

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Filter	ADDMISSINGITEMS	ADDMISSINGITEMS(<showAllColumn>[, <showAllColumn>]..., <table>[, <groupingColumn>[, <groupingColumn>]...[, filterTable]...)	Adds combinations of items from multiple columns to a table if they do not already exist. The determination of which item combinations to add is based on referencing source columns which contain all the possible values for the columns.	
Filter	ALL	ALL({<table> <column>[, <column>[, <column>[,...]]]})	Returns all the rows in a table, or all the values in a column, ignoring any filters that might have been applied. This function is useful for clearing filters and creating calculations on all the rows in a table.	
Filter	ALLEXCEPT	ALLEXCEPT(<table>,<column>[,<column>[,...]])	Removes all context filters in the table except filters that have been applied to the specified columns.	
Filter	ALLNOBLANKROW	ALLNOBLANKROW({<table> <column>[, <column>[, <column>[,...]]]})	From the parent table of a relationship, returns all rows but the blank row, or all distinct values of a column but the blank row, and disregards any context filters that might exist.	
Filter	CROSSFILTER	CROSSFILTER(<columnName1>, <columnName2>, <direction>)	Specifies the cross-filtering direction to be used in a calculation for a relationship that exists between two columns.	The function returns no value; the function only sets the cross-filtering direction for the indicated relationship, for the duration of the query.
Filter	DISTINCT	DISTINCT(<column>)	Returns a one-column table that contains the distinct values from the specified column. In other words, duplicate values are removed and only unique values are returned.	
Filter	FILTERS	FILTERS(<columnName>)	Returns the values that are directly applied as filters to columnName.	The values that are directly applied as filters to columnName.
Filter	HASONEFILTER	HASONEFILTER(<columnName>)	Returns TRUE when the number of directly filtered values on columnName is one; otherwise returns FALSE.	TRUE when the number of directly filtered values on columnName is one; otherwise returns FALSE.
Filter	HASONEVALUE	HASONEVALUE(<columnName>)	Returns TRUE when the context for columnName has been filtered down to one distinct value only. Otherwise is FALSE.	TRUE when the context for columnName has been filtered down to one distinct value only. Otherwise is FALSE.
Filter	ISCROSSFILTERED	ISCROSSFILTERED(<columnName>)	Returns TRUE when columnName or another column in the same or related table is being filtered.	TRUE when columnName or another column in the same or related table is being filtered. Otherwise returns FALSE.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Filter	ISFILTERED	ISFILTERED(<columnName>)	Returns TRUE when columnName is being filtered directly. If there is no filter on the column or if the filtering happens because a different column in the same table or in a related table is being filtered then the function returns FALSE.	TRUE when columnName is being filtered directly.
Filter	USERELATIONSHIP	USERELATIONSHIP(<columnName1>,<columnName2>)	Specifies the relationship to be used in a specific calculation as the one that exists between columnName1 and columnName2.	
Filter	VALUES	VALUES(<TableNameOrColumnName>)	Returns a one-column table that contains the distinct values from the specified table or column. In other words, duplicate values are removed and only unique values are returned.	
Information	CUSTOMDATA	CUSTOMDATA()	Returns the content of the CustomData property in the connection string.	
Information	USERNAME	USERNAME()	Returns the domain name and username from the credentials given to the system at connection time	
Logical	SWITCH	SWITCH(<expression>, <value>, <result>[, <value>, <result>]...[, <else>])	Evaluates an expression against a list of values and returns one of multiple possible result expressions.	resultAny scalar expression to be evaluated if the results of expression match the corresponding value.
Math and Trig	ACOSH	ACOSH(number)	Returns the inverse hyperbolic cosine of a number. The number must be greater than or equal to 1. The inverse hyperbolic cosine is the value whose hyperbolic cosine is number, so ACOSH(COSH(number)) equals number.	Returns the inverse hyperbolic cosine of a number.
Math and Trig	ASINH	ASINH(number)	Returns the inverse hyperbolic sine of a number. The inverse hyperbolic sine is the value whose hyperbolic sine is number, so ASINH(SINH(number)) equals number.	Returns the inverse hyperbolic sine of a number.
Math and Trig	ATANH	ATANH(number)	Returns the inverse hyperbolic tangent of a number. Number must be between -1 and 1 (excluding -1 and 1). The inverse hyperbolic tangent is the value whose hyperbolic tangent is number, so ATANH(TANH(number)) equals number.	Returns the inverse hyperbolic tangent of a number.
Math and Trig	CURRENCY	CURRENCY(<value>)	Evaluates the argument and returns the result as currency data type.	The value of the expression evaluated and returned as a currency type value.
Math and Trig	DEGREES	DEGREES(angle)	Converts radians into degrees.	

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Math and Trig	EXP	EXP(<number>)	Returns e raised to the power of a given number. The constant e equals 2.71828182845904, the base of the natural logarithm.	A decimal number.
Math and Trig	PRODUCT	PRODUCT(<column>)	Returns the product of the numbers in a column.	
Math and Trig	PRODUCTX	PRODUCTX(<table>, <expression>)	Returns the product of an expression evaluated for each row in a table.	
Math and Trig	RADIANS	RADIANS(angle)	Converts degrees to radians.	
Math and Trig	SIGN	SIGN(<number>)	Determines the sign of a number, the result of a calculation, or a value in a column. The function returns 1 if the number is positive, 0 (zero) if the number is zero, or -1 if the number is negative.	A whole number. The possible return values are 1, 0, and -1.
Other	INTERSECT	INTERSECT(<table_expression1>, <table_expression2>)	Returns the row intersection of two tables, retaining duplicates.	A table that contains all the rows in table_expression1 that are also in table_expression2
Other	VAR (DAX)			
Parent and Child	PATH	PATH(<ID_columnName>, <parent_columnName>)	Returns a delimited text string with the identifiers of all the parents of the current identifier, starting with the oldest and continuing until current.	
Parent and Child	PATHCONTAINS	PATHCONTAINS(<path>, <item>)	Returns TRUE if the specified item exists within the specified path.	
Parent and Child	PATHITEM	PATHITEM(<path>, <position>[, <type>])	Returns the item at the specified position from a string resulting from evaluation of a PATH function. Positions are counted from left to right.	type(Optional)An enumeration that defines the data type of the result:
Parent and Child	PATHITEMREVERSE	PATHITEMREVERSE(<path>, <position>[, <type>])	Returns the item at the specified position from a string resulting from evaluation of a PATH function. Positions are counted backwards from right to left.	type(Optional) An enumeration that defines the data type of the result:
Statistical	BETA.INV	BETA.INV(probability,alpha,beta,[A],[B])	Returns the inverse of the beta cumulative probability density function (BETA.DIST).	
