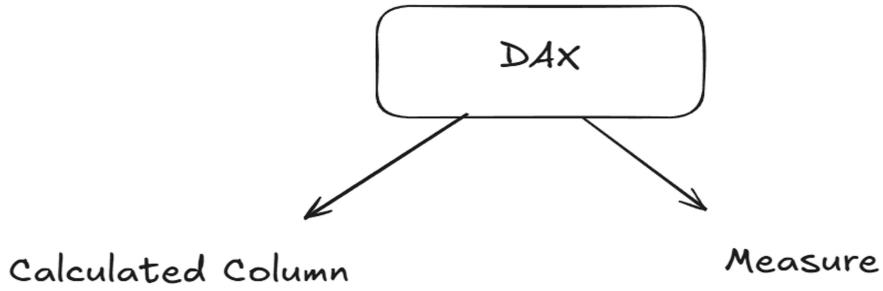
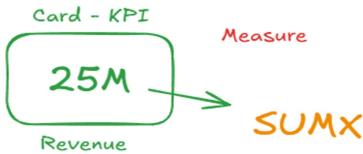


Power BI - DAX - p1

Row Context

1 Sales = 'Sales Record 2020-2022'[OrderQuantity] * 'Sales Record 2020-2022'[Retail Price]										
OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity	Quantity Type	Retail Price	Sales
16-12-2021	28-09-2021	SO60148	528	19077	10	1	3	Multiple Items	\$4.99	\$14.97
15-12-2021	18-08-2021	SO60071	528	15849	1	1	3	Multiple Items	\$4.99	\$14.97
15-12-2021	09-09-2021	SO60062	528	23241	1	1	3	Multiple Items	\$4.99	\$14.97
14-12-2021	02-10-2021	SO60010	528	16284	8	1	3	Multiple Items	\$4.99	\$14.97
14-12-2021	19-10-2021	SO60011	528	17791	8	1	3	Multiple Items	\$4.99	\$14.97
08-12-2021	31-10-2021	SO59620	528	12101	4	1	3	Multiple Items	\$4.99	\$14.97



- Creating a new column by row context

- Filter Context

New Table : Measure Table

I store all the measure I calculate here.

Implicit

Data

Values

Sum of OrderQuantity

Sum of ReturnQuantity

Aggregation

Sum

Explicit Measure

- To perform Complex Calculation

[total Orders] - SUM(orders)

Further complex measure can be created.
[Bikes]

- CALCULATE(measure , filter);
 - Calculate[[Total Orders], 'Category Lookup'[Category Name] = "Bikes"]

CALCULATED COLUMNS

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

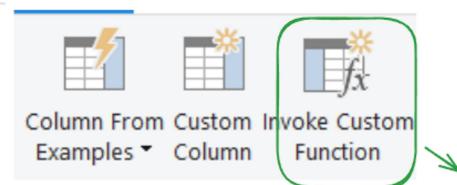
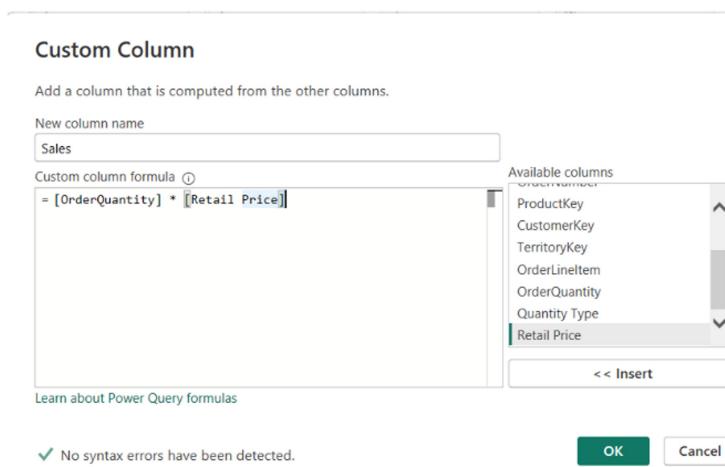
- Calculated columns refer to entire tables or columns (no A1-style cell references).
- Calculated columns generate values for each row, which are visible within tables in the Data view.
- Calculated columns understand **row context**; they're great for defining properties based on information in each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

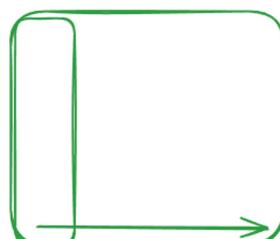
PRO TIP:

