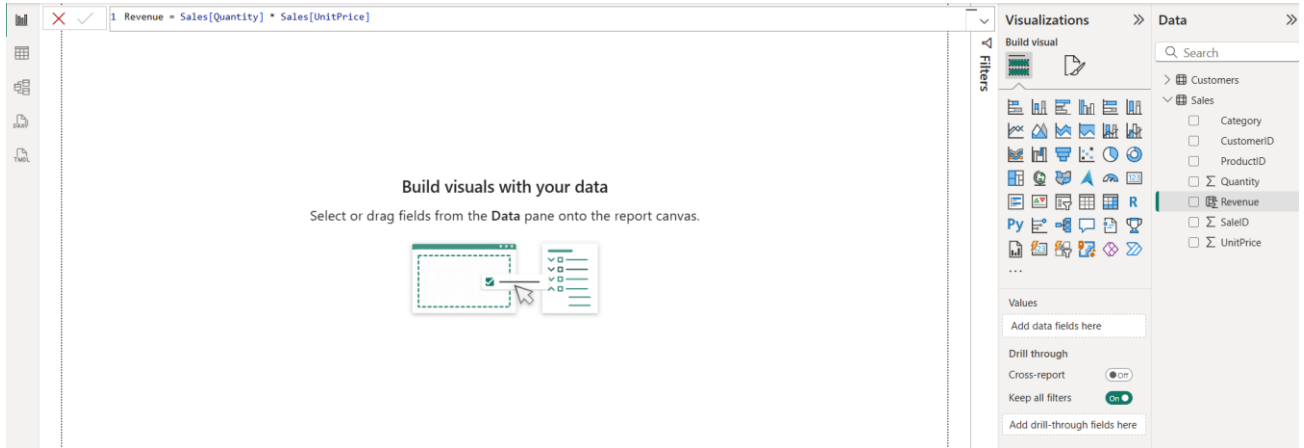


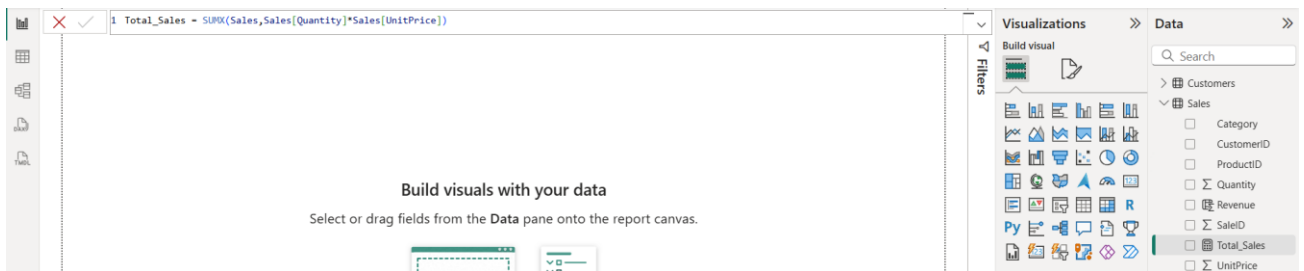
1. What is row context? Give an example in a calculated column.

Row context means that when Power BI (DAX) calculates a **calculated column**, it processes the formula **row by row**, and in each row it automatically “knows” the values from that row without you having to filter for them.



