Lesson 10

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

- It returns a **table** that includes **only rows** from the `Sales` table where `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)
)
```

---

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 Region** — retains filtering by `Region` only.

---

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`?

- `ALLSELECTED` keeps **user-made selections from slicers**, but **removes filters** from visuals (e.g., matrix rows).
- Useful for **calculating totals** based on what the user **actively selected**, not what's visible.

---

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

```
Regional Sales % =
DIVIDE(
    SUM(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 slicer connected to a disconnected table `MeasureSelector[MeasureType]` with values "Sum", "Average", "Count":

```
Selected Measure =
SWITCH(
    SELECTEDVALUE(MeasureSelector[MeasureType]),
    "Sum", SUM(Sales[Amount]),
    "Average", AVERAGE(Sales[Amount]),
    "Count", COUNT(Sales[Amount])
)
```

---

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

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

---

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

- Because it removes **visual-level filters**, the context inside a **matrix row or column** can be lost.
- May return **total values** where row-level behavior is expected — especially if rows are filtered but slicers are not used.

---

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

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

---

11. Optimize this measure by replacing `FILTER` with a Boolean filter

Original:

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

Optimized:

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

☑ Same result, better performance. `FILTER` returns a table; Boolean filters are more efficient.

---

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

```
Top 2 Product Sales =
CALCULATE(
    SUM(Sales[Amount]),
    TOPN(2, VALUES(Products[ProductName]), CALCULATE(SUM(Sales[Amount])),
DESC)
)
```

---

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

```
Sales % of Selected =
DIVIDE(
    SUM(Sales[Amount]),
    CALCULATE(SUM(Sales[Amount]), ALLSELECTED())
)
```

---

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

- Common reasons:
    - You use `SELECTEDVALUE()` which returns blank when **multiple values** exist.
    - Fix: Add default or fallback logic:

      ```
      SELECTEDVALUE(Table[Column], "Default")
      ```

    - Or use `MAX()` or other aggregation to handle multiple values.

---

15. Simulate a "Reset Filters" button using `ALL` in a measure

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