

## DAX Functions - P2

---

## MEET DAX

---

Data Analysis Expressions (commonly known as DAX) is the formula language that drives the Power BI front-end. With DAX, you can:

- Go beyond the capabilities of traditional spreadsheet formulas, with powerful and flexible functions built specifically to work with relational data models.
- Add calculated columns (for filtering) and measures (for aggregation) to enhance data models.



## M VS. DAX

---

M and DAX are two distinct functional languages used within Power BI Desktop:

- M is used in the Power Query editor, and is designed specifically for extracting, transforming and loading data.
- DAX is used in the Power BI front-end, and is designed specifically for analyzing relational data models

## CALCULATED COLUMNS

Calculated columns allow you to add new, formula-based columns to tables in a model

- Calculated columns refer to entire tables or columns (no A1-style cell references).
- Calculated columns generate values for each row, which are visible within tables in the Data view.
- Calculated columns understand row context; they're great for defining properties based on information in each row, but generally useless for aggregation (sum, count, etc.)

### HEY THIS IS IMPORTANT!

As a rule of thumb, use calculated columns to "stamp" static, fixed values to each row in a table (or go upstream and use the Query Editor!) DO NOT use calculated columns for aggregation – this is what measures are for!

### PRO TIP:

Calculated columns are typically used for filtering & grouping data, rather than creating aggregate numerical values

## DAX MEASURES

Measures are DAX formulas used to generate new calculated values

- Like calculated columns, measures reference entire tables or columns (no A1-style cell references).
- Unlike calculated columns, measures aren't visible within tables; they can only be "seen" within a visualization like a chart or matrix (similar to a calculated field in a PivotTable).
- Measures evaluate based on filter context, which means they recalculate when the fields or filters around them change.

Filter Context

## Filter Context

### HEY THIS IS IMPORTANT!

As a rule of thumb, use measures when a single row can't give you the answer, or when you need to aggregate values across multiple rows in a table

### PRO TIP:

Use measures to create numerical, calculated values that can be analyzed in the "values" field of a report visual

STATUS	Large	Medium	Small	Total
Cancelled		1247	791	2038
Disputed	296	184	117	597
In Process	153	762	575	1490
On Hold	373	1097	409	1879
Resolved	45	1010	605	1660
Shipped	6547	48219	36637	91403
<b>Total</b>	<b>7414</b>	<b>52519</b>	<b>39134</b>	<b>99067</b>

Aggregated Results

## IMPLICIT VS. EXPLICIT MEASURES

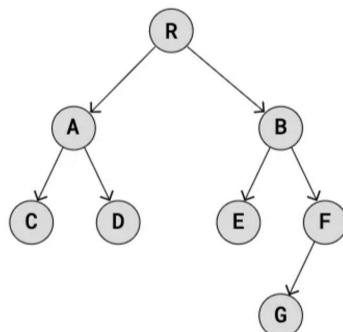
Implicit measures are created when you drag raw numerical fields into a report visual and manually select an aggregation mode (Sum, Average, Min, Max, Count, etc.)

Explicit measures are created when you actually write a DAX formula and define a new measure that can be used within the model

