✅ Introduction to DAX Basics & Calculated Columns vs. Measures

1. What does DAX stand for?

Answer:
DAX stands for **Data Analysis Expressions**.
It's a formula language used in Power BI, Power Pivot, and SSAS Tabular to define custom calculations and queries.

2. Write a DAX formula to sum the Sales column.

Answer:

```
Total Sales = SUM(Sales[SalesAmount])
```

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

Answer:

Calculated Column	Measure
Computed row-by-row in the data table	Computed based on filters and context
Stored in the data model	Calculated on demand
Takes up more space	Efficient, lightweight
Useful in slicers and relationships	Used in aggregations, KPIs, visuals

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

Answer:

```
Profit Margin = DIVIDE(Sales[Profit], Sales[SalesAmount])
```

`DIVIDE()` is safer than `/` because it handles division by zero.

5. What does COUNTROWS() do in DAX?

Answer:
`COUNTROWS()` returns the **number of rows in a table**.
Example:

```
Total Transactions = COUNTROWS(Sales)
```

---

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

Answer:

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

---

7. Write a measure to calculate Average Sales per Product.

Answer:

```
Avg Sales per Product =
DIVIDE(SUM(Sales[SalesAmount]), DISTINCTCOUNT(Sales[Product]))
```

---

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

Answer (Calculated Column):

```
Profit Tag = IF(Sales[Profit] > 1000, "High Profit", "Low Profit")
```

---

9. What is a circular dependency error in a calculated column?

Answer:
A **circular dependency** occurs when **a column refers to itself (directly or indirectly)** in its own formula, causing infinite logic. Power BI cannot resolve such calculations.

---

10. Explain row context vs. filter context.

Answer:

- **Row context**: Evaluates calculations **row by row** (applies in calculated columns, iterators like `SUMX`).
- **Filter context**: Comes from filters/slicers applied in **visuals, CALCULATE()**, or measures.

---

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

Answer:

```
YTD Sales = TOTALYTD(SUM(Sales[SalesAmount]), Sales[Date])
```

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

Answer:
Assume you have a slicer table with measure names:

```
Selected Metric =
SWITCH(
    SELECTEDVALUE(MetricSelector[Metric]),
    "Sales", SUM(Sales[SalesAmount]),
    "Profit", [Total Profit],
    "Margin", [Profit Margin]
)
```

13. Optimize a slow DAX measure using variables (VAR).

Example:

```
Optimized Measure =
VAR TotalSales = SUM(Sales[SalesAmount])
VAR TotalCost = SUM(Sales[Cost])
RETURN
TotalSales - TotalCost
```

✅ Variables prevent repeated calculations and improve readability & performance.

14. Use CALCULATE() to override a filter

Example:

```
Sales West Region =
CALCULATE(SUM(Sales[SalesAmount]), Region[RegionName] = "West")
```

15. Write a measure that returns the highest sales amount

Answer:

```
Max Sales = MAX(Sales[SalesAmount])
```