Lesson 8

1. What does DAX stand for?

DAX = **Data Analysis Expressions**
It's the formula language used in Power BI, Excel Power Pivot, and SSAS for data modeling and calculations.

---

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

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

---

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

Calculated Column	Measure
Calculated **row-by-row**	Calculated **based on filter context**
Stored in memory (adds data)	Calculated only when used in visuals
Useful for filtering/grouping	Used for summaries, KPIs, calculations

---

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

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

Safer than `/` — handles divide-by-zero.

---

5. What does COUNTROWS() do in DAX?

It returns the **number of rows** in a table.

```
Product Count = COUNTROWS(Products)
```

---

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

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

---

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

```
Avg Sales per Product =
DIVIDE(
    SUM(Sales[SalesAmount]),
    COUNTROWS(VALUES(Sales[ProductID]))
)
```

---

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

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

---

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

It happens when a column depends on itself — **directly or indirectly** — causing an endless loop.

Example:

```
ColumnA = Table[ColumnB]
ColumnB = Table[ColumnA]  ← ✗ Circular reference!
```

---

10. Explain row context vs. filter context.

Type	Meaning
Row context	Calculations **row by row** (in calculated columns, iterators like `SUMX`)
Filter context	Filters **applied by visuals, slicers, or CALCULATE**

Example:

- In a column: `Price * Quantity` → row context
- In a measure with slicers → filter context

---

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

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

---

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

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

You need a **Metrics table** with values: `"Sales"`, `"Profit"`, `"Margin"`.

---

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

Instead of repeating expensive calculations:

```
Optimized Profit Margin =
VAR TotalSales = SUM(Sales[SalesAmount])
VAR TotalProfit = SUM(Sales[Profit])
RETURN
DIVIDE(TotalProfit, TotalSales)
```

---

14. Use CALCULATE() to override a filter

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

🔁 Overrides filters on `Region`.

---

15. Write a measure that returns the highest sales amount.

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

Or for a **measure across a table**:

```
Top Sale =
MAXX(
    Sales,
    Sales[SalesAmount]
)
```