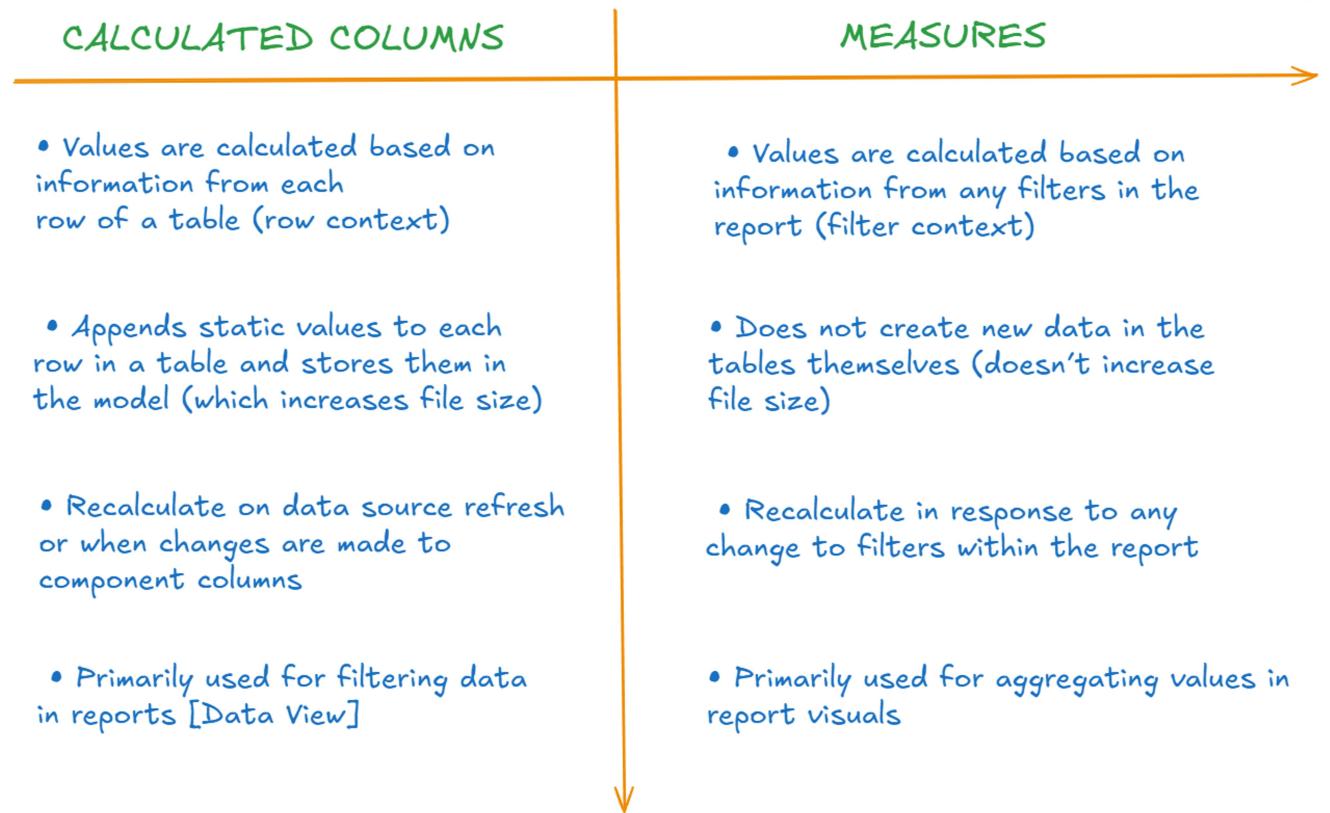
### HEY THIS IS IMPORTANT!

Implicit measures are only accessible within the specific visualization in which they were created, and cannot be referenced elsewhere.

Explicit measures can be used anywhere in the report, and referenced by other DAX calculations to create "measure trees".



## RECAP: CALCULATED COLUMNS VS. MEASURES



In Power BI, slicers apply filters to the report view, which affects the filter context of the visuals in the report. When a slicer is used, it modifies the filter context by adding or removing data points based on the selections made in the slicer.

The screenshot shows the Power BI Data View interface. On the left, the 'Queries [8]' pane lists several queries including 'Customers', 'Sales - 1', 'Product Information Po...', 'Students Info', 'Sales - 2', 'Sales Record', 'Sales detail', and 'Vehicle Orders'. The 'Vehicle Orders' query is selected. The main area displays a table with columns: PRODUCTLINE, CITY, STATE, POSTALCODE, COUNTRY, TERRITORY, and CONTACTLAST. Each column has a corresponding histogram and a percentage breakdown of 'Valid', 'Error', and 'Empty' values. The table contains 46 rows of motorcycle sales data. On the right, the 'Query Settings' pane shows the 'Vehicle Orders' query settings with the 'Name' field set to 'Vehicle Orders'. The 'APPLIED STEPS' section lists the following steps: Source, Navigation, Promoted Headers, Changed Type, Filtered Rows, Replaced Value, and Replaced Value1. The 'Replaced Value1' step is currently selected.

