DAX - Data Analysis Expression

3 Operations

1. New Measure
2. New Column
3. New table

Functions

1. Sum
2. Sumx
3. Min
4. Minx
5. Max
6. Mina
7. Maxa
8. Maxx
9. Count
   1. It will count the count of records in a Column for this the return type as a Scalar value
   2. The Count function Counts rows for below data type Columns. a) Numbers b) Dates c) Strings
10. COUNTA
    1. The CountA fn Counts the number of cells in a Column for this also return type as Scalar value
    2. This is Similar to Count function but is Support the rows in a table

       Syntax: Countrows (tablename)

12. Countax
13. Distinct Count
    1. Counts the number of distint values in a column
    Syntax : DISTINCTCOUNT(columnname)
14. Count blank.
    Counts the number of blanks in a column
    Syntax : COUNTBLANK (column name)
15. Calculate
16. Calculate table
17. AND
18. OR
19. NOT
20. All
21. All Except
22. All Selected
23. Related
24. Related Table
25. len
26. left
27. Right
28. lower
29. UPPer
30. If
31. Selected value
32. Filter
33. Calender
34. Date
35. Date Diff

COUNT

3. When the fn finds no rows to count, it returns a blank.
4. Blank values are skipped
5. TRUE/FALSE values are not supported.

36. Date values
37. Use Relationship
38. DAY
39. Edate
40. EOMonth
41. Hour
42. Minute
43. Now
44. Month
45 Quarter
46. Second
47. Time
48. Time value
49. Today
50. Utc Now
51 Week Day
52. Week Num
53. Year
54. Dates YTD
55 Dates QTD
56. Dates MTD
57. Total YTD
58. Total QTD
59. Total MTD
60. Dates in Period
61. SamePeriod last Year
62. Previous Day
63. Previous Month
64. Previous Quarter
65 Previous Year
66 Next Day
67 Next Month
68 Next Quarter
69 Next Year
70 Blank
71 Usu name
72 Usu Principalname

Dax : data Analysis expression

We use on Several Places functional Complete

1) SSAS
2) Powa BI
3) Excel
4) Azure Analysis
5) Sunvice

Coming to Powabi we Perform 3 Operations

1) New Measure
2) New Column
3) New Table

NEW MEASURE	NEW COLUMN
1) When we want output as a Scalar value then we will go to New Measure.	1) When we want to Perform Row by Row Operations then we will go to new Column
2) If we create new Measure will not occupy Space anywhere	2) If we Create new Column will occupy the Memory on application
3) If we did any Calculation on new Measure it will do that Calculation on fly	3) If we do any kind of Calculation on new Column while Reloading the application it will do that Calculation
4) Better to create New Measure in application.	4) If we create More Column it will degrade the Performance

New Table

- When we want output as a row's & column's information then we will go to new table.
- This new table also stored at application level.
- This is also while refreshing the application it will do that calculation.
- We should minimize column's & Tables to Increase Performance

AGGREGATION FUNCTIONS.

1) SUM
2) SUMX
3) MIN, MINA, MINX.
4) MAX, MAXA, MAXX.
5) AVG, AVGA, AVGX
6) COUNT, COUNTROWS, COUNTX, COUNTA, COUNTAX, COUNTBLANK(), DISTINCT COUNT().

1) What is the Difference b/w SUM & SUMX.

When we want to Perform SUM Operation on Particular Column then we will go to SUM function.

Syntax: SUM(Column name)

SUMX

This SUMX is a Iteration function

When we want to Perform SUM Operation on top of expression then we will go to SUMX function.

Sytax: SUMX(table name, Expression) .... SUMX(Table, Sales x Qty)

Note:

X - Means Expression on top of expression to do that Particular functionality we will go with X function

MIN

When we want to fetch Min value on top of column then we will go to Min function.

Syt : Min(columnname) ... Min(Sales)

This Min function on top of 2 Scalar values also we can find the Min value.

Syt : Min(Scalar1, Scalar2) .... Min(10,20) ... >10

MINA

This MinA is Similar to Min function but it will allow Bulian data types

This MinA function will not allows Scalar values

Syntax : MINA(columnname)

Note:

Whereever we found 'A' A will allows Bulian values (Data type), Bulian data type value Means (True, False)

True Means the value is 1.

