What does DAX stand for?

DAX stands for **Data Analysis Expressions**.

It is the **formula language** used in **Power BI**, **Excel Power Pivot**, and **Analysis Services** to crea te **measures**, **calculated columns**, and **custom calculations** for data analysis.

Write a DAX formula to sum the Sales column.

Total Sales = SUM(Sales)

What is the difference between a calculated column and a measure?

Calculated Column

- **Calculated once** during data **refresh**.
- The result is **stored** in the data model (takes memory).
- Works **row-by-row** in the table.
- Useful when you need a value to **filter**, **group**, or build **relationships**.

This adds a new column

Measure

- **Calculated dynamically** based on the **current filter context** (slicers, rows in visuals, etc .).
- **Not stored** → calculated **on the fly** (efficient).
- Used for **aggregations**, totals, ratios, KPIs.

Use the DIVIDE function to calculate Profit Margin (Profit/Sales).

Profit Margin = DIVIDE(Profit, Sales)

Why Use DIVIDE Instead of / ?

Method Problem Advantage of DIVIDE

Profit / Sales

Causes **errors** if Sales = 0

DIVIDE returns **BLANK or alternate value** safely

DIVIDE(Profit, Sales)

No errors

More stable & reliable

What does COUNTROWS() do in DAX?

COUNTROWS() is a DAX function that **counts the number of rows** in a table.

COUNTROWS() returns how many rows exist in a **table** or in a **filtered table**.

Number of Orders = COUNTROWS(Sales)

Orders 2024 =

COUNTROWS(

FILTER(Sales, YEAR(Sales[OrderDate]) = 2024)

)

Create a measure: Total Profit that subtracts total cost from total sales

```
Total Profit = sum(Sheet1[Sales]) - SUM(Sheet1[Cost])
```

Write a measure to calculate Average Sales per Product.

Average Sales per Product =

DIVIDE(

SUM(Sales[SalesAmount]),

DISTINCTCOUNT(Sales[Product])

)

Use IF() to tag products as "High Profit" if Profit > 1000.

```
Status of Profit = IF(((Sheet1[Sales]) - (Sheet1[Cost]))> 1000,  "High Profit", "Low Profit")
```

What is a circular dependency error in a calculated column?

A **circular dependency error** happens when a calculated column depends on itself **directly or i ndirectly**, creating a loop that Power BI cannot resolve.

Imagine you have two columns:

- **Column A** depends on **Column B**
- **Column B** depends on **Column A**

Power BI tries to calculate Column A → needs Column B → but Column B needs Column A firs t → **loop** → error.

How to Fix It

- **Remove the dependency** between the two columns
- Move logic into a **measure** instead of a calculated column (measures do NOT store data, so they do not cause circular dependencies)

Explain row context vs. filter context.

Row Context

Row context = Power BI is evaluating one row at a time.

- It occurs in **calculated columns** and **iterators** (like SUMX, FILTER, AVERAGEX).
- It knows which row is currently being calculated and can read other columns in the same row.

Total Price = Sales[Quantity] * Sales[UnitPrice]

Here, Power BI calculates **row-by-row**, multiplying quantity and price for *each row*.

This is **row context**.

Filter Context

Filter context = which rows are included based on report filters, slicers, visuals, and relatio nships.

- It occurs in **measures**.
- It controls *which subset of rows* are included when calculating.

Example (Measure):

Total Sales = SUM(Sales[Total Price])

If you select **Region = Asia** in a slicer, the measure will only sum rows from Asia.

That's **filter context**.

Write a measure to calculate YTD Sales using TOTALYTD().

to calculate **Year-To-Date (YTD) Sales** using **TOTALYTD():**

YTD Sales =

TOTALYTD(

    [Total Sales],   -- your existing total sales measure

    'Calendar'[Date]   -- your date column from the Date table

)

Create a dynamic measure that switches between Sales, Profit, and Margin.

done

Optimize a slow DAX measure using variables (VAR).

```
Profit Margin (Optimized) =

VAR TotalProfit =

    SUM(Sheet1[Profit])

VAR TotalSales =

    SUM(Sheet1[Sales])

RETURN

    DIVIDE(TotalProfit, TotalSales)
```

Use CALCULATE() to override a filter

Total Sales All Regions =

CALCULATE(

   [Total Sales],

   ALL(Sales[Region])   -- removes region filter

)

Write a measure that returns the highest sales amount

```
Highest Sales =

MAX(Sheet1[Sales])
```