Statistical	CHISQ.INV	CHISQ.INV(probability,deg_freedom)	Returns the inverse of the left-tailed probability of the chi-squared distribution.	
Statistical	CHISQ.INV.RT	CHISQ.INV.RT(probability,deg_freedom)	Returns the inverse of the right-tailed probability of the chi-squared distribution.	

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Statistical	CROSSJOIN	CROSSJOIN(<table>, <table>[, <table>]...)	Returns a table that contains the Cartesian product of all rows from all tables in the arguments. The columns in the new table are all the columns in all the argument tables.	A table that contains the Cartesian product of all rows from all tables in the arguments.
Statistical	GENERATE	GENERATE(<table1>, <table2>)	Returns a table with the Cartesian product between each row in table1 and the table that results from evaluating table2 in the context of the current row from table1.	
Statistical	GENERATEALL	GENERATEALL(<table1>, <table2>)	Returns a table with the Cartesian product between each row in table1 and the table that results from evaluating table2 in the context of the current row from table1.	
Statistical	GEOMEAN	GEOMEAN(<column>)	Returns the geometric mean of the numbers in a column.	
Statistical	GEOMEANX	GEOMEANX(<table>, <expression>)	Returns the geometric mean of an expression evaluated for each row in a table.	
Statistical	MAX	MAX(<column>)	Returns the largest numeric value in a column, or between two scalar expressions.	
Statistical	MEDIAN	MEDIAN(<column>)	Returns the median of numbers in a column.	
Statistical	MEDIANX	MEDIANX(<table>, <expression>)	Returns the median number of an expression evaluated for each row in a table.	
Statistical	MIN	MIN(<column>)	Returns the smallest numeric value in a column, or between two scalar expressions. Ignores logical values and text.	
Statistical	NORM.DIST	NORM.DIST(X, Mean, Standard_dev, Cumulative)	Returns the normal distribution for the specified mean and standard deviation.	The normal distribution for the specified mean and standard deviation.
Statistical	NORM.INV	NORM.INV(Probability, Mean, Standard_dev)	The inverse of the normal cumulative distribution for the specified mean and standard deviation.	Returns the inverse of the normal cumulative distribution for the specified mean and standard deviation.
Statistical	NORM.S.DIST	NORM.S.DIST(Z, Cumulative)	Returns the standard normal distribution (has a mean of zero and a standard deviation of one).	The standard normal distribution (has a mean of zero and a standard deviation of one).
Statistical	NORM.S.INV (DAX)	NORM.S.INV(Probability)	Returns the inverse of the standard normal cumulative distribution. The distribution has a mean of zero and a standard deviation of one.	The inverse of the standard normal cumulative distribution. The distribution has a mean of zero and a standard deviation of one.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Statistical	PERCENTILE.EXC	PERCENTILE.EXC(<column>, <k>)	Returns the k-th percentile of values in a range, where k is in the range 0..1, exclusive.	
Statistical	PERCENTILE.INC	PERCENTILE.INC(<column>, <k>)	Returns the k-th percentile of values in a range, where k is in the range 0..1, inclusive.	
Statistical	PERCENTILEX.EXC	PERCENTILEX.EXC(<table>, <expression>, k)	Returns the percentile number of an expression evaluated for each row in a table.	
Statistical	PERCENTILEX.INC	PERCENTILEX.INC(<table>, <expression>;, k)	Returns the percentile number of an expression evaluated for each row in a table.	
Statistical	RANK.EQ	RANK.EQ(<value>, <columnName>[, <order>])	Returns the ranking of a number in a list of numbers.	order(Optional) A value that specifies how to rank number, low to high or high to low:
Statistical	ROW	ROW(<name>, <expression>[[, <name>, <expression>]...])	Returns a table with a single row containing values that result from the expressions given to each column.	
Statistical	SAMPLE	SAMPLE(<n_value>, <table>, <orderBy_expression>, [<order>], <orderBy_expression>, [<order>]...])	Returns a sample of N rows from the specified table.	orderBy_expression(Optional) Any scalar DAX expression where the result value is evaluated for each row of table.
Statistical	STDEV.P	STDEV.P(<ColumnName>)	Returns the standard deviation of the entire population.	A number representing the standard deviation of the entire population.
Statistical	STDEV.P	=STDEV.P(InternetSales_USD[SalesAmount_USD])	and n is the population size	
Statistical	STDEV.S	STDEV.S(<ColumnName>)	Returns the standard deviation of a sample population.	A number that represents the standard deviation of a sample population.
Statistical	STDEV.S	=STDEV.S(InternetSales_USD[SalesAmount_USD])	and n is the population size	
Statistical	STDEVX.P	STDEVX.P(<table>, <expression>)	Returns the standard deviation of the entire population.	
Statistical	STDEVX.P	=STDEVX.P(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD] – (InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))	and n is the population size	
Statistical	STDEVX.S	STDEVX.S(<table>, <expression>)	Returns the standard deviation of a sample population.	

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Statistical	STDEVX.S	=STDEVX.S(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD] – (InternetSales_USD[DiscountAmount_USD]/Inter netSales_USD[OrderQuantity]))	and n is the population size	
Statistical	SUMMARIZE	SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, <name>, <expression>]...)	Returns a summary table for the requested totals over a set of groups.	nameThe name given to a total or summarize column, enclosed in double quotes.
Statistical	T.DIST	T.DIST(X,Deg_freedom,Cumulative)	Returns the Student's left-tailed t-distribution.	The Student's left-tailed t-distribution.
Statistical	T.DIST.2T	T.DIST.2T(X,Deg_freedom)	Returns the two-tailed Student's t-distribution.	The two-tailed Student's t-distribution.
Statistical	T.DIST.RT	T.DIST.RT(X,Deg_freedom)	Returns the right-tailed Student's t-distribution.	The right-tailed Student's t-distribution.
Statistical	T.INV	T.INV(Probability,Deg_freedom)	Returns the left-tailed inverse of the Student's t-distribution.	The left-tailed inverse of the Student's t- distribution.
Statistical	T.INV.2t	T.INV.2T(Probability,Deg_freedom)	Returns the two-tailed inverse of the Student's t-distribution.	The two-tailed inverse of the Student's t- distribution.
