Math and Statistical Functions

INT

INT Rounds a number down to the nearest integer.

Syntax

INT (<number>)

Example

INT Example = INT (Orders [Sales])

12.34 =12 12.78 = 12 12.55=12

ROUND

Round function will Rounds a number to the given number of digits.

Syntax

ROUND (<number>, <num_digits>)

Example

Round Example = ROUND (Orders [Sales], 0)

12.34 =12 12.78 = 13 12.55=13

Round Example = ROUND (Orders [Sales], 1)

12.34 =12.3 12.78 = 12.8 12.55=12.6

ROUNDUP

Roundup will Rounds a number to next integer value if num_digits = 0.

Syntax

ROUNDUP (<number>, <num_digits>)

Example

ROUNDUP Example = ROUNDUP (Orders [Sales], 0)

12.34 =13 12.78 = 13 12.55=13

ROUNDUP Example = ROUNDUP (Orders [Sales], 1)

12.34 =12.4 12.78 = 12.8 12.55=12.6

ROUNDDOWN

ROUNDDOWN Rounds a number down, toward zero.

Syntax

ROUNDDOWN (<number>, <num_digits>)

Example

ROUNDDOWN Example = ROUNDDOWN (Orders [Sales], 0)

12.34 =12 12.78 = 12 12.55=12

ROUNDDOWN Example = ROUNDDOWN (Orders [Sales], 1)

12.34 =12.3 12.78 = 12.7 12.55=12.5

DIVIDE

Performs division and returns alternate result or BLANK () on division by 0.

Syntax

DIVIDE (<numerator>, <denominator>, [<alternateresult>])

Example

Divide Example = DIVIDE (Orders [Sales], Orders [Quantity], 0)

Divide Example = DIVIDE (200,4,0) = 50

Divide Example = DIVIDE (100,0, -1) = -1

EVEN

Returns number **rounded up** to the nearest even integer.

Syntax

EVEN (number)

Example

Even Example = EVEN (Orders [Sales])

12.34 =14 12.78 = 14 12.55=14

13.34 =14 13.78 = 14 13.55=14

ODD

Returns number **rounded up** to the nearest odd integer.

Syntax

ODD (number)

Example

Odd Example = ODD (Orders [Sales])

12.34 =13	12.78 = 13	12.55=13
13.34 =15	13.78 = 15	13.55=15

POWER

Returns the result of a number raised to a power.

Syntax

POWER (<number>, <power>)

Example

Power Example = POWER (Orders [Quantity], Orders [Quantity])

Power Example = POWER (Orders [Quantity],2)

Power Example = POWER (4,2) →16

Power Example = POWER (3,3) →27

SIGN

SIGN 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.

Syntax

SIGN (<number>)

Example

Sign Example = SIGN (Orders [Sales])

SQRT

SQRT returns the square root of a number. If the number is negative, the SQRT function returns an error.

Syntax

SQRT (<number>)

Example

SQRT Example = SQRT (Orders [Quantity])

SQRT Example = ROUND (SQRT (Orders [Quantity]), 2)

FACT

Returns the factorial of a number, equal to the series 1*2*3*..., ending in the given number.

Syntax

FACT (<number>)

Example

FACT Example = FACT (Orders [Quantity])

SUM

SUM Function Adds all the numbers in a column.

Syntax

SUM (<column>)

Example

Sum of Sal = SUM (EMP [SAL])

SUMX

Returns the sum of an expression evaluated for each row in a table.

Syntax

SUMX (<table>, <expression>)

SUMX of Sal = SUMX (EMP, EMP [SAL] +EMP [COMM])

MIN

Returns the smallest numeric value in a column.

Syntax

MIN (<column>)

Example

Min of Sal = MIN(EMP[SAL])

MINX

Returns the smallest numeric value that results from evaluating an expression for each row of a table.

Syntax

MINX (<table>, <expression>)

Example

MINX of Sal = MINX (EMP, EMP [SAL]+ EMP [COMM])

MAX

Returns the largest numeric value in a column.

