

## A. What is difference between calculated column and calculated measure?

**ANS:**

### 1. CALCULATED COLUMN:

Data is stored in the xVelocity in-memory database, meaning that a calculation is made before the model is ever queried by the user. From a performance perspective, calculated columns are positive given that there is a smaller virtual memory requirement as the user interacts with the report. The downside is that calculated columns take up more storage in your database.

e.g. NewColumn:=UPPER(ExistingColumnName)

### 2. CALCULATED MEASURE:

They are used to calculate aggregates, such as the sum or average of a column. Measures are calculated at the time of your query, which means that they aren't stored in your database, but use processing power to execute a query at the time of your request. Because measures are not stored in memory, they are generally faster.

e.g. TotalSales:=SUM(DailySales)

Honestly saying, it is important to recognize the tradeoff between utilizing in-memory storage or processing power in an instance where either a measure or a calculated column could be used.

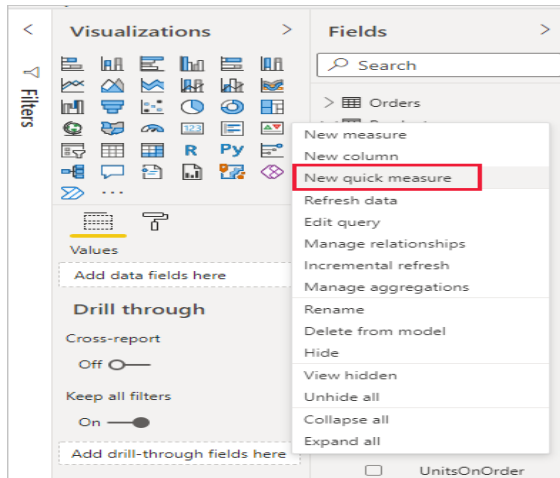
<i>Calculated column</i>	<i>Calculated measure</i>
➤ Calculated based on the information in each row	➤ Calculated based on the information from any filter in the report
➤ Appends static values in each row in the table	➤ Does not create a new data in the tables
➤ Table file size increases in power bi	➤ Does not increases the file size
➤ Mostly used as rows /columns/ filter in reports	➤ Always used within the values fields

## B. Explain Quick Measures?

**ANS:**

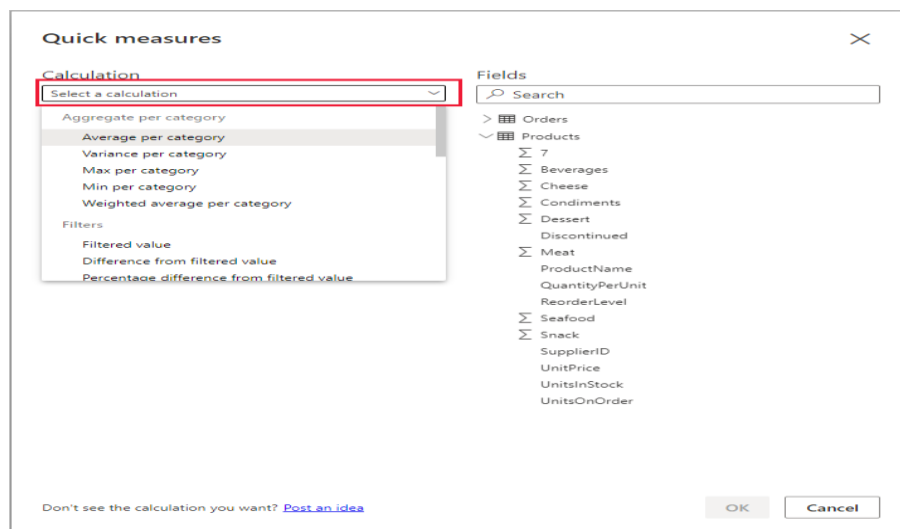
### Step1:

To create a quick measure in Power BI Desktop, right-click or select the ellipsis ... next to any item in the Fields pane, and select New quick measure from the menu that appears.