Statistical	VAR.P	VAR.P(<columnName>)	Returns the variance of the entire population.	A number with the variance of the entire population.
Statistical	VAR.P	=VAR.P(InternetSales_USD[SalesAmount_USD])	and n is the population size	
Statistical	VAR.S	VAR.S(<columnName>)	Returns the variance of a sample population.	A number with the variance of a sample population.
Statistical	VAR.S	=VAR.S(InternetSales_USD[SalesAmount_USD])	and n is the population size	
Statistical	VARX.P	VARX.P(<table>, <expression>)	Returns the variance of the entire population.	
Statistical	VARX.P	=VARX.P(InternetSales_USD, InternetSales_USD[UnitPrice_USD] –(InternetSales_USD[DiscountAmount_USD]/Inte rnetSales_USD[OrderQuantity]))	and n is the population size	
Statistical	VARX.S	VARX.S(<table>, <expression>)	Returns the variance of a sample population.	

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Statistical	VARX.S	=VARX.S(InternetSales_USD, InternetSales_USD[UnitPrice_USD] – (InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))	and n is the population size	
Text	BLANK	BLANK()	Returns a blank.	Blanks are not equivalent to nulls. DAX uses blanks for both database nulls and for blank cells in Excel. For more information, see Data Types Supported (SSAS Tabular).
Text	CODE	CODE(text)	Returns a numeric code for the first character in a text string. The returned code corresponds to the character set used by your computer.	
Text	FORMAT	FORMAT(<value>, <format_string>)	Converts a value to text according to the specified format.	A string containing value formatted as defined by format_string.
Text	RIGHT	RIGHT(<text>, <num_chars>)	RIGHT returns the last character or characters in a text string, based on the number of characters you specify.	
Text	SEARCH	SEARCH(<find_text>, <within_text>[, <start_num>][, <NotFoundValue>])	Returns the number of the character at which a specific character or text string is first found, reading left to right. Search is case-insensitive and accent sensitive.	The number of the starting position of the first text string from the first character of the second text string.
Date and Time	CALENDAR	CALENDAR(<start_date>, <end_date>)	Returns a table with a single column named “Date” that contains a contiguous set of dates. The range of dates is from the specified start date to the specified end date, inclusive of those two dates.	Returns a table with a single column named “Date” containing a contiguous set of dates. The range of dates is from the specified start date to the specified end date, inclusive of those two dates.
Date and Time	CALENDARAUTO	CALENDARAUTO([fiscal_year_end_month])	Returns a table with a single column named “Date” that contains a contiguous set of dates. The range of dates is calculated automatically based on data in the model.	Returns a table with a single column named “Date” that contains a contiguous set of dates. The range of dates is calculated automatically based on data in the model.
Date and Time	DATE	DATE(<year>, <month>, <day>)	Returns the specified date in datetime format.	Returns the specified date (datetime).
Date and Time	DATEDIFF	DATEDIFF(<start_date>, <end_date>, <interval>)	Returns the count of interval boundaries crossed between two dates.	The count of interval boundaries crossed between two dates.
Date and Time	DATEVALUE	DATEVALUE(date_text)	Converts a date in the form of text to a date in datetime format.	A date in datetime format.
Date and Time	DAY	DAY(<date>)	Returns the day of the month, a number from 1 to 31.	An integer number indicating the day of the month.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Date and Time	EDATE	EDATE(<start_date>, <months>)	Returns the date that is the indicated number of months before or after the start date. Use EDATE to calculate maturity dates or due dates that fall on the same day of the month as the date of issue.	A date (datetime).
Date and Time	EOMONTH	EOMONTH(<start_date>, <months>)	Returns the date in datetime format of the last day of the month, before or after a specified number of months. Use EOMONTH to calculate maturity dates or due dates that fall on the last day of the month.	A date (datetime).
Date and Time	HOUR	HOUR(<datetime>)	Returns the hour as a number from 0 (12:00 A.M.) to 23 (11:00 P.M.).	An integer number from 0 to 23.
Date and Time	MINUTE	MINUTE(<datetime>)	Returns the minute as a number from 0 to 59, given a date and time value.	An integer number from 0 to 59.
Date and Time	MONTH	MONTH(<datetime>)	Returns the month as a number from 1 (January) to 12 (December).	An integer number from 1 to 12.
Date and Time	NOW	NOW()	Returns the current date and time in datetime format.	
Date and Time	SECOND	SECOND(<time>)	Returns the seconds of a time value, as a number from 0 to 59.	An integer number from 0 to 59.
Date and Time	TIME	TIME(hour, minute, second)	Converts hours, minutes, and seconds given as numbers to a time in datetime format.	A time (datetime).
Date and Time	TIMEVALUE	TIMEVALUE(time_text)	Converts a time in text format to a time in datetime format.	A date (datetime).
Date and Time	TODAY	TODAY()	Returns the current date.	The TODAY function is useful when you need to have the current date displayed on a worksheet, regardless of when you open the workbook. It is also useful for calculating intervals.
Date and Time	WEEKDAY	WEEKDAY(<date>, <return_type>)	Returns a number from 1 to 7 identifying the day of the week of a date. By default the day ranges from 1 (Sunday) to 7 (Saturday).	An integer number from 1 to 7.
Date and Time	WEEKNUM	WEEKNUM(<date>, <return_type>)	Returns the week number for the given date and year according to the return_type value. The week number indicates where the week falls numerically within a year.	An integer number.
Date and Time	YEAR	YEAR(<date>)	Returns the year of a date as a four digit integer in the range 1900-9999.	An integer in the range 1900-9999.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Date and Time	YEARFRAC	YEARFRAC(<start_date>, <end_date>, <basis>)	Calculates the fraction of the year represented by the number of whole days between two dates. Use the YEARFRAC worksheet function to identify the proportion of a whole year's benefits or obligations to assign to a specific term.	A decimal number. The internal data type is a signed IEEE 64-bit (8-byte) double-precision floating-point number.
Time Intelligence	CLOSINGBALANCEMONTH	CLOSINGBALANCEMONTH(<expression>,<dates>[,<filter>])	Evaluates the expression at the last date of the month in the current context.	A scalar value that represents the expression evaluated at the last date of the month in the current context.
Time Intelligence	CLOSINGBALANCEQUARTER	CLOSINGBALANCEQUARTER(<expression>,<dates>[,<filter>])	Evaluates the expression at the last date of the quarter in the current context.	A scalar value that represents the expression evaluated at the last date of the quarter in the current context.
