**What is DAX?**

DAX is a collection of functions, operators, and constants that can be used in a formula, or expression, to calculate and return one or more values. Stated more simply, DAX helps you create new information from data already in your model.

**Why is DAX so important?**

It’s easy to create a new Power BI Desktop file and import some data into it. You can even create reports that show valuable insights without using any DAX formulas at all. But, what if you need to analyze growth percentage across product categories and for different date ranges? Or, you need to calculate year-over-year growth compared to market trends? DAX formulas provide this capability and many other important capabilities as well. Learning how to create effective DAX formulas will help you get the most out of your data. When you get the information you need, you can begin to solve real business problems that affect your bottom line. This is the power of Power BI, and DAX will help you get there.

#### **DAX Calculation Types**

So, apparently, the DAX formulas can also be called as calculations as they calculate an input value and return a resultant value. You can create two types of expressions or calculations using DAX in Power BI; calculated columns and calculated measures.

* **Calculated Columns:** The calculated columns create a new column in your existing table. The only difference between a regular column and a calculated column is that it is necessary to have at least one function in the calculated column. These are used when you want to create a column with filtered or sorted information.

To create a calculated column:

1. Go to the **Modeling** tab in Power BI Desktop.
2. Then select**New Column** option. A Formula bar will open showing “Column =”. You can replace the “Column” word with the column name you want.
3. After this, enter the expression for the calculated column on the right of the equals to sign.

* **Calculated Measures:** A calculated measure creates a field having aggregated values such a sum, ratios, percentages, averages, etc.

To create a calculated measure:

1. Go to the **Modeling** tab in Power BI Desktop.
2. Then select **New Measure** option. A Formula bar will open showing “Measure =”. You can replace the “Measure” word with the measure name you want.
3. After this, enter the expression for the calculated measure on the right of the equals to sign.
4. Once you create the measure, you can modify your measure name with a calculator icon next to it, under the table name you created the measure in.

### **DAX Functions**

A DAX function is a predefined formula which performs calculations on values provided to it in arguments. The arguments in a function need to be in a particular order and can be a column reference, numbers, text, constants, another formula or function, or a logical value such as TRUE or FALSE. Every function performs a particular operation on the values enclosed in an argument. You can use more than one argument in a DAX formula.

##### Key Points about DAX Functions

Here are some unique facts about DAX functions that you must know in order to understand them better:

* Any DAX function always refers to a complete column/field or a table. It will never refer to individual values. If you want to use the functions on separate values within a column, you need to apply filters in a DAX formula.
* DAX functions provide the flexibility to create a formula that is applied on a row-by-row basis. The calculations or formulas get applied as per the context of the values in each row.
* In some cases, DAX functions return a full table which can be used in other DAX formulas that need a complete set of values. However, you cannot display this table’s contents.
* DAX functions have a category known as time intelligence functions. Such functions are used to calculate time/date ranges and periods.

#### **Types of DAX functions**

##### 1. Date and Time Functions

The date time functions carry out calculations on the date and time values. The data type of these values is always datetime data type.

* CALENDAR
* CALENDARAUTO
* DATE
* DATEDIFF
* DATEVALUE
* DAY
* EOMONTH
* HOUR
* MINUTE
* MONTH
* NOW
* SECOND
* TIME
* TIMEVALUE
* TODAY
* WEEKDAY
* WEEKNUM
* YEAR
* YEARFRAC

##### 2. Time Intelligence Functions

The time-intelligence functions are used to evaluate values over a fixed period such as days, weeks, months, quarter, years, etc. You can specify a time period using these functions and compare two scenarios in your report.

* CLOSINGBALANCEMONTH
* CLOSINGBALANCEQUARTER
* CLOSINGBALANCEYEAR
* DATEADD
* DATESBETWEEN
* DATESINPERIOD
* DATESMTD
* DATESQTD
* DATESYTD
* ENDOFMONTH
* ENDOFQUARTER
* ENDOFYEAR
* FIRSTDATE
* FIRSTNONBLANK
* LASTDATE
* NEXTQUARTER
* LASTNONBLANK
* NEXTDAY
* NEXTMONTH

##### 3. Information Functions

The information functions are used to provide certain information on the data values contained in rows and columns. It evaluates the given condition in a function for the value given and return TRUE or FALSE. For instance, the function ISERROR will return TRUE if the value evaluated contains an error.

