# Microsoft Power BI:DAX functions

- ## Aggregation functions

| Function | Description | Syntax Example |
|---|---|---|
| APPROXIMATEDISTINCTCOUNT | Estimated count of unique values | APPROXIMATEDISTINCTCOUNT(Sales[CustomerID]) |
| AVERAGE | Average of numbers in a column | AVERAGE(Sales[SalesAmount]) |
| AVERAGEA | Average including text/logical | AVERAGEA(Sales[Quantity]) |
| AVERAGEX | Avg of expression over table | AVERAGEX(Sales, Sales[SalesAmount]*Sales[Quantity]) |
| COUNT | Count numeric non-blanks | COUNT(Sales[Quantity]) |
| COUNTA | Count all non-blanks (incl. text, Boolean) | COUNTA(Sales[CustomerName]) |
| COUNTAX | Count non-blank results of expression | COUNTAX(Sales, Sales[Quantity]*2) |
| COUNTBLANK | Count blank values | COUNTBLANK(Sales[Remarks]) |
| COUNTROWS | Count rows in table | COUNTROWS(Sales) |
| COUNTX | Count numeric expression results | COUNTX(Sales, Sales[Quantity]*Sales[UnitPrice]) |
| DISTINCTCOUNT | Count unique values | DISTINCTCOUNT(Sales[CustomerID]) |
| DISTINCTCOUNTNOBLANK | Count unique values excluding blanks | DISTINCTCOUNTNOBLANK(Sales[CustomerID]) |
| MAX | Largest numeric value | MAX(Sales[SalesAmount]) |
| MAXA | Largest including logical/text | MAXA(Sales[Score]) |
| MAXX | Max of expression per row | MAXX(Sales, Sales[Quantity]*Sales[UnitPrice]) |
| MIN | Smallest numeric value | MIN(Sales[SalesAmount]) |
| MINA | Smallest incl. logical/text | MINA(Sales[Score]) |
| MINX | Min of expression per row | MINX(Sales, Sales[Quantity]*Sales[UnitPrice]) |
| PRODUCT | Product of all values | PRODUCT(Sales[Quantity]) |
| PRODUCTX | Product of expression per row | PRODUCTX(Sales, Sales[Quantity]*Sales[UnitPrice]) |
| SUM | Sum of numbers in column | SUM(Sales[SalesAmount]) |
| SUMX | Sum of expression per row | SUMX(Sales, Sales[Quantity]*Sales[UnitPrice]) |

# Microsoft Power BI:DAX functions

- ## Date and time functions

| Function | Description | Syntax Example |
|---|---|---|
| CALENDAR | Returns a contiguous set of dates in a table | CALENDAR(DATE(2024,1,1), DATE(2024,12,31)) |
| CALENDARAUTO | Creates a contiguous date table based on data model | CALENDARAUTO() |
| DATE | Creates a date in datetime format | DATE(2025,8,17) |
| DATEDIFF | Difference between two dates (unit: DAY, MONTH, YEAR, etc.) | DATEDIFF(DATE(2024,1,1), DATE(2024,12,31), DAY) |
| DATEVALUE | Converts text to date | DATEVALUE("2025-08-17") |
| DAY | Extracts day from date | DAY(DATE(2025,8,17)) |
| EDATE | Date shifted by given months | EDATE(DATE(2025,8,17), 2) |
| EOMONTH | Last day of month, shifted by N months | EOMONTH(DATE(2025,8,17), 1) |
| HOUR | Extracts hour from datetime | HOUR(NOW()) |
| MINUTE | Extracts minute from datetime | MINUTE(NOW()) |
| MONTH | Extracts month number | MONTH(DATE(2025,8,17)) |
| NETWORKDAYS | Number of working days between two dates (excludes weekends/holidays) | NETWORKDAYS(DATE(2025,8,1), DATE(2025,8,31)) |
| NOW | Current date & time | NOW() |
| QUARTER | Extracts quarter number (1–4) | QUARTER(DATE(2025,8,17)) |
| SECOND | Extracts seconds from datetime | SECOND(NOW()) |
| TIME | Creates time from hours, minutes, seconds | TIME(14, 30, 0) |
| TIMEVALUE | Converts text time to datetime | TIMEVALUE("2:45 PM") |
| TODAY | Current date | TODAY() |
| UTCNOW | Current UTC date & time | UTCNOW() |
| UTCTODAY | Current UTC date | UTCTODAY() |
| WEEKDAY | Returns day of week (1=Sunday, 7=Saturday by default) | WEEKDAY(DATE(2025,8,17)) |
| WEEKNUM | Week number of a date | WEEKNUM(DATE(2025,8,17), 2) |
| YEAR | Extracts year | YEAR(DATE(2025,8,17)) |
| YEARFRAC | Fraction of year between two dates | YEARFRAC(DATE(2025,1,1), DATE(2025,8,17)) |