Time Intelligence	CLOSINGBALANCEYEAR	CLOSINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])	Evaluates the expression at the last date of the year in the current context.	A scalar value that represents the expression evaluated at the last date of the year in the current context.
Time Intelligence	DATEADD	DATEADD(<dates>,<number_of_intervals>,<interval>)	Returns a table that contains a column of dates, shifted either forward or backward in time by the specified number of intervals from the dates in the current context.	A table containing a single column of date values.
Time Intelligence	DATESBETWEEN	DATESBETWEEN(<dates>,<start_date>,<end_date>)	Returns a table that contains a column of dates that begins with the start_date and continues until the end_date.	A table containing a single column of date values.
Time Intelligence	DATESINPERIOD	DATESINPERIOD(<dates>,<start_date>,<number_of_intervals>,<interval>)	Returns a table that contains a column of dates that begins with the start_date and continues for the specified number_of_intervals.	A table containing a single column of date values.
Time Intelligence	DATESMTD	DATESMTD(<dates>)	Returns a table that contains a column of the dates for the month to date, in the current context.	A table containing a single column of date values.
Time Intelligence	DATESQTD	DATESQTD(<dates>)	Returns a table that contains a column of the dates for the quarter to date, in the current context.	A table containing a single column of date values.
Time Intelligence	DATESYTD	DATESYTD(<dates> [,<year_end_date>])	Returns a table that contains a column of the dates for the year to date, in the current context.	A table containing a single column of date values.
Time Intelligence	ENDOFMONTH	ENDOFMONTH(<dates>)	Returns the last date of the month in the current context for the specified column of dates.	A table containing a single column and single row with a date value.
Time Intelligence	ENDOFQUARTER	ENDOFQUARTER(<dates>)	Returns the last date of the quarter in the current context for the specified column of dates.	A table containing a single column and single row with a date value.
Time Intelligence	ENDOFYEAR	ENDOFYEAR(<dates> [,<year_end_date>])	Returns the last date of the year in the current context for the specified column of dates.	A table containing a single column and single row with a date value.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Time Intelligence	FIRSTDATE	FIRSTDATE(<dates>)	Returns the first date in the current context for the specified column of dates.	A table containing a single column and single row with a date value.
Time Intelligence	FIRSTNONBLANK	FIRSTNONBLANK(<column>,<expression>)	Returns the first value in the column, column, filtered by the current context, where the expression is not blank.	A table containing a single column and single row with the computed first value.
Time Intelligence	LASTDATE	LASTDATE(<dates>)	Returns the last date in the current context for the specified column of dates.	A table containing a single column and single row with a date value.
Time Intelligence	LASTNONBLANK	LASTNONBLANK(<column>,<expression>)	Returns the last value in the column, column, filtered by the current context, where the expression is not blank.	A table containing a single column and single row with the computed last value.
Time Intelligence	NEXTDAY	NEXTDAY(<dates>)	Returns a table that contains a column of all dates from the next day, based on the first date specified in the dates column in the current context.	A table containing a single column of date values.
Time Intelligence	NEXTMONTH	NEXTMONTH(<dates>)	Returns a table that contains a column of all dates from the next month, based on the first date in the dates column in the current context.	A table containing a single column of date values.
Time Intelligence	NEXTQUARTER	NEXTQUARTER(<dates>)	Returns a table that contains a column of all dates in the next quarter, based on the first date specified in the dates column, in the current context.	A table containing a single column of date values.
Time Intelligence	NEXTYEAR	NEXTYEAR(<dates>[,<year_end_date>])	Returns a table that contains a column of all dates in the next year, based on the first date in the dates column, in the current context.	A table containing a single column of date values.
Time Intelligence	OPENINGBALANCEMONTH	OPENINGBALANCEMONTH(<expression>,<dates>[,<filter>])	Evaluates the expression at the first date of the month in the current context.	A scalar value that represents the expression evaluated at the first date of the month in the current context.
Time Intelligence	OPENINGBALANCEQUARTER	OPENINGBALANCEQUARTER(<expression>,<dates>[,<filter>])	Evaluates the expression at the first date of the quarter, in the current context.	A scalar value that represents the expression evaluated at the first date of the quarter in the current context.
Time Intelligence	OPENINGBALANCEYEAR	OPENINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])	Evaluates the expression at the first date of the year in the current context.	A scalar value that represents the expression evaluated at the first date of the year in the current context.
Time Intelligence	PARALLELPERIOD	PARALLELPERIOD(<dates>,<number_of_intervals>,<interval>)	Returns a table that contains a column of dates that represents a period parallel to the dates in the specified dates column, in the current context, with the dates shifted a number of intervals either forward in time or back in time.	A table containing a single column of date values.
Time Intelligence	PREVIOUSDAY	PREVIOUSDAY(<dates>)	Returns a table that contains a column of all dates representing the day that is previous to the first date in the dates column, in the current context.	A table containing a single column of date values.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Time Intelligence	PREVIOUSMONTH	PREVIOUSMONTH(<dates>)	Returns a table that contains a column of all dates from the previous month, based on the first date in the dates column, in the current context.	A table containing a single column of date values.
Time Intelligence	PREVIOUSQUARTER	PREVIOUSQUARTER(<dates>)	Returns a table that contains a column of all dates from the previous quarter, based on the first date in the dates column, in the current context.	A table containing a single column of date values.
Time Intelligence	PREVIOUSYEAR	PREVIOUSYEAR(<dates>[,<year_end_date>])	Returns a table that contains a column of all dates from the previous year, given the last date in the dates column, in the current context.	A table containing a single column of date values.
Time Intelligence	SAMEPERIODLASTYEAR	SAMEPERIODLASTYEAR(<dates>)	Returns a table that contains a column of dates shifted one year back in time from the dates in the specified dates column, in the current context.	A single-column table of date values.
Time Intelligence	STARTOFMONTH	STARTOFMONTH(<dates>)	Returns the first date of the month in the current context for the specified column of dates.	A table containing a single column and single row with a date value.
Time Intelligence	STARTOFQUARTER	STARTOFQUARTER(<dates>)	Returns the first date of the quarter in the current context for the specified column of dates.	A table containing a single column and single row with a date value.
Time Intelligence	STARTOFYEAR	STARTOFYEAR(<dates>)	Returns the first date of the year in the current context for the specified column of dates.	A table containing a single column and single row with a date value.