False Means the value is 0.

MINX.

On top of expression to finding the Minimum value we will go with Min function

Syntax : Minx(tablename, expression)

Ex : Minx(tablename, ~~Bad~~ unitPrice x Quantity)

MAX

This Max function is Similar to Min function there we will get Min value here we will get Maximum values.

[Implicit filter & Explicit filter] (G)

Categorization of Power BI

[Connectivity Data Modelling]

One file from Excel (Import), SQL (Direct), Extract

[CONTEXT & ROWTEXT FILTERS]

SUMX → ROWTEXT FILTERS

Score Board → Card

COUNT

It will count the Count of records in a Column for this the return type as a Scalar value

Syntax : Count (Column Name)

COUNTA

1. The COUNTA function counts the number of cells in a column. for this also return type as Scalar value.
2. This is Similar to Count function but is Support Boolean datatype.

COUNTROWS.... Count(*)

1. This Countrows function will Count the rows in a table

Syntax : Countrows (Tablename)

DISTINCT COUNT

Counts the number of distinct counts in a column

Syntax : DISTINCT COUNT (<column>)

COUNT BLANK

Counts the number of blanks in a column

Syntax : COUNTBLANK (<column name>)

CALCULATE

On top of expression to filter the data then we will go with Calculate function.

To find the country wise Sales

En visualization

Country	Sales	Measure(F)	Measure(F)2	Measure (3)
A-C	1	2	-	C-G & Sg-G
B-F	2	2	2	
C-G	3	2	-	
D-M	4	2	-	
E-USA	5	2	-	

Measure(F) = Calculate (Sum (financials (Sales)),
financials [country] = "France")

(or)

Measure(F)2 = Calculate (Sum (financials (Sales)),
FILTER (financials, financials [country] = "France"))

AND

When we write Multiple Conditions. to club Multiple Conditions. [double Ambresant &&]

```
Measure (3) = Calculate (Sum (financials[Sales]),
                financial[country] = "Germany" &&
                financial[Segment] = "Government")
```

OR

To get either one or two in a Particular Column Indicates || (double Pipe Symbol)

```
Measure(4) = Calculate (Sum(financials(Sales)),
                financials(country) = "Germany" ||
                financials(country) = "France")
```

(In double coat it behaves a & Case Sensitive)

All

All function will Show Complete Information. all will have 3 Categorization

1) All (Column name)
2) All ()
3) All (Table name)

All X All Except (All is Completely opposite of All Except)

En visualization Table.

Country	Sales	(All) Measure(1)
C	1	15
F	2	15
G	3	15
M	4	15
USA	5	15
Total	15	15

All

- Measure(1) : Calculate (Sum (financials (Sales)), all (financials))

  Note : If we apply slicer on country wise it wont apply for Measure(1)

  It will remove all the filters on a specific table which you have been took.

- Measure(2) : Calculate (Sum (financial (Sales)), all ()

  Note : It won't affect a entire table or data model

3) Measure (3) : Calculate (Sum (financials (Sales)), all (financals (Country))

Note : If I Select country wise, it wont change that information, which I Select Segment wise it will change. it means it removes the filter for a Selected Column

All Except

Measure 5 : Calculate (Sum (financials Sales)),
All Except (financials, financials[country]))

"Completely opposite of All"

Note :, In Slicer If I Select any Country it will change accordingly because which I have taken in Syntax. it work as accordingly.

- Based on the Selection will work on Column level

All SELECTED.

Selection Based Implementation

Measure 6 = Calculate (Sum (financials (sales)),
All Selected (financials))

In All & All Selected will work both on table level & column level

If we apply any Slicer it will change Accordingly based on the Selection as normal way

"Work as Normal Functionality"

Mainly if we take Date Slicer "Date's between"
To Select Selected Range value.  ← Date Slicer

→ Table = All Selected (financial (Date))
or
→ Measure 7 = Calculate (sum (financials (Date)), All Selected (financial (Date)))

All () .... It won't affect conflict datamodel

All (tablename) ..... Except that table for remaining table columns it will change the data.

All (columnname) .... Except selected column, it will change for other columns

Pure opposite to this is All (Except)

All & All except is one category

All Selected

It is used mainly for selected range

By removing all filters, for selected range values

Don't club All Selected into All & All except it is other category.