1 Sales Value = 'Vehicle Orders'[QUANTITYORDERED] \* 'Vehicle Orders'[PRICEEACH]

QUANTITYORDERED	PRICEEACH	DELIVERYDATETIME	STATUS	PRODUCTLINE	CITY	STATE	POSTALCODE	COUNTRY	TERRITORY	CONTACTLASTNAME	CONTACTFIRSTNAME	DEALSIZE	Sales Value
39	\$100	01-11-2013 02:20:49	Shipped	Vintage Cars	NYC	NY	10022	USA	NA	Frick	Michael	Medium	\$3,900
42	\$100	30-05-2012 19:21:13	Shipped	Classic Cars	NYC	NY	10022	USA	NA	Young	Jeff	Medium	\$4,200
48	\$100	03-08-2014 18:54:11	Shipped	Trucks and Buses	NYC	NY	10022	USA	NA	Young	Jeff	Medium	\$4,800
21	\$100	27-08-2016 10:00:06	Shipped	Trucks and Buses	NYC	NY	10022	USA	NA	Yu	Kwai	Small	\$2,100
46	\$100	24-07-2013 18:30:11	Shipped	Trucks and Buses	NYC	NY	10022	USA	NA	Young	Jeff	Small	\$2,400
33	\$100	30-08-2012 01:37:01	Shipped	Trucks and Buses	NYC	NY	10022	USA	NA	Hernandez	Maria	Medium	\$4,600
26	\$100	04-09-2016 23:20:20	Shipped	Classic Cars	NYC	NY	10022	USA	NA	Yu	Kwai	Medium	\$3,300
42	\$100	15-06-2013 04:53:57	Shipped	Classic Cars	NYC	NY	10022	USA	NA	Young	Jeff	Medium	\$2,600
24	\$100	17-09-2012 09:42:37	Shipped	Classic Cars	NYC	NY	10022	USA	NA	Young	Jeff	Large	\$4,200
30	\$100	18-10-2013 13:55:59	Shipped	Classic Cars	NYC	NY	10022	USA	NA	Young	Jeff	Large	\$4,600
32	\$100	02-05-2012 08:20:00	Shipped	Motorcycles	NYC	NY	10022	USA	NA	Yu	Kwai	Medium	\$2,400
21	\$100	07-08-2015 19:17:43	Shipped	Motorcycles	NYC	NY	10022	USA	NA	Kuo	Kee	Medium	\$3,000
46	\$100	24-02-2015 05:35:28	Shipped	Motorcycles	NYC	NY	10022	USA	NA	Frick	Michael	Medium	\$3,200
26	\$100	30-11-2015 01:19:05	Shipped	Motorcycles	NYC	NY	10022	USA	NA	Yu	Kwai	Medium	\$2,100
39	\$100	12-04-2016 08:44:09	Shipped	Classic Cars	NYC	NY	10022	USA	NA	Young	Jeff	Large	\$4,600
27	\$100	02-03-2014 10:47:13	Shipped	Motorcycles	NYC	NY	10022	USA	NA	Yu	Kwai	Medium	\$2,600
20	\$100	07-11-2014 01:15:44	Shipped	Motorcycles	NYC	NY	10022	USA	NA	Frick	Michael	Large	\$3,900
21	\$100	16-04-2012 06:32:28	Shipped	Motorcycles	NYC	NY	10022	USA	NA	Yu	Kwai	Medium	\$2,700
42	\$100	03-07-2014 19:18:00	Shipped	Motorcycles	NYC	NY	10022	USA	NA	Yu	Kwai	Medium	\$2,000
23	\$100	05-03-2013 19:24:20	Shipped	Classic Cars	NYC	NY	10022	USA	NA	Hernandez	Maria	Medium	\$2,100

## Measure Tables

### Option - 1

File Home Help Table tools Column tools

Paste Cut Copy Get data v workbook catalog v OneLake Server Enter data Data Transform Refresh data v Queries Manage relationships Relationships New measure measure column New New table Manage roles View as Security

1 Sales Value = 'Vehicle Orders'[QUANTITYORDERED] \* 'Vehicle Orders'[PRICEEACH]

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	ORDERDATETIME	DELIVERYDATETIME	STATUS	PRODUCT	
10204		42	\$100		17 29-05-2012 00:15:00	30-05-2012 19:21:13	Shipped	Classic Ca
10349		48	\$100		9 01-08-2014 22:32:00	03-08-2014 18:54:11	Shipped	Trucks and
10292		21	\$100		8 24-08-2016 23:51:00	27-08-2016 10:00:06	Shipped	Trucks and