Syntax

MAX (<column>)

Example

MAX of Sal = MAX(EMP[SAL])

MAXX

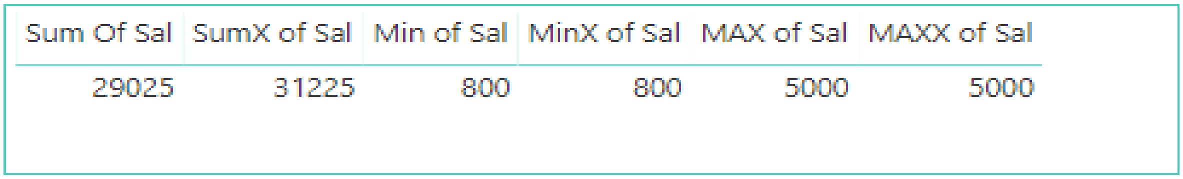
Evaluates an expression for each row of a table and returns the largest numeric value.

Syntax

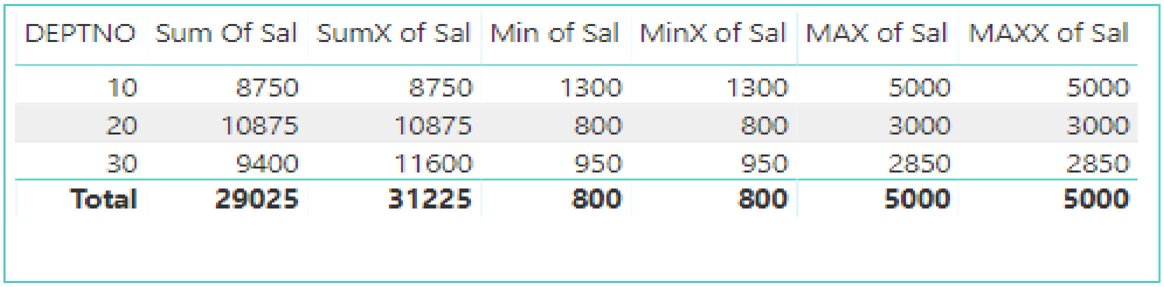
MAXX (<table>, <expression>)

Example

MAXX of Sal = MAXX (EMP, EMP [SAL]+ EMP [COMM])



Sum Of Sal	SumX of Sal	Min of Sal	MinX of Sal	MAX of Sal	MAXX of Sal
29025	31225	800	800	5000	5000



DEPTNO	Sum Of Sal	SumX of Sal	Min of Sal	MinX of Sal	MAX of Sal	MAXX of Sal
10	8750	8750	1300	1300	5000	5000
20	10875	10875	800	800	3000	3000
30	9400	11600	950	950	2850	2850
Total	**29025**	**31225**	**800**	**800**	**5000**	**5000**

COUNT

Counts the number of cells in a column that contain numbers.

Syntax

COUNT (<column>)

Example

COUNT Example = COUNT(EMP[SAL])

Count Example1 = COUNT (EMP [COMM])

COUNTX

Counts the number of rows that contain a number or an expression that evaluates to a number, when evaluating an expression over a table.

Syntax

COUNTX (<table>, <expression>)

Example

COUNTX Example = COUNTX (EMP, EMP[SAL]+EMP[COMM])

AVERAGE

AVERAGE Function Will Returns the average of all the numbers in a column.

Syntax

AVERAGE (<column>)

Example

Average Example = AVERAGE(EMP[SAL])

Average of COMM = AVERAGE(EMP[COMM])

AVERAGEX

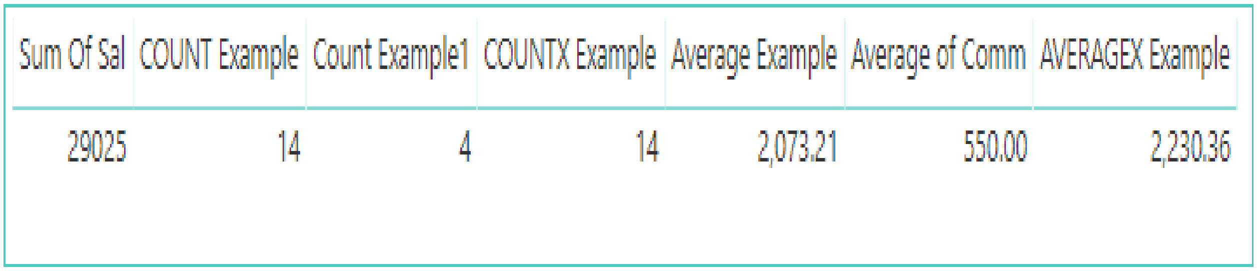
Calculates the average of a set of expressions evaluated over a table.