* CONTAINS
* CUSTOMDATA
* IN Operator / CONTAINSROW function
* ISBLANK
* ISERROR
* ISEVEN
* ISINSCOPE
* ISLOGICAL
* ISNONTEXT
* ISNUMBER
* ISODD
* ISONORAFTER
* ISTEXT
* LOOKUPVALUE
* USERNAME

##### 4. Logical Functions

The logical functions are used to evaluate an expression or argument logically and return TRUE or FALSE if the condition is met or not.

* AND
* FALSE
* IF
* IFERROR
* IN
* NOT
* OR
* SWITCH
* TRUE

##### 5. Mathematical and Trigonometric Functions

The mathematical and trig functions are used to perform all sorts of mathematical functions on the referred values. Given below, is a list of all the available math and trig DAX functions in Power BI.

* ABS
* ACOS
* ACOSH
* ASIN
* ASINH
* ATAN
* ATANH
* CEILING
* COMBIN
* COMBINA
* COS
* COSH
* CURRENCY
* DEGREES
* DIVIDE
* EVEN
* EXP
* FACT
* FLOOR

##### 6. Statistical Functions

These functions carry out statistical and aggregation functions on data values in a DAX expression in Power BI. The list of available statistical functions is given below.

* ADDCOLUMNS
* APPROXIMATEDISTINCTCOUNT
* AVERAGE
* AVERAGEA
* AVERAGEX
* BETA.DIST
* BETA.INV
* CHISQ.INV
* CHISQ.INV.RT
* CONFIDENCE.NORM
* CONFIDENCE.T
* COUNT
* COUNTA
* COUNTAX
* COUNTBLANK
* COUNTROWS
* COUNTX
* CROSSJOIN
* DISTINCTCOUNT
* EXPON.DIST
* GENERATE
* GENERATEALL
* GEOMEAN
* GEOMEANX
* MAX
* MAXA
* MAXX
* MEDIAN
* MEDIANX
* MIN
* MINA
* MINX

##### 7. Text Functions

The text functions in Power BI are very similar to the string functions of Excel. These functions evaluate string values.

* BLANK
* CODE
* COMBINEVALUES
* CONCATENATE
* CONCATENATEX
* EXACT
* FIND
* FIXED
* FORMAT
* LEFT
* LEN
* LOWER
* MID
* REPLACE
* REPT
* RIGHT
* SEARCH
* SUBSTITUTE
* TRIM
* UNICHAR
* UPPER
* VALUE

##### 8. Parent-Child functions

The parent and child functions are used for data values that are a part of a parent-child hierarchy.

* PATH
* PATHCONTAINS
* PATHITEM
* PATHITEMREVERSE
* PATHLENGTH

##### 9. Other functions

There a bunch of functions that do not fit in any particular category. These are also very useful functions.

* DATATABLE
* ERROR
* EXCEPT
* GENERATESERIES
* GROUPBY
* INTERSECT
* ISEMPTY
* ISSELECTEDMEASURE
* NATURALINNERJOIN
* NATURALLEFTOUTERJOIN
* SELECTEDSMEASURE
* SELECTEDMEASUREFORMATSTRING
* SELECTEDSMEASURENAME
* SUMMARIZECOLUMNS
* Table Constructor
* TREATAS
* UNION
* VAR

##### 10. Table functions

The table functions in DAX formulas for [Power BI](https://powerbi.microsoft.com/en-us/partner-showcase/) are used to apply operations and conditions on entire tables. The output of table functions is used as inputs in other expressions or arguments in a DAX formula. The results of these functions retain the relationships between columns of that table.

* FILTER
* ALL
* VALUES
* DISTINCT
* RELATEDTABLE

Total sale = sum(sales[salesamount])

**I.** **Total Sales**is the measure name.

**II.** The **equals sign operator (=)** indicates the beginning of the formula.

**III.** **SUM** adds up all of the numbers in the column, **Sales[SalesAmount]**.

**IV.** There are these **parentheses** **()** that surround an expression containing one or more arguments. All functions require at least one argument.