# Microsoft Power BI:DAX functions

- **Filter functions**

| Function | Description | Syntax Example |
|---|---|---|
| ALL | Returns all rows in a table/column, ignoring filters | CALCULATE(SUM(Sales[Amount]), ALL(Sales)) |
| ALLCROSSFILTERED | Clears filters from related tables | CALCULATE(SUM(Sales[Amount]), ALLCROSSFILTERED(Product)) |
| ALLEXCEPT | Clears all filters except specified | CALCULATE(SUM(Sales[Amount]), ALLEXCEPT(Sales, Sales[Region])) |
| ALLNOBLANKROW | Returns all rows except blank row | CALCULATE(COUNTROWS(Customer), ALLNOBLANKROW(Customer)) |
| ALLSELECTED | Clears filters only from current selection (visual) | CALCULATE(SUM(Sales[Amount]), ALLSELECTED(Sales)) |
| CALCULATE | Evaluates expression with modified filter context | CALCULATE(SUM(Sales[Amount]), Sales[Region]="West") |
| CALCULATETABLE | Returns a table with modified filter context | CALCULATETABLE(FILTER(Sales, Sales[Amount]>1000), Sales[Region]="West") |
| EARLIER | Gets column value from earlier row context | FILTER(Sales, Sales[Amount] > EARLIER(Sales[Amount])) |
| EARLIEST | Similar to EARLIER, earliest outer evaluation | Used inside nested row contexts |
| FILTER | Returns a filtered table | FILTER(Sales, Sales[Amount]>1000) |
| FIRST | Visual calc: first row value on axis | In matrix: FIRST(SUM(Sales[Amount])) |
| INDEX | Returns row at absolute position | INDEX(3, Sales, ORDERBY(Sales[Amount], ASC)) |
| KEEPFILTERS | Keeps filters instead of overriding | CALCULATE(SUM(Sales[Amount]), KEEPFILTERS(Sales[Amount]>1000)) |
| LAST | Visual calc: last row value on axis | LAST(SUM(Sales[Amount])) |
| LOOKUP | Visual calc: lookup value with filters | Works in matrix context |
| LOOKUPWITHTOTALS | Visual calc: lookup with explicit filters only | Similar to LOOKUP |
| LOOKUPVALUE | Returns value matching condition(s) | LOOKUPVALUE(Product[Price], Product[ProductID], 101) |
| MATCHBY | Defines match columns in window functions | MATCHBY(Sales[Region]) |
| MOVINGAVERAGE | Returns moving average over axis | MOVINGAVERAGE(SUM(Sales[Amount]), -3, 0) |
| NEXT | Visual calc: value from next row | NEXT(SUM(Sales[Amount])) |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| OFFSET | Returns row offset by N | OFFSET(-1, Sales, ORDERBY(Sales[Date], ASC)) |
| ORDERBY | Defines order in window functions | ORDERBY(Sales[Date], ASC) |
| PARTITIONBY | Defines partition columns in window functions | PARTITIONBY(Sales[Region]) |
| PREVIOUS | Visual calc: value from previous row | PREVIOUS(SUM(Sales[Amount])) |
| RANGE | Returns interval of rows (like WINDOW) | RANGE(-2, 0, ORDERBY(Sales[Date], ASC)) |
| RANK | Ranking within partition | RANK(SUM(Sales[Amount]), ORDERBY(Sales[Amount], DESC)) |
| REMOVEFILTERS | Clears filters from table/column | CALCULATE(SUM(Sales[Amount]), REMOVEFILTERS(Sales)) |
| ROWNUMBER | Unique row number within partition | ROWNUMBER(ORDERBY(Sales[Amount], DESC)) |
| RUNNINGSUM | Running total along axis | RUNNINGSUM(SUM(Sales[Amount])) |
| SELECTEDVALUE | Returns value if one value is selected | SELECTEDVALUE(Sales[Region], "Multiple") |
| WINDOW | Returns multiple rows in interval | WINDOW(-2, 0, ORDERBY(Sales[Date], ASC)) |

# Microsoft Power BI:DAX functions

- ## Financial functions

| Function | Description | Syntax Example |
|---|---|---|
| ACCRINT | Accrued interest for periodic-paying security | ACCRINT(issue, first_int, settlement, rate, par, freq) |
| ACCRINTM | Accrued interest for maturity-paying security | ACCRINTM(issue, settlement, rate, par) |
| AMORDEGRC | Depreciation with coefficient | AMORDEGRC(cost, date_purch, first_per, salvage, period, rate) |
| AMORLINC | Linear depreciation | AMORLINC(cost, date_purch, first_per, salvage, period, rate) |
| COUPDAYBS | Days from coupon start → settlement | COUPDAYBS(settlement, maturity, freq) |
| COUPDAYS | Total days in coupon period | COUPDAYS(settlement, maturity, freq) |
| COUPDAYSNC | Days from settlement → next coupon | COUPDAYSNC(settlement, maturity, freq) |
| COUPNCD | Next coupon date after settlement | COUPNCD(settlement, maturity, freq) |
| COUPNUM | No. of coupons until maturity | COUPNUM(settlement, maturity, freq) |
| COUPPCD | Previous coupon date before settlement | COUPPCD(settlement, maturity, freq) |
| CUMIPMT | Cumulative interest in period | CUMIPMT(rate, nper, pv, start, end, type) |
| CUMPRINC | Cumulative principal in period | CUMPRINC(rate, nper, pv, start, end, type) |
| DB | Declining balance depreciation | DB(cost, salvage, life, period) |
| DDB | Double-declining depreciation | DDB(cost, salvage, life, period, factor) |
| DISC | Discount rate for security | DISC(settlement, maturity, pr, redemption, basis) |
| DOLLARDE | Convert fractional price → decimal | DOLLARDE(1.02, 16) |
| DOLLARFR | Convert decimal → fractional price | DOLLARFR(1.125, 16) |
| DURATION | Macauley duration | DURATION(settlement, maturity, coupon, yld, freq, basis) |
| EFFECT | Effective annual interest rate | EFFECT(nominal_rate, npery) |
| FV | Future value of investment | FV(rate, nper, pmt, [pv], [type]) |
| INTRATE | Interest rate for fully invested | INTRATE(settlement, maturity, investment, redemption, |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| | security | basis) |
| IPMT | Interest payment for a period | IPMT(rate, per, nper, pv) |
| ISPMT | Interest in specific period | ISPMT(rate, per, nper, pv) |
| MDURATION | Modified Macauley duration | MDURATION(settlement, maturity, coupon, yld, freq, basis) |
| NOMINAL | Nominal rate given effective | NOMINAL(effect_rate, npery) |
| NPER | Number of periods for investment | NPER(rate, pmt, pv, [fv], [type]) |
| ODDFPRICE | Price of bond w/ odd first period | ODDFPRICE(settlement, maturity, issue, first_coupon, rate, yld, redemption, freq, basis) |
| ODDFYIELD | Yield of bond w/ odd first period | ODDFYIELD(settlement, maturity, issue, first_coupon, rate, pr, redemption, freq, basis) |
| ODDLPRICE | Price of bond w/ odd last period | ODDLPRICE(settlement, maturity, last_interest, rate, yld, redemption, freq, basis) |
| ODDLYIELD | Yield of bond w/ odd last period | ODDLYIELD(settlement, maturity, last_interest, rate, pr, redemption, freq, basis) |
| PDURATION | Periods to reach target FV | PDURATION(rate, pv, fv) |
| PMT | Loan payment (constant payments) | PMT(rate, nper, pv) |
| PPMT | Principal part of payment | PPMT(rate, per, nper, pv) |
| PRICE | Price of bond (periodic interest) | PRICE(settlement, maturity, rate, yld, redemption, freq, basis) |
| PRICEDISC | Price of discount security | PRICEDISC(settlement, maturity, discount, redemption, basis) |
| PRICEMAT | Price of bond paying at maturity | PRICEMAT(settlement, maturity, issue, rate, yld, basis) |
| PV | Present value of investment | PV(rate, nper, pmt, [fv], [type]) |
| RATE | Interest rate per period | RATE(nper, pmt, pv, [fv], [type]) |
| RECEIVED | Amount received at maturity | RECEIVED(settlement, maturity, investment, discount, basis) |
| RRI | Equivalent growth rate | RRI(nper, pv, fv) |
| SLN | Straight-line depreciation | SLN(cost, salvage, life) |
| SYD | Sum-of-years digits depreciation | SYD(cost, salvage, life, per) |
| TBILLEQ | Bond-equivalent yield for T-Bill | TBILLEQ(settlement, maturity, discount) |
| TBILLPRICE | Price per $100 T-Bill | TBILLPRICE(settlement, maturity, discount) |
| TBILLYIE | Yield for T-Bill | TBILLYIELD(settlement, maturity, pr) |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| LD | | |
| VDB | Variable declining balance depreciation | VDB(cost, salvage, life, start, end, [factor], [no_switch]) |
| XIRR | IRR for irregular cashflows | XIRR(values, dates, [guess]) |
| XNPV | NPV for irregular cashflows | XNPV(rate, values, dates) |
| YIELD | Yield of periodic-paying bond | YIELD(settlement, maturity, rate, pr, redemption, freq, basis) |
| YIELDDISC | Yield of discount security | YIELDDISC(settlement, maturity, pr, redemption, basis) |
| YIELDMAT | Yield of bond paying at maturity | YIELDMAT(settlement, maturity, issue, rate, pr, redemption, basis) |

