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

Row context is the “current row” available during row-by-row calculations (e.g., calculated columns, iterators).

Example (calculated column in Sales):

DAX

КопироватьРедактировать

TotalPrice = Sales[Quantity] * Sales[UnitPrice]

The formula uses the values from the current row.

---

2) Measure that finds total sales

DAX

КопироватьРедактировать

Total Sales = SUMX(Sales, Sales[Quantity] * Sales[UnitPrice])

---

3) Use RELATED to fetch the Name from Customers into Sales

Calculated column in Sales:

DAX

КопироватьРедактировать

Customer Name = RELATED(Customers[Name])

---

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

It returns the total Quantity for rows where Category = "Electronics", applying other existing filters but replacing/overriding the Category filter with Electronics.

---

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

VAR defines a variable (number/table/text) to reuse; RETURN specifies the final expression that uses those variables. Every VAR block must end with a RETURN.

---

6) Create a calculated column TotalPrice using row context

DAX

КопироватьРедактировать

TotalPrice = Sales[Quantity] * Sales[UnitPrice]

---

7) Measure: Electronics Sales using CALCULATE

DAX

КопироватьРедактировать

Electronics Sales =

CALCULATE([Total Sales], Sales[Category] = "Electronics")

---

8) Use ALL(Sales[Category]) to show total sales ignoring category filters

DAX

КопироватьРедактировать

Total Sales (Ignore Category) =

CALCULATE([Total Sales], ALL(Sales[Category]))

---

9) Fix: calculated column uses RELATED(Customers[Region]) but returns blanks

Likely causes: no active one-to-many relationship from Customers[CustomerID] → Sales[CustomerID], data types don’t match, or relationship is inactive.

Fix: ensure a single active relationship (Customers = “one”, Sales = “many”), matching data types; then RELATED works. (Alternative: LOOKUPVALUE if you cannot relate.)

---

10) Why does CALCULATE override existing filters?

CALCULATE modifies the filter context: it adds, removes, or replaces filters. When a filter argument targets the same column as an existing filter, it replaces it.

---

11) Measure that returns average unit price of products

If you need the per-product average, then average across products:

DAX

КопироватьРедактировать

Average UnitPrice =

AVERAGEX(VALUES(Sales[ProductID]), AVERAGE(Sales[UnitPrice]))

(If you just need overall row average: AVERAGE(Sales[UnitPrice]).)

---

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

DAX

КопироватьРедактировать

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

---

13) Measure % of Category Sales (each sale's contribution to its category total)

DAX

КопироватьРедактировать

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

---

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

DAX

КопироватьРедактировать

```
Total Sales (Ignore All) =
CALCULATE( [Total Sales], ALL(Sales) )
-- or equivalently:
-- CALCULATE( [Total Sales], REMOVEFILTERS(Sales) )
```

---

15) Troubleshoot: a CALCULATE measure ignores a slicer — likely cause?

Common reasons:

The measure uses ALL / REMOVEFILTERS (or ALLEXCEPT) on the slicer's column, clearing it.

The slicer comes from a disconnected/unrelated table.

The relationship is inactive or wrong direction; or Edit Interactions set to "None" for that visual.