3 Related

When we want to take some extra column information from another table to current table, then we will use with Related table. [We can use Costum column or calculated column or Merge joints]

RELATED	RELATED TABLE
1) Returns a single value from another table that is related to the current row	1) Returns a table with all the rows related to the current row
2) Related < (column Name) >	2) Related Table (< Table Name >)
3) Requires Relationship b/w 2 tables, 1 to 1, or Many to 1	3) Requires Relationship b/w 2 tables 1-1 or M-1 or 1-M
4) Related (Product Table [Product Name])	4) SUMX [Related Table (Store Stock Table) Store Stock Table (Stock)]

DAX

CustomerName = Related [(CustomerID)]

↳ New column

Orders (Table 1)

ID	CustomerID	Product	Qty	Price	Sales	CustomerName
1	1	A	6	2	12	James
2	1	B	1	3	3	James
3	1	C	2	4	8	James
4	2	D	1	5	5	Jony
5	2	E	3	7	21	Jony
6	3	F	10	9	90	Rocky

(1) (*)

Sales (Table 2)

Sales	~~Tech~~ ID	Total Sales	Remarks.
James	1	23	DAX
Jony	2	26	Total Sales = SUMX [Related Table §, (orders), Orders (Sales)]
Rocky	3	90	

- In some cases we can use Merge join types
- Or Custom column condition

Length

When we want to find the length of the String then we will go with len fn.

Take a Table visual.

NAME	(LEN) Column	R Column2	L Column3	Column4	Column5
SRI	3	SRI	RI	Sri	SRI
LOKI	4	LOK	KI	loki	LOKI
MAHI	4	MAH	HI	Mahi	MAHI
TEJ	3	TEJ	EJ	tej	TEJ
RAJ	3	RAJ	AJ	raj	RAJ
SHIVA	5	SHI	VA	Shva	SHIVA
VINOD	5	VIN	OD	vinod	VINOD
BIESY	5	BIE	SY	blesy	BLESY

Column = LEN (emp(NAME))

RIGHT

R Column2 = RIGHT (employee[ENAME], 3)

L column3 = LEFT (employee [ENAME), 2)

LOWER

When we want Convert String as lower then we will go with Lower. column4 = LOWER[employee (ename))

UPPER

When we Want Convert String as UPPER then we will go with UPPER.

Column5 = UPPER [employee (ename)]

SELECTED VALUE

To change the value or text as per Slicer in the Card.

Slicer] Product

```
1 Measure 1 =
2 Var Product = SELECTEDVALUE ( financals(Product)]
3 Return
4 Product
```

To change based on Sales & Product

```
1 Measure 1 =
2 Var Product = SELECTED VALUE ( financals (Product)]
3 Var Selected_Sum = CALCULATE(SUM (financials (sales)),
                    financials [Product] = Product
4 return
5 Selected_Sum.
```

In visualization

Product Slicer

Table

Country	Sales	Measure 1	Measure 1

> To change country wise title Dynamically

Slicer country | Column chart

Country vs Sales

This will take it as only one value or 1 return

Measure

1. Dynamic title = "Sales by country as" &
SELECTEDVALUE (financials[country]).

Show this in a card u can see changes

& go to format click on Title [fx]

Format by

Field value

Based on field

Dynamic title

FILTER.

> When we want to filter the data then we will go with these filters fn.

> Basically for these filter the Return type is a table.

> This we can use inside the Calculation function or without Calculate function.

> for outside or Endside Calculate fn can use.

> for Enside the ~~Unit~~ Return type is Scalar

> for Outside the Return type is table

for ex: Table = FILTER (employee, employee(DEPTNO) = 10)

or for Scalar

Measure = Calculate (Sum (employee (sal)),
FILTER (employee, employee [DEPTNO] = 10))

USE RELATIONSHIP.

Whenever it is having Inactive Relationship b/w the 2 tables. If we want use that Inactive Relationship then we will go with use Relationship.

CALENDER.

To Creating date table, the use of this Calender is to Creating date table.