# Microsoft Power BI:DAX functions

- **INFO functions**

| Function | Description | Syntax Example |
|---|---|---|
| INFO.VIEW.COLUMNS | Returns a list of all columns in the current model. | EVALUATE INFO.VIEW.COLUMNS() |
| INFO.VIEW.MEASURES | Returns a list of all measures in the current model. | EVALUATE INFO.VIEW.MEASURES() |
| INFO.VIEW.RELATIONSHIPS | Returns a list of all relationships in the current model. | EVALUATE INFO.VIEW.RELATIONSHIPS() |
| INFO.VIEW.TABLES | Returns a list of all tables in the current model. | EVALUATE INFO.VIEW.TABLES() |
| INFO.ALTERNATEOFDEFINITIONS | (No description provided – alternate object definitions). | EVALUATE INFO.ALTERNATEOFDEFINITIONS() |
| INFO.ANNOTATIONS | Returns list of all annotations in the model. | EVALUATE INFO.ANNOTATIONS() |
| INFO.ATTRIBUTEHIERARCHIES | DMV query for attribute hierarchies. | EVALUATE INFO.ATTRIBUTEHIERARCHIES() |
| INFO.ATTRIBUTEHIERARCHYSTORAGES | Returns attribute hierarchy storages. | EVALUATE INFO.ATTRIBUTEHIERARCHYSTORAGES() |
| INFO.CALCDEPENDENCY | Shows calculation dependency information for a DAX query. | EVALUATE INFO.CALCDEPENDENCY() |
| INFO.CALCULATIONGROUPS | Returns all calculation groups in the model. | EVALUATE INFO.CALCULATIONGROUPS() |
| INFO.CALCULATIONITEMS | Returns all calculation items in the model. | EVALUATE INFO.CALCULATIONITEMS() |
| INFO.CATALOGS | DMV query for catalogs. | EVALUATE INFO.CATALOGS() |
| INFO.CHANGEDPROPERTIES | DMV query for changed properties. | EVALUATE INFO.CHANGEDPROPERTIES() |
| INFO.COLUMNPARTITIONSTORAGES | Returns column partition storages. | EVALUATE INFO.COLUMNPARTITIONSTORAGES() |
| INFO.COLUMNPERMISSIONS | Returns all column permissions in the model. | EVALUATE INFO.COLUMNPERMISSIONS() |
| INFO.COLUMNS | Returns all columns with schema rowset. | EVALUATE INFO.COLUMNS() |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| INFO.COLUMNSTORAGES | Returns all column storages in the model. | EVALUATE INFO.COLUMNSTORAGES() |
| INFO.CSDLMETADATA | Returns metadata in XML format. | EVALUATE INFO.CSDLMETADATA() |
| INFO.CULTURES | Returns a list of all cultures in the model. | EVALUATE INFO.CULTURES() |
| INFO.DATACOVERAGEDEFINITIONS | Returns data coverage definitions. | EVALUATE INFO.DATACOVERAGEDEFINITIONS() |
| INFO.DATASOURCES | DMV query for data sources. | EVALUATE INFO.DATASOURCES() |
| INFO.DELTATABLEMETADATASTORAGES | Returns delta table metadata storages. | EVALUATE INFO.DELTATABLEMETADATASTORAGES() |
| INFO.DEPENDENCIES | Shows dependencies between calculations. | EVALUATE INFO.DEPENDENCIES() |
| INFO.DETAILROWSDEFINITIONS | Returns all detail rows definitions. | EVALUATE INFO.DETAILROWSDEFINITIONS() |
| INFO.DICTIONARYSTORAGES | Returns dictionary storages. | EVALUATE INFO.DICTIONARYSTORAGES() |
| INFO.EXCLUDEDARTIFACTS | DMV query for excluded artifacts. | EVALUATE INFO.EXCLUDEDARTIFACTS() |
| INFO.EXPRESSIONS | Returns all expressions in the model. | EVALUATE INFO.EXPRESSIONS() |
| INFO.EXTENDEDPROPERTIES | Returns extended properties. | EVALUATE INFO.EXTENDEDPROPERTIES() |
| INFO.FORMATSTRINGDEFINITIONS | Returns format string definitions. | EVALUATE INFO.FORMATSTRINGDEFINITIONS() |
| INFO.FUNCTIONS | Returns list of DAX functions. | EVALUATE INFO.FUNCTIONS() |
| INFO.GENERALSEGMENTMAPSEGMENTMETADATASTORAGES | Returns general segment metadata. | EVALUATE INFO.GENERALSEGMENTMAPSEGMENTMETADATASTORAGES() |
| INFO.GROUPBYCOLUMNS | Returns group-by columns. | EVALUATE INFO.GROUPBYCOLUMNS() |
| INFO.HIERARCHIES | DMV query for hierarchies. | EVALUATE INFO.HIERARCHIES() |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| INFO.HIERARCHYSTORAGES | Returns hierarchy storages. | EVALUATE INFO.HIERARCHYSTORAGES() |
| INFO.KPIS | Returns all KPIs in the model. | EVALUATE INFO.KPIS() |
| INFO.LEVELS | Returns all levels in the model. | EVALUATE INFO.LEVELS() |
| INFO.LINGUISTICMETADATA | DMV query for linguistic metadata. | EVALUATE INFO.LINGUISTICMETADATA() |
| INFO.MEASURES | Returns all measures in the model. | EVALUATE INFO.MEASURES() |
| INFO.MODEL | DMV query for the model itself. | EVALUATE INFO.MODEL() |
| INFO.OBJECTTRANSLATIONS | Returns object translations. | EVALUATE INFO.OBJECTTRANSLATIONS() |
| INFO.PARQUETFILESTORAGES | Returns parquet file storages. | EVALUATE INFO.PARQUETFILESTORAGES() |
| INFO.PARTITIONS | DMV query for partitions. | EVALUATE INFO.PARTITIONS() |
| INFO.PARTITIONSTORAGES | Returns partition storages. | EVALUATE INFO.PARTITIONSTORAGES() |
| INFO.PERSPECTIVECOLUMNS | Returns perspective columns. | EVALUATE INFO.PERSPECTIVECOLUMNS() |
| INFO.PERSPECTIVEHIERARCHIES | Returns perspective hierarchies. | EVALUATE INFO.PERSPECTIVEHIERARCHIES() |
| INFO.PERSPECTIVEMEASURES | Returns perspective measures. | EVALUATE INFO.PERSPECTIVEMEASURES() |
| INFO.PERSPECTIVES | Returns perspectives. | EVALUATE INFO.PERSPECTIVES() |
| INFO.PERSPECTIVETABLES | Returns perspective tables. | EVALUATE INFO.PERSPECTIVETABLES() |
| INFO.PROPERTIES | DMV query for properties. | EVALUATE INFO.PROPERTIES() |
| INFO.QUERYGROUPS | Returns query groups. | EVALUATE INFO.QUERYGROUPS() |
| INFO.REFRESHPOLICIES | Returns refresh policies. | EVALUATE INFO.REFRESHPOLICIES() |
| INFO.RELATEDCOLUMNDETAILS | Returns related column details. | EVALUATE INFO.RELATEDCOLUMNDETAILS() |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| INFO.RELATIONSHIPINDEXSTORAGES | Returns relationship index storages. | EVALUATE INFO.RELATIONSHIPINDEXSTORAGES() |
| INFO.RELATIONSHIPS | DMV query for relationships. | EVALUATE INFO.RELATIONSHIPS() |
| INFO.RELATIONSHIPSTORAGES | Returns relationship storages. | EVALUATE INFO.RELATIONSHIPSTORAGES() |
| INFO.ROLEMEMBERSHIPS | Returns role memberships. | EVALUATE INFO.ROLEMEMBERSHIPS() |
| INFO.ROLES | Returns roles in the model. | EVALUATE INFO.ROLES() |
| INFO.SEGMENTMAPSTORAGES | Returns segment map storages. | EVALUATE INFO.SEGMENTMAPSTORAGES() |
| INFO.SEGMENTSTORAGES | Returns segment storages. | EVALUATE INFO.SEGMENTSTORAGES() |
| INFO.STORAGEFILES | Returns storage files. | EVALUATE INFO.STORAGEFILES() |
| INFO.STORAGEFOLDERS | Returns storage folders. | EVALUATE INFO.STORAGEFOLDERS() |
| INFO.STORAGETABLECOLUMNS | Returns column statistics of in-memory tables. | EVALUATE INFO.STORAGETABLECOLUMNS() |
| INFO.STORAGETABLECOLUMNSEGMENTS | Returns column segment storage info. | EVALUATE INFO.STORAGETABLECOLUMNSEGMENTS() |
| INFO.STORAGETABLES | Returns in-memory table statistics. | EVALUATE INFO.STORAGETABLES() |
| INFO.TABLEPERMISSIONS | Returns table permissions. | EVALUATE INFO.TABLEPERMISSIONS() |
| INFO.TABLES | Returns list of all tables in the model. | EVALUATE INFO.TABLES() |
| INFO.TABLESTORAGES | Returns table storages. | EVALUATE INFO.TABLESTORAGES() |
| INFO.VARIATIONS | Returns variations in the model. | EVALUATE INFO.VARIATIONS() |

