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

- It doesn’t calculate anything on its own — just **filters rows**.

Example:

```
FILTER(Sales, Sales[Amount] > 1000)
```

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

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

This measure sums `Amount` only for rows where `Amount > 1000`.

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

- `ALL(Sales)` removes **all filters** on the Sales table.
- `ALLEXCEPT(Sales, Sales[Region])` removes all filters **except** the one on `Region`.

Example: Use `ALLEXCEPT` when you want % by Region but ignore other slicers.

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

```
Amount Category =
SWITCH(
    TRUE(),
    Sales[Amount] > 1000, "High",
    Sales[Amount] >= 500, "Medium",
    "Low"
)
```

5. What is the purpose of `ALLSELECTED`?

It returns **all values** that are **selected by slicers** or **manual selections** — ignoring filters **only from visuals**.

Useful for dynamic % calculations where you want slicer context **but not visual-level filters**.

6. Write a measure `Regional Sales %` showing each sale's contribution to its region's total

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

7. Create a dynamic measure using `SWITCH` to toggle between SUM, AVERAGE, and COUNT of Amount

Assume a disconnected table `MetricSelector[Measure]` with values `"Sum"`, `"Average"`, `"Count"`:

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

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

```
Exclude Furniture Sales =
CALCULATE(
    SUM(Sales[Amount]),
    FILTER(Products, Products[Category] <> "Furniture")
)
```

Make sure `Products` and `Sales` are related on `ProductID`.

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

Because it:

- Keeps slicer filters
- **Ignores visual-level filters**
  So if you use `ALLSELECTED()` in a visual with **nested groups (like Region → Product)**, results can appear **inconsistent or ambiguous**.

10. Write a measure that calculates total sales and ignores filters from region

```
Sales Without Region =
CALCULATE(
    SUM(Sales[Amount]),
    REMOVEFILTERS(Sales[Region])
)
```

Or:

```
CALCULATE(SUM(Sales[Amount]), ALL(Sales[Region]))
```

11. Optimize this measure:

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

Optimized using boolean expression:

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

This is faster — no need to wrap `FILTER` unless you're doing row-by-row logic.

12. Write a measure `Top 2 Products` using `TOPN` and `FILTER` to show the highest-grossing products

```
Top 2 Products Sales =
CALCULATE(
    SUM(Sales[Amount]),
    FILTER(
        TOPN(2, SUMMARIZE(Sales, Sales[ProductID], "TotalAmt",
SUM(Sales[Amount])), [TotalAmt], DESC),
        TRUE()
    )
)
```

13. Use `ALLSELECTED()` with no parameters to respect slicers but ignore visual-level filters

```
Total Selected Sales =
CALCULATE(
    SUM(Sales[Amount]),
    ALLSELECTED()
)
```

14. Debug: A `SWITCH` measure returns incorrect values when fields are added to a matrix visual

Possible causes:

- `SELECTEDVALUE()` returns **blank or multiple values** when grouped
- Matrix rows might create unexpected context

Fix:
Use `HASONEVALUE()` or design the logic to handle multiple rows.

15. Simulate a "reset filters" button using `ALL` in a measure

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