Table = Calender (Min (financial (Date)), Max (financials (Date))
or
Table = Calender (date (2012, 12, 29), date (2020, 12, 31))

YEAR

column = Year (table (Date))

MONTH

column = Month (table (Date))

FORMAT

column = FORMAT (table (Date), "MMM")
= FORMAT (table (Date), "DDD")

QUARTER

column = Quarter (table (Date))
= "Quarter" & Quarter (table (Date))

DAY

Column = day (table (Date))

NOW

It return as a Scalar value as a Current time

Measure = NOW ()

In real time we use to show last Refresh time. click on Refresh.

TODAY

Measure = TODAY ()

It Show's only date

UTC NOW

To fetch Universal time

Measure = UTC NOW ()

UTC TODAY

when we want to fetch Universal time on card

Measure = UTC TODAY ()

C-4

Date Diff

To finding the difference b/w 2 dates Ex: to find Age

DATE1 DATE2 Interval

Measure = DATEDIFF (1997, 9, 11, 2021, 9, 11) Month

= DATEDIFF(DATE(1997, 9, 11), DATE(2021, 9, 11) Month)

DATE VALUE

Basically this date value function will convert String Information into date.

1-Aug-2021 → 1-8-2021

New Column : DATE VALUE (financials(Date))

EDATE

It take Current date to next Month Same date or Previous Month Same date or 2 Months next or 2 Months Previous Same date.

New Column = EDATE (financials(Date), 1) Add's 1 Month for Current(D)

= EDATE (financials(Date), -1) last Month for Current Date

= EDATE (financials(Date), 0) Same as it is

3 Parameters for EDATE

1) Same Date with Previous Month (-1)
2) Same Date with next Month (+1)
3) Same Date for Current Date (0)

Syntax = EDATE (Start Date, Months)

EOMONTH

End of the Month

For Current date to Showing the End of the Month Information we will use this EOMonth fn.

New Column : EOMONTH (Start_date , Months)

: EOMONTH (financials (Date), 0) → Current Month End date

: EOMONTH (financials (Date), 1) → Next Month End date

: EOMONTH (financials (Date), -1) → Previous Month End date

WEEKDAY

To Add Weekday for a Particular date column

New Column : WEEKDAY (Date, Returntype)

: WEEKDAY (~~Dat~~ financials (Date), 1) Starts from Sunday as 1

: FORMAT (financials (Date), "dddd")

If we keep '0' Means Shows ERROR

New Column : WEEKDAY (financials (Date), 2) Stars from Monday as 1

if → 1 Means	if → 2 Means	if → 3 Means
Sun - 1	Mon - 1	Tue - 1
Mon - 2	Tue - 2	Wed - 2
Tue - 3	Wed - 3	Thu - 3
Sat - 7	Sun - 7	Mon - 7

WEEKNUM

To find the Weeknum according to Month or Yearwise

New column = fn. WEEKNUM (Date, (ReturnTiPe) as Pe Day(column)

= WEEKNUM (financials (Date), 1)

SAME PERIOD LAST YEAR

To See the Comparision between Current Year & Previous Year we go this fn. for YOY case.

DATES IN PERIOD.

When we want to take Some Periods of Informahon like last 6 Months Informahon from Current date. In this Case we go with this fn.

SAME PERIOD LAST YEAR.

In visualizahon Table

Date, Measure CY, Measure PY

Measure CY = Calculate (Sum (final (Sales)), DATESYTD (financials (Date)).

Measure PY = Calculate (Sum (final (Sales)), SAMEPERIODLY (financial (Date))

for YOY

```
1 Measure YOY =
2 Var CY = Calculate (Sum (finan (Sales)), DatesYTD (finan (date)))
3 Var PY = Calculate (Sum (fina (Sales)), SAMEPERIODLY (financial (date)))
4 Var YOY = CY - PY
5 Return
6 YOY
```

To find % wise

1. Measure YOY =
2. Var CY = Calculate (Sum (financials (Sales)), DATESYTD (financials (Date))
3. Var PY = Calculate (Sum (financials (Sales)), SAMEPLY (financials (Date))
4. Var YOY = CY - PY
5. Var YOYPercent = DIVIDE (YOY, PY, 0)
6. Return
7. YOY Percent.

If we get Infinity in %, can use alternate value as 0, can use 0 by IF Condition also.

PREVIOUS DAY.

