

CHEAT SHEET - DAX

Power BI (DAX) cheat sheet

The DAX functions that power measures and calculated columns - aggregations, filter context, time intelligence - with the notes that explain why a measure returns the wrong number.

87

FUNCTIONS

10

CATEGORIES

DAX

LANGUAGE

Aggregations 11 functions

The plain functions aggregate one column. The X variants iterate a table row by row, evaluating an expression per row before aggregating.

SUM	Total a column <code>SUM(Sales[Amount])</code>
SUMX	Sum a per-row expression <code>SUMX(Sales, Sales[Qty] * Sales[Price])</code>
AVERAGE	Mean of a column <code>AVERAGE(Sales[Amount])</code>
AVERAGEX	Mean of a per-row expression <code>AVERAGEX(Sales, Sales[Qty] * Sales[Price])</code>
MIN / MAX	Smallest or largest value <code>MAX(Sales[Amount])</code>
MINX / MAXX	Min/max of a per-row expression <code>MAXX(Sales, Sales[Qty] * Sales[Price])</code>
COUNT	Count numeric values <code>COUNT(Sales[OrderID])</code>
COUNTA	Count non-blank values <code>COUNTA(Sales[Notes])</code>
COUNTROWS	Count rows in a table <code>COUNTROWS(Sales)</code>
DISTINCTCOUNT	Count distinct values <code>DISTINCTCOUNT(Sales[CustomerID])</code>
COUNTBLANK	Count blank values <code>COUNTBLANK(Sales[Notes])</code>

Filter context 13 functions

CALCULATE is the engine of DAX. It evaluates an expression after changing the filter context, and it forces context transition - turning the current row into an equivalent filter. Almost every non-trivial measure goes through it.

CALCULATE	Evaluate with modified filters <code>CALCULATE(SUM(Sales[Amount]), Sales[Region] = "West")</code>
CALCULATETABLE	Like CALCULATE, returns a table <code>CALCULATETABLE(Sales, Sales[Year] = 2026)</code>
FILTER	Return rows matching a condition <code>FILTER(Sales, Sales[Amount] > 1000)</code>
ALL	Ignore filters on a table/column <code>CALCULATE(SUM(Sales[Amount]), ALL(Sales))</code>
ALLEXCEPT	Ignore all filters except listed <code>CALCULATE(SUM(Sales[Amount]), ALLEXCEPT(Sales, Sales[Region]))</code>
ALLSELECTED	Respect slicer/outer selection <code>CALCULATE(SUM(Sales[Amount]), ALLSELECTED(Sales[Region]))</code>

REMOVEFILTERS	Clear filters (modern ALL) <code>CALCULATE(SUM(Sales[Amount]), REMOVEFILTERS(Sales[Region]))</code>
KEEPFILTERS	Add a filter without overriding <code>CALCULATE([Sales], KEEPFILTERS(Sales[Region] = "West"))</code>
VALUES	Distinct values as a table <code>VALUES(Sales[Region])</code>
SELECTEDVALUE	The single value in context <code>SELECTEDVALUE(Sales[Region], "All regions")</code>
HASONEVALUE	True if exactly one value <code>HASONEVALUE(Sales[Region])</code>
ISFILTERED	True if a column is filtered <code>ISFILTERED(Sales[Region])</code>
ISINSCOPE	True if a column is a grouping level <code>ISINSCOPE(Sales[Region])</code>

Iterators & variables 7 functions

VAR / RETURN	Store a value for reuse <code>VAR Total = SUM(Sales[Amount]) RETURN Total * 1.1</code>
RANKX	Rank rows by an expression <code>RANKX(ALL(Product), [Total Sales])</code>
CONCATENATEX	Join an expression across rows <code>CONCATENATEX(VALUES(Product[Name]), Product[Name], ", ")</code>
ADDCOLUMNS	Add computed columns to a table <code>ADDCOLUMNS(VALUES(Region[Name]), "Sales", [Total Sales])</code>
SELECTCOLUMNS	Pick and rename columns <code>SELECTCOLUMNS(Sales, "Region", Sales[Region])</code>
SUMMARIZE	Group rows and summarize <code>SUMMARIZE(Sales, Region[Name], "Sales", SUM(Sales[Amount]))</code>
SUMMARIZECOLUMNS	Group and summarize (modern, faster) <code>SUMMARIZECOLUMNS(Region[Name], "Sales", SUM(Sales[Amount]))</code>

Time intelligence 10 functions

These need a dedicated date table that is contiguous (no gaps), covers full years, and is marked as a date table. Without one, they silently return wrong or blank results.

TOTALYTD	Year-to-date total <code>TOTALYTD(SUM(Sales[Amount]), 'Date'[Date])</code>
DATESYTD	Year-to-date date set <code>CALCULATE(SUM(Sales[Amount]), DATESYTD('Date'[Date]))</code>
SAMEPERIODLASTYEAR	Shift context back one year <code>CALCULATE(SUM(Sales[Amount]), SAMEPERIODLASTYEAR('Date'[Date]))</code>
DATEADD	Shift by an interval <code>CALCULATE(SUM(Sales[Amount]), DATEADD('Date'[Date], -1, MONTH))</code>