Time Intelligence	TOTALMTD	TOTALMTD(<expression>,<dates>[,<filter>])	Evaluates the value of the expression for the month to date, in the current context.	A scalar value that represents the expression evaluated for the dates in the current month-to-date, given the dates in dates.
Time Intelligence	TOTALQTD	TOTALQTD(<expression>,<dates>[,<filter>])	Evaluates the value of the expression for the dates in the quarter to date, in the current context.	A scalar value that represents the expression evaluated for all dates in the current quarter to date, given the dates in dates.
Time Intelligence	TOTALYTD	TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])	Evaluates the year-to-date value of the expression in the current context.	A scalar value that represents the expression evaluated for the current year-to-date dates.
Filter	ALLSELECTED	ALLSELECTED([<tableName> <columnName>])	Removes context filters from columns and rows in the current query, while retaining all other context filters or explicit filters.	columnNameThe name of an existing column using standard DAX syntax, usually fully qualified. It cannot be an expression. This parameter is optional.
Filter	CALCULATE	CALCULATE(<expression>,<filter1>,<filter2>...)	Evaluates an expression in a context that is modified by the specified filters.	The following restrictions apply to Boolean expressions that are used as arguments:

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Filter	CALCULATETABLE	CALCULATETABLE(<expression>,<filter1>,<filter2>,...)	Evaluates a table expression in a context modified by the given filters.	The following restrictions apply to Boolean expressions that are used as arguments:
Filter	EARLIER	EARLIER(<column>, <number>)	Returns the current value of the specified column in an outer evaluation pass of the mentioned column.	
Filter	EARLIEST	EARLIEST(<column>)	Returns the current value of the specified column in an outer evaluation pass of the specified column.	A column with filters removed.
Filter	FILTER	FILTER(<table>,<filter>)	Returns a table that represents a subset of another table or expression.	A table containing only the filtered rows.
Filter	KEEPFILTERS	KEEPFILTERS(<expression>)	Modifies how filters are applied while evaluating a CALCULATE or CALCULATETABLE function.	No return value.
Filter	RELATED	RELATED(<column>)	Returns a related value from another table.	A single value that is related to the current row.
Filter	RELATEDTABLE	RELATEDTABLE(<tableName>)	Evaluates a table expression in a context modified by the given filters.	A table of values.
Filter	SELECTEDVALUE	SELECTEDVALUE(<columnName>[, <alternateResult>])	Returns the value when the context for columnName has been filtered down to one distinct value only. Otherwise returns alternateResult.	The value when the context for columnName has been filtered down to one distinct value only. Else, alternateResult.
Filter	SUBSTITUTEWITHINDEX	SUBSTITUTEWITHINDEX(<table>, <indexColumnName>, <indexColumnsTable>, [<orderBy_expression>, [<order>]], [<orderBy_expression>, [<order>]]...)	Returns a table which represents a left semijoin of the two tables supplied as arguments. The semijoin is performed by using common columns, determined by common column names and common data type . The columns being joined on are replaced with a single column in the returned table which is of type integer and contains an index. The index is a reference into the right join table given a specified sort order.	
Information	CONTAINS	CONTAINS(<table>, <columnName>, <value>[, <columnName>, <value>]...)	Returns true if values for all referred columns exist, or are contained, in those columns; otherwise, the function returns false.	valueAny DAX expression that returns a single scalar value, that is to be sought in columnName. The expression is to be evaluated exactly once and before it is passed to the argument list.
Information	ISBLANK	ISBLANK(<value>)	Checks whether a value is blank, and returns TRUE or FALSE.	A Boolean value of TRUE if the value is blank; otherwise FALSE.
Information	ISERROR	ISERROR(<value>)	Checks whether a value is an error, and returns TRUE or FALSE.	A Boolean value of TRUE if the value is an error; otherwise FALSE.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Information	ISEVEN	ISEVEN(number)	Returns TRUE if number is even, or FALSE if number is odd.	Returns TRUE if number is even, or FALSE if number is odd.
Information	ISLOGICAL	ISLOGICAL(<value>)	Checks whether a value is a logical value, (TRUE or FALSE), and returns TRUE or FALSE.	TRUE if the value is a logical value; FALSE if any value other than TRUE OR FALSE.
Information	ISNONTEXT	ISNONTEXT(<value>)	Checks if a value is not text (blank cells are not text), and returns TRUE or FALSE.	TRUE if the value is not text or blank; FALSE if the value is text.
Information	ISNUMBER	ISNUMBER(<value>)	Checks whether a value is a number, and returns TRUE or FALSE.	TRUE if the value is numeric; otherwise FALSE.
Information	ISONORAFTER	ISONORAFTER(<scalar_expression>, <scalar_expression>sort_order] [,scalar_expression>, <scalar_expression>, [sort_order][,...])	A boolean function that emulates the behavior of a 'Start At' clause and returns true for a row that meets all of the condition parameters.	
Information	ISTEXT	ISTEXT(<value>)	Checks if a value is text, and returns TRUE or FALSE.	TRUE if the value is text; otherwise FALSE
Information	LOOKUPVALUE	LOOKUPVALUE(<result_columnName>, <search_columnName>, <search_value>[, <search_columnName>, <search_value>]...)	Returns the value in result_columnName for the row that meets all criteria specified by search_columnName and search_value.	search_valueA scalar expression that does not refer to any column in the same table being searched.
Logical	AND	AND(<logical1>,<logical2>)	Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE. Otherwise returns false.	Returns true or false depending on the combination of values that you test.
Logical	FALSE	FALSE()	Returns the logical value FALSE.	The word FALSE is also interpreted as the logical value FALSE.
Logical	IF	IF(logical_test>,<value_if_true>, value_if_false)	Checks if a condition provided as the first argument is met. Returns one value if the condition is TRUE, and returns another value if the condition is FALSE.	Any type of value that can be returned by an expression.
Logical	IFERROR	IFERROR(value, value_if_error)	Evaluates an expression and returns a specified value if the expression returns an error; otherwise returns the value of the expression itself.	A scalar of the same type as value
Logical	NOT	NOT(<logical>)	Changes FALSE to TRUE, or TRUE to FALSE.	TRUE OR FALSE.
Logical	OR	OR(<logical1>,<logical2>)	Checks whether one of the arguments is TRUE to return TRUE. The function returns FALSE if both arguments are FALSE.	A Boolean value. The value is TRUE if any of the two arguments is TRUE; the value is FALSE if both the arguments are FALSE.