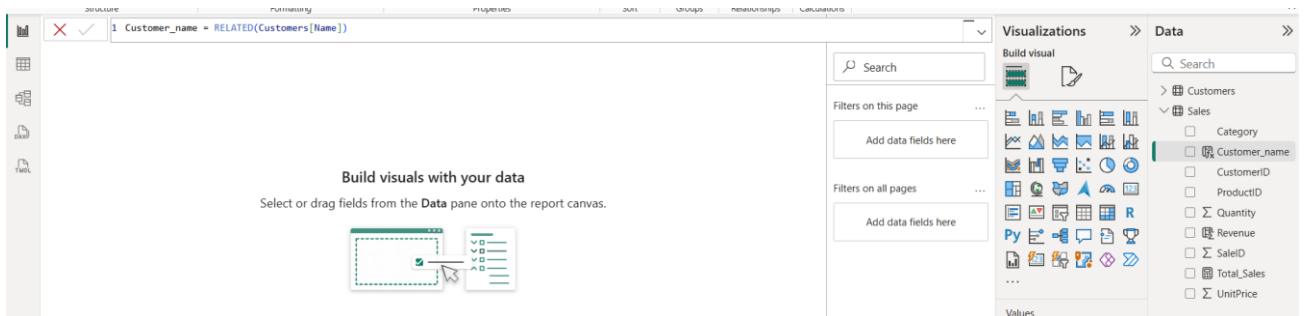
2. Write a measure that finds total sales

`Total_Sales = SUMX(Sales, Sales[Quantity]*Sales[UnitPrice])`



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

`Customer_name = RELATED(Customers[Name])`



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



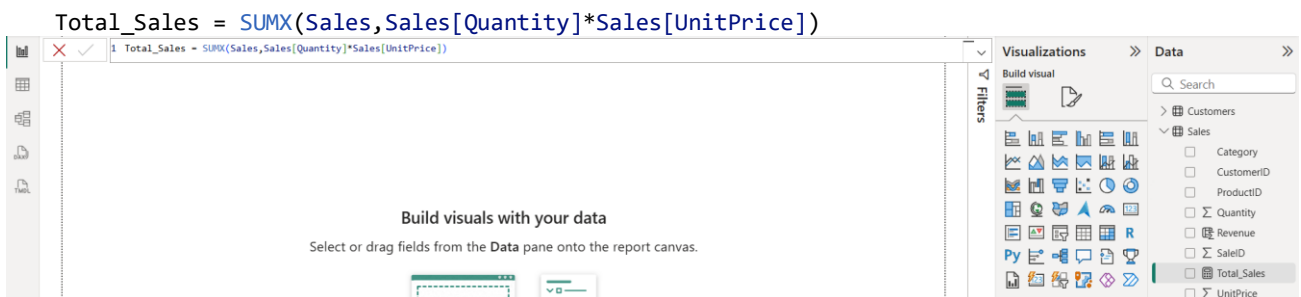
It returns the `total_quantity` of "Electronics" category.

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

`VAR` is used to **define a variable** inside your DAX formula. It stores an intermediate result (like a number, table, or string) so you can reuse it later in the calculation.

`RETURN` specifies **what the formula should output**.

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



7. Write a measure `Electronics Sales` using `CALCULATE` to sum sales only for the "Electronics" category.
`CALCULATE(SUM(Sales[Quantity]), Sales[Category] = "Electronics")`