# Microsoft Power BI:DAX functions

- ## Information functions

| Function | Description | Syntax Example |
|---|---|---|
| COLUMNSTATISTICS | Returns a table of statistics regarding every column in every table in the model. | COLUMNSTATISTICS() |
| CONTAINS | Returns TRUE if values for all referred columns exist in those columns. | CONTAINS(Product, Product[ProductID], 1) |
| CONTAINSROW | Returns TRUE if a row of values exists in a table. | CONTAINSROW({1,2,3}, 2) |
| CONTAINSSTRING | Returns TRUE if one string contains another string. | CONTAINSSTRING("Power BI", "BI") |
| CONTAINSSTRINGEXACT | Returns TRUE if one string contains another (case-sensitive). | CONTAINSSTRINGEXACT("Power BI", "bi") |
| CUSTOMDATA | Returns the CustomData property in the connection string. | CUSTOMDATA() |
| HASONEFILTER | Returns TRUE if only one value is directly filtered. | HASONEFILTER(Sales[Region]) |
| HASONEVALUE | Returns TRUE if only one distinct value exists in context. | HASONEVALUE(Sales[CustomerID]) |
| ISAFTER | Boolean function to check if row comes after a certain value. | ISAFTER(Date[Date], DATE(2024,1,1)) |
| ISBLANK | Checks whether a value is blank. | ISBLANK(SUM(Sales[Amount])) |
| ISCROSSFILTERED | Returns TRUE if a column is being filtered due to cross-filter. | ISCROSSFILTERED(Product[Category]) |
| ISEMPTY | Returns TRUE if a table is empty. | ISEMPTY(FILTER(Sales, Sales[Amount] > 10000)) |
| ISERROR | Check if the value is an error. | ISERROR(DIVIDE(1,0)) |
| ISEVEN | Returns TRUE if the number is even. | ISEVEN(10) |
| ISFILTERED | Returns TRUE if the column is being filtered directly. | ISFILTERED(Sales[ProductID]) |
| ISINSCOPE | Returns TRUE if the column is currently in scope in the hierarchy. | ISINSCOPE(Product[Category]) |
| ISLOGICAL | Returns TRUE if value is logical (TRUE/FALSE). | ISLOGICAL(TRUE()) |
| ISNONTEXT | Returns TRUE if the value is not text. | ISNONTEXT(123) |
| ISNUMBER | Returns TRUE if the value is a number. | ISNUMBER(25) |
| ISODD | Returns TRUE if the number is odd. | ISODD(5) |
| ISONORAFTER | Boolean function like “Start At” clause. | ISONORAFTER(Date[Date], DATE(2023,12,31)) |
| ISSELECTEDME | Checks if the current measure in context is one | ISSELECTEDMEASURE([Sales |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| ASURE | of the specified. | Amount], [Profit]) |
| ISSUBTOTAL | Returns TRUE if row is a subtotal in SUMMARIZE. | ISSUBTOTAL(Sales[Region]) |
| ISTEXT | Returns TRUE if the value is text. | ISTEXT("Hello") |
| NONVISUAL | Marks a value filter in SUMMARIZECOLUMNS as non-visual. | SUMMARIZECOLUMNS(Product[ Category], NONVISUAL(Sales[Region])) |
| SELECTEDMEA SURE | Returns the measure that is currently in context. | SELECTEDMEASURE() |
| SELECTEDMEA SUREFORMATS TRING | Returns the format string of the measure in context. | SELECTEDMEASUREFORMATST RING() |
| SELECTEDMEA SURENAME | Returns the name of the measure in context. | SELECTEDMEASURENAME() |
| USERCULTURE | Returns the locale for the current user. | USERCULTURE() |
| USERNAME | Returns domain and username. | USERNAME() |
| USEROBJECTID | Returns the user’s Object ID or SID. | USEROBJECTID() |
| USERPRINCIPAL NAME | Returns the user principal name. | USERPRINCIPALNAME() |