### Step2:

You can also right-click or select the drop-down arrow next to any value in the Values well for an existing visual, and select New quick measure from the menu. When you select New quick measure, the Quick measures window appears, letting you select the calculation you want and the fields to run the calculation against. Select the Select a calculation field to see a long list of available quick measures.



You can also right-click or select the drop-down arrow next to any value in the Values. The five quick measure calculation types, with their calculations, are:

#### 1. Aggregate per category

- Average per category
- Variance per category
- Max per category
- Min per category
- Weighted average per category

#### 2. Filters

- Filtered value
- Difference from filtered value
- Percentage difference from filtered value
- Sales from new customers

#### 3. Time intelligence

- Year-to-date total
- Quarter-to-date total
- Month-to-date total
- Year-over-year change
- Quarter-over-quarter change
- Month-over-month change
- Rolling average

#### 4. Totals

- Running total
- Total for category (filters applied)
- Total for category (filters not applied)

#### 1. Mathematical operations

- Addition
- Subtraction
- Multiplication
- Division
- Percentage difference
- Correlation coefficient
- Text
- Star rating
- Concatenated list of values...
- well for an existing visual, and select New quick measure from the menu.

## C. Explain types of operators?

**ANS:**

### A. Arithmetic operators :

+ (plus sign) - Addition

Example => 9+3

– (minus sign) - Subtraction or sign

Example => 9-3

\* (asterisk) - Multiplication

Example => 9\*3

/ (Forward slash) - Division

Example => 9/3

## **B. Comparison operators :**

= - Equal to

Example => [Name] = "Roy"

== - Strict Equal to

Example => [Name] == "Roy"

> - Greater than

Example => [Salary] > 100000

>= - Greater than equal to

Example => [Salary] >= 100000

< - Less than

Example => [Salary] < 100000

<= - Less than equal to

Example => [Salary] <= 100000

## **C. Text concatenation operator :**

& (ampersand) => Concatenation

Example => [First Name] & ' ' & [Last Name]

## **D. Logical operators :**

&& (double ampersand) - Creates an AND condition between two expressions

Example => [Name] = "Roy" && [Salary] > 10000

|| (double pipe symbol) - Creates an OR condition between two expressions

Example => [Name] = "Roy" || [Salary] > 10000

## D. Explain DAX syntax?

### ANS:

A great advantage of quick measures is that they show you the DAX formula that implements the measure. When you select a quick measure in the Fields pane, the Formula bar appears, showing the DAX formula that Power BI created to implement the measure.

The formula bar not only shows you the formula behind the measure, but perhaps more importantly, lets you see how to create the DAX formulas underlying quick measures.

Imagine you need to do a year-over-year calculation, but you're not sure how to structure the DAX formula, or you have no idea where to start. Instead of banging your head on the desk, you can create a quick measure using the Year-over-year change calculation, and see how it appears in your visual and how the DAX formula works. Then you can either make changes directly to the DAX formula, or create a similar measure that meets your needs and expectations. It's like having a teacher that immediately responds to what-if questions you ask with a few clicks.

You can always delete quick measures from your model if you don't like them. That's as easy as right-clicking or selecting the ... next to the measure and selecting Delete from model. You can also rename a quick measure whatever you like by selecting Rename from the menu.

There are a few considerations and limitations to keep in mind.

- You can use quick measures added to the **Fields** pane with any visual in the report.
- You can always see the DAX associated with a quick measure by selecting the measure in the **Fields** list and looking at the formula in the formula bar.
- Quick measures are only available if you can modify the model. That isn't the case when you're working with some Live connections. SSAS tabular live connections are supported, as previously described.
- You can't create time intelligence quick measures when working in Direct Query mode. The DAX functions used in these quick measures have performance implications when translated into the T-SQL statements that are sent to your data source.
- DAX statements for quick measures use only commas for argument separators. If your version of Power BI Desktop is in a language that uses commas as decimal separators, quick measures will not work properly.