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

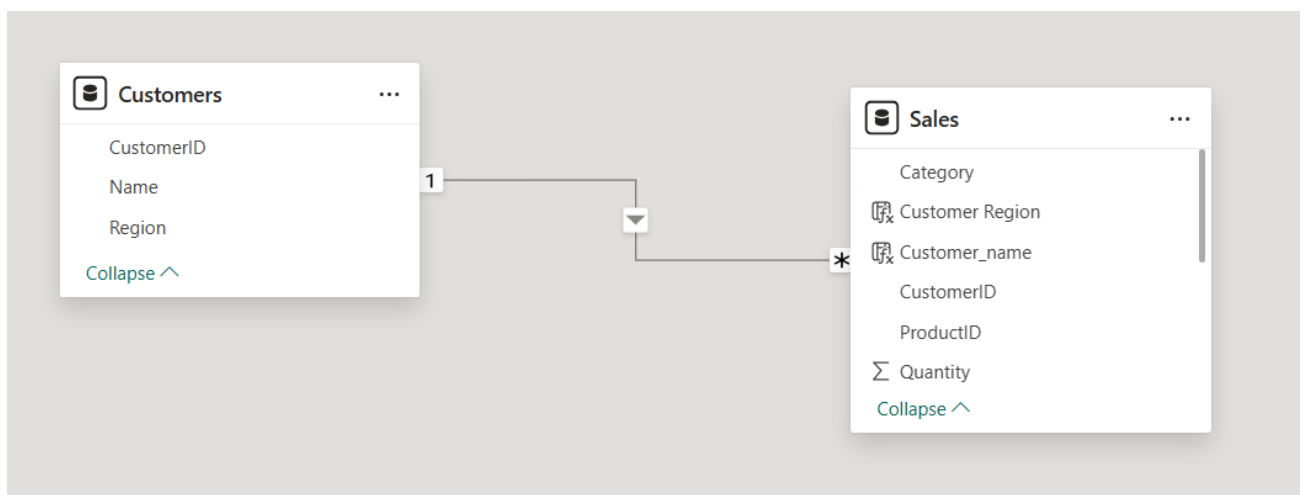
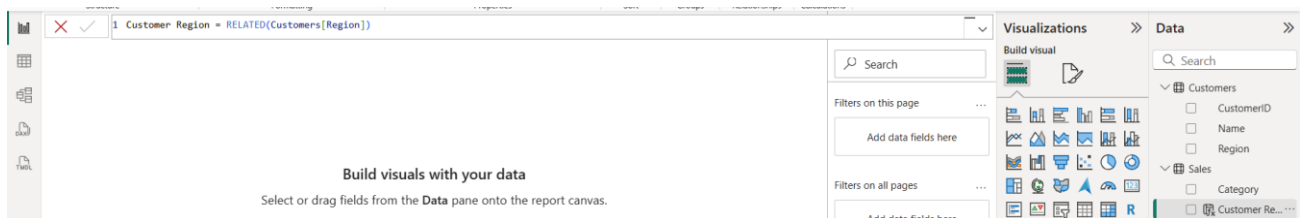


9. Fix this error: A calculated column in Sales uses RELATED(Customers[Region]) but returns blanks.

When RELATED(Customers[Region]) in a **calculated column** returns **blank**, it means Power BI cannot find a matching row in the Customers table for that Sales row.

There is no relationship between Sales[CustomerID] and Customers[CustomerID]

The relationship exists, but the Sales table contains CustomerIDs that don't exist in Customers (or the data types are different)

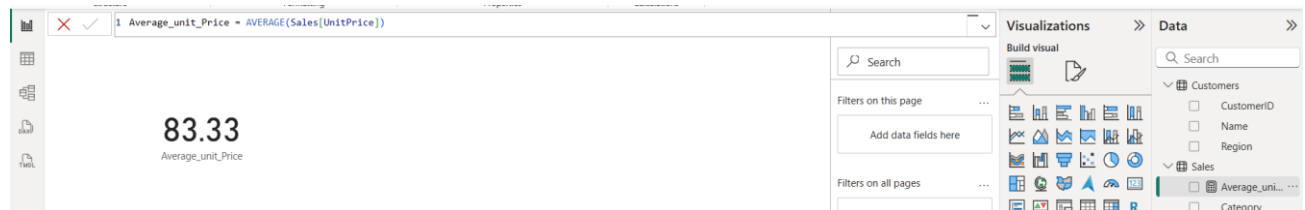


10. Why does CALCULATE override existing filters?

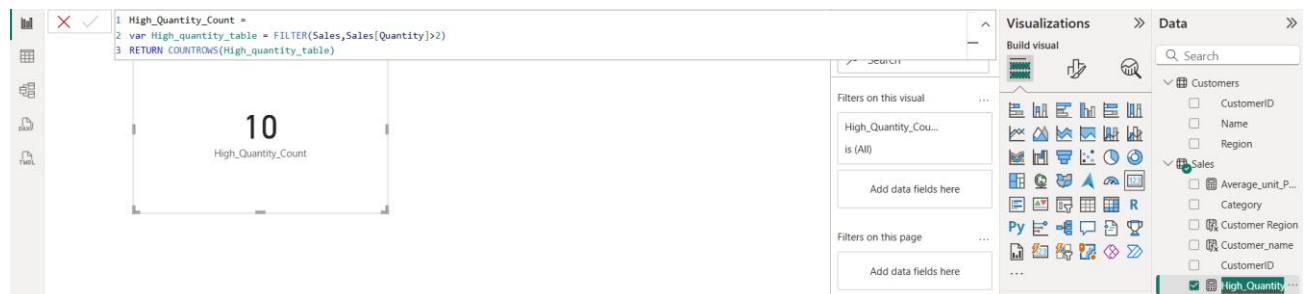
CALCULATE() overrides existing filters because that's exactly its job — it changes the filter context in which an expression is evaluated.