**V.** **Sales**is the table referenced.

**VI.** An **argument** passes a value to a function. The referenced column **[SalesAmount]** is an argument with which, the SUM function knows the column on which it has to aggregate a SUM.

Simply put, you can read it as, **“For the measure named Total Sales, calculate (=) the SUM of values in the [SalesAmount ] column in the Sales table.”**

## ****1. Aggregate Functions****

### **MIN**

This DAX function returns the minimum numeric value in a column, or between two scalar expressions.

**Syntax**

MIN(<column>)

**Example**

=MIN([ResellerMargin])

### **MINA**

This DAX function returns the minimum value in a column, including any logical values and numbers represented as text.

**Syntax**

MINA(<column>)

**Example**

=MINA(([PostalCode])

### **MINX**

This DAX function returns the minimum numeric value that results from evaluating an expression for each row of a table.

**Syntax**

MINX(<table>, < expression evaluated for each row>)

**Example**

=MINX( FILTER(InternetSales, InternetSales[SalesTerritoryKey] = 5), InternetSales[Freight] + InternetSales[TaxAmt])

### **MAX**

This DAX function returns the maximum value in a column, including any logical values and numbers represented as text.

**Syntax**

MAX(<column>)

**Example**

=MAX([ResellerMargin])

### **MAXA**

This DAX function returns the maximum value in a column, including any logical values and numbers represented as text.

**Syntax**

MAXA(<column>)

**Example**

=MAXA(([PostalCode])

### **MAXX**

This DAX function returns the maximum numeric value that results from evaluating an expression for each row of a table.

**Syntax**

MAXX(<table>, < expression evaluated for each row>)

**Example**

=MAXX( FILTER(InternetSales, InternetSales[SalesTerritoryKey] = 5), InternetSales[Freight] + InternetSales[TaxAmt])

### **SUM**

This DAX function adds all the numbers in a column.

**Syntax**

SUM(<column>)

**Example**

=SUM(Sales[Amt])

### **AVERAGE**

This DAX function returns the arithmetic mean of the values in a column.

**Syntax**

AVERAGE(<column>)

**Example**

=AVERAGE(InternetSales[ExtendedSalesAmount])

### **SUMX**

This DAX function returns the sum of an expression evaluated for each row in a table.

**Syntax**

SUMX(<table>, <expression evaluated for each row>)

**Example**

=SUMX(FILTER(InternetSales, InternetSales[SalesTerritoryID]=5),[Freight])

### **AVERAGEX**

This DAX function calculates the arithmetic mean of a set of expressions evaluated over a table.

**Syntax**

AVERAGEX(<table>, <expression evaluated for each row>)

**Example**

=AVERAGEX(InternetSales, InternetSales[Freight]+ InternetSales[TaxAmt])

## ****2. Count Functions****

### **DISTINCTCOUNT**

This is a DAX function used to return the distinct count of items in a column. So, if there are multiple numbers of the same item,  this function will count it as a single item.

**Syntax**

DISTINCTCOUNT(<column>)

**Example**

=DISTINCTCOUNT(ResellerSales\_USD[SalesOrderNumber])

### **COUNT**

This is a DAX function used to return the count of items in a column. So, if there are multiple numbers of the same item,  this function will count it as separate items and not a single item.

**Syntax**

COUNT(<column>)

**Example**

=COUNT([ShipDate])

### **COUNTA**

This is a DAX function used to return the count of items, in a column, that is not empty.

**Syntax**

COUNTA(<column>)

**Example**

=COUNTA('Reseller'[Phone])

### **COUNTROWS**

This is a DAX function that counts the number of rows in the specified table, or in a table defined by an expression.

**Syntax**

COUNTROWS(<table>)

**Example**

=COUNTROWS('Orders')

### **COUNTBLANK**

This is a DAX function that counts the number of blank cells in a column.

**Syntax**

COUNTBLANK(<column>)

**Example**

=COUNTBLANK(Reseller[BankName])

## ****3. Date-Time Functions****

### **DATE**

This DAX function returns the specified date in Date-Time format.

**Syntax**

DATE(<year>, <month>, <day>)

**Example**

=DATE(2019,12,17)

### **HOUR**

This DAX function returns the specified hour as a number from 0 to 23 (12:00 A.M. to 11:00 P.M.).