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



Rolling Calendar

1/1/2020 - rolling calendar



T/F

BirthDate	MaritalStatus	Gender	EmailAddress	AnnualIncome	TotalChildren	EducationLevel	Occupation	HomeOwner	Domain Name	Parent
01-04-1964	S	M	shannon38@adventure-works.com	\$70,000	0	Bachelors	Professional	N	Adventure Works	No
11-09-1964	S	M	jessie9@adventure-works.com	\$70,000	0	Bachelors	Professional	N	Adventure Works	No
05-11-1963	S	M	ruben1@adventure-works.com	\$70,000	0	Bachelors	Professional	N	Adventure Works	No
09-11-1974	S	M	ruben30@adventure-works.com	\$70,000	0	Bachelors	Professional	N	Adventure Works	No
07-09-1965	S	M	joe14@adventure-works.com	\$70,000	0	Bachelors	Professional	N	Adventure Works	No
20-12-1963	S	M	jarrod0@adventure-works.com	\$70,000	0	Bachelors	Professional	N	Adventure Works	No
09-10-1975	S	M	dustin20@adventure-works.com	\$70,000	0	Bachelors	Professional	N	Adventure Works	No
08-07-1976	S	M	clayton29@adventure-works.com	\$70,000	0	Bachelors	Professional	N	Adventure Works	No
04-03-1976	S	M	irving0@adventure-works.com	\$70,000	0	Bachelors	Professional	N	Adventure Works	No
11-02-1974	S	M	alan10@adventure-works.com	\$70,000	0	Bachelors	Professional	N	Adventure Works	No

Martial Status - "M", Total Children > 0 ---- Parent.



Parent 1 = IF(AND('Customer Lookup'[MaritalStatus] = "M" , 'Customer Lookup'[TotalChildren] > 0), "Yes" , "No")

EXAMPLE: CALCULATED COLUMNS

X ✓ 1 Parent = IF('Customer Lookup'[Total Children]>0,"Yes","No")										
Fields										
Emma Address										
barry20@adventure-works.com										
martha13@adventure-works.com										
tamara16@adventure-works.com										
gerald21@adventure-works.com										
alex8@adventure-works.com										
jack53@adventure-works.com										
ricky1@adventure-works.com										
keith4@adventure-works.com										
latoya19@adventure-works.com										

In this case we've added a calculated column named Parent, which equals "Yes" if the [Total Children] field is greater than 0, and "No" otherwise

- Since calculated columns understand row context, a new value is calculated in each row based on the value in the [Total Children] column.
- This is a valid use of calculated columns; it creates a new row "property" that we can use to filter or segment any related data within the model.

Here we're using an aggregation function (SUM) to calculate a new column named Total Quantity

X ✓ 1 TotalQuantity = SUM('Sales Data'[Order Quantity])										
Fields										
Order Date										
Order Number										
Product Key										
Customer Key										
Territory Key										
Order Line Item										
Order Quantity										
Index										
TotalQuantity										

- Since this is an aggregation function, the same grand total is returned in every row of the table
- This is not a valid use of calculated columns; these values are statically "stamped" onto the table and can't be filtered, sliced, etc.

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.

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

IMPLICIT VS. EXPLICIT MEASURES

The screenshot shows the 'Build a visual' interface in Power BI. On the left, under 'X-axis', there is a field 'Year'. Under 'Y-axis', there is a field 'Sum of Order ...'. A yellow box highlights the 'Sum of Order ...' field. To the right, a 'Select data' pane is open, showing various data sources and fields. A yellow box highlights the field 'Σ Order Quantity'. An arrow points from the highlighted 'Sum of Order ...' field to the highlighted 'Σ Order Quantity' field in the 'Select data' pane.

Example of an implicit measure

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.

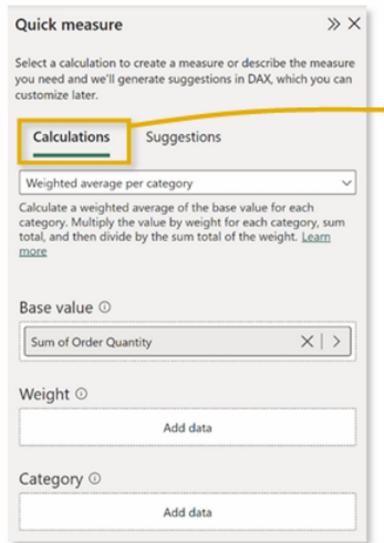
Explicit measures can be used anywhere in the report, and referenced by other DAX calculations to create "measure trees".

