

## Lesson 9

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

**Row context** occurs in **calculated columns** or iterators (like SUMX), where each row is evaluated **one at a time**.

 **Example (Calculated Column):**

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

This uses row context to multiply `Quantity` and `UnitPrice` on each row of the `Sales` table.

---

### ☒ 2. Write a measure that finds total sales

```
dax
КопироватьРедактировать
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])
```

 Make sure there's a **relationship** between `Sales` and `Customers`.

---

### ☒ 4. What does this return?

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

This returns the **total quantity sold** where the **category is Electronics**.

But it needs to be used **inside a row context or filter expression** properly. Fixed version:

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

---

### ☒ 5. Difference between VAR and RETURN

- VAR lets you **define variables** to simplify logic
- RETURN is used to **output the result**

 **Example:**

```
HighProfit =  
VAR profit = Sales[UnitPrice] * Sales[Quantity]  
RETURN  
IF(profit > 1000, "High", "Low")
```

---

## ☒ 6. Create a calculated column in Sales called TotalPrice

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

This uses **row context**.

---

## ☒ 7. Measure: Electronics Sales using CALCULATE

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

---

## ☒ 8. Use ALL(Sales[Category]) to ignore category filters

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

---

## ☒ 9. Fix: RELATED returns blanks

If RELATED(Customers[Region]) returns blanks:

-  Check if there's a **relationship** between Sales and Customers
  - ☒ Ensure each Sales[CustomerID] exists in Customers[CustomerID]
- 

## ☒ 10. Why does CALCULATE override filters?

Because CALCULATE **replaces the current filter context** with what you specify inside it. That's how it can **override slicers or visuals**.

---

## ☒ 11. Average Unit Price

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

---

## ☒ 12. Use VAR to store high-quantity sales, then count

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

---

## ☒ 13. % of Category Sales

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

Shows each sale's % of its category total.

---

## ☒ 14. Simulate a "remove filters" button using ALL

```
Total Sales (No Filters) =  
CALCULATE(  
    SUM(Sales[SalesAmount]),  
    ALL(Sales)  
)
```

Bind this to a **card or bookmark** to simulate filter reset.

---

## ☒ 15. CALCULATE measure ignores slicer – why?

 Most common causes:

- CALCULATE uses **ALL()** or REMOVEFILTERS() → ignoring slicer
- **Slicer table isn't related** to the visual's table
- You're using a **disconnected table**

☒ Fix:

- Check **relationships** in **Model view**
- Remove ALL() or fix filter context logic