1 Measure Table (DAX) = {" "}

Value Data

Q. Search

> Customers

> Measure Table (DAX)

Value

> Product Information Power BI

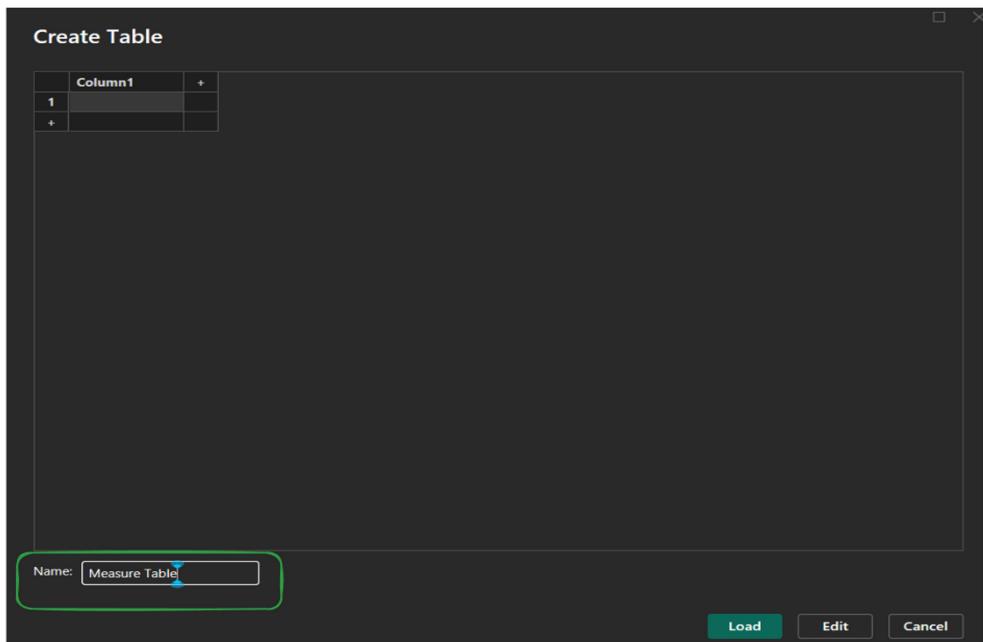
### Option - 2

File Home Insert Modeling View Optimize Help

Paste Cut Copy Get data v workbook catalog v OneLake Server Enter Data Data

STATUS Large Medium Small Total

Cancelled	1247	791	20:
-----------	------	-----	-----



Revenue

Measure Table (DAX)

Structure

1 Revenue = SUM('Vehicle Orders'[Sales Value])

Format Currency

\$ \$ % .00 Auto

Data category Uncategorized

Properties

Measure Table (DAX)

Revenue

Value

## SUMX - Iterative Functions [Sum Product]

Measure

Measure Table (DAX)

Structure

1 Total Re

02 SUMX(

SUMX(Table, Expression)

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

Total Revenues

Measure Table (DAX)

Structure

1 Total Revenues =

02 SUMX(

03 'Vehicle Orders',

04 'Vehicle Orders'[QUANTITYORDERED] \* 'Vehicle Orders'[PRICEEACH])

\$8.29M

Total Revenues

New Quick measure measure Calculations

QUANTITYORDERED	PRICEEACH
39	\$100
42	\$100
48	\$100
21	\$100
24	\$100
46	\$100
33	\$100
26	\$100
42	\$100
46	\$100
24	\$100
30	\$100
32	\$100

$$(39 * 100) + (42 * 100) + (48 * 100) + \dots$$

`=SUMPRODUCT(A2:A16,B2:B16)`

A	B	C	D
Quantity	Price	Sales	Revenue
10	100	1000	1240000
20	200	4000	
30	300	9000	
40	400	16000	
50	500	25000	
60	600	36000	
70	700	49000	1240000
80	800	64000	
90	900	81000	
100	1000	100000	
110	1100	121000	
120	1200	144000	
130	1300	169000	
140	1400	196000	
150	1500	225000	