CALCULATE Evaluate this measure but under these filter conditions, ignoring whatever filters were applied before. It **modifies** that filter context

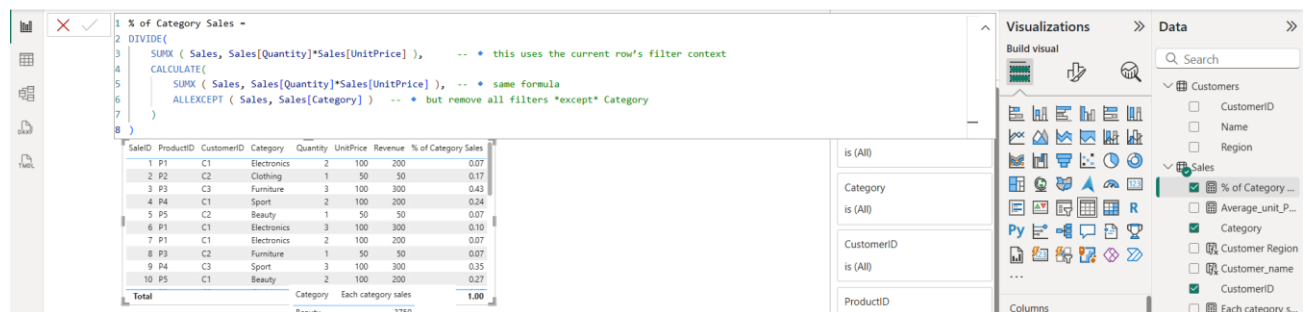
11. Write a measure that returns average unitprice of products



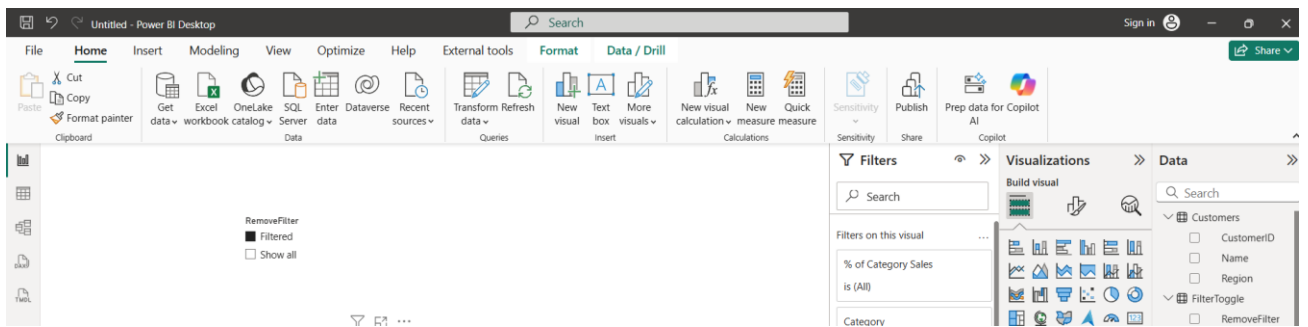
12. Use VAR to store a temporary table of high-quantity sales (Quantity > 2), then count rows.



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



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



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

Usually this happens when the CALCULATE expression contains a filter function that removes the slicer's filter.

Most commonly the measure uses something like:

ALL()

REMOVEFILTERS()

ALLEXCEPT()

These functions override the filter context coming from the slicer — so the slicer does apply a filter, but CALCULATE removes it, which makes it look like the slicer isn't working.