Advanced Transformation DAX - 5

18 September 2024 21:07

Advanced DAX Transformation - 5

Perform the below mentioned operations in Power BI:

- Create a sub-table of customers table only showing customers which met the conditions in the expression.
- Calculate the revenue contributed by each customer.
- Calculate the percentage of orders of each country.
- Calculate the previous month's revenue corresponding to each month of a year.
- Set the background color of the canvas, and the image on the canvas page.
- Download a them and use it, attach screenshot of that them.
- Show year-wise revenue using a column chart

Calculate - (measure , filter);
Calculate - (measure , filter[Remove] - ALL)

ALL - Removing a filter.

BASIC MATH & STATS FUNCTIONS

SUM	Evaluates the sum of a column	=SUM(ColumnName)
AVERAGE	Returns the average (arithmetic mean) of all the numbers in a column	=AVERAGE(ColumnName)
MAX	Returns the largest value in a column or between two scalar expressions	=MAX(ColumnNameOrScalar1 , [Scalar2])
MIN	Returns the smallest value in a column or between two scalar expressions	=MIN(ColumnNameOrScalar1 , [Scalar2])
DIVIDE	Performs division and returns the alternate result (or blank) if DIV/0	=DIVIDE(Numerator, Denominator, [AlternateResult])

COUNTING FUNCTIONS

COUNTING FUNCTIONS

COUNT	Counts the number of non-empty cells in a column(excluding Boolean values)	=COUNT(ColumnName)
COUNTA	Counts the number of non-empty cells in a column (including Boolean values)	=COUNTA(ColumnName)
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

SWITCH

SWITCH

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

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

Any DAX expression that returns a single scalar value, evaluated multiples times.
Examples:
- Calendar[Month ID]
- 'Product Lookup'[category]

List of values produced by the expression, each paired with a result to return for rows/cases that match.
Examples:
=SWITCH(Calendar[Month ID],
1, "January",
2, "February"

Value returned if the expression doesn't match any value argument

PRO TIP
SWITCH(TRUE) is a common DAX pattern to replace multiple nested IF statements

```
Month Number (DAX) =
    IF(
        'Calendar Lookup'[Month Name] = "January" , "1",
        IF(
            'Calendar Lookup'[Month Name] = "February" , "2",
            IF(
                'Calendar Lookup'[Month Name] = "March" , "3",
                IF('Calendar Lookup'[Month Name] = "April" , "4", "Other"
                ) )))
```

Nested If , Can easily be handle with Switch Statement.

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	=SUBSTITUTE(Text, OldText,

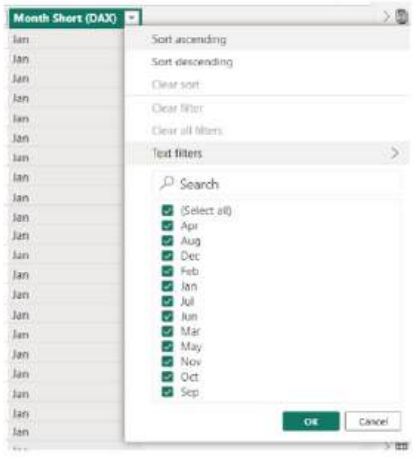
SUBSTITUTE — Replaces an instance of existing text with new text in a string

=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])

"Geekster" → "-1"




```
Month Short (DAX) =
    LEFT(
        'Calendar Lookup'[Month Name] ,
        3)
```

```
Customer Full Name =
    'Customer Lookup'[Prefix] & " " &'Customer Lookup'[FirstName] & " " & 'Customer Lookup'[LastName]
```

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

=EOMONTH(StartDate, Months)

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)

```
Weekend =
    IF(
        'Calendar Lookup'[Week Of Day] IN {6,7},
        "Weekend",
        "Weekday")
```

```
Week Of Day =
    WEEKDAY(
        'Calendar Lookup'[Date],
        2)
```

RELATED

RELATED() :-
Returns related values in each row of a table based on relationships with other tables

=RELATED(ColumnName)

The column from a related table containing the values you want to retrieve
Examples:
- 'Product Lookup'[Product Name]
- 'Territory Lookup'[Country]

HEY THIS IS IMPORTANT!

- RELATED works like a VLOOKUP function in Excel – it uses the relationship between tables (defined by primary and foreign keys) to pull values from one table into a new column of another.

- Since this function requires row context, it can only be used as a calculated column or as part of an iterator function that cycles through all rows in a table (FILTER, SUMX, MAXX, etc.)

