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

- **Row context** = When DAX evaluates a formula **row by row** in a table.
- Example: In a calculated column, each row has its own "context," so you can multiply values in that row.

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

Here, for each row in `DAX_Practice_Data`, DAX uses the row's own `Quantity` and `UnitPrice`.

---

2. Write a measure that finds total sales

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

Measures work on **filter context**, not row context.

---

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

In the `DAX_Practice_Data` table, create a calculated column:

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

This works if there's a relationship (`Sales[CustomerID]` → `Customers[CustomerID]`).

---

4. What does this return?

```
CALCULATE(
    SUM(DAX_Practice_Data[Quantity]),
    DAX_Practice_Data[Category] = "Electronics"
)
```

Returns the **total Quantity of Electronics sales**, ignoring the current filter for Category and replacing it with `"Electronics"`.

---

5. Explain the difference between VAR and RETURN in DAX

- **VAR**: Store a value or table temporarily (like a variable).
- **RETURN**: Defines what the measure will finally output.

Example:

```
HighQuantity =
VAR q = SUM(DAX_Practice_Data[Quantity])
RETURN
    q * 2
```

---

6. Create a calculated column TotalPrice using row context

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

---

7. Write a measure Electronics Sales using CALCULATE

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

---

8. Use ALL(Sales[Category]) in a measure

```
Total Sales (Ignore Category) =
CALCULATE(
    SUM(DAX_Practice_Data[Sales]),
    ALL(DAX_Practice_Data[Category])
)
```

This shows **total sales across all categories**, even if a Category filter is active.

---

9. Fix this error: RELATED(Customers[Region]) returns blanks

Causes:

- No relationship between `DAX_Practice_Data[CustomerID]` and `Customers[CustomerID]`.
- Or relationship is inactive.

Fix: Ensure a proper **one-to-many relationship** exists and is active.

---

10. Why does CALCULATE override existing filters?

Because `CALCULATE` **modifies filter context**:

- If the same column is filtered already, CALCULATE **replaces** that filter.
- If not filtered, CALCULATE **adds** a filter.

---

11. Write a measure that returns average unit price of products

```
Average Unit Price =
AVERAGE(DAX_Practice_Data[UnitPrice])
```

---

12. Use VAR to store a temporary table of high-quantity sales

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

---

13. % of Category Sales (each category's contribution to total sales)

```
% of Category Sales =
DIVIDE(
    SUM(DAX_Practice_Data[Sales]),
    CALCULATE(
        SUM(DAX_Practice_Data[Sales]),
        ALLEXCEPT(DAX_Practice_Data, DAX_Practice_Data[Category])
    )
)
```

Shows each row/category's % within its **category total**.

---

14. Simulate a "remove filters" button using ALL

```
Total Sales (Remove Filters) =
CALCULATE(
    SUM(DAX_Practice_Data[Sales]),
    ALL(DAX_Practice_Data)
)
```

Ignores slicers/filters and shows the **grand total**.

---

15. Troubleshoot: A CALCULATE measure ignores a slicer. Likely cause?

- The slicer field is from a **disconnected table** (no relationship).
- Or the slicer is applied on a column not used in the CALCULATE filters.
- Or ALL() / REMOVEFILTERS() inside CALCULATE is overriding slicers.