The screenshot shows the 'Build a visual' interface in Power BI. Under 'Values', there are two fields: 'Sum of OrderQuantity' and 'Sum of ReturnQuantity'. A blue box highlights the 'Sum of OrderQuantity' field. To the right, a dropdown menu is open, showing various aggregation functions: Sum, Average, Minimum, Maximum, Count (Distinct), Count, Standard deviation, Variance, and Median. A blue box highlights the 'Sum' option at the top of the list. An arrow points from the highlighted 'Sum of OrderQuantity' field to the highlighted 'Sum' option in the dropdown menu.

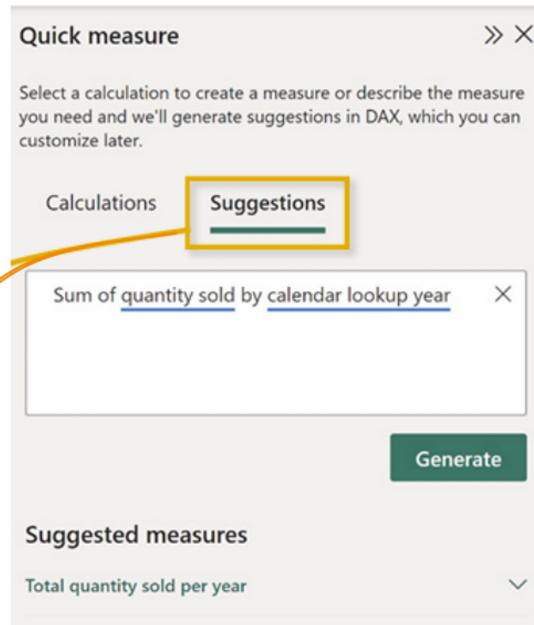
Implicit Measures.

QUICK MEASURES

Quick measures automatically create formulas based on pre-built templates or natural language prompts



Quick measure calculations can be used to build measures using predefined templates (weighted averages, percent difference, time intelligence, etc.)



Quick measure suggestions can be used to find suggested measures based on natural language queries (i.e. "sum of quantity sold by calendar year")

PRO TIP:

Quick measures can be a great learning tool for beginners or for building more complex formulas but use them with caution; mastering DAX requires a deep understanding of the underlying theory!

RECAP: CALCULATED COLUMNS VS. MEASURES

CALCULATED COLUMNS

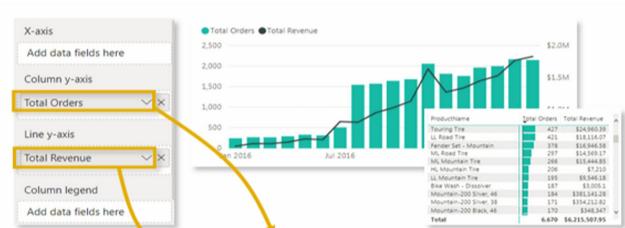
- Values are calculated based on information from each row of a table (row context)
- Appends static values to each row in a table and stores them in the model (which increases file size)
- Recalculate on data source refresh or when changes are made to component columns
- Primarily used for filtering data in reports

	Birth Date	Marital Status	Email Address	Annual Income	Total Children	Education Level	Parent
9/3/1943	M	emma32@adventure-works.com	70000	5	Bachelors	Yes	
9/4/1967	M	barny20@adventure-works.com	40000	3	High School	Yes	
8/5/1945	M	marth11@adventure-works.com	70000	3	High School	Yes	
6/4/1946	S	tamar16@adventure-works.com	40000	5	High School	Yes	
10/16/1970	S	gerald12@adventure-works.com	130000	5	Bachelors	Yes	
5/10/1945	M	alexa10@adventure-works.com	40000	3	High School	Yes	
9/24/1998	M	jacks13@adventure-works.com	70000	3	Graduate Degree	Yes	
7/23/1965	S	rich13@adventure-works.com	200000	3	BA - College	Yes	
1/9/1962	M	laura19@adventure-works.com	70000	5	Partial College	Yes	
8/15/1962	M	larry19@adventure-works.com	70000	5	Bachelors	Yes	
1/26/1967	S	michael11@adventure-works.com	70000	5	Bachelors	Yes	
3/8/1946	M	mindy22@adventure-works.com	80000	5	Partial College	Yes	
6/21/1960	M	teresa18@adventure-works.com	70000	5	Partial College	Yes	

Calculated columns "live" in tables

MEASURES

- Values are calculated based on information from any filters in the report (filter context)
- Does not create new data in the tables themselves (doesn't increase file size)
- Recalculate in response to any change to filters within the report
- Primarily used for aggregating values in report visuals