PRO TIP:
Instead of using RELATED to create extra columns (which increases file size),

PRO TIP:
Instead of using RELATED to create extra columns (which increases file size), nest it within measures like FILTER or SUMX

```
Retail Price =
    RELATED(
        'Product Lookup'[ProductPrice])
```

```
Revnue =
    'Sales Data'[Retail Price] * 'Sales Data'[OrderQuantity]
```

CALCULATE

CALCULATE()
Evaluates an expression in a context that is modified by filters

=CALCULATE(Expression, [Filter1], [Filter2],...)

Name of an existing measure or a DAX formula for a valid measure
Examples:
- [Total Orders]
- SUM('Returns Data'[Return Quantity])

A Boolean (True/False) expression or a table expression that defines a filter.
Note: these require fixed values or aggregation functions that return a scalar value (you cannot create filters based on measures)
Examples:
- 'Territory Lookup'[Country] = "USA"
- Calendar[Year] <> MAX(Calendar[Year])

PRO TIP:
Think of CALCULATE as a filter modifier; it allows you to overrule existing report filters and "force" new filter context

EXAMPLE: CALCULATE

```
1 Red Sales = CALCULATE( [Quantity Sold], 'Product Lookup'[Product Color] = "Red" )
```

- Here we've defined a new measure named Red Sales,

• Here we've defined a new measure named Red Sales, which evaluates the Quantity Sold measure under a filter context where the product color is "Red"

Product Color	Quantity Sold	Red Sales
Black	10,590	4,011
Multi	5,756	4,011
Red	4,011	4,011
Silver	3,257	4,011
Total	**23,614**	**4,011**

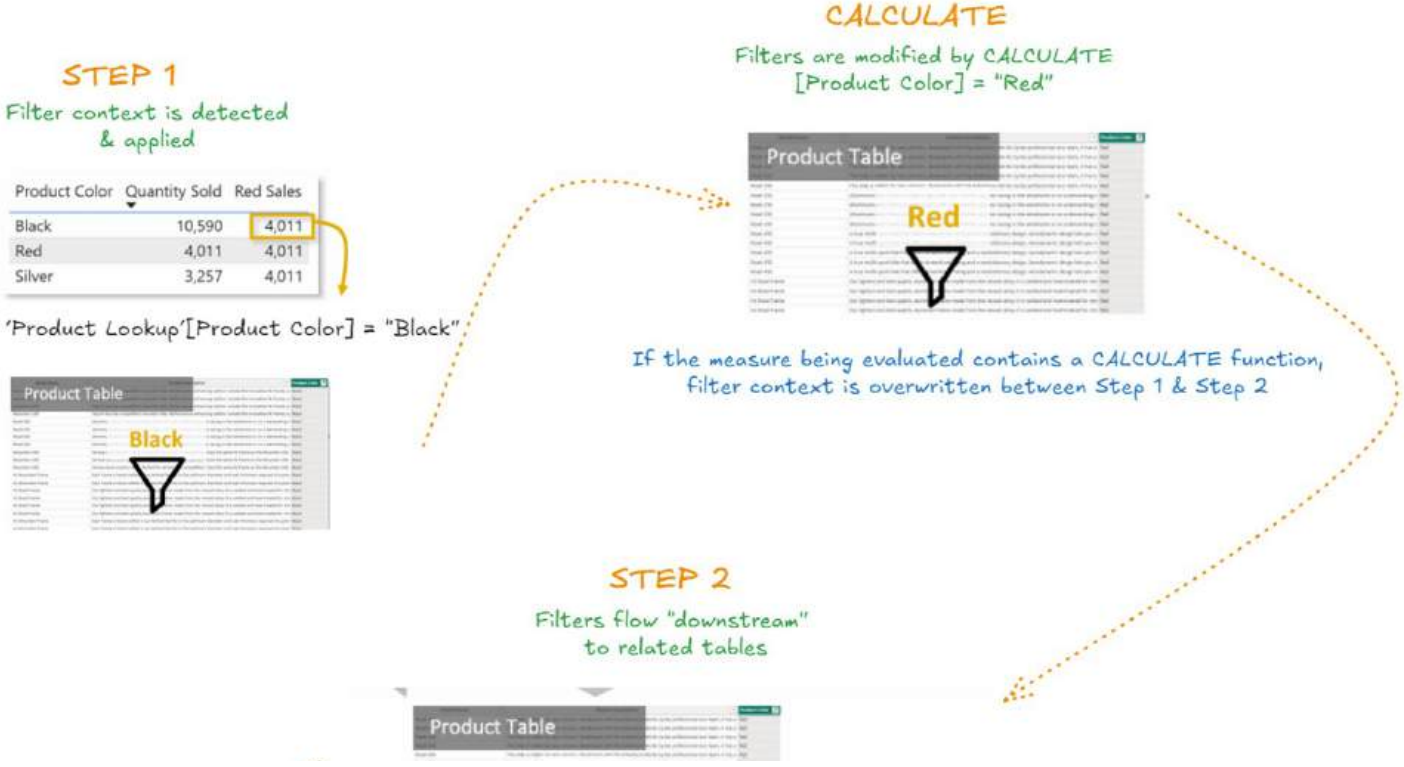
Note how we see the the same repeated values for each product color, and even the total!