Logical	TRUE	TRUE()	Returns the logical value TRUE.	The word TRUE is also interpreted as the logical value TRUE.
Math and Trig	ABS	ABS(<number>)	Returns the absolute value of a number.	A decimal number.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Math and Trig	ACOS	ACOS(number)	Returns the arccosine, or inverse cosine, of a number. The arccosine is the angle whose cosine is number. The returned angle is given in radians in the range 0 (zero) to pi.	Returns the arccosine, or inverse cosine, of a number.
Math and Trig	ASIN	ASIN(number)	Returns the arcsine, or inverse sine, of a number. The arcsine is the angle whose sine is number. The returned angle is given in radians in the range -pi/2 to pi/2.	Returns the arcsine, or inverse sine, of a number.
Math and Trig	ATAN	ATAN(number)	Returns the arctangent, or inverse tangent, of a number. The arctangent is the angle whose tangent is number. The returned angle is given in radians in the range -pi/2 to pi/2.	Returns the inverse hyperbolic tangent of a number.
Math and Trig	CEILING	CEILING(<number>, <significance>)	Rounds a number up, to the nearest integer or to the nearest multiple of significance.	A number rounded as specified.
Math and Trig	COMBIN	COMBIN(number, number_chosen)	Returns the number of combinations for a given number of items. Use COMBIN to determine the total possible number of groups for a given number of items.	Returns the number of combinations for a given number of items.
Math and Trig	COMBINA	COMBINA(number, number_chosen)	Returns the number of combinations (with repetitions) for a given number of items.	Returns the number of combinations (with repetitions) for a given number of items.
Math and Trig	COS	COS(number)	Returns the cosine of the given angle.	Returns the cosine of the given angle.
Math and Trig	COSH	COSH(number)	Returns the hyperbolic cosine of a number.	The hyperbolic cosine of a number.
Math and Trig	DIVIDE	DIVIDE(<numerator>, <denominator> [, <alternateresult>])	Performs division and returns alternate result or BLANK() on division by 0.	A decimal number.
Math and Trig	EVEN	EVEN(number)	Returns number rounded up to the nearest even integer. You can use this function for processing items that come in twos. For example, a packing crate accepts rows of one or two items. The crate is full when the number of items, rounded up to the nearest two, matches the crate's capacity.	Returns number rounded up to the nearest even integer.
Math and Trig	FACT	FACT(<number>)	Returns the factorial of a number, equal to the series 1*2*3*...*, ending in the given number.	A decimal number.
Math and Trig	FLOOR	FLOOR(<number>, <significance>)	Rounds a number down, toward zero, to the nearest multiple of significance.	A decimal number.
Math and Trig	GCD	GCD(number1, [number2], ...)	Returns the greatest common divisor of two or more integers. The greatest common divisor is the largest integer that divides both number1 and number2 without a remainder.	The greatest common divisor of two or more integers.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Math and Trig	INT	INT(<number>)	Rounds a number down to the nearest integer.	A whole number.
Math and Trig	ISO.CEILING	ISO.CEILING(<number>[, <significance>])	Rounds a number up, to the nearest integer or to the nearest multiple of significance.	A number, of the same type as the number argument, rounded as specified.
Math and Trig	LCM	LCM(number1, [number2], ...)	Returns the least common multiple of integers. The least common multiple is the smallest positive integer that is a multiple of all integer arguments number1, number2, and so on. Use LCM to add fractions with different denominators.	Returns the least common multiple of integers.
Math and Trig	LN	LN(<number>)	Returns the natural logarithm of a number. Natural logarithms are based on the constant e (2.71828182845904).	A decimal number.
Math and Trig	LOG	LOG(<number>,<base>)	Returns the logarithm of a number to the base you specify.	A decimal number.
Math and Trig	LOG10	LOG10(<number>)	Returns the base-10 logarithm of a number.	A decimal number.
Math and Trig	INT	INT(<number>)	Rounds a number down to the nearest integer.	A whole number.
Math and Trig	MROUND	MROUND(<number>, <multiple>)	Returns a number rounded to the desired multiple.	A decimal number.
Math and Trig	ODD	ODD(number)	Returns number rounded up to the nearest odd integer.	Returns number rounded up to the nearest odd integer.
Math and Trig	PI	PI()	Returns the value of Pi, 3.14159265358979, accurate to 15 digits.	Pi is a mathematical constant. In DAX, Pi is represented as a real number accurate to 15 digits, the same as Excel.
Math and Trig	POWER	POWER(<number>, <power>)	Returns the result of a number raised to a power.	A decimal number.
Math and Trig	QUOTIENT	QUOTIENT(<numerator>, <denominator>)	Performs division and returns only the integer portion of the division result. Use this function when you want to discard the remainder of division.	A whole number.
Math and Trig	RAND	RAND()	Returns a random number greater than or equal to 0 and less than 1, evenly distributed. The number that is returned changes each time the cell containing this function is recalculated.	In Power Pivot workbooks, recalculation depends on various factors, including whether the workbook is set to Manual or Automatic recalculation mode, and whether data has been refreshed. This is different from Microsoft Excel, where you can control when RAND generates a new random number by turning off recalculation.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Math and Trig	RANDBETWEEN	RANDBETWEEN(<bottom>,<top>)	Returns a random number in the range between two numbers you specify.	A whole number.
Math and Trig	ROUND	ROUND(<number>, <num_digits>)	Rounds a number to the specified number of digits.	A decimal number.
Math and Trig	ROUNDDOWN	ROUNDDOWN(<number>, <num_digits>)	Rounds a number down, toward zero.	A decimal number.
Math and Trig	ROUNDUP	ROUNDUP(<number>, <num_digits>)	Rounds a number up, away from 0 (zero).	A decimal number.
Math and Trig	SQRT	SQRT(<number>)	Returns the square root of a number.	A decimal number.
Math and Trig	SUM	SUM(<column>)	Adds all the numbers in a column.	A decimal number.
Math and Trig	SUMX	SUMX(<table>, <expression>)	Returns the sum of an expression evaluated for each row in a table.	A decimal number.
Math and Trig	TRUNC	TRUNC(<number>,<num_digits>)	Truncates a number to an integer by removing the decimal, or fractional, part of the number.	A whole number.
Other	DATATABLE	DATATABLE (ColumnName1, DataType1, ColumnName2, DataType2..., {{Value1, Value2...}, {ValueN, ValueN+1...}...})	Provides a mechanism for declaring an inline set of data values.	A table declaring an inline set of values.