Measures "live" in visuals

ProductColor	Sum of OrderQuantity	Quantity Sold	Red Product Qty Sold	Black Product Qty Sold
Black	10590	10590	4011	10590
Blue	3779	3779	4011	10590
Grey			4011	10590
Multi	5756	5756	4011	10590
NA	51080	51080	4011	10590
Red	4011	4011	4011	10590
Silver	3257	3257	4011	10590
Silver/Black			4011	10590
White	1063	1063	4011	10590
Yellow	4638	4638	4011	10590
Total	84174	84174	4011	10590

4011 10590

Red Product Qty Sold

Black Product Qty Sold

Calling out existing measure.....

```
Red Product Qty Sold = CALCULATE([Quantity Sold], 'Product Lookup'[ProductColor] = "Red")
```

```
Black Product Qty Sold = CALCULATE([Quantity Sold], 'Product Lookup'[ProductColor] = "Black")
```

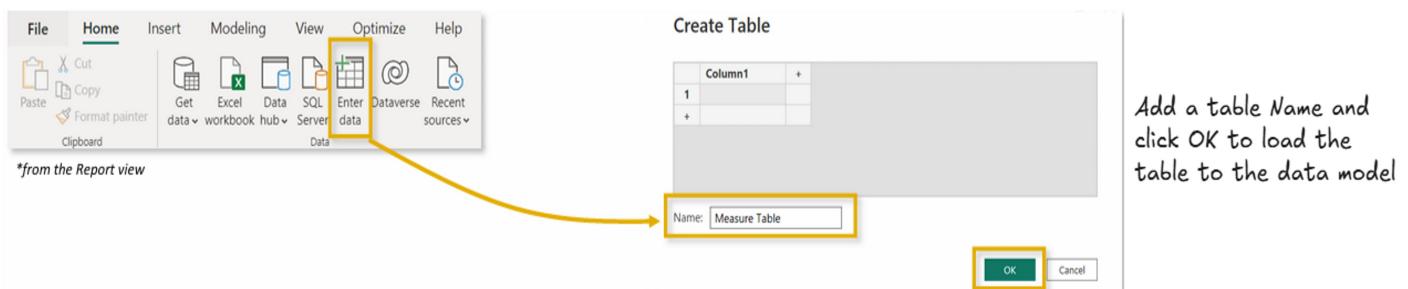
```
Quantity Sold = SUM('Sales Record 2020-2022'[OrderQuantity])
```

Explicit Measure help us to do complex calculations. I hope you remember the difference :)

PRO TIP: MEASURE TABLES

It's a common best practice to create a dedicated table to store your measures; this will help you stay organized, find measures quickly, and allow you to group related measures into folders.

Option 1: Enter Data into Power Query (loads the table to the data model – table is visible in Power Query)



Option 2: Create a calculated table using DAX directly in the model (table is not visible in Power Query)



FILTER CONTEXT

Measures are evaluated based on filter context, which means that they recalculate whenever the fields or filters around them change

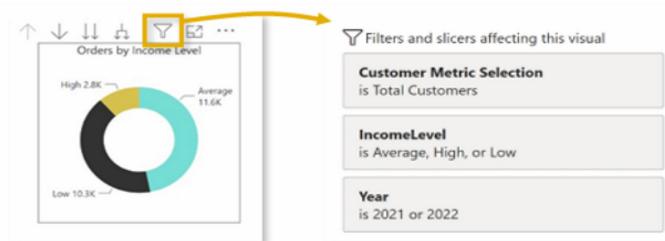
Top 10 Products	Orders	Revenue	Return %
Water Bottle - 30 oz.	3,983	\$39,755	1.95%
Patch Kit/8 Patches	2,952	\$13,506	1.61%
Mountain Tire Tube	2,846	\$28,333	1.64%
Road Tire Tube	2,173	\$17,265	1.55%
Sport-100 Helmet, Red	2,099	\$73,444	3.33%
AWC Logo Cap	2,062	\$35,865	1.11%
Sport-100 Helmet, Blue	1,995	\$67,112	3.31%
Fender Set - Mountain	1,975	\$87,041	1.36%
Sport-100 Helmet, Black	1,940	\$65,262	2.68%
Mountain Bottle Cage	1,896	\$38,062	2.02%
Total	15,587	\$465,644	1.85%

For this value in the matrix (2,846), the Orders measure is calculated based on the following filter context: Products[Product Name] = "Mountain Tire Tube"

- This allows the measure to return the total order quantity for each product specifically (or whatever context the row and column labels dictate – years, countries, categories, customer names, etc.)