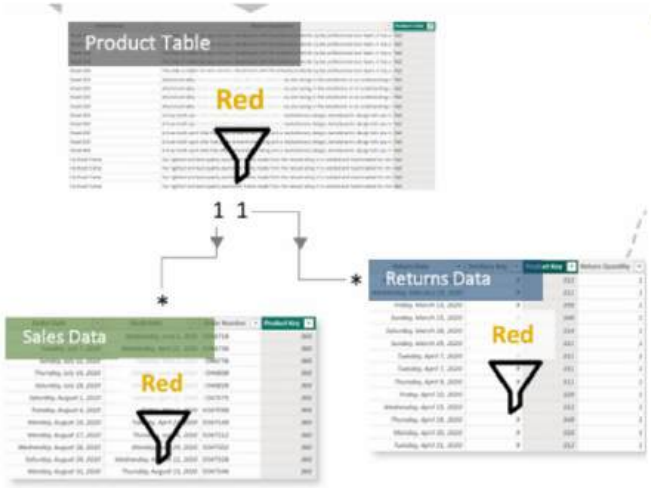
HEY THIS IS IMPORTANT!

• The CALCULATE function modifies and overrules any competing filter context!

• In this matrix, the "Black" row has competing filter context: Product Color = Black (from the row label) and Product Color= "Red" (from the CALCULATE function)

• Both can't be true at the same time, so the "Red" filter from CALCULATE takes priority





STEP 3

Measure evaluates against the filtered table

```
1 Quantity Sold =
2 SUM( 'Sales Data'[Order Quantity] )
```

- Sum of the Order Quantity column in the Sales Data table, filtered to rows where the product color is "Red"

= 4,011

```
1 Bulk Orders =
2     CALCULATE(
3         [Total Orders],
4         'Sales Data'[OrderQuantity] > 1
5 )
```

Filter

```
Weekend Orders =
    CALCULATE(
        [Total Orders],
        'Calendar Lookup'[Weekend] = "Weekend")
```

ALL

ALL() :-
Returns all rows in a table, or all values in a column, ignoring any filters that have been applied.

ALL() :-
Returns all rows in a table, or all values in a column, ignoring any filters that have been applied.

=ALL(Table or Column, [Column2], [Column3],..)

The table or column that you want to clear filters on
Examples:
- Transactions
- Products[Category]

Additional columns that you want to clear filters on (optional)
- Cannot specify columns if your first parameter is a table
- All columns must include the table name and come from the same table

Examples:
- 'Customer Lookup'[City], 'Customer Lookup'[Country]
- Products[Product Name]

PRO TIP:
Instead of adding filter context, the ALL function removes it. This is often used in "% of Total" calculations, when the denominator needs to remain fixed regardless of filter context.

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

```
% of All Orders =
    DIVIDE(
        [Total Orders],
        [All Orders])
```

```
Overall Average Price =
    CALCULATE(
        [Average Retail Price],
        ALL(
            'Product Lookup'))
```

CategoryName	Weekend Orders	Total Orders	All Orders	% of All Orders	Average Retail Price	Overall Average Price
		1	25,165	0.00%		$714.44
Accessories	4,913	16,983	25,165	67.49%	$34.26	$714.44
Bikes	3,995	13,929	25,165	55.35%	$1,541.38	$714.44
Clothing	1,962	6,976	25,165	27.72%	$50.68	$714.44
Components			25,165		$432.19	$714.44
Total	**7,214**	**25,165**	**25,165**	**100.00%**	**$714.44**	**$714.44**

