



DAX

Data Analytics Expression

What is DAX?



- ▶ Data Analysis Expressions (DAX) is a programming language that is used throughout Microsoft Power BI for creating calculated columns, measures, and custom tables.
- ▶ It is a collection of functions, operators, and constants that can be used in a formula, or expression, to calculate and return one or more values

What is Measure in DAX?



Measures are DAX formulas used to generate new calculated values

- ▶ Unlike calculated columns, measure values aren't visible within tables; they can only be “seen” within a visualization like a chart or matrix (similar to a calculated field in an Excel pivot)

PRO TIP :

- ▶ Use measures to create **numerical, calculated values** that can be analyzed in the “**values**” field of a report visual

What is Calculated Column in DAX?



Calculated columns allow you to add new, formula-based columns to tables

- ▶ Calculated columns generate values for each row, which are visible within tables in the Data view



Scalar Functions

Scalar Function



- ▶ Scalar Function return a single value, rather than a column or table, common examples include aggregation, conversion, rounding and logical functions.

Aggregation Functions



SUM()	Evaluates the sum of the column	=SUM (ColumnName)
AVERAGE()	Return the average of all the numbers in column	=AVERAGE (ColumnName)
MAX()	Return the largest value in a column	=MAX (ColumnName)
MIN()	Return the smallest value in a column	=MIN (ColumnName)
COUNT()	Count the number of cells in a column that contain numbers	=COUNT (ColumnName)
DISTINCTCOUNT()	Counts the number of distinct or unique values in a column	=DISTINCTCOUNT (ColumnName)
COUNTROWS()	Count the number of rows in the specified table	=COUNTROWS (Table)

Rounding Functions



INT()	Rounds a number down to the nearest integer	=INT (Number)
ROUND()	Round a number to a specific number of digits	=ROUND (Number,NumberOfDigits)
ROUNDUP()	Rounds a number up, away from zero	=ROUNDUP (Number, NumberOfDigits)
ROUNDDOWN()	Rounds a number down, towards zero	=ROUNDDOWN (Number, NumberOfDigits)
MROUND()	Rounds a number of the desired multiple	=MROUND (Number, Multiple)
TRUNC()	Truncates a number to an integer by removing the decimal part of the number	=TRUNC (Number,[NumberOfDigits])
FIXED()	Rounds number down to specified number of decimals and returns results text.	=FIXED (Number, [Decimals], [No.Commas])
CEILING()	Rounds a number up, to the nearest integer or nearest unit of significance	=CEILING (Number, Significance)
FLOOR()	Rounds a number down, toward zero, to the nearest multiple of significance	=FLOOR (Number, Significance)

Information Functions



ISBLANK()	Check whether a value is blank, and returns TRUE or FALSE	=ISBLANK(Value)
ISERROR()	Check whether a value is an error, and returns TRUE or FALSE	=ISERROR(Value)
ISLOGICAL()	Check whether a value is a logical value(TRUE or FALSE), and returns TRUE or FALSE	=ISLOGICAL(Value)
ISNUMBER()	Check whether a value is a number, and returns TRUE or FALSE	=ISNUMBER(Value)
ISNONTTEXT()	Check whether a value is not text (blank cells are not text), and returns TRUE or FALSE	=NONTTEXT(Value)
ISTEXT()	Check whether a value is a text, and returns TRUE or FALSE	=ISTEXT(Value)

Conversion Functions



CURRENCY()	Evaluates the argument and returns the result as a currency data type	=CURRENCY(Value)
FORMAT()	Converts a value to text un the specified number format	=FORMAT(Value, Format)
DATE()	Returns the specified date in datetime format	=DATE(Year, Month, Day)
TIME()	Converts hours, minutes, and seconds given as numbers to a time in datetime format	=TIME(Hours, Minute, Second)
DATEVALUE()	Converts a date in the form of text to a date in datetime format	=DATEVALUE(Date Text)
VALUE()	Converts a text string that represents to a number	=VALUE(Text)

Logical Functions



- ▶ IF(): Check if a given condition is met, and return one value if the condition is TRUE, and another if the condition is FALSE

=IF(LogicalTest, ResultIfTrue, [ResultIfFalse])

- ▶ Check whether both arguments are TRUE, and return TRUE if both the arguments are TRUE, otherwise returns FALSE

=AND(Logical1, Logical2)

- ▶ Check whether one of the arguments is TRUE to return TRUE, and return FALSE if both the arguments are FALSE.

