Power BI - lesson 10

1. What does `FILTER(Sales, Sales[Amount] > 1000)` return?

It returns a table with only rows where **Amount > 1000**.

🔷2. Write a measure `High Sales` that sums Amount where Amount > 1000 using FILTER.

```
dax
CopyEdit
High Sales =
CALCULATE(
    SUM(Sales[Amount]),
    FILTER(Sales, Sales[Amount] > 1000)
)
```

🔷3. How does `ALLEXCEPT(Sales, Sales[Region])` differ from `ALL(Sales)`?

- `ALL(Sales)`: Removes all filters.
- `ALLEXCEPT(Sales, Sales[Region])`: Removes all filters **except Region**.

🔷4. Use SWITCH to categorize Amount: "Medium" if 500–1000, "High" if > 1000

```
dax
CopyEdit
Amount Category =
SWITCH(TRUE(),
    Sales[Amount] > 1000, "High",
```

```
    Sales[Amount] >= 500, "Medium",
    "Low"
)
```

🔷5. What is the purpose of `ALLSELECTED`?

It keeps slicer filters but ignores filters from visuals (like charts or tables).

🔷6. Measure: `Regional Sales %` using ALLEXCEPT

```
dax
CopyEdit
Regional Sales % =
DIVIDE(
    SUM(Sales[Amount]),
    CALCULATE(SUM(Sales[Amount]), ALLEXCEPT(Sales, Sales[Region]))
)
```

🔷7. Dynamic measure using SWITCH to toggle between SUM, AVERAGE, and COUNT

Assuming a slicer on a table called `MetricSelector[Metric]` with values "Sum", "Average", "Count":

```
dax
CopyEdit
Dynamic Measure =
SWITCH(SELECTEDVALUE(MetricSelector[Metric]),
    "Sum", SUM(Sales[Amount]),
    "Average", AVERAGE(Sales[Amount]),
    "Count", COUNT(Sales[Amount])
```

```
)
```

🔷8. Use FILTER inside CALCULATE to exclude "Furniture" sales

```
dax
CopyEdit
Non-Furniture Sales =
CALCULATE(
    SUM(Sales[Amount]),
    FILTER(Products, Products[Category] <> "Furniture")
)
```

🔷9. Why might `ALLSELECTED` behave unexpectedly in a pivot table?

Because it reacts to slicers *and* report-level filters, but not always to all visual filters—can return unexpected totals.

🔷10. Measure: Total Sales ignoring Region filters

```
dax
CopyEdit
Total Sales (Ignore Region) =
CALCULATE(
    SUM(Sales[Amount]),
    ALL(Sales[Region])
)
```

🔷11. Optimize this measure:

❌ **Original:**

```dax
CopyEdit
High Sales = CALCULATE(SUM(Sales[Amount]), FILTER(Sales, Sales[Amount] > 1000))
```

✅ **Optimized:**

```dax
CopyEdit
High Sales = CALCULATE(SUM(Sales[Amount]), Sales[Amount] > 1000)
```

Using a Boolean filter inside `CALCULATE` is faster than using `FILTER()`.

🔷12. Measure: Top 2 Products using TOPN and FILTER

```dax
CopyEdit
Top 2 Products Sales =
CALCULATE(
    SUM(Sales[Amount]),
    TOPN(2, VALUES(Sales[ProductID]), CALCULATE(SUM(Sales[Amount])), DESC)
)
```

🔷13. Use `ALLSELECTED()` with no parameters

```dax
CopyEdit
Selected Total =
```

```
CALCULATE(SUM(Sales[Amount]), ALLSELECTED())
```

→ Keeps slicers, ignores visual-level filters.

🔷14. **Debug: SWITCH returns wrong values in matrix**

Reason: `SELECTEDVALUE()` might return blank if multiple values are selected.

Fix: Use `ISFILTERED()` or default values inside `SELECTEDVALUE()`:

```
dax
CopyEdit
SWITCH(SELECTEDVALUE(MetricSelector[Metric], "Sum"), ...)
```

🔷15. **Simulate "Reset Filters" button using ALL**

```
dax
CopyEdit
Total Sales (Reset) =
CALCULATE(SUM(Sales[Amount]), ALL(Sales))
```

→ Ignores **all** filters and gives grand total.