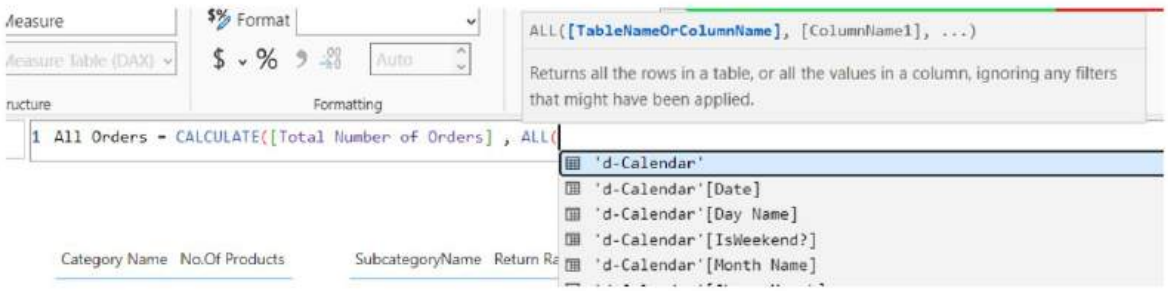
- Calculate - measure , ALL(remove filter)
- ALL Syntax - Individual Usage as well.

ALL Syntax - Individual Usage as well.

```
1 all orders = CALCULATE([total no:of orders],ALL('f-sales'))
```

Back to report

Country	total no:of orders	all orders	% of all orders
Australia	6060	25164	24.08%
Canada	3024	25164	12.02%
France	2315	25164	9.20%
Germany	2294	25164	9.12%
United Kingdom	2771	25164	11.01%
United States	8700	25164	34.57%
Total	**25164**	**25164**	**100.00%**




```
1 All Orders = CALCULATE([Total Number of Orders] , ALL('F-Sales'))
```

12954

All Orders

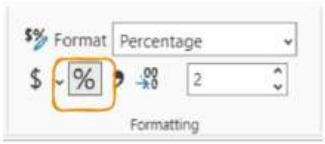
Country	Total Number of Orders	All Orders	% of All Orders
Australia	3120	12954	24.09%
Canada	1864	12954	14.39%
France	1101	12954	8.50%
Germany	1116	12954	8.62%
United Kingdom	1381	12954	10.66%
United States	4372	12954	33.75%
Total	**12954**	**12954**	**100.00%**

```
1 % of All Orders = DIVIDE([Total Number of Orders], [All Orders] , 0)
```

12954

All Orders

Country	Total Number of Orders	All Orders	% of All Orders
Australia	3120	12954	24.09%
Canada	1864	12954	14.39%
France	1101	12954	8.50%
Germany	1116	12954	8.62%
United Kingdom	1381	12954	10.66%
United States	4372	12954	33.75%
Total	**12954**	**12954**	**100.00%**



SQL - Window Function - 1. Lead / Lag

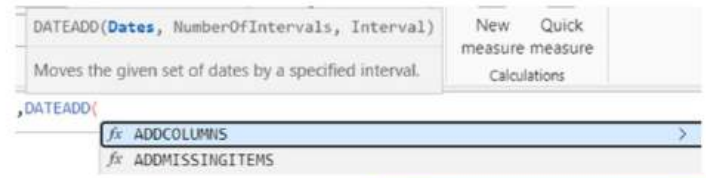
- Calculate the previous month's revenue corresponding to each month of a year.
- Set the background color of the canvas, and the image on the canvas page.
- Download a them and use it, attach screenshot of that them.
- Show year-wise revenue using a column chart

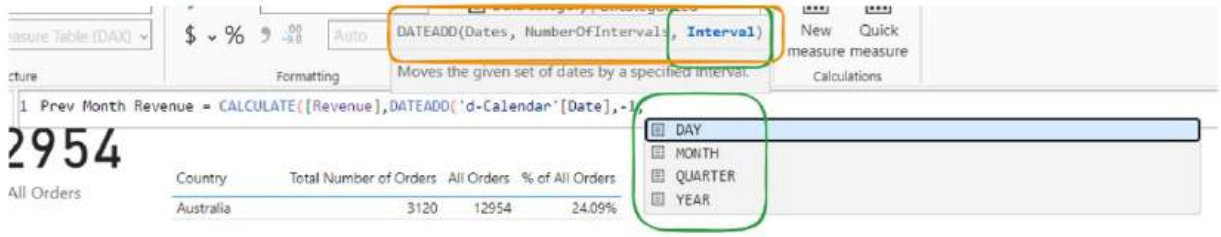
```
1 prev month revenue = CALCULATE([revenue],DATEADD('d-calendar'[Date],-1,MONTH))
```