Other	ERROR	ERROR(<text>)	Raises an error with an error message.	None
Other	EXCEPT	EXCEPT(<table_expression1>, <table_expression2>)	Returns the rows of one table which do not appear in another table.	A table that contains the rows of one table minus all the rows of another table.
Other	GENERATESERIES	GENERATESERIES(<startValue>, <endValue>[, <incrementValue>])	Returns a single column table containing the values of an arithmetic series, that is, a sequence of values in which each differs from the preceding by a constant quantity. The name of the column returned is Value.	A single column table containing the values of an arithmetic series. The name of the column is Value.
Other	GROUPBY	GROUPBY (<table>, [<groupBy_columnName1>], [<name>, <expression>]...)	The GROUPBY function is similar to the SUMMARIZE function. However, GROUPBY does not do an implicit CALCULATE for any extension columns that it adds. GROUPBY permits a new function, CURRENTGROUP(), to be used inside aggregation functions in the extension columns that it adds. GROUPBY attempts to reuse the data that has been grouped making it highly performant.	A table with the selected columns for the groupBy_columnName arguments and the grouped by columns designated by the name arguments.
Other	ISEMPTY	ISEMPTY(<table_expression>)	Checks if a table is empty.	True if the table is empty (has no rows), if else, False.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Other	NATURALINNERJOIN	NATURALINNERJOIN(<leftJoinTable>, <rightJoinTable>)	Performs an inner join of a table with another table. The tables are joined on common columns (by name) in the two tables. If the two tables have no common column names, an error is returned.	A table which includes only rows for which the values in the common columns specified are present in both tables. The table returned will have the common columns from the left table and other columns from both the tables.
Other	NATURALLEFTOUTERJOIN	NATURALLEFTOUTERJOIN(<leftJoinTable>, <rightJoinTable>)	Performs an inner join of a table with another table. The tables are joined on common columns (by name) in the two tables. If the two tables have no common column names, an error is returned.	A table which includes only rows from rightJoinTable for which the values in the common columns specified are also present in leftJoinTable. The table returned will have the common columns from the left table and the other columns from both the tables.
Other	SUMMARIZECOLUMNS	SUMMARIZECOLUMNS(<groupBy_columnName> [, <groupBy_columnName >]..., [<filterTable>]...[, <name>, <expression>]...)	Returns a summary table over a set of groups.	A table which includes combinations of values from the supplied columns, based on the grouping specified. Only rows for which at least one of the supplied expressions return a non-blank value are included in the table returned. If all expressions evaluate to BLANK/NULL for a row, that row is not included in the table returned.
Other	Table Constructor (DAX)	{ <scalarExpr1>, <scalarExpr2>, ... }	Returns a table of one or more columns.	A table of one or more columns. When there is only one column, the name of the column is Value. When there are N columns where N > 1, the names of the columns from left to right are Value1, Value2, ..., ValueN.
		{ (<scalarExpr1>, <scalarExpr2>, ...), (<scalarExpr1>, <scalarExpr2>, ...), ... }		
Other	TREATAS	TREATAS(table_expression, <column>[, <column>[, <column>[,...]]])	Applies the result of a table expression as filters to columns from an unrelated table.	A table that contains all the rows in column(s) that are also in table_expression.
Other	UNION	UNION(<table_expression1>, <table_expression2> [,<table_expression>]...)	Creates a union (join) table from a pair of tables.	A table that contains all the rows from each of the two table expressions.
Parent and Child	PATHLENGTH	PATHLENGTH(<path>)	Returns the number of parents to the specified item in a given PATH result, including self.	The number of items that are parents to the specified item in a given PATH result, including the specified item.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Statistical	ADDCOLUMNS	ADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]...)	Adds calculated columns to the given table or table expression.	expressionAny DAX expression that returns a scalar expression, evaluated for each row of table.
Statistical	AVERAGE	AVERAGE(<column>)	Returns the average (arithmetic mean) of all the numbers in a column.	Returns a decimal number that represents the arithmetic mean of the numbers in the column.
Statistical	AVERAGEA	AVERAGEA(<column>)	Returns the average (arithmetic mean) of the values in a column. Handles text and non-numeric values.	A decimal number.
Statistical	AVERAGEX	AVERAGEX(<table>,<expression>)	Calculates the average (arithmetic mean) of a set of expressions evaluated over a table.	A decimal number.
Statistical	BETA.DIST	BETA.DIST(x,alpha,beta,cumulative,[A],[B])	Returns the beta distribution. The beta distribution is commonly used to study variation in the percentage of something across samples, such as the fraction of the day people spend watching television.	Returns the beta distribution.
Statistical	CONFIDENCE.NORM	CONFIDENCE.NORM(alpha,standard_dev,size)	The confidence interval is a range of values. Your sample mean, x, is at the center of this range and the range is $x \pm \text{CONFIDENCE.NORM}$. For example, if x is the sample mean of delivery times for products ordered through the mail, $x \pm \text{CONFIDENCE.NORM}$ is a range of population means. For any population mean, μ_0 , in this range, the probability of obtaining a sample mean further from μ_0 than x is greater than alpha; for any population mean, μ_0 , not in this range, the probability of obtaining a sample mean further from μ_0 than x is less than alpha. In other words, assume that we use x, standard_dev, and size to construct a two-tailed test at significance level alpha of the hypothesis that the population mean is μ_0 . Then we will not reject that hypothesis if μ_0 is in the confidence interval and will reject that hypothesis if μ_0 is not in the confidence interval. The confidence interval does not allow us to infer that there is probability $1 - \alpha$ that our next package will take a delivery time that is in the confidence interval.	A range of values
Statistical	CONFIDENCE.T	CONFIDENCE.T(alpha,standard_dev,size)	Returns the confidence interval for a population mean, using a Student's t distribution.	Returns the confidence interval for a population mean, using a Student's t distribution.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Statistical	COUNT	COUNT(<column>)	The COUNT function counts the number of cells in a column that contain numbers.	A whole number.
Statistical	COUNTA	COUNTA(<column>)	The COUNTA function counts the number of cells in a column that are not empty. It counts not just rows that contain numeric values, but also rows that contain nonblank values, including text, dates, and logical values.	A whole number.
Statistical	COUNTAX	COUNTAX(<table>,<expression>)	The COUNTAX function counts nonblank results when evaluating the result of an expression over a table. That is, it works just like the COUNTA function, but is used to iterate through the rows in a table and count rows where the specified expressions results in a nonblank result.	A whole number.