Syntax

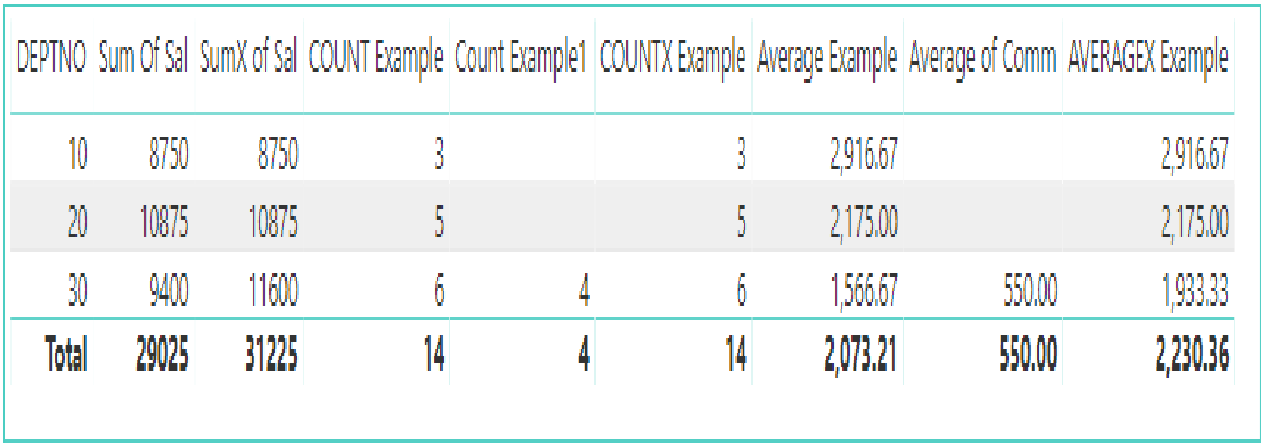
AVERAGEX (<table>, <expression>)

Example

AVERAGEX Example = AVERAGEX (EMP, EMP[SAL]+EMP[COMM])



Sum Of Sal	COUNT Example	Count Example1	COUNTX Example	Average Example	Average of Comm	AVERAGEX Example
29025	14	4	14	2,073.21	550.00	2,230.36



DEPTNO	Sum Of Sal	SumX of Sal	COUNT Example	Count Example1	COUNTX Example	Average Example	Average of Comm	AVERAGEX Example
10	8750	8750	3		3	2,916.67		2,916.67
20	10875	10875	5		5	2,175.00		2,175.00
30	9400	11600	6	4	6	1,566.67	550.00	1,933.33
Total	**29025**	**31225**	**14**	**4**	**14**	**2,073.21**	**550.00**	**2,230.36**

COUNTROWS

Counts the number of rows in the specified table, or in a table defined by an expression.

Syntax

COUNTROWS (<table>)

Example

COUNTROWS Example = COUNTROWS(EMP)

COUNTBLANK

Counts the number of blank cells in a column.

Syntax

COUNTBLANK (<column>)

Example

COUNTBLANK Example = COUNTBLANK(EMP[COMM])

RANKX

The RANKX function is used in DAX to create rankings.

Ranks allow you to easily compare products, salespeople or anything else that you want to evaluate the performance.

RANKX is a scalar function and it is also an iterator. The RANKX function can optionally take a Value argument that represents a scalar value whose rank is to be found. The optional Order argument specifies how to rank Value, descending (0) or ascending (1). The optional Ties argument defines how to determine ranking when there are ties (Same Ranks). Skip (default) will use the next rank value after a tie, and Dense will use the next rank value (i.e. there will be no gaps in the rank numbers).

Optional arguments might be skipped by placing an empty comma (,) in the argument list, i.e. RANKX = RANKX (EMP, EMP[SAL],,,Dense)

Syntax

RANKX (TABLE, EXPRESSION [, VALUE] [, ORDER] [, TIES]…)

Dense – will not Skip Between Rank Numbers

Examples

RANKX = RANKX (EMP, EMP[SAL])

```
Category Rank =
        -- STEP 3. Rank my expression against the sorted list from 2.
        RANKX (
                -- STEP 1. Loop the rows in this table
                ALL(EMP[DEPTNO]),
                -- STEP 2. Run this expression for each loop
                CALCULATE (
                       SUM(EMP[SAL])
                       )
           )
```