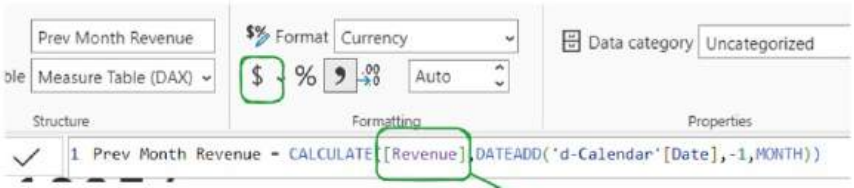
Back to report

Start of Month	revenue	previous month revenue	prev month revenue
Thursday, January 01, 2015	585,277.00		
Sunday, February 01, 2015	532,190.00	585,277.00	585,277.00

Start of Month	revenue	previous month revenue	prev month revenue
Thursday, January 01, 2015	585,277.00		
Sunday, February 01, 2015	532,190.00	585,277.00	585,277.00
Sunday, March 01, 2015	643,397.00	532,190.00	532,190.00
Wednesday, April 01, 2015	653,324.00	643,397.00	643,397.00
Friday, May 01, 2015	659,287.00	653,324.00	653,324.00
Monday, June 01, 2015	669,948.00	659,287.00	659,287.00
Wednesday, July 01, 2015	486,113.00	669,948.00	669,948.00
Saturday, August 01, 2015	536,447.00	486,113.00	486,113.00
Tuesday, September 01, 2015	344,053.00	536,447.00	536,447.00
Thursday, October 01, 2015	404,261.00	344,053.00	344,053.00
Sunday, November 01, 2015	326,595.00	404,261.00	404,261.00
Tuesday, December 01, 2015	563,737.00	326,595.00	326,595.00

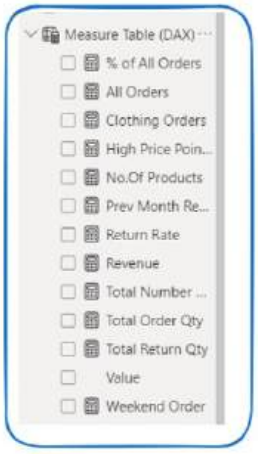






r	Quarter	Month	Day	Sum of Revenue	Prev Month Revenue
15	Qtr 1	January	1	$2,81,315.9	
15	Qtr 1	February	1	$2,32,492.7	$2,81,315.9
15	Qtr 1	March	1	$4,30,181.3	$2,32,492.7
15	Qtr 2	April	1	$4,13,281.4	$4,30,181.3
15	Qtr 2	May	1	$3,95,593.2	$4,13,281.4
15	Qtr 2	June	1	$4,41,041.3	$3,95,593.2
15	Qtr 3	July	1	$2,71,593.9	$4,41,041.3
15	Qtr 3	August	1	$2,99,520.4	$2,71,593.9
15	Qtr 3	September	1	$1,72,243.4	$2,99,520.4
15	Qtr 4	October	1	$2,15,215.7	$1,72,243.4
15	Qtr 4	November	1	$1,52,688.6	$2,15,215.7
15	Qtr 4	December	1	$2,92,106.4	$1,52,688.6
al				$1,31,53,587.3	$1,22,83,513

Total Orders ,
Order Quantity ,
Return Quantity.



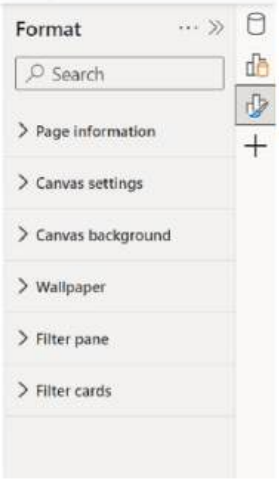
1st oct - 1st Nov - Revenue
2nd oct - 2nd Nov - Revenue

→ if you drop date , and find revenue on monthly basis. It means you are adding 1 day and removing another day

1st oct - 1st Nov - Revenue
2nd oct - 2nd Nov - Revenue
3rd Oct - 3rd Nov - Revenue.

→ If you drop date, and find revenue on monthly basis. It means you are adding 1 day and removing another day to find the revenue So it keeps on changing the value.

Report Making →



After applying background to the canvas.
you have to insert adventure logo in it.
Insert > Images > Choose the images from your local System.