This total (15,587) does NOT calculate by summing the values above; it evaluates as an independent measure with no filter context applied

- IMPORTANT:** Every measure value in a report evaluates independently (like an island) and calculates based on its own filter context



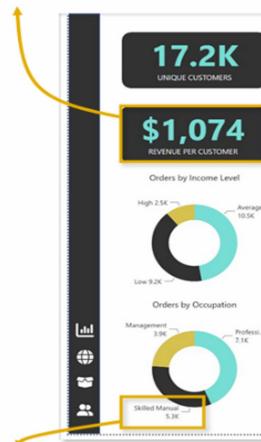
PRO TIP: Clicking the filter icon will show you the filters currently applied to a selected visual

EXAMPLE: FILTER CONTEXT

MEASURE: Revenue Per Customer

FILTER CONTEXT:

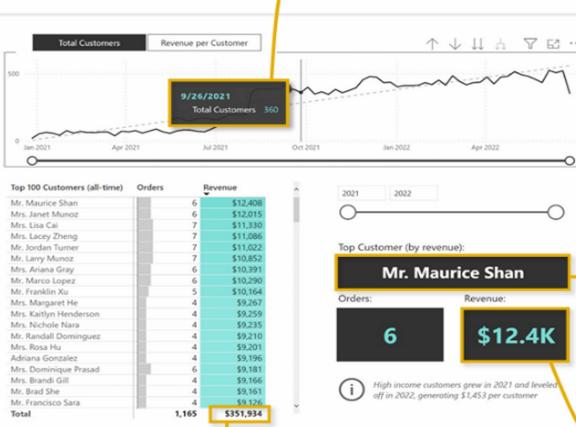
- Calendar[Year] = 2021 or 2022



MEASURE: Total Customers

FILTER CONTEXT:

- Calendar[Date] = September 26, 2021

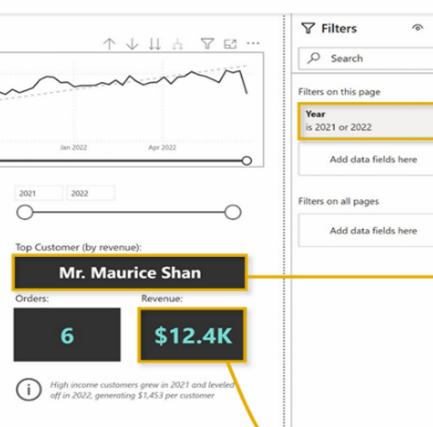


This is a **page-level filter**, which impacts **ALL** visuals on this report page (more on this later!)

COLUMN: Customer Full Name

FILTER CONTEXT:

- Calendar[Year] = 2021 or 2022
- Customer[Full Name] = Top 100 by Total Revenue



MEASURE: Total Orders

FILTER CONTEXT:

- Calendar[Year] = 2021 or 2022
- Customers[Occupation] = Skilled Manual

MEASURE: Total Revenue

FILTER CONTEXT:

- Calendar[Year] = 2021 or 2022
- Customer[Full Name] = Top 100 by Total Orders

MEASURE: Total Revenue

FILTER CONTEXT:

- Calendar[Year] = 2021 or 2022
- Customer[Full Name] = Mr. Maurice Shan

STEP-BY-STEP MEASURE CALCULATION

Product Color	Quantity Sold
Black	10,590
Red	4,011
Yellow	4,638

How exactly is this measure value calculated?

- NOTE: This all happens instantly behind the scenes, every time the filter context changes

STEP 1

Filter context is detected & applied



Product Color	Quantity Sold
Black	10,590
Red	4,011
Yellow	4,638

'Product Lookup'[Product Color] = "Black"

The screenshot shows the 'Product Lookup Table' in Power BI. A funnel icon is positioned over the 'Black' row, which is highlighted with a yellow box. The 'Search' pane on the right shows the filter context: 'Product Color' is set to 'Black'. Other filters like 'Category' and 'Subcategory' are also visible.

STEP 2

Filters flow “downstream” to related tables



The screenshot displays several tables in Power BI: 'Sales Data', 'Returns Data', and 'Order Details'. Each table has a funnel icon over its 'Black' row, indicating that the filter context from Step 1 has been applied downstream. The 'Order Details' table specifically shows a breakdown of sales by date and order number, all filtered for the 'Black' product.

STEP 3

Measure evaluates against the
filtered table



```
1 Quantity Sold =  
2 SUM(  
3 |   'Sales Data'[Order Quantity]  
4 )
```

*Sum of values in the **Order Quantity** column of the **Sales Data** table, filtered to rows where the product color is “**Black**”*

= **10,590**