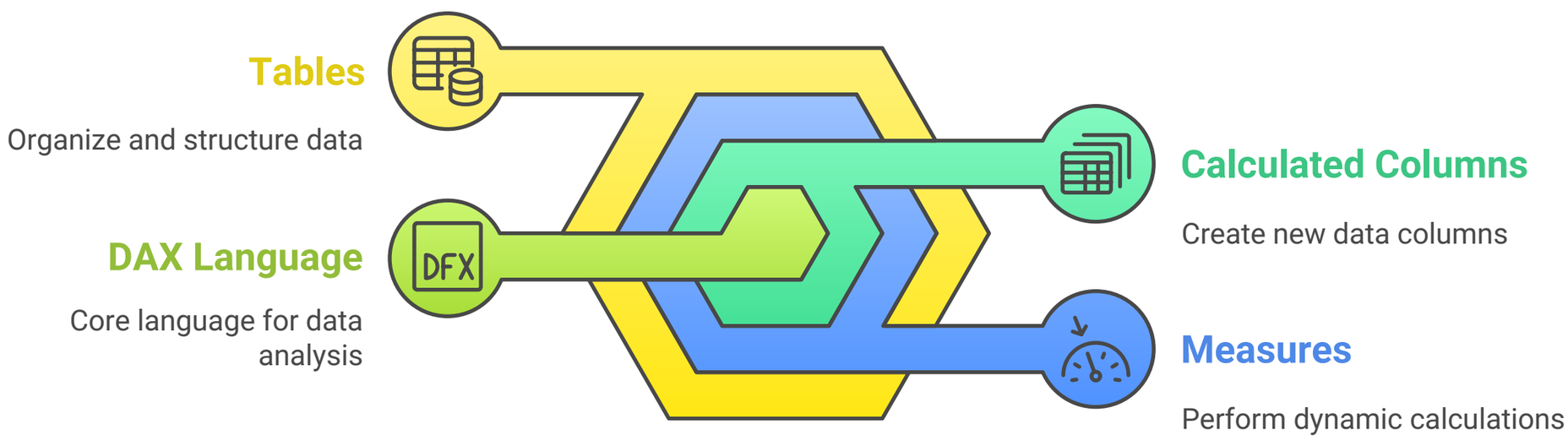
# Top 10 DAX Interview Questions and Answers with Examples 🚀

## 1. What is DAX, and why is it used in Power BI?

**Answer:**DAX (Data Analysis Expressions) is a formula language used in Power BI, Power Pivot, and SSAS Tabular models. It allows users to create **calculated columns, measures, and tables** for data analysis and reporting.




## 2. What is the difference between a Calculated Column and a Measure?

| Feature | Calculated Column | Measure |
|---|---|---|
| **Evaluation** | Row-by-row | Aggregated level |
| **Storage** | Stored in memory | Calculated on the fly |
| **Performance** | Uses more memory | More efficient |

- **Example:**✅ **Calculated Column:** Adds a new column to the table

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

✅ **Measure:** Works dynamically with aggregations
**Total Sales = SUM(Sales[Quantity] * Sales[UnitPrice])**

## 3. What is the difference between SUM and SUMX?

**Answer:**

- **SUM** adds up all values of a single column.
- **SUMX** iterates over a table and performs row-wise calculations.

- **Example:**✅ **SUM**: Adds values from a single column

**Total Quantity = SUM(Sales[Quantity])**

✅ **SUMX**: Performs row-wise multiplication before summing
**Total Sales = SUMX(Sales, Sales[Quantity] * Sales[Unit Price])**

## 4. What is the difference between RELATED and LOOKUPVALUE?

**Answer:**

- **RELATED()** is used with a **one-to-many** (1:*) relationship to fetch a related value from another table.
- **LOOKUPVALUE()** is used when there is **no direct relationship** between tables.
  - **Example:**✅ Using **RELATED()** when there is a relationship

**CategoryName = RELATED(Category[Category Name])**

✅ Using **LOOKUPVALUE()** when no direct relationship exists

**CategoryName = LOOKUPVALUE(Category[Category Name], Category[Category ID], Sales[CategoryID])**

## 5. How does the CALCULATE function work in DAX?

**Answer:**The **CALCULATE** function modifies the filter context of an expression by applying specified filters.

- **Example:**✅ Calculate sales only for “Electronics”

**Electronics Sales = CALCULATE(SUM(Sales[Sales Amount]), Sales[Category] = "Electronics")**
This function sums up the sales only where the category is “Electronics”.