**Syntax**

HOUR(<datetime>)

**Example**

=HOUR('Orders'[TransactionTime])

### **TODAY**

This DAX function returns the current date.

**Syntax**

TODAY()

### **NOW**

This DAX function returns the current date and time in Date-Time format.

**Syntax**

NOW()

### **EOMONTH**

This DAX function returns the date in Date-Time format of the last day of the month, before or after a specified number of months.

**Syntax**

EOMONTH(<start\_date>, <months>)

**Example**

=EOMONTH("March 3, 2008",1.5)

## ****4. Mathematical Functions****

### **ABS**

This DAX function returns the absolute value of the number given.

**Syntax**

ABS(<number>)

**Example**

=ABS([DealerPrice]-[ListPrice])

### **EXP**

This DAX function returns the value of e raised to the power of the given number.

**Syntax**

EXP(<number>)

**Example**

=EXP([Power])

### **FACT**

This DAX function returns the factorial of a number.

**Syntax**

FACT(<number>)

**Example**

=FACT([Values])

### **LN**

This DAX function returns the natural log of the given number.

**Syntax**

LN(<number>)

**Example**

=LN([Values])

### **LOG**

This DAX function returns the log with the base of the given number.

**Syntax**

LOG(<number>,<base>)

**Example**

All the following return the same result, 2.

=LOG(100,10)

=LOG(100)

=LOG10(100)

### **PI**

This DAX function returns the value of Pi.

**Syntax**

PI()

### BI and Visualization Training

### [TABLEAU TRAINING AND CERTIFICATION](https://www.edureka.co/tableau-certification-training" \o "Tableau Training and Certification" \t "_blank)

### [Tableau Training and Certification](https://www.edureka.co/tableau-certification-training" \o "Tableau Training and Certification" \t "_blank)

*[Reviews](https://www.edureka.co/tableau-certification-training" \o "Tableau Training and Certification" \t "_blank)*

**[5](https://www.edureka.co/tableau-certification-training" \o "Tableau Training and Certification" \t "_blank)**[(22390)](https://www.edureka.co/tableau-certification-training" \o "Tableau Training and Certification" \t "_blank)

### [MICROSOFT POWER BI TRAINING](https://www.edureka.co/power-bi-certification-training" \o "Microsoft Power BI Training" \t "_blank)

### [Microsoft Power BI Training](https://www.edureka.co/power-bi-certification-training" \o "Microsoft Power BI Training" \t "_blank)

*[Reviews](https://www.edureka.co/power-bi-certification-training" \o "Microsoft Power BI Training" \t "_blank)*

**[5](https://www.edureka.co/power-bi-certification-training" \o "Microsoft Power BI Training" \t "_blank)**[(8582)](https://www.edureka.co/power-bi-certification-training" \o "Microsoft Power BI Training" \t "_blank)

### [MICROSOFT BI CERTIFICATION TRAINING](https://www.edureka.co/microsoft-bi-certification" \o "Microsoft BI Certification Training" \t "_blank)

### [Microsoft BI Certification Training](https://www.edureka.co/microsoft-bi-certification" \o "Microsoft BI Certification Training" \t "_blank)

*[Reviews](https://www.edureka.co/microsoft-bi-certification" \o "Microsoft BI Certification Training" \t "_blank)*

**[5](https://www.edureka.co/microsoft-bi-certification" \o "Microsoft BI Certification Training" \t "_blank)**[(11817)](https://www.edureka.co/microsoft-bi-certification" \o "Microsoft BI Certification Training" \t "_blank)

### [ADVANCED MS EXCEL 2016 CERTIFICATION TRAINING](https://www.edureka.co/advanced-ms-excel-self-paced" \o "Advanced MS Excel 2016 Certification Training" \t "_blank)

### [Advanced MS Excel 2016 Certification Training](https://www.edureka.co/advanced-ms-excel-self-paced" \o "Advanced MS Excel 2016 Certification Training" \t "_blank)

*[Reviews](https://www.edureka.co/advanced-ms-excel-self-paced" \o "Advanced MS Excel 2016 Certification Training" \t "_blank)*

**[5](https://www.edureka.co/advanced-ms-excel-self-paced" \o "Advanced MS Excel 2016 Certification Training" \t "_blank)**[(6799)](https://www.edureka.co/advanced-ms-excel-self-paced" \o "Advanced MS Excel 2016 Certification Training" \t "_blank)