# Microsoft Power BI:DAX functions

- **Logical functions**

| Function | Description | Syntax Example |
|---|---|---|
| AND | Checks whether both arguments are TRUE, and returns TRUE if both are TRUE. | AND( Sales[Amount] > 1000, Sales[Quantity] > 10 ) |
| BITAND | Returns a bitwise 'AND' of two numbers. | `BITAND(6, 3) → 2` |
| BITLSHIFT | Returns a number shifted left by the specified number of bits. | `BITLSHIFT(5, 2) → 20` |
| BITOR | Returns a bitwise 'OR' of two numbers. | `BITOR(6, 3) → 7` |
| BITRSHIFT | Returns a number shifted right by the specified number of bits. | `BITRSHIFT(20, 2) → 5` |
| BITXOR | Returns a bitwise 'XOR' of two numbers. | `BITXOR(6, 3) → 5` |
| COALESCE | Returns the first expression that is not BLANK. | COALESCE(Sales[Discount], 0) |
| FALSE | Returns the logical value FALSE. | FALSE |
| IF | Checks a condition, and returns one value when TRUE, otherwise another. | IF(Sales[Amount] > 1000, "High", "Low") |
| IF.EAGER | Same as IF but evaluates both TRUE & FALSE expressions (eager execution). | IF.EAGER(Sales[Amount] > 1000, "High", "Low") |
| IFERROR | Evaluates an expression and returns a specified value if error occurs. | IFERROR(DIVIDE(Sales[Amount], Sales[Quantity]), 0) |
| NOT | Changes FALSE to TRUE, or TRUE to FALSE. | NOT(Sales[Amount] > 1000) |
| OR | Returns TRUE if one of the arguments is TRUE. | OR(Sales[Amount] > 1000, Sales[Quantity] > 10) |
| SWITCH | Evaluates an expression against a list of values and returns a matching result. | #ERROR! |
| TRUE | Returns the logical value TRUE. | TRUE |

# Microsoft Power BI:DAX functions

- **Math and trig functions**

| Function | Description | Syntax Example |
|---|---|---|
| ABS | Returns the absolute value of a number. | ABS(<number>) |
| ACOS | Returns the arccosine of a number. | ACOS(<number>) |
| ACOSH | Returns the inverse hyperbolic cosine of a number. | ACOSH(<number>) |
| ACOT | Returns the arccotangent of a number. | ACOT(<number>) |
| ACOTH | Returns the inverse hyperbolic cotangent of a number. | ACOTH(<number>) |
| ASIN | Returns the arcsine of a number. | ASIN(<number>) |
| ASINH | Returns the inverse hyperbolic sine of a number. | ASINH(<number>) |
| ATAN | Returns the arctangent of a number. | ATAN(<number>) |
| ATANH | Returns the inverse hyperbolic tangent of a number. | ATANH(<number>) |
| CEILING | Rounds a number up, to the nearest integer or significance. | CEILING(<number>, <significance>) |
| CONVERT | Converts an expression from one data type to another. | CONVERT(<number>, <from_unit>, <to_unit>) |
| COS | Returns the cosine of an angle. | COS(<number>) |
| COSH | Returns the hyperbolic cosine of a number. | COSH(<number>) |
| COT | Returns the cotangent of an angle in radians. | COT(<number>) |
| COTH | Returns the hyperbolic cotangent of a number. | COTH(<number>) |
| CURRENCY | Returns value as currency data type. | CURRENCY(<number>) |
| DEGREES | Converts radians into degrees. | DEGREES(<radians>) |
| DIVIDE | Division with alternate result if divisor = 0. | DIVIDE(<numerator>, <denominator>[, <alternateResult>]) |
| EVEN | Rounds a number up to nearest even integer. | EVEN(<number>) |
| EXP | Returns e^n. | EXP(<number>) |
| FACT | Returns factorial. | FACT(<number>) |
| FLOOR | Rounds down to nearest multiple. | FLOOR(<number>, <significance>) |
| GCD | Greatest common divisor of integers. | GCD(<number1>, <number2>, …) |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| INT | Rounds down to nearest integer. | INT(<number>) |
| ISO.CEILING | Rounds up to nearest integer/multiple. | ISO.CEILING(<number>[, <significance>]) |
| LCM | Least common multiple. | LCM(<number1>, <number2>, …) |
| LN | Natural logarithm. | LN(<number>) |
| LOG | Logarithm with base. | LOG(<number>[, <base>]) |
| LOG10 | Base-10 logarithm. | LOG10(<number>) |
| MOD | Returns remainder. | MOD(<number>, <divisor>) |
| MROUND | Rounds to given multiple. | MROUND(<number>, <multiple>) |
| ODD | Rounds up to nearest odd integer. | ODD(<number>) |
| PI | Returns value of π. | PI() |
| POWER | Number raised to a power. | POWER(<number>, <power>) |
| QUOTIENT | Returns integer division result. | QUOTIENT(<numerator>, <denominator>) |
| RADIANS | Converts degrees to radians. | RADIANS(<degrees>) |
| RAND | Random number between 0–1. | RAND() |
| RANDBETWEEN | Random number in range. | RANDBETWEEN(<bottom>, <top>) |
| ROUND | Rounds to digits. | ROUND(<number>, <num_digits>) |
| ROUNDDOWN | Rounds down toward zero. | ROUNDDOWN(<number>, <num_digits>) |
| ROUNDUP | Rounds up away from zero. | ROUNDUP(<number>, <num_digits>) |
| SIGN | Returns -1, 0, or 1 depending on sign. | SIGN(<number>) |
| SIN | Returns sine of an angle. | SIN(<number>) |
| SINH | Hyperbolic sine. | SINH(<number>) |
| SQRT | Square root. | SQRT(<number>) |
| SQRTPI | √(nπ). | SQRTPI(<number>) |
| TAN | Tangent of an angle. | TAN(<number>) |
| TANH | Hyperbolic tangent. | TANH(<number>) |
| TRUNC | Truncates decimal part. | TRUNC(<number>[, <num_digits>]) |