=OR(Logical1, Logical2)

SWITCH()



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

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



Table & Filter Functions

CALCULATED TABLES



DAX functions with **table** arguments can typically accept either **physical** tables (*i.e.* 'Sales by Store') or **calculated, virtual** tables (*with functions like FILTER, VALUES, etc.*)

=**SUMX**(Table, Expression)

```
1 Total Sales =  
2 SUMX(  
3     'Sales by Store',  
4     'Sales by Store'[quantity_sold])
```

```
1 Total Sales =  
2 SUMX(  
3     FILTER(  
4         'Sales by Store',  
5         'Sales by Store'[quantity_sold]) > 3  
6     ),  
7     'Sales by Store'[quantity_sold]) )
```

DISTINCT



Returns a **single column** table of unique values when a **column name** is given. If a **table** is supplied, *DISTINCT* returns all unique combinations of values.

=**DISTINCT**(ColumnNameor TableExpression)

VALUES()



Returns a **single column** table of unique values when a **column name** is given. If a table name is supplied, *VALUES* returns the entire table (including duplicates) plus a blank row.

=**VALUES**(ColumnName or TableExpression)



SELECTEDVALUE()

Returns a value when there's **only one value** in a specified column, otherwise returns an (optional) alternate result.

=**SELECTEDVALUE**(ColumnName,[AlternateResult])

SELECTCOLUMNS()



Returns a table with selected columns from the table plus any new columns specified by the DAX expressions

=SELECTCOLUMNS(Table, Name, Expression, [...])



ADDCOLUMNS()

Returns a table with selected columns from the table plus any new columns specified by the DAX expressions

=**ADDCOLUMNS**(Table, Name, Expression, [...])



Calculated Join Functions

CALCULATED TABLE JOINS



Calculated table joins are used to combine two or more tables of data, common examples include CROSSJOIN, UNION, EXCEPT and INTERSECT.

Common Use cases:

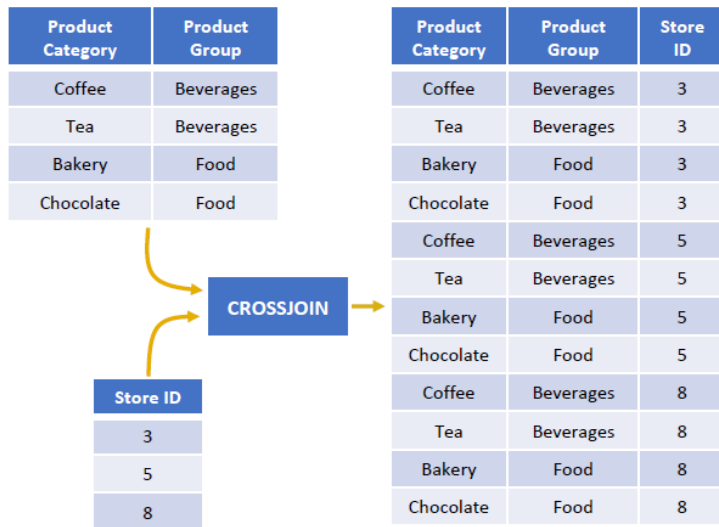
- ▶ Combining data across multiple tables.
- ▶ Querying tables to troubleshoot errors or better understand relationships in data model.

CROSSJOIN()



Return a table that contains the Cartesian product of the specified tasks.

`=CROSSJOIN(Table, Table, [...])`



Resulting table contains **12 rows** (4*3)
and **3 columns** (2+1)

UNION()



Combines or "stacks" rows from two or more tables sharing the same column structure

=**UNION**(Table, Table, [...])

