

## DAX Functions

---

- What is DAX.
- DAX Vs M-Code.
- Calculated Column Vs Measure.
- Implicit Measure Vs Explicit Measure.
- Filter Context Vs Row Context.
- Different Categories of DAX Functions.
- Understanding the Syntax of Formula Language.

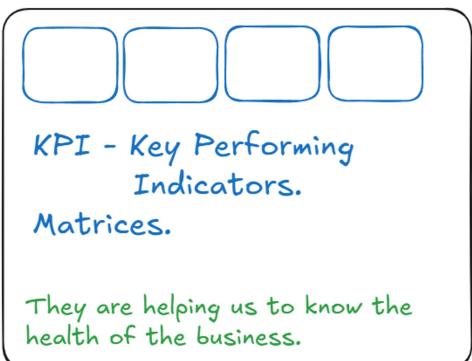
"Vehicle Orders"

DAX

→ Data Analysis Expression.

- It's a Formula Language used in Power BI, to derive complex calculation which are not present in our raw dataset.

- Its largely being used to find the insights from your dashboard.

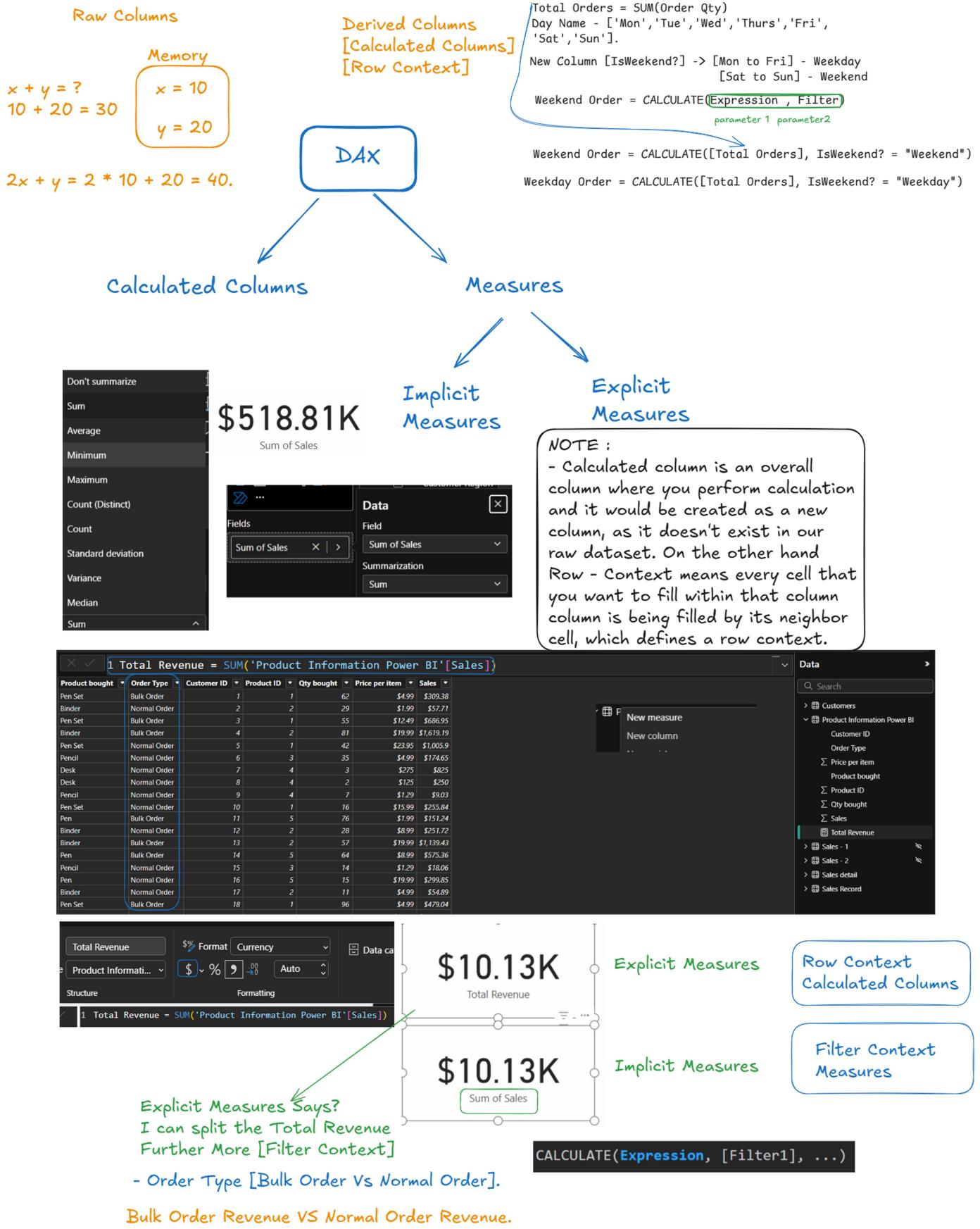


Order Quantity	Product Cost	Product Price
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--
--	--	--

