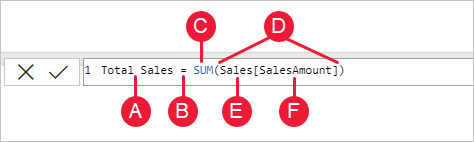
**What is DAX?**

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.

DAX around three fundamental concepts: Syntax, Functions, and Context.



The measure name, **Total Sales**.

**B.** The equals sign operator (**=**), which indicates the beginning of the formula. When calculated, it will return a result.

**C.** The DAX function **SUM**, which adds up all of the numbers in the **Sales[SalesAmount]** column. You’ll learn more about functions later.

**D.** Parenthesis **()**, which surround an expression that contains one or more arguments. Most functions require at least one argument. An argument passes a value to a function.

**E.** The referenced table, **Sales**.

**F.** The referenced column, **[SalesAmount]**, in the Sales table. With this argument, the SUM function knows on which column to aggregate a SUM.

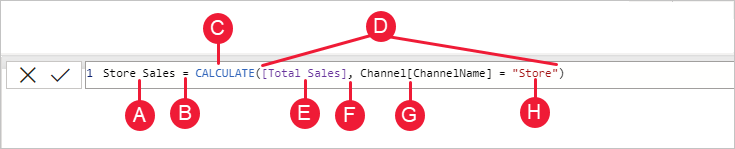
DAX formulas can contain up to 64 nested functions. It’s unlikely a formula would ever contain so many nested functions. In fact, such a formula would be difficult to create and debug, and it probably wouldn’t be very fast either.

DAX includes the following categories of functions: [Date and Time](https://learn.microsoft.com/en-us/dax/date-and-time-functions-dax), [Time Intelligence](https://learn.microsoft.com/en-us/dax/time-intelligence-functions-dax), [Information](https://learn.microsoft.com/en-us/dax/information-functions-dax), [Logical](https://learn.microsoft.com/en-us/dax/logical-functions-dax), [Mathematical](https://learn.microsoft.com/en-us/dax/math-and-trig-functions-dax), [Statistical](https://learn.microsoft.com/en-us/dax/statistical-functions-dax), [Text](https://learn.microsoft.com/en-us/dax/text-functions-dax), [Parent/Child](https://learn.microsoft.com/en-us/dax/parent-and-child-functions-dax), and [Other](https://learn.microsoft.com/en-us/dax/other-functions-dax) functions.

* A DAX function always references a complete column or a table. If you want to use only particular values from a table or column, you can add filters to the formula.
* If you need to customize calculations on a row-by-row basis, DAX provides functions that let you use the current row value or a related value as a kind of argument, to perform calculations that vary by context.
* DAX includes many functions that return a table rather than a value. The table isn't displayed, but is used to provide input to other functions. For example, you can retrieve a table and then count the distinct values in it, or calculate dynamic sums across filtered tables or columns.
* DAX includes a variety of time intelligence functions. These functions let you define or select date ranges, and perform dynamic calculations based on them. For example, you can compare sums across parallel periods.

Excel has a popular function, VLOOKUP. DAX functions don’t take a cell or cell range as a reference like VLOOKUP does in Excel. DAX functions take a column or a table as a reference. Keep in mind, in Power BI Desktop you’re working with a relational data model.

 filter context can most easily be applied by adding fields to a visualization, filter context can also be applied in a DAX formula by defining a filter using functions such as ALL, RELATED, FILTER, CALCULATE, by relationships, and by other measures and columns. For example, let’s look at the following formula in a measure named Store Sales:



**A.** The measure name, **Store Sales**.

**B.** The equals sign operator (**=**), which indicates the beginning of the formula.

**C.** The **CALCULATE** function, which evaluates an expression, as an argument, in a context that is modified by the specified filters.

**D.** Parenthesis **()**, which surround an expression containing one or more arguments.

**E.** A measure **[Total Sales]** in the same table as an expression. The Total Sales measure has the formula: =SUM(Sales[SalesAmount]).

**F.** A comma (**,**), which separates the first expression argument from the filter argument.

**G.** The fully qualified referenced column, **Channel[ChannelName]**. This is our row context. Each row in this column specifies a channel, such as Store or Online.

**H.** The particular value, **Store**, as a filter. This is our filter context.

This formula ensures only sales values defined by the Total Sales measure are calculated only for rows in the Channel[ChannelName] column, with the value Store used as a filter.

# APPROXIMATEDISTINCTCOUNT

Returns the approximate number of rows that contain distinct values in a column. This function can query large amounts of data with potentially better performance than DISTINCTCOUNT, with slight deviation from the exact result.

APPROXIMATEDISTINCTCOUNT(<columnName>)

# **AVERAGE**

Returns the average (arithmetic mean) of all the numbers in a column.

Nonnumeric values in the column are handled as follows:

If the column contains text, no aggregation can be performed, and the functions returns blanks.

If the column contains logical values or empty cells, those values are ignored.

Cells with the value zero are included.

When you average cells, you must keep in mind the difference between an empty cell and a cell that contains the value 0 (zero). When a cell contains 0, it is added to the sum of numbers and the row is counted among the number of rows used as the divisor. However, when a cell contains a blank, the row is not counted.

# **AVERAGEA**

Returns the average (arithmetic mean) of the values in a column. Handles text and non-numeric values.