Take new table column as CalendarAuto. Title (D)

Column = PREVIOUSDAY (Table (Date))

Measure = Calculate (Sum (Sales), Previousday (date))

PREVIOUS MONTH

Column = PREVIOUS MONTH (Table (Date))

PREVIOUS QUARTER

Column = PREVIOUS QUARTER (Table (Date))

PREVIOUS YEAR

It takes only last year 2020 total

YTD works like Cummulative Sum.

Max we use SAMEPERIODLASTYEAR in Place of Previous Year

Next DAY

Same as Privious

column = Calculate (Sum (Sales, nextday (date))

In Realtime we use Max New Measure.

BLANK.

To ouwnte In Black Cell.

Measure = If (date = blank(), 0, Sum(Sales)

Black Mean null in Backend

```
1 Measure =
2 Var Cy = Calculate (Sum (Sales), datecytd (date))
3 Var Py = Calculate (Sum (Sales), SamePLY (date))
4 YOY = Cy - Py
5 Return
6 If (Py = blank(), 0, YOY/Py)
```

Note

Null + any value = Null

USER NAME & USER PRINCIPALNAME

We use this functions while ~~dev~~ Implementing RLS, Mainly In dynamic RLS we use this functions,

In Power bi ~~if~~ we use USERNAME only for like this without Domain

Seenukm97 → USER NAME

But for USER PRINCIPAL the Domain is Mandatory

Seenukm97@gmail.com → USER PRINCIPAL NAME

DATES IN PERIOD

When we want information between 2 dates we go for this fn. like from current date to last 6 months.

ex: Month
: Year

Syntax : DATESINPERIOD(date column, Start date, number of Intervals, Interval type)

Measure

1) var Selected date = Selected value (date)
2) Expression : Calculate (Sum(Sales), Dates In Period (date column, Selected date, -3, Months)

1) Measure (Rolling Months) =
2) var Selected date = SELECTEDVALUE (financials (Date))
3) Return
4) Calculate (Sum (financial (Sales)), DATESINPERIOD (financial (date), Selected date, -3, Month))

In visualization

Date, Sales, Measure (rolling mths)

PARAMETERS.

To give the dynamic connections or to pass any dynamic values. to Implement dynamic value.

Go to Power Query Editor

In Applied Steps

Source ⚙ ← Settings Click.

opens a Windows

• Basic ○ Advanced

File Path

[v]

New Parameter

Parameter 1

Current value

Copy & Paste the Path ↵

Now the file Path in Parameter 1

Point : 2

To filter the data

> Go to Manage Parameter

> New Parameter

Name : Parameter 2

Current value Canada [OK]

Now go to financials table (Country) filter ▼ (dropdown)

Apply TEXT FILTERS Equals [ABC] Parameter 2

Now You can find only Canada Data in a table

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>])

Ex = New Measure =

Calculate (SUM (Financials (sales)),

DATESYTD (financials (Date))

DATESMTD

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

DATESQTD

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

Total YTD

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

Sytax

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

Ex : Measure =

Total YTD (SUM (financials (sales), financial (Date),

All (Date), "6/30")

In this example, year_end_date can be specified as "6/30", "Jun 30", "30 June", or any string that resolves to a Month/day. However, it is recommended you specify year_end-date using "Month/day" (as shown) to ensure the string resolves to a date

TOTAL MTD

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

Syntax

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

= TOTAL MTD (SUM (financials (sales)), financials (Date)),

DATESYTD	TOTAL YTD
Datesytd will return a column with the dates up to now. (You can use datesytd to manipulate the date)	TotalYTD is an aggregate function that will return a scalar (single) value. For instance used with a SUM, you will get the SUM of value from beginning of the year up to now.

IF

Checks a Condition, and returns one value when it's True, otherwise it returns a Second value.

Syntax

New Column = IF(<logical_test>, <value_if_true>
[, <value_if_false>])

Return value

Either value if true, value if false or Blank.

Ex: in New Column

1) Price Group = IF('Product'[List Price] < 500, "Low")

2) Price Group = IF('Product'[List Price] < 500, "Low", "HIGH")

3) Price Group = IF('Product'[List Price] < 500, "Low",
IF('Product'[List Price] < 1500,
"Medium", "HIGH"))