DAX Formula  
 $Sales = Order Qty * P.P$   
 $Expense = Order Qty * P.C$   
 $Total Revenue = Sum(Sales)$

[Measures] [Aggregation]

Order Qty	Product Cost	Product Price	Sales	Expenses	Total Revenue	Total Expenses	Total Profit
2	10	25	50	20	19250	7700	11550
4	20	50	200	80			
6	30	75	450	180			
8	40	100	800	320			
10	50	125	1250	500			
12	60	150	1800	720			
14	70	175	2450	980			
16	80	200	3200	1280			
18	90	225	4050	1620			
20	100	250	5000	2000			



```
Bulk Order Revenue =
    CALCULATE(
        [Total Revenue],
        'Product Information Power BI'[Order Type] = "Bulk Order")
```

**\$6.93K**  
Bulk Order Revenue  
Filter

```
Normal Order Revenue =
    CALCULATE(
        [Total Revenue],
        'Product Information Power BI'[Order Type] = "Normal Order")
```

**\$3.2K**  
Normal Order Revenue  
Filter

Order Type	Total Revenue	Sum of Sales	Normal Order Revenue	Bulk Order Revenue
Bulk Order	\$6,926.08	\$6,926.08	\$3,202.65	\$6,926.08
Normal Order	\$3,202.65	\$3,202.65	\$3,202.65	\$6,926.08
Total	<b>\$10,128.73</b>	<b>\$10,128.73</b>	<b>\$3,202.65</b>	<b>\$6,926.08</b>

Filter Being applied under Calculate function is having highest priority than any other filter.

- SELECT a
- Binder
  - Desk
  - Pen
  - Pen Set
  - Pencil

2 above Measure \* 5 more Filters  
[10 more Measure]

Filter

**\$4.64K**  
Binders Under Bulk Order Revenue

```
Binders Under Bulk Order Revenue =
    CALCULATE(
        [Total Revenue],
        'Product Information Power BI'[Order Type] = "Bulk Order",
        'Product Information Power BI'[Product bought] = "Binder")
```

Binders Under Bulk Order Revenue =
 CALCULATE(
 [Bulk Order Revenue],
 'Product Information Power BI'[Product bought] = "Binder")

1 Sales Values =

2 RELATED(

3 'Product Information Power BI'[Sales Value])

Customer ID	Sales Representative	Date of sale	Sales Channel	Sales Values
1	Naveen Menon	01-01-2024	Online	\$309.38
2	Preeti Khatri	02-01-2024	In-Store	\$57.71
3	Arjun Mehta	03-01-2024	In-Store	\$686.95
4	Aisha Kapoor	04-01-2024	In-Store	\$1,619.19
5	Siddharth Sharma	05-01-2024	Online	\$1,005.9
6	Neha Singh	06-01-2024	In-Store	\$174.65
7	Rajiv Verma	07-01-2024	In-Store	\$625
8	Roshni Patel	08-01-2024	Online	\$250
9	Vikrant Reddy	09-01-2024	Online	\$9.03
10	Anusha Kumar	10-01-2024	In-Store	\$255.84
11	Rahul Kapoor	11-01-2024	In-Store	\$151.24
12	Ashwarya Nair	12-01-2024	In-Store	\$251.72
13	Akshay Mishra	23-01-2024	Online	\$1,159.43
14	Divya Khurana	27-01-2024	Online	\$575.36
15	Karthik Reddy	05-02-2024	Online	\$18.06
16	Shruti Mehra	16-02-2024	Online	\$299.85
17	Varun Khera	17-02-2024	In-Store	\$54.89
18	Sneha Kapoor	18-02-2024	Online	\$479.04
19	Ravi Malhotra	19-02-2024	Online	\$1,879.06
20	Tanvi Sharma	20-02-2024	Online	\$86.43

Total Revenue

Bulk Order Revenue

- SELECT a
- Binder
  - Desk
  - Pen
  - Pen Set
  - Pencil

**\$4.08K**  
In-Store Revenue =

```
CALCULATE(
    SUM('Sales detail'[Sales Values]),
    'Sales detail'[Sales_Channel] = "In-Store")
```

**\$6.05K**  
Online Revenue =

```
CALCULATE(
    SUM('Sales detail'[Sales Values]),
    'Product Information Power BI'[Sales Channel] = "Online")
```

## 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.

### 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



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



### 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

### 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

