SQL and POWERBI

Dhanush S-G21SQL

Assessment:3

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

DAX is a formula language used to create custom calculations in Power BI, Excel Power Pivot, and SSAS Tabular. It's essential for creating **measures** (like sum, average, percentage) and **calculated columns** (new fields in a table). In Power BI, DAX lets you transform raw data into meaningful business insights and complex reports.

2. Explain the difference between calculated columns and measures in DAX.

A calculated column adds a new field to your existing data table and computes a value for *every single row*. This value is stored in the data model. A measure, on the other hand, is calculated on the fly when you drag it into a visual (like a table or chart). It aggregates data (like Sum or Average) based on the **filters** applied to that visual.

3. What is row context and filter context in DAX?

Row context is the current row being evaluated; it's the default context for calculated columns, meaning the formula looks at the values in that specific row. **Filter context** is the set of filters applied to a calculation, usually from slicers, visuals, or filter arguments in DAX functions. It determines the *subset* of data the formula will aggregate.

4. How does the RELATED function work in DAX?

The related function retrieves a value from a **column in another table**. To work, a relationship must already exist between the two tables in your data model. It only works in a row context, meaning you primarily use it in calculated columns to bring corresponding data from a "one" side of a relationship to the "many" side.

5. What is the purpose of the CALCULATE function in DAX?

Calculate is arguably the most important function in DAX; it's the **function that changes the** filter context. It takes an expression (like a measure) and one or more filter arguments. These filters temporarily override or modify any existing filters to force the expression to be calculated under a new set of conditions.

6. Describe the difference between ALL and ALLEXCEPT functions in DAX.

The ALL function is a table function that **removes all filters** from a table or specified columns. This is often used inside CALCULATE to compare a value to the total (e.g.,

calculating a percentage of total). ALLEXCEPT is similar but only removes all filters *except* those applied to the specified columns.

7. Explain the difference between VALUES and DISTINCT functions in DAX.

The **VALUES** function returns a **table of unique values** from a column, including the special **blank** row if it exists in the data. The **DISTINCT** function also returns a **table of unique values** from a column but does **not** include the special blank row. Both are often used to define a set of items for iteration or filtering.

8. What is the role of the SUMMARIZE function in DAX?

The **SUMMARIZE** function is used to create a **new summary table** by grouping the original table's data. You specify the columns you want to group by and then define new columns using aggregation expressions (like SUM or COUNT). It's primarily used for internal calculations within other DAX expressions, not typically to create tables for visuals.

9. How do you handle date and time calculations in DAX?

DAX has many built-in functions for date and time handling. You use functions like **DATEDIFF** to find the difference between dates, **DATEADD** to shift dates, and **YEAR, MONTH, DAY** to extract parts of a date. A crucial practice is using a dedicated **Date Table** to manage date-related filters and calculations efficiently.

10. How do you use the IF function in DAX? Provide an example.

The **IF** function checks if a condition is true, then returns one value if it is true, and a different value if it is false. It follows the structure: IF(logical_test, value_if_true, value_if_false). For example, IF([Sales Amount] > 1000, "High", "Low") checks if sales exceed 1000.

11. Difference between the SUM and SUMX functions in DAX?

SUM is an aggregation function that simply adds up the values in a specified column. It ignores row context. **SUMX** is an **iterator function** that performs a row-by-row calculation first, and *then* sums up the results. This is essential for calculating totals for complex expressions that vary per row, like a profit: SUMX('Sales', 'Sales'[Quantity] * 'Sales'[Price]).

12. Difference between the SUMMARIZE and ADDCOLUMNS functions in DAX?

SUMMARIZE creates a **new summary table** by grouping the data based on specified columns, similar to a SQL GROUP BY. **ADDCOLUMNS** does not change the table's structure but simply **adds new calculated columns** to an *existing* or derived table. They are often used together to group data and then add a calculation to the grouped result.

13. How do you calculate the number of days between two dates using DAX?

You use the **DATEDIFF** function. This function takes two date columns or expressions and the interval you want to calculate the difference in. To get the number of days, the syntax would be: DATEDIFF(start_date, end_date, DAY).

14. How do you add or subtract a specific number of days from a date using DAX?

You use the **DATEADD** function or simple addition/subtraction. **DATEADD** shifts a column of dates by a specified number of intervals (like days, months, or years): DATEADD('Date'[Date], 7, DAY). You can also just add or subtract a number to a date column, as DAX treats dates as numbers: 'Date'[Date] + 7.

15. How do you format a date value using DAX?

The **FORMAT** function is used to convert a date (or any value) into a text string in a specific style. You provide the date expression and a format string. For example, to display a date as "Month Day, Year", you'd use: FORMAT('Date'[Date], "mmm dd, yyyy").

16. How do you remove leading or trailing spaces from a text string using DAX?

You use the **TRIM** function. This function takes a single text column or expression and returns a text string with spaces removed from the beginning and end. This is a common step for cleaning up data imported from external sources.

17. How can you apply multiple filters to a table using DAX?

You use the **CALCULATE** function, which is designed to modify the filter context. Inside CALCULATE, you separate multiple filter arguments with a **comma**. Each filter argument can be a boolean expression or another table function, and they are combined using a logical **AND** condition.