### [MICROSOFT EXCEL VBA CERTIFICATION TRAINING](https://www.edureka.co/excel-vba-training-course" \o "Microsoft Excel VBA Certification Training" \t "_blank)

### [Microsoft Excel VBA Certification Training](https://www.edureka.co/excel-vba-training-course" \o "Microsoft Excel VBA Certification Training" \t "_blank)

*[Reviews](https://www.edureka.co/excel-vba-training-course" \o "Microsoft Excel VBA Certification Training" \t "_blank)*

**[5](https://www.edureka.co/excel-vba-training-course" \o "Microsoft Excel VBA Certification Training" \t "_blank)**[(5989)](https://www.edureka.co/excel-vba-training-course" \o "Microsoft Excel VBA Certification Training" \t "_blank)

### [QLIKVIEW CERTIFICATION TRAINING](https://www.edureka.co/qlikview-certification-training" \o "QlikView Certification Training" \t "_blank)

### [QlikView Certification Training](https://www.edureka.co/qlikview-certification-training" \o "QlikView Certification Training" \t "_blank)

*[Reviews](https://www.edureka.co/qlikview-certification-training" \o "QlikView Certification Training" \t "_blank)*

**[5](https://www.edureka.co/qlikview-certification-training" \o "QlikView Certification Training" \t "_blank)**[(9505)](https://www.edureka.co/qlikview-certification-training" \o "QlikView Certification Training" \t "_blank)

### [PENTAHO BI CERTIFICATION TRAINING](https://www.edureka.co/pentaho-business-intelligence-training-self-paced" \o "Pentaho BI Certification Training" \t "_blank)

### [Pentaho BI Certification Training](https://www.edureka.co/pentaho-business-intelligence-training-self-paced" \o "Pentaho BI Certification Training" \t "_blank)

*[Reviews](https://www.edureka.co/pentaho-business-intelligence-training-self-paced" \o "Pentaho BI Certification Training" \t "_blank)*

**[4](https://www.edureka.co/pentaho-business-intelligence-training-self-paced" \o "Pentaho BI Certification Training" \t "_blank)**[(3365)](https://www.edureka.co/pentaho-business-intelligence-training-self-paced" \o "Pentaho BI Certification Training" \t "_blank)

### [MICROSTRATEGY 10 CERTIFICATION TRAINING](https://www.edureka.co/microstrategy-10-bi" \o "MicroStrategy 10 Certification Training" \t "_blank)

### [MicroStrategy 10 Certification Training](https://www.edureka.co/microstrategy-10-bi" \o "MicroStrategy 10 Certification Training" \t "_blank)

*[Reviews](https://www.edureka.co/microstrategy-10-bi" \o "MicroStrategy 10 Certification Training" \t "_blank)*

**[5](https://www.edureka.co/microstrategy-10-bi" \o "MicroStrategy 10 Certification Training" \t "_blank)**[(1736)](https://www.edureka.co/microstrategy-10-bi" \o "MicroStrategy 10 Certification Training" \t "_blank)

Next

### **POWER**

This DAX function returns the value of the first argument raised to the power of the second argument.

**Syntax**

POWER(<number>, <power>)

**Example**

=POWER(5,2)

### **QUOTIENT**

This DAX function performs division returns the integer part of the quotient.

**Syntax**

QUOTIENT(<dividend>, <divisor>)

**Example**

=QUOTIENT(5,2)

### **SIGN**

This DAX function returns the sign of a given number.

**Syntax**

SIGN(<number>)

**Example**

=SIGN( ([Sale Price] - [Cost Price]) )

### **SQRT**

This DAX function returns the square root of the given number.

**Syntax**

SQRT(<number>)

**Example**

=SQRT(25)

## ****5. Logical Functions****

### **AND**

This DAX function performs logical AND(conjunction) on two expressions. For AND to return true, both conditions specified have to be fulfilled.

**Syntax**

AND(<logical argument1>,<logical argument2>)

**Example**

=IF(AND(10 > 9, -10 < -1), "All true", "One or more false"

Because both conditions, passed as arguments, to the AND function are true, the formula returns "All True".