DATESINPERIOD	A rolling window of dates <code>CALCULATE(SUM(Sales[Amount]), DATESINPERIOD('Date'[Date], MAX('Date'[Date]), -3, MONTH))</code>
PARALLELPERIOD	A whole shifted period <code>CALCULATE(SUM(Sales[Amount]), PARALLELPERIOD('Date'[Date], -1, YEAR))</code>
PREVIOUSMONTH	The prior month <code>CALCULATE(SUM(Sales[Amount]), PREVIOUSMONTH('Date'[Date]))</code>
DATESBETWEEN	An explicit date range <code>CALCULATE(SUM(Sales[Amount]), DATESBETWEEN('Date'[Date], DATE(2026,1,1), DATE(2026,3,31)))</code>
FIRSTDATE / LASTDATE	First/last date in context <code>LASTDATE('Date'[Date])</code>
ENDOFMONTH	Last day of the month in context <code>ENDOFMONTH('Date'[Date])</code>

Logical & error handling 11 functions

IF	Branch on a condition <code>IF([Sales] > 0, "Yes", "No")</code>
SWITCH	Match one value against many <code>SWITCH([Grade], "A", 4, "B", 3, 0)</code>
SWITCH(TRUE())	If / else-if chain <code>SWITCH(TRUE(), [Margin] > 0.3, "High", [Margin] > 0.1, "Mid", "Low")</code>
AND &&	All conditions true <code>[Active] && [Approved]</code>
OR 	Any condition true <code>[IsAdmin] [IsOwner]</code>
NOT	Invert a condition <code>NOT(ISBLANK([Sales]))</code>
DIVIDE	Safe division <code>DIVIDE([Profit], [Sales], 0)</code>
COALESCE	First non-blank value <code>COALESCE([Sales], 0)</code>
IFERROR	Catch and replace an error <code>IFERROR([Ratio], 0)</code>
ISBLANK	Test for blank <code>ISBLANK([Sales])</code>
BLANK	Return an explicit blank <code>IF([Sales] = 0, BLANK(), [Sales])</code>

Relationships 6 functions

RELATED	Pull a value across a many-to-one relationship <code>RELATED(Product[Category])</code>
----------------	---

RELATEDTABLE	Get related rows (one-to-many) COUNTROWS(RELATEDTABLE(Sales))
USERELATIONSHIP	Activate an inactive relationship CALCULATE([Sales], USERELATIONSHIP(Sales[ShipDate], 'Date'[Date]))
CROSSFILTER	Set cross-filter direction CALCULATE([Sales], CROSSFILTER(Sales[CustID], Customer[CustID], BOTH))
LOOKUPVALUE	Look up a value with no relationship LOOKUPVALUE(Rate[USD], Rate[Date], Sales[Date])
TREATAS	Apply one table as a filter on another CALCULATE([Sales], TREATAS(VALUES(Budget[Region]), Sales[Region]))

Text 10 functions

FORMAT	Format a value as text FORMAT([Sales], "#,##0.00")
CONCATENATE &	Join text values "Region: " & SELECTEDVALUE(Region[Name])
LEFT / RIGHT	Characters from one end LEFT([Code], 3)
MID	Characters from a position MID([Code], 4, 2)
LEN	Count characters LEN([Code])
UPPER / LOWER	Change case UPPER([Code])
TRIM	Remove extra spaces TRIM([Name])
SUBSTITUTE	Replace text by match SUBSTITUTE([Code], "-", "")
SEARCH / FIND	Position of a substring SEARCH("@", [Email], 1, BLANK())
VALUE	Text to number VALUE([CodeText])

Math & statistics 9 functions

ROUND	Round to N decimals ROUND([Sales], 2)
ROUNDUP / ROUNDDOWN	Round away from / toward zero ROUNDUP([Sales], 0)
INT	Truncate to an integer INT([Value])

ABS	Absolute value <code>ABS([Variance])</code>
MOD	Remainder after division <code>MOD([Row], 2)</code>
CEILING / FLOOR	Round to a multiple <code>CEILING([Price], 0.05)</code>
MEDIAN	Median of a column <code>MEDIAN(Sales[Amount])</code>
PERCENTILE.INC	Percentile of a column <code>PERCENTILE.INC(Sales[Amount], 0.9)</code>
RANK.EQ	Rank a value within a column <code>RANK.EQ([Sales], ALL(Sales[Amount]))</code>

Tables & set operations 7 functions

These return tables. Use them as the table argument of an iterator or filter, or inside CALCULATETABLE - you cannot put a table directly in a card visual.

TOPN	Top N rows by an expression <code>TOPN(5, Product, [Total Sales], DESC)</code>
UNION	Stack rows from tables <code>UNION(Actuals, Budget)</code>
INTERSECT	Rows present in both tables <code>INTERSECT(VALUEA(A[ID]), VALUEA(B[ID]))</code>
EXCEPT	Rows in the first table only <code>EXCEPT(VALUEA(A[ID]), VALUEA(B[ID]))</code>
DISTINCT	Distinct rows of a table <code>DISTINCT(Sales[Region])</code>
GENERATESERIES	Build a numeric series <code>GENERATESERIES(1, 12, 1)</code>
ROW	Build a single-row table <code>ROW("Total", SUM(Sales[Amount]))</code>

Security & context info 3 functions

Used mainly in row-level security (RLS) rules.

USERPRINCIPALNAME	Current user's UPN <code>Customer[Email] = USERPRINCIPALNAME()</code>
USERNAME	Current user (domain\user) <code>USERNAME()</code>
CUSTOMDATA	Read a value passed in the connection <code>CUSTOMDATA()</code>