# Microsoft Power BI:DAX functions

- **Other functions**

| Function | Description | Syntax Example |
|---|---|---|
| BLANK | Returns a blank. | BLANK() |
| ERROR | Raises an error with an error message. | ERROR("This is a custom error message") |
| EVALUATEANDLOG | Returns the value of the first argument and logs it in a DAX Evaluation Log. | EVALUATEANDLOG(SUM(Sales[Amount])) |
| TOCSV | Returns a table as a string in CSV format. | TOCSV(SUMMARIZE(Sales, Sales[Product], "Total Sales", SUM(Sales[Amount]))) |
| TOJSON | Returns a table as a string in JSON format. | TOJSON(SUMMARIZE(Sales, Sales[Product], "Total Sales", SUM(Sales[Amount]))) |

# Microsoft Power BI:DAX functions

- ## Parent and child functions

| Function | Description | Syntax Example |
|---|---|---|
| PATH | Returns a delimited text string with the identifiers of all parents of a node, starting with the top-level parent and ending with the current row. | PATH(<id_column>, <parent_id_column>) |
| PATHCONTAINS | Returns TRUE/FALSE if the specified item exists in the given path. Useful for checking if a node belongs to a particular branch. | PATHCONTAINS(<path>, <item>) |
| PATHITEM | Returns the item at the specified position in a path (1 = top parent, n = bottom child). type (optional): TEXT (default) or INTEGER. | PATHITEM(<path>, <position>[, <type>]) |
| PATHITEMREVERSE | Returns the item at the specified position counting from the end of the path. (1 = current node, 2 = parent, etc.) | PATHITEMREVERSE(<path>, <position>[, <type>]) |
| PATHLENGTH | Returns the number of items in a path (i.e., hierarchy depth). | PATHLENGTH(<path>) |

# Microsoft Power BI:DAX functions

- **Relationship functions**

| Function | Description | Syntax Example |
|---|---|---|
| CROSSFILTER | Changes the filter direction between two columns in a relationship for a specific calculation. direction can be None, OneWay, or Both. | CROSSFILTER(<columnName1>, <columnName2>, <direction>) |
| RELATED | Returns a value from a related table by following an existing many-to-one relationship. Works like a lookup. | RELATED(<columnName>) |
| RELATEDTABLE | Returns a table of related rows from another table, based on the current row context. | RELATEDTABLE(<tableName>) |
| USERELATIONSHIP | Activates an inactive relationship between two tables for the duration of a calculation. | CALCULATE(<expression>, USERELATIONSHIP(<columnName1>, <columnName2>)) |

# Microsoft Power BI:DAX functions

- **Statistical functions**

| Function | Description | Syntax |
|---|---|---|
| BETA.DIST | Returns the Beta distribution. If cumulative=TRUE → CDF, else PDF. | BETA.DIST(x, alpha, beta, cumulative [, A], [B]) |
| BETA.INV | Returns the inverse of Beta CDF. | BETA.INV(probability, alpha, beta [, A], [B]) |
| CHISQ.DIST | Returns the Chi-squared distribution (CDF or PDF). | CHISQ.DIST(x, deg_freedom, cumulative) |
| CHISQ.DIST.RT | Returns the right-tailed probability of the Chi-squared distribution. | CHISQ.DIST.RT(x, deg_freedom) |
| CHISQ.INV | Returns the inverse left-tailed Chi-squared distribution. | CHISQ.INV(probability, deg_freedom) |
| CHISQ.INV.RT | Returns the inverse right-tailed Chi-squared distribution. | CHISQ.INV.RT(probability, deg_freedom) |
| COMBIN | Returns number of combinations without repetition. | COMBIN(number, number_chosen) |
| COMBINA | Returns number of combinations with repetition. | COMBINA(number, number_chosen) |
| CONFIDENCE.NORM | Returns the confidence interval using normal distribution. | CONFIDENCE.NORM(alpha, standard_dev, size) |
| CONFIDENCE.T | Returns the confidence interval using Student's t-distribution. | CONFIDENCE.T(alpha, standard_dev, size) |
| EXPON.DIST | Returns the exponential distribution. | EXPON.DIST(x, lambda, cumulative) |
| GEOMEAN | Returns the geometric mean of a column. | GEOMEAN(<column>) |
| GEOMEANX | Evaluates expression for each row in table → returns geometric mean. | GEOMEANX(<table>, <expression>) |
| LINEST | Returns parameters (slope, intercept) of linear regression. | LINEST(y-values, x-values [, const]) |
| LINESTX | Regression across rows of a table. | LINESTX(table, y-expression, x-expression [, const]) |
| MEDIAN | Returns median of values in column. | MEDIAN(<column>) |
| MEDIANX | Returns median of expression evaluated over a table. | MEDIANX(<table>, <expression>) |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| NORM.DIST | Returns normal distribution (CDF or PDF). | NORM.DIST(x, mean, standard_dev, cumulative) |
| NORM.INV | Returns the inverse normal distribution. | NORM.INV(probability, mean, standard_dev) |
| NORM.S.DIST | Returns standard normal distribution (mean=0, std=1). | NORM.S.DIST(z, cumulative) |
| NORM.S.INV | Returns the inverse of standard normal distribution. | NORM.S.INV(probability) |
| PERCENTILE.EXC | Returns k-th percentile (exclusive method). | PERCENTILE.EXC(<column>, k) |
| PERCENTILE.INC | Returns k-th percentile (inclusive method). | PERCENTILE.INC(<column>, k) |
| PERCENTILEX.EXC | Evaluates expression for each row, then percentile (exclusive). | PERCENTILEX.EXC(<table>, <expression>, k) |
| PERCENTILEX.INC | Same as above but inclusive. | PERCENTILEX.INC(<table>, <expression>, k) |
| PERMUT | Returns number of permutations without repetition. | PERMUT(number, number_chosen) |
| POISSON.DIST | Returns Poisson distribution probability. | POISSON.DIST(x, mean, cumulative) |
| RANK.EQ | Returns the rank of a number in a list. | RANK.EQ(<number>, <column>[, <order>]) |
| RANKX | Returns the rank of a calculated expression within a table. | RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]]) |
| SAMPLE | Returns n random rows from a table. | SAMPLE(n, table [, orderBy_expression]) |
| STDEV.P | Population standard deviation. | STDEV.P(<column>) |
| STDEV.S | Sample standard deviation. | STDEV.S(<column>) |
| STDEVX.P | Population standard deviation over evaluated expression. | STDEVX.P(<table>, <expression>) |
| STDEVX.S | Sample standard deviation over evaluated expression. | STDEVX.S(<table>, <expression>) |
| T.DIST | Returns Student's t-distribution (CDF or PDF). | T.DIST(x, deg_freedom, cumulative) |
| T.DIST.2T | Returns two-tailed Student's t-distribution. | T.DIST.2T(x, deg_freedom) |
| T.DIST.RT | Returns right-tailed Student's t-distribution. | T.DIST.RT(x, deg_freedom) |
| T.INV | Returns inverse left-tailed Student's t-distribution. | T.INV(probability, deg_freedom) |

