✅ DAX – Row Context, CALCULATE, VAR, and Filters

1. What is row context? Give an example in a calculated column.

Answer:
Row context occurs when Power BI evaluates an expression **row by row**, usually in **calculated columns**.

Example:

```
LineTotal = Sales[Quantity] * Sales[UnitPrice]
```

This formula uses **row context** because it multiplies fields for **each individual row** in the `Sales` table.

2. Write a measure that finds total sales.

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

3. Use RELATED to fetch the Name from the Customers table into the Sales table.

```
Customer Name = RELATED(Customers[Name])
```

✅ This is used in a **calculated column**, assuming a relationship exists between `Sales[CustomerID]` and `Customers[CustomerID]`.

4. What does `CALCULATE(SUM(Sales[Quantity]), Sales[Category] = "Electronics")` return?

Answer:
It returns the **total quantity sold** but only for rows where `Category = "Electronics"`, overriding any existing filters on `Sales[Category]`.

5. Explain the difference between VAR and RETURN in DAX.

- `VAR` is used to **define temporary variables** to store values or expressions.

- `RETURN` tells DAX **what to output**, often using variables defined above.

Example:

```
ExampleMeasure =
VAR TotalQ = SUM(Sales[Quantity])
RETURN TotalQ * 2
```

---

6. Create a calculated column in Sales called `TotalPrice` using row context (`Quantity * UnitPrice`).

```
TotalPrice = Sales[Quantity] * Sales[UnitPrice]
```

✅ Because it's a **calculated column**, DAX evaluates it **row by row** using row context.

---

7. Write a measure `Electronics Sales` using CALCULATE to sum sales only for the "Electronics" category.

```
Electronics Sales =
CALCULATE(
    SUM(Sales[SalesAmount]),
    Sales[Category] = "Electronics"
)
```

---

8. Use `ALL(Sales[Category])` in a measure to show total sales ignoring category filters.

```
Total Sales All Categories =
CALCULATE(SUM(Sales[SalesAmount]), ALL(Sales[Category]))
```

✅ This **removes any slicers or filters** on the Category column.

---

9. Fix this error: A calculated column in Sales uses `RELATED(Customers[Region])` but returns blanks.

Cause:
There’s likely **no active relationship** between `Sales` and `Customers`.

Fix:
Check **Manage Relationships** and ensure:

- Relationship exists between `Sales[CustomerID]` and `Customers[CustomerID]`
- It’s **active**

10. Why does CALCULATE override existing filters?

Answer:
`CALCULATE()` is designed to **change the filter context**. When you add a new condition inside CALCULATE, it **replaces** or **adds to** the current filters, depending on context.

11. Write a measure that returns average UnitPrice of products.

```
Average UnitPrice = AVERAGE(Sales[UnitPrice])
```

12. Use VAR to store a temporary table of high-quantity sales (`Quantity > 2`), then count rows.

```
High Quantity Count =
VAR HighQtyTable =
    FILTER(Sales, Sales[Quantity] > 2)
RETURN
    COUNTROWS(HighQtyTable)
```

13. Write a measure `% of Category Sales` that shows each sale's contribution to its category total.

```
% of Category Sales =
DIVIDE(
    SUM(Sales[SalesAmount]),
    CALCULATE(SUM(Sales[SalesAmount]), ALL(Sales[Product]))
)
```

✅ This gives the **row's sales** divided by **total sales in the same category**, ignoring product-level filters.

14. Simulate a "remove filters" button using ALL in a measure.

```
RemoveFilter Sales =
CALCULATE(SUM(Sales[SalesAmount]), ALL(Sales))
```

This acts like a **reset button**, showing total sales regardless of slicers.

15. Troubleshoot: A CALCULATE measure ignores a slicer. What’s the likely cause?

Answer:
One of the following:

- You're using `ALL()` or `REMOVEFILTERS()` inside the measure, which **removes slicer filters**.
- Slicer field and measure field are from **unrelated tables** (no relationship).
- Slicer filters a **column not involved** in the measure’s calculation.