Statistical	COUNTBLANK	COUNTBLANK(<column>)	Counts the number of blank cells in a column.	A whole number. If no rows are found that meet the condition, blanks are returned.
Statistical	COUNTROWS	COUNTROWS(<table>)	The COUNTROWS function counts the number of rows in the specified table, or in a table defined by an expression.	A whole number.
Statistical	COUNTX	COUNTX(<table>,<expression>)	Counts the number of rows that contain a number or an expression that evaluates to a number, when evaluating an expression over a table.	An integer.
Statistical	DATATABLE Function	DATATABLE (ColumnName1, DataType1, ColumnName2, DataType2..., {{Value1, Value2...}, {ValueN, ValueN+1...}})	Provides a mechanism for declaring an inline set of data values.	A table declaring an inline set of values.
Statistical	DISTINCTCOUNT	DISTINCTCOUNT(<column>)	The DISTINCTCOUNT function counts the number of distinct values in a column.	The number of distinct values in column.
Statistical	EXPON.DIST	EXPON.DIST(x,lambda,cumulative)	Returns the exponential distribution. Use EXPON.DIST to model the time between events, such as how long an automated bank teller takes to deliver cash. For example, you can use EXPON.DIST to determine the probability that the process takes at most 1 minute.	Returns the exponential distribution.
Statistical	MAXA	MAXA(<column>)	Returns the largest value in a column. Logical values and blanks are counted.	A decimal number.
Statistical	MAXX	MAXX(<table>,<expression>)	Evaluates an expression for each row of a table and returns the largest numeric value.	A decimal number.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Statistical	MINA	MINA(<column>)	Returns the smallest value in a column, including any logical values and numbers represented as text.	A decimal number.
Statistical	MINX	MINX(<table>, < expression>)	Returns the smallest numeric value that results from evaluating an expression for each row of a table.	A decimal number.
Statistical	POISSON.DIST	POISSON.DIST(x,mean,cumulative)	Returns the Poisson distribution. A common application of the Poisson distribution is predicting the number of events over a specific time, such as the number of cars arriving at a toll plaza in 1 minute.	Returns the Poisson distribution.
Statistical	RANKX	RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])	Returns the ranking of a number in a list of numbers for each row in the table argument.	value(Optional) Any DAX expression that returns a single scalar value whose rank is to be found. See the remarks section to understand the function's behavior when value is not found in the expression.
Statistical	SELECTCOLUMNS	SELECTCOLUMNS(<table>, <name>, <scalar_expression> [, <name>, <scalar_expression>]...)	Adds calculated columns to the given table or table expression.	expressionAny expression that returns a scalar value like a column reference, integer, or string value.
Statistical	SIN	SIN(number)	Returns the sine of the given angle.	Returns the sine of the given angle.
Statistical	SINH	SINH(number)	Returns the hyperbolic sine of a number.	Returns the hyperbolic sine of a number.
Statistical	SQRTPI	SQRTPI(number)	Returns the square root of (number * pi).	Returns the square root of (number * pi).
Statistical	TAN	TAN(number)	Returns the tangent of the given angle.	Returns the tangent of the given angle.
Statistical	TANH	TANH(number)	Returns the hyperbolic tangent of a number.	Returns the hyperbolic tangent of a number.
Statistical	TOPN	TOPN(<n_value>, <table>, <orderBy_expression>, [<order>[, <orderBy_expression>, [<order>]]...])	Returns the top N rows of the specified table.	See the remarks section to understand when an empty table is returned.
Statistical	XIRR	XIRR(<table>, <values>, <dates>, [guess])	Returns the internal rate of return for a schedule of cash flows that is not necessarily periodic.	Internal rate of return for the given inputs. If the calculation fails to return a valid result, an error is returned.
Statistical	XNPV	XNPV(<table>, <values>, <dates>, <rate>)	Returns the present value for a schedule of cash flows that is not necessarily periodic.	Net present value.
Text	CONCATENATE	CONCATENATE(<text1>, <text2>)	Joins two text strings into one text string.	The concatenated string.
Text	CONCATENATEX	CONCATENATEX(<table>, <expression>, [delimiter])	Concatenates the result of an expression evaluated for each row in a table.	A text string.

Function Type	DAX Function Name	DAX Function Syntax	DAX Function Description	DAX Function Returns
Text	EXACT	EXACT(<text1>,<text2>)	Compares two text strings and returns TRUE if they are exactly the same, FALSE otherwise. EXACT is case-sensitive but ignores formatting differences. You can use EXACT to test text being entered into a document.	True or false. (Boolean)
Text	FIND	FIND(<find_text>, <within_text>[, [<start_num>]], [<NotFoundValue>]])	Returns the starting position of one text string within another text string. FIND is case-sensitive.	Number that shows the starting point of the text string you want to find.
Text	FIXED	FIXED(<number>, <decimals>, <no_commas>)	Rounds a number to the specified number of decimals and returns the result as text. You can specify that the result be returned with or without commas.	A number represented as text.
Text	LEFT	LEFT(<text>, <num_chars>)	Returns the specified number of characters from the start of a text string.	A text string.
Text	LEN	LEN(<text>)	Returns the number of characters in a text string.	A whole number indicating the number of characters in the text string.
Text	LOWER	LOWER(<text>)	Converts all letters in a text string to lowercase.	Text in lowercase.
Text	MID	MID(<text>, <start_num>, <num_chars>)	Returns a string of characters from the middle of a text string, given a starting position and length.	A string of text of the specified length.
Text	REPLACE	REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)	REPLACE replaces part of a text string, based on the number of characters you specify, with a different text string.	A text string.
Text	REPT	REPT(<text>, <num_times>)	Repeats text a given number of times. Use REPT to fill a cell with a number of instances of a text string.	A string containing the changes.
Text	SUBSTITUTE	SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)	Replaces existing text with new text in a text string.	A string of text.
Text	TRIM	TRIM(<text>)	Removes all spaces from text except for single spaces between words.	The string with spaces removed.
Text	UNICHAR	UNICHAR(number)	Returns the Unicode character referenced by the numeric value.	A character represented by the Unicode number
Text	UPPER	UPPER (<text>)	Converts a text string to all uppercase letters	Same text, in uppercase.
Text	VALUE	VALUE(<text>)	Converts a text string that represents a number to a number.	The converted number in decimal data type.