| | | |
|---|---|---|
| T.INV.2T | Returns inverse two-tailed Student's t-distribution. | T.INV.2T(probability, deg_freedom) |
| VAR.P | Returns population variance. | VAR.P(<column>) |
| VAR.S | Returns sample variance. | VAR.S(<column>) |
| VARX.P | Returns population variance over evaluated expression. | VARX.P(<table>, <expression>) |
| VARX.S | Returns sample variance over evaluated expression. | VARX.S(<table>, <expression>) |

# Microsoft Power BI:DAX functions

- **Table manipulation functions**

| Function | Description | Syntax |
|---|---|---|
| ADDCOLUMNS | Adds calculated columns to the given table. | ADDCOLUMNS(Table, ColumnName, Expression, …) |
| ADDMISSINGITEMS | Adds combinations of items from multiple columns if they don't already exist. | ADDMISSINGITEMS(Table, ColumnName, …) |
| CROSSJOIN | Returns the Cartesian product of all rows from all tables. | CROSSJOIN(Table1, Table2, …) |
| CURRENTGROUP | Returns the set of rows in the current group when used inside GROUPBY. | Used inside GROUPBY → GROUPBY(Table, GroupBy_Column, "Column", Expression, …) with CURRENTGROUP() |
| DATATABLE | Creates an inline static table with given values. | DATATABLE(ColumnName, DataType, { {Row1}, {Row2}, … }) |
| DETAILROWS | Returns the data defined in the Detail Rows Expression for a measure. | DETAILROWS([Measure]) |
| DISTINCT (column) | Returns a one-column table of distinct values from a column. | DISTINCT(ColumnName) |
| DISTINCT (table) | Removes duplicate rows from a table. | DISTINCT(Table) |
| EXCEPT | Returns rows from Table1 that do not appear in Table2. | EXCEPT(Table1, Table2) |
| FILTERS | Returns the filters applied directly on a column. | FILTERS(ColumnName) |
| GENERATE | Cartesian product of Table1 and evaluated Table2 in current row context. | GENERATE(Table1, Table2) |
| GENERATEALL | Similar to GENERATE, but keeps unmatching rows as well. | GENERATEALL(Table1, Table2) |
| GENERATESERIES | Returns a single column table containing values of an arithmetic series. | GENERATESERIES(Start, End, Increment) |
| GROUPBY | Groups data without implicit CALCULATE (lighter than SUMMARIZE). | GROUPBY(Table, GroupBy_Column, "Column", Expression, …) |
| IGNORE | Omits specific expressions from blank/null evaluation. | IGNORE(Expression) (used inside SUMMARIZECOLUMNS) |
| INTERSECT | Returns intersection rows of two | INTERSECT(Table1, Table2) |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| | tables. | |
| NATURALINNERJOIN | Performs an inner join between two tables. | NATURALINNERJOIN(Table1, Table2) |
| NATURALLEFTOUTERJOIN | Performs a left outer join between two tables. | NATURALLEFTOUTERJOIN(Table1, Table2) |
| ROLLUP | Adds subtotal/rollup rows to a grouped result. | ROLLUP(Column1, Column2, …) (inside SUMMARIZE) |
| ROLLUPADDISSUBTOTAL | Adds rollup/subtotal rows in SUMMARIZECOLUMNS. | ROLLUPADDISSUBTOTAL(Table, Column) |
| ROLLUPISSUBTOTAL | Flags subtotal rows created by ROLLUPADDISSUBTOTAL. | ROLLUPISSUBTOTAL(Column) |
| ROLLUPGROUP | Adds rollup groups inside SUMMARIZE / SUMMARIZECOLUMNS. | ROLLUPGROUP(Column1, Column2, …) |
| ROW | Returns a single row table. | ROW("Column1", Expression1, "Column2", Expression2, …) |
| SELECTCOLUMNS | Creates a table with new calculated columns. | SELECTCOLUMNS(Table, "NewColumnName", Expression, …) |
| SUBSTITUTEWITHINDEX | Returns a table with a left semi join + adds index. | SUBSTITUTEWITHINDEX(Table1, Table2, OrderBy_Column) |
| SUMMARIZE | Creates a summary table over groups. | SUMMARIZE(Table, GroupBy_Column, "Column", Expression, …) |
| SUMMARIZECOLUMNS | Creates a summary table with filters applied. | SUMMARIZECOLUMNS(GroupBy_Column, …, [Measure]) |
| Table Constructor | Creates an inline table with one or more columns. | { (Value1), (Value2), … } or { (1,"A"), (2,"B") } |
| TOPN | Returns top N rows of a table. | TOPN(N, Table, OrderBy_Expression, Order) |
| TREATAS | Applies a table as filters to unrelated columns. | TREATAS(Table, Column1, Column2, …) |
| UNION | Returns combined rows from tables. | UNION(Table1, Table2, …) |
| VALUES | Returns distinct values of a column or one-column version of a table. | VALUES(ColumnName) OR VALUES(Table) |

