1. What is DAX? Why it is used?

   Ans: **DAX (Data Analysis Expressions)** is a formula language used in Power BI to perform powerful data analysis and modeling.

   - It's used to create **calculated columns, measures,** and **custom tables.**

   - Helps analyze complex datasets by applying logic, filters, and aggregations.

2. What are the data types supported by DAX?

   Ans: DAX supports the following key data types:

   - **Whole Number** – integers

   - **Decimal Number** – floating-point values

   - **Currency** – similar to decimals but formatted for financial calculations

   - **Date/Time** – date and time values

   - **Boolean** – TRUE/FALSE

   - **String (Text)** – alphanumeric values

   - **Blank** – similar to NULL in databases

3. What is the difference between a calculated column and a measure?

   Ans:

Feature	Calculated Column	Measure
Stored in model	Stored as a column	Calculated on the fly
Calculation	Row-by-row	Aggregations
Evaluation Context	Row Context	Filter Context
Performance	Slower for large datasets	Faster and more optimized
Use Case	Filtering, Relationships, New data fields etc.	KPIs, Aggregations(SUM, AVERAGE, Dashboards etc.
Example	"Profit = Sales - Cost" for each row	"Total Sales = SUM(Sales[SalesAmount])"

4. How to exclude all filters while calculations on any measure.

   Ans: To ignore filters during measure calculations, wrap your expression in the **ALL()** function.

   TotalSalesAll = CALCULATE(SUM(Sales[SalesAmount]), ALL(Sales))

   This removes *all filters* from the Sales table, allowing the measure to compute across the full dataset.

5. What are circular dependencies? How can you avoid creating circular dependencies in your DAX expressions?

   Ans: **Circular dependency** happens when:

   - Column A depends on Column B

   - Column B depends on Column C

   - Column C depends on Column A

   This loop confuses the model and breaks calculations.

   This is also called **Transitive Dependency.**

   To avoid circular dependencies:

   - Use **measures** instead of calculated columns when possible.

   - Break expressions into **independent components**.

   - Avoid linking calculations too tightly—use intermediate steps if needed.

Aditya Mohanty

B-316