## 6. What is the difference between SUMX and CALCULATE(SUM())?

✅ **SUMX**: Iterates row by row and performs calculation before aggregation.✅
**CALCULATE(SUM(Column), Filter)**: First filters the data and then applies SUM.

- **Example:**✅ **SUMX** applies row-by-row calculation

**Total Revenue = SUMX(Sales, Sales[Quantity] * Sales[Unit Price])**

✅ **CALCULATE(SUM())** applies a filter first, then sums
**Sales in 2024 = CALCULATE(SUM(Sales[Sales Amount]), Sales[Year] = 2024)**

## 7. What is the difference between ALL, ALLEXCEPT, and REMOVEFILTERS?

| Function | Description |
|---|---|
| `ALL(Table[Column])` | Ignores all filters on a table or column |
| `ALLEXCEPT(Table, Table[Column])` | Ignores all filters **except** for the column(s) specified |
| `REMOVEFILTERS(Table[Column])` | Removes filters from one or multiple columns/tables |

- **Example:**✅ Ignore all filters on **Sales** table

**Total Sales All = CALCULATE(SUM(Sales[Amount]), ALL(Sales))**

✅ Remove all filters **except Category**
**Sales by Product = CALCULATE(SUM(Sales[Amount]), ALLEXCEPT(Sales, Sales[Category]))**

## 8. What is the difference between Calculated Columns and Virtual Columns in DAX?

**Answer:**

- **Calculated Column:** Created at the **table level** and stored in the dataset.
- **Virtual Column:** Created within functions like **SUMX()**, **FILTER()**, and **ADDCOLUMNS()**, but not stored in the model.

◈ **Example:**✅ **Calculated Column:** Stored permanently in the model

**TotalPrice = Sales[Quantity] * Sales[Unit Price]**

✅ **Virtual Column using SUMX():** Created only when needed

**Total Revenue = SUMX(Sales, Sales[Quantity] * Sales[Unit Price])**

◈ **Key Difference:**

- Calculated **columns take up storage**, while virtual columns are created dynamically during calculation.

## 9. What is the difference between EARLIER and SELECTEDVALUE?

**Answer:**

- **EARLIER()**: Refers to **previous row context** inside **CALCULATE()** or **FILTER()**.
- **SELECTEDVALUE()**: Returns a **single selected value** from a column. If multiple values exist, it returns **BLANK().**

◈ **Example:**✅ Using **EARLIER()** in a **row context loop**

**Running Total =**
**CALCULATE( SUM(Sales[Sales Amount]),**
**FILTER(Sales, Sales[Date] <= EARLIER(Sales[Date])) )**

✅ Using **SELECTEDVALUE()** to return a single value from a column

**Selected Year = SELECTEDVALUE(Sales[Year], "Multiple Years Selected")**

## 10. How does the RANKX function work in DAX?

**Answer:**The **RANKX()** function ranks values in a column based on an expression.

◈ **Example:**✅ Rank Products by Sales

**ProductRank = RANKX(ALL(Sales), SUM(Sales[Amount]), , DESC, DENSE)**

- **DESC**: Ranks in descending order.
- **DENSE**: Ensures no gaps in ranking (e.g., 1, 2, 2, 3 instead of 1, 2, 3, 4)

## 11. How does the DISTINCT function work in DAX?

**Answer:DISTINCT()** returns a unique list of values from a column or table.

◈ **Example:**✅ Get unique categories

**Unique Categories = DISTINCT(Sales[Category])**

✅ Count unique customers

**Unique Customers = DISTINCTCOUNT(Sales[CustomerID])**