 - 25 - Regular Order

Average Order / Bulk Order - Average Order

Large Order - Bulk Order

A+ve , A+, A positive, A +ive , --> A+

```
Order Type (3 Filters) =
SWITCH(
    'Vehicle Orders'[Order Type (Switch)],
    "Average Order" , "Average Order",
    "Regular Order" , "Regular Order",
    "Bulk Order" , "Regular Order",
    "Large Order" , "Bulk Order")
```



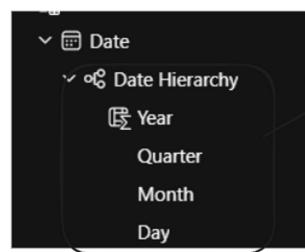
```
1 Calendar Table =
2 CALENDAR(
3 |   "01/01/2024" , TODAY()
4 )
```

Date
01-01-2024 00:00:00
02-01-2024 00:00:00
03-01-2024 00:00:00
04-01-2024 00:00:00
05-01-2024 00:00:00
06-01-2024 00:00:00
07-01-2024 00:00:00
08-01-2024 00:00:00
09-01-2024 00:00:00
10-01-2024 00:00:00
11-01-2024 00:00:00
12-01-2024 00:00:00

"Helping me to create a rolling calendar"

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

Date	Year
01 January 2024	2024
02 January 2024	2024
03 January 2024	2024



"drill down" approach

```
1 Month = MONTH('Calendar Table'[Date])
```

Date	Year	Month
01 January 2024	2024	1
02 January 2024	2024	1

```
1 Day = DAY('Calendar Table'[Date])
```

Date	Year	Month	Day
------	------	-------	-----

```
1 MonthName = FORMAT(['Calendar Table'[Date],"MMMM")
```

Year	Month	Day	MonthName
2024	1	1	January
2024	1	2	January

```
ShortMonthName = FORMAT('Calendar Table'[Date],"MMM")
```

Year Month Day MonthName ShortMonthName DayName

	Year	Month	Day	MonthName	ShortMonthName	DayName
24	2024	1	1	January	Jan	Monday
24	2024	1	2	January	Jan	Tuesday
24	2024	1	3	January	Jan	Wednesday

	Year	Month	Day	MonthName	ShortMonthName	DayName	Weekday
	2024	1	1	January	Jan	Monday	2
	2024	1	2	January	Jan	Tuesday	3
	2024	1	3	January	Jan	Wednesday	4
	2024	1	4	January	Jan	Thursday	5
	2024	1	5	January	Jan	Friday	6
	2024	1	6	January	Jan	Saturday	7
	2024	1	7	January	Jan	Sunday	1
	2024	1	8	January	Jan	Monday	2
	2024	1	9	January	Jan	Tuesday	3

Return Type - 1
[Sun - 1 till Sat - 7]

	Year	Month	Day	MonthName	ShortMonthName	DayName	Weekday
	2024	1	1	January	Jan	Monday	1
	2024	1	2	January	Jan	Tuesday	2
	2024	1	3	January	Jan	Wednesday	3
	2024	1	4	January	Jan	Thursday	4
	2024	1	5	January	Jan	Friday	5
	2024	1	6	January	Jan	Saturday	6
	2024	1	7	January	Jan	Sunday	7
	2024	1	8	January	Jan	Monday	1
	2024	1	9	January	Jan	Tuesday	2

Return Type - 2
[Mon- 1 till Sun - 7]

	Year	Month	Day	MonthName	ShortMonthName	DayName	Weekday
	2024	1	1	January	Jan	Monday	0
	2024	1	2	January	Jan	Tuesday	1
	2024	1	3	January	Jan	Wednesday	2
	2024	1	4	January	Jan	Thursday	3
	2024	1	5	January	Jan	Friday	4
	2024	1	6	January	Jan	Saturday	5
	2024	1	7	January	Jan	Sunday	6
	2024	1	8	January	Jan	Monday	0
	2024	1	9	January	Jan	Tuesday	1

Return Type - 3
[Mon- 0 till Sun - 6]

	Year	Month	Day	MonthName	ShortMonthName	DayName	Weekday	IsWeekend?
	2024	1	1	January	Jan	Monday	1	Weekday
	2024	1	2	January	Jan	Tuesday	2	Weekday
	2024	1	3	January	Jan	Wednesday	3	Weekday
	2024	1	4	January	Jan	Thursday	4	Weekday
	2024	1	5	January	Jan	Friday	5	Weekday
	2024	1	6	January	Jan	Saturday	6	Weekend
	2024	1	7	January	Jan	Sunday	7	Weekend
	2024	1	8	January	Jan	Monday	1	Weekday
	2024	1	9	January	Jan	Tuesday	2	Weekday