# Microsoft Power BI:DAX functions

- **Text functions**

| Function | Description | Example DAX Formula |
|---|---|---|
| COMBINEVALUES | Joins two or more text strings into one text string. | COMBINEVALUES("-", Sales[Region], Sales[Product]) → "East-TV" |
| CONCATENATE | Joins two text strings into one text string. | CONCATENATE("Hello ", "World") → "Hello World" |
| CONCATENATEX | Concatenates the result of an expression evaluated for each row in a table. | CONCATENATEX(VALUES(Sales[Product]), Sales[Product], ", ") → "TV, Laptop, Mobile" |
| EXACT | Compares two text strings and returns TRUE if they are exactly the same. | EXACT("PowerBI", "powerbi") → FALSE |
| FIND | Returns the starting position of one text string within another. | FIND("BI", "PowerBI", 1, -1) → 6 |
| FIXED | Rounds a number to the specified decimals and returns text. | FIXED(1234.567, 2) → "1,234.57" |
| FORMAT | Converts a value to text in the specified format. | FORMAT(TODAY(), "DD-MMM-YYYY") → "17-Aug-2025" |
| LEFT | Returns characters from the start of a string. | LEFT("PowerBI", 5) → "Power" |
| LEN | Returns the number of characters in a text string. | LEN("Power BI") → 8 |
| LOWER | Converts text to lowercase. | LOWER("POWER BI") → "power bi" |
| MID | Returns characters from the middle of a string. | MID("PowerBI", 2, 3) → "owe" |
| REPLACE | Replaces part of a string with another string. | REPLACE("PowerBI", 1, 5, "Excel") → "ExcelBI" |
| REPT | Repeats text a given number of times. | REPT("BI ", 3) → "BI BI BI " |
| RIGHT | Returns characters from the end of a string. | RIGHT("PowerBI", 2) → "BI" |
| SEARCH | Returns the position of a substring, case-insensitive. | SEARCH("bi", "PowerBI", 1, -1) → 6 |
| SUBSTITUTE | Replaces existing text with new text. | SUBSTITUTE("Power-BI", "-", " ") → "Power BI" |
| TRIM | Removes all extra spaces from text. | TRIM(" Power BI ") → "Power BI" |
| UNICHAR | Returns a Unicode character from its numeric value. | UNICHAR(9733) → "★" |
| UNICODE | Returns the numeric Unicode of the first character. | UNICODE("★") → 9733 |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| UPPER | Converts text to uppercase. | UPPER("power bi") → "POWER BI" |
| VALUE | Converts a text string to a number. | VALUE("123.45") → 123.45 |

# Microsoft Power BI:DAX functions

- **Time intelligence functions**

| Function | Description | Example DAX Formula |
|---|---|---|
| CLOSINGBALANCEMONTH | Evaluates expression at the last date of the month. | CLOSINGBALANCEMONTH(SUM(Sales[Amount]), 'Date'[Date]) |
| CLOSINGBALANCEQUARTER | Evaluates expression at the last date of the quarter. | CLOSINGBALANCEQUARTER(SUM(Sales[Amount]), 'Date'[Date]) |
| CLOSINGBALANCEYEAR | Evaluates expression at the last date of the year. | CLOSINGBALANCEYEAR(SUM(Sales[Amount]), 'Date'[Date]) |
| DATEADD | Shifts dates forward/backward by specified intervals. | CALCULATE(SUM(Sales[Amount]), DATEADD('Date'[Date], -1, MONTH)) |
| DATESBETWEEN | Returns dates between a start and end date. | CALCULATE(SUM(Sales[Amount]), DATESBETWEEN('Date'[Date], DATE(2024,1,1), DATE(2024,12,31))) |
| DATESINPERIOD | Returns dates starting at a date for a given interval. | CALCULATE(SUM(Sales[Amount]), DATESINPERIOD('Date'[Date], MAX('Date'[Date]), -3, MONTH)) |
| DATESMTD | Returns month-to-date dates. | CALCULATE(SUM(Sales[Amount]), DATESMTD('Date'[Date])) |
| DATESQTD | Returns quarter-to-date dates. | CALCULATE(SUM(Sales[Amount]), DATESQTD('Date'[Date])) |
| DATESYTD | Returns year-to-date dates. | CALCULATE(SUM(Sales[Amount]), DATESYTD('Date'[Date])) |
| ENDOFMONTH | Returns last date of month in context. | ENDOFMONTH('Date'[Date]) |
| ENDOFQUARTER | Returns last date of quarter in context. | ENDOFQUARTER('Date'[Date]) |
| ENDOFYEAR | Returns last date of year in context. | ENDOFYEAR('Date'[Date]) |
| FIRSTDATE | Returns first date in context. | FIRSTDATE('Date'[Date]) |
| LASTDATE | Returns last date in context. | LASTDATE('Date'[Date]) |
| NEXTDAY | Returns next day from current context. | CALCULATE(SUM(Sales[Amount]), NEXTDAY('Date'[Date])) |
| NEXTMONTH | Returns all dates in next month. | CALCULATE(SUM(Sales[Amount]), NEXTMONTH('Date'[Date])) |
| NEXTQUARTER | Returns all dates in next quarter. | CALCULATE(SUM(Sales[Amount]), NEXTQUARTER('Date'[Date])) |
| NEXTYEAR | Returns all dates in next year. | CALCULATE(SUM(Sales[Amount]), NEXTYEAR('Date'[Date])) |
| OPENINGBALANCEMONTH | Evaluates expression at first date of month. | OPENINGBALANCEMONTH(SUM(Sales[Amount]), 'Date'[Date]) |

# Microsoft Power BI:DAX functions

| | | |
|---|---|---|
| OPENINGBALANCEQUARTER | Evaluates expression at first date of quarter. | OPENINGBALANCEQUARTER(SUM(Sales[Amount]), 'Date'[Date]) |
| OPENINGBALANCEYEAR | Evaluates expression at first date of year. | OPENINGBALANCEYEAR(SUM(Sales[Amount]), 'Date'[Date]) |
| PARALLELPERIOD | Returns parallel period shifted by intervals. | CALCULATE(SUM(Sales[Amount]), PARALLELPERIOD('Date'[Date], -1, YEAR)) |
| PREVIOUSDAY | Returns previous day. | CALCULATE(SUM(Sales[Amount]), PREVIOUSDAY('Date'[Date])) |
| PREVIOUSMONTH | Returns all dates in previous month. | CALCULATE(SUM(Sales[Amount]), PREVIOUSMONTH('Date'[Date])) |
| PREVIOUSQUARTER | Returns all dates in previous quarter. | CALCULATE(SUM(Sales[Amount]), PREVIOUSQUARTER('Date'[Date])) |
| PREVIOUSYEAR | Returns all dates in previous year. | CALCULATE(SUM(Sales[Amount]), PREVIOUSYEAR('Date'[Date])) |
| SAMEPERIODLASTYEAR | Returns same period one year back. | CALCULATE(SUM(Sales[Amount]), SAMEPERIODLASTYEAR('Date'[Date])) |
| STARTOFMONTH | Returns first date of month. | STARTOFMONTH('Date'[Date]) |
| STARTOFQUARTER | Returns first date of quarter. | STARTOFQUARTER('Date'[Date]) |
| STARTOFYEAR | Returns first date of year. | STARTOFYEAR('Date'[Date]) |
| TOTALMTD | Returns month-to-date calculation. | TOTALMTD(SUM(Sales[Amount]), 'Date'[Date]) |
| TOTALQTD | Returns quarter-to-date calculation. | TOTALQTD(SUM(Sales[Amount]), 'Date'[Date]) |
| TOTALYTD | Returns year-to-date calculation. | TOTALYTD(SUM(Sales[Amount]), 'Date'[Date]) |