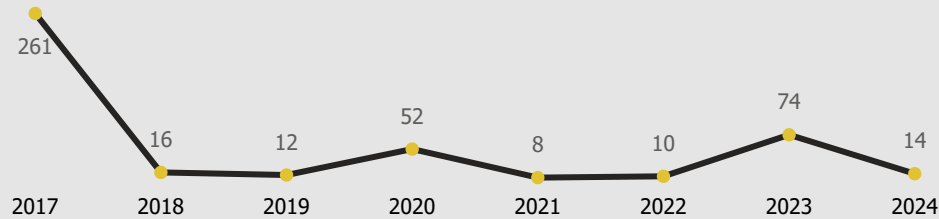




# POWER BI DAXpedia

The DAX (Data Analysis Expressions) language was specifically designed for managing and analyzing data models through formulas and expressions. It is utilized across several Microsoft products, including Microsoft Power BI, Microsoft Analysis Services, and Microsoft Power Pivot for Excel. These tools leverage the same internal engine, known as Tabular, to ensure consistency and performance in data modeling and analysis.

No. of Functions Released Yearly



Most Recent Released Functions

Released Month: Sep-24

- INFO.VIEW.COLUMNS
- INFO.VIEW.RELATIONSHIPS
- INFO.VIEW.TABLES
- INFO.VIEW.MEASURES

DAXpedia includes **15** DAX Function Types and a comprehensive library of **447** DAX Functions, empowering dynamic data modeling and analysis.

Reference: [DAX GUIDE](#)

## Aggregation Functions

Aggregation functions return a scalar value by applying an aggregation function to a column or to an expression evaluated by iterating a table expression.

## Date and Time Functions

Date and time functions help create calculations based on dates and time. Many of the functions in DAX are similar to the Excel date and time functions.

## Filter Functions

Filter functions manipulate table and filter contexts.

## Financial Functions

Financial functions corresponding to Excel functions with the same name.

## Information Functions

Information functions provide information about the data type or filter context of the argument provided.

## Logical Functions

Logical functions act upon an expression, to return information about the values or sets in the expression.

## Math and Trig Functions

The mathematical functions in DAX are very similar to the Excel mathematical and trigonometric functions.

## Other Functions

These are special functions that cannot be classified in other categories.

## Parent-child Functions

These functions help flatten a parent-child hierarchy into a regular hierarchy.

## Relationship management Functions

These functions manage and manipulate relationships between tables.

## Statistical Functions

Statistical aggregation functions.

## Table manipulation Functions

These functions manipulate and return tables.

## Text Functions

Text functions manipulate strings.

## Time Intelligence Functions

Time intelligence functions support calculations to compare and aggregate data over time periods, supporting days, months, quarters, and years.

## Visual calculations Functions

Visual calculations functions can only be used in Power BI visual calculations and cannot be used in regular measures, calculated columns, nor calculated tables.



Select your DAX function

- 
- ☐

BITLSHIFT

☐

BITOR

☐

BITRSHIFT

☐

BITXOR

☐

BLANK

☒

CALCULATE

☐

CALCULATETABLE

☐

CALENDAR

☐

CALENDARAUTO

☐

CEILING

☐

CHISQ.DIST

☐

CHISQ.DIST.RT

☐

CHISQ.INV

☐

CHISQ.INV.RT

☐

CLOSINGBALANCEMONTH

☐

CLOSINGBALANCEQUARTER

☐

CLOSINGBALANCEYEAR

☐

COALESCE

☐

COLLAPSE

☐

COLLAPSEALL

# CALCULATE

Released Date: Before 2017-01-01

## DAX Function (Filter)

Evaluates an expression in a context modified by filters.

### Syntax

CALCULATE ( <Expression> [, <Filter> [, <Filter> [, ... ] ] ] )

### Return values

Scalar

A single value of any type. The value is the result of the expression evaluated in a modified filter context.

### Remarks

Every filter argument can be either a filter removal (such as ALL, ALLEXCEPT, ALLNOBLANKROW), a filter restore (ALLSELECTED), or a table expression returning a list of values for one or more columns or for an entire expanded table. When a filter argument has the form of a predicate with a single column reference, the expression is embedded into a FILTER expression that filters all the values of the referenced column. For example, the predicate shown in the first expression is internally converted in the second expression. CALCULATE ( <expression>, table[column] = 10 ) CALCULATE ( <expression>, FILTER ( ALL ( table[column] ), table[column] = 10 )) A filter argument overrides the existing corresponding filters over the same column(s), unless it is embedded within KEEPFILTERS. CALCULATE evaluation follow these steps: CALCULATE evaluates all the explicit filter arguments in the original evaluation context, each one independently from the others. This includes both the original row contexts (if any) and the original filter context. Once this evaluation is finished, CALCULATE starts building the new filter context. CALCULATE makes a copy of the original filter context to prepare the new filter context. It discards the original row contexts, because the new evaluation context will not contain any row context. CALCULATE performs the context transition. It uses the current value of columns in the original row contexts to provide a filter with a unique value for all the columns currently being iterated in the original row contexts. This filter may or may not contain one individual row. There is no guarantee that the new filter context contains a single row at this point. If there are no row contexts active, this step is skipped. Once all implicit filters created by the context transition are applied to the new filter context, CALCULATE moves on to the next step. CALCULATE evaluates the CALCULATE modifiers used in filter arguments: USERRELATIONSHIP, CROSSFILTER, ALL, ALLEXCEPT, ALLSELECTED, and ALLNOBLANKROW. This step happens after step 3. This is very important, because it means that one can remove the effects of the context transition by using ALL as a filter argument. The CALCULATE modifiers are applied after the context transition, so they can alter the effects of the context transition. CALCULATE applies the explicit filter arguments evaluated at 1. to the new filter context generated after step 4. These filter arguments are applied to the new filter context once the context transition has happened so they can overwrite it, after filter removal — their filter is not removed by any ALL\* modifier — and after the relationship architecture has been updated. However, the evaluation of filter arguments happens in the original filter context, and it is not affected by any other modifier or filter within the same CALCULATE function. If a filter argument is modified by KEEPFILTERS, the filter is added to the filter context without overwriting existing filters over the same column(s). The filter context generated after point (5) is the new filter context used by CALCULATE in the evaluation of its expression.