### **OR**

This DAX function performs logical OR(disjunction) on two expressions. For OR to return true, either of the two conditions specified has to be fulfilled.

**Syntax**

OR(<logical argument1>,<logical argument2>)

**Example**

=IF(OR(10 > 9, -10 >-1), "True", "False"

Because one of the conditions, passed as arguments, to the OR function is true, the formula returns "True".

### **NOT**

This DAX function performs logical NOT (negation) on given expression.

**Syntax**

NOT(<logical argument>)

**Example**

=NOT([CalculatedColumn1])

For each row in Calculated Column1, the NOT function returns the logical opposite of the given value.

### **IF**

This DAX function tests a series of inputs for the one that fulfills the condition specified in the argument.

**Syntax**

IF(logical\_test>,<value\_if\_true>, value\_if\_false)

**Example**

=IF([Calls]<200,"low",IF([Calls]<300,"medium","high"))

### **IFERROR**

This DAX function evaluates an expression and returns a specified value if the expression returns an error.

**Syntax**

IFERROR(value, value\_if\_error)

**Example**

=IFERROR(25/0,9999)

## ****6. Information Functions****

### **ISBLANK**

This DAX function returns TRUE or FALSE after checking whether a value is blank.

**Syntax**

ISBLANK(<value>)

**Example**

=IF( ISBLANK('CalculatedMeasures'[PreviousYearTotalSales]) , BLANK() , ( 'CalculatedMeasures'[Total Sales]-'CalculatedMeasures'[PreviousYearTotalSales] ) /'CalculatedMeasures'[PreviousYearTotalSales])

### **ISNUMBER**

This DAX function returns TRUE or FALSE after checking whether a value is numeric.

**Syntax**

ISNUMBER(<value>)

**Example**

=IF(ISNUMBER(0), "Is number", "Is Not number")

### **ISTEXT**

This DAX function returns TRUE or FALSE after checking whether a value is a text.

**Syntax**

ISTEXT(<value>)

**Example**

=IF(ISTEXT("text"), "Is Text", "Is Non-Text")

### **ISNONTEXT**

This DAX function returns TRUE or FALSE after checking whether a value is non-text.

**Syntax**

ISNONTEXT(<value>)

**Example**

=IF(ISNONTEXT("text"), "Is Non-Text", "Is Text")

### **ISERROR**

This DAX function returns TRUE or FALSE after checking whether a value is an error.

**Syntax**

ISERROE(<value>)

**Example**

=IF( ISERROR( SUM('ResellerSales\_USD'[SalesAmount\_USD]) /SUM('InternetSales\_USD'[SalesAmount\_USD]) ) , BLANK() , SUM('ResellerSales\_USD'[SalesAmount\_USD]) /SUM('InternetSales\_USD'[SalesAmount\_USD]) )

## ****7. Text Functions****

### **CONCATENATE**

This DAX function joins two text strings into one.

**Syntax**

CONCATENATE(<text1>, <text2>)

**Example**

=CONCATENATE("Hello ", "World")

### **CONCATENATEX**

This DAX function the result of an expression evaluated for each row in a table.

**Syntax**

CONCATENATEX(<table>, <expression>, [delimiter])

**Example**

=CONCATENATEX(Employees, [FirstName] & “ “ & [LastName], “,”)

### **FIXED**

This DAX function rounds a number to the specified number of decimals and returns the result as text.

**Syntax**

FIXED(<number>, <decimals>, <no\_commas>)

**Example**

=FIXED([PctCost],3,1)

### **REPLACE**

This DAX function replaces part of a text string, based on the number of characters you specify, with a different text string.

**Syntax**

REPLACE(<old\_text>, <start\_num>, <num\_chars>, <new\_text>)

**Example**

=REPLACE('New Products'[Product Code],1,2,"OB")

### **SEARCH**

This DAX function returns the number of the characters at which a specific text string is first found.

**Syntax**

SEARCH(<find\_text>, <within\_text>[, [<start\_num>][, <NotFoundValue>]])

**Example**

=SEARCH("n","printer")

The formula returns 4 because "n" is the fourth character in the word "printer."

### **UPPER**

This DAX function returns a text string in all uppercase letters.

**Syntax**

UPPER (<text>)

**Example**

=UPPER(['New Products'[Product Code])