Date	ABC	XYZ	123
5/1/19	A	X	1
5/1/19	B	Y	2
5/1/19	C	Z	3

Date	ABC	XYZ	123
5/1/20	A	X	1
5/1/20	B	Y	2
5/1/20	C	Z	3

UNION

Date	ABC	XYZ	123
5/1/19	A	X	1
5/1/19	B	Y	2
5/1/19	C	Z	3
5/1/20	A	X	1
5/1/20	B	Y	2
5/1/20	C	Z	3

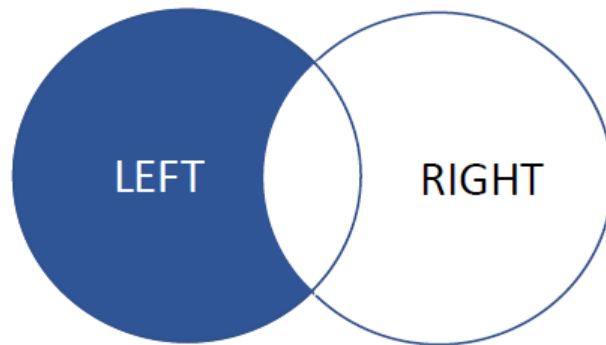
UNION stacks tables together,
just like **append**!

EXCEPT()



Returns all rows from the left table which do not appear in the right table

=**EXCEPT**(LeftTable, RightTable)



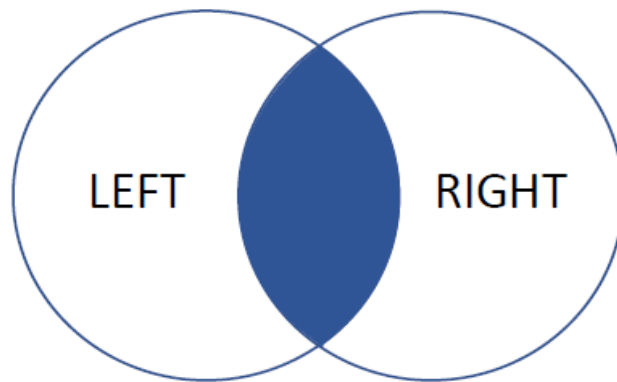
*Resulting table contains rows which
ONLY appear in the **left table***

INTERSECT()



Returns all the rows from the left table which also appear in the right table

=INTERSECT(LeftTable, RightTable)



*Resulting table contains rows
which appear in **BOTH** tables*



Relationship Functions



Relationship Functions

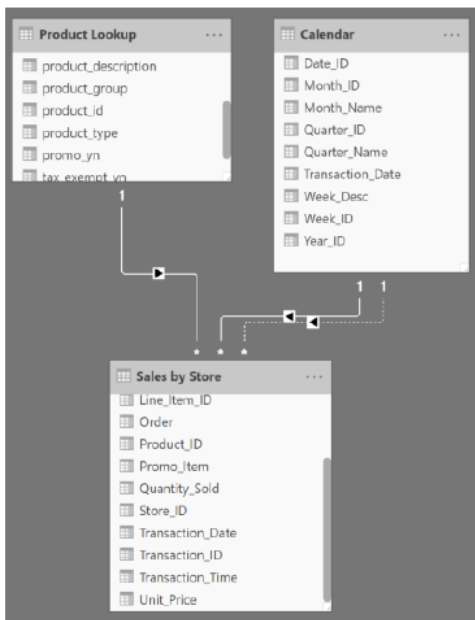
Relationship functions allow you to access fields within DAX measures or calculated columns through either *physical* or *virtual* relationships between tables

Relationship Functions



There are two key types of table relationships: **PHYSICAL** and **VIRTUAL**

- ▶ **Physical** relationships are manually created, and visible in your data model
- ▶ **Virtual** relationships are temporary, and defined using DAX expressions



Physical Relationships

```
1 Bean Goal (TREATAS) =
2 CALCULATE(
3     SUM(
4         'Target Sales UNION Example'[Bean/Teas Goal]
5     ),
6     TREATAS(
7         VALUES(
8             'Calendar'[Year_ID]
9         ),
10        'Target Sales UNION Example'[Year]
11    ),
12    TREATAS(
13        VALUES(
14            'Calendar'[Month_Name]
15        ),
16        'Target Sales UNION Example'[Month]
17    )
18 )
```

Virtual Relationships

Related()



Returns a value from a related table in the data model

=**RELATED**(ColumnName)



USERELATIONSHIP()

Specifies an existing relationship to be used in the evaluation of a DAX expression, defined by naming, as arguments, the two columns that serve as endpoints.

=USERELATIONSHIP(ColumnName1,ColumnName2)



Tips and Key Practices



Thank You !!

