**Q. 1: What is DAX?**

Answer: The acronym of DAX is Data Analysis Expressions. It is known as a functional language. It means, calculations mostly use functions to generate the results. It is also called an expression language.

**Q. 2: Why are you using DAX? or What is the purpose of DAX?**

DAX is designed for enhancing data modelling, reporting and analytics capability. DAX helps to get more insights into your data that you wouldn’t otherwise be able to get. For example, enabling time intelligence capability.

**Q. 3: Are there specific data types for DAX?**

Yes, some data types are supported by DAX. Those are 7 types, whole number, decimal number, boolean, text, date, currency, N/A (Blank).

**Q. 4: Is it possible to define data type in DAX? Is there any benefit to that?**

Yes, it is possible to define the data type for each column based on the requirement.

The selection of the best data type helps to reduce the size of a data model and improve the performance when to refresh data and use of any report.

**Q. 5: How many types of operators are we using in DAX?**

There are 4 categories of DAX operators, comparison operators, arithmetic operators, text operators, and logical operators.

**Q. 6: How many function categories are there in DAX?**

As per Microsoft documentation, there are 14 types of function categories in DAX. Out of them, we are using mostly 9 function categories in Power BI.

* Date and Time
* Time Intelligence
* Information
* Logical
* Mathematical
* Statistical
* Text
* Parent/Child
* Other

**Q. 7: Are you following any data model concept when you are writing DAX functions? If yes, do you know how many components are there?**

Yes, the DAX function supports a data model concept. Just like one data model, it has data, tables, columns, relationships, measures, hierarchies. As we know, a data model consists of data, calculations and formatting rules and it combines to create an object. This object helps to explore and understand the dataset.

**Q. 8: Do you know about M? Is there any difference between M and DAX?**

Yes, in Power Query, we are writing the script which is known as M language or Power Query Formula language.

There is some difference between M and DAX.

M language is normally used for data transformation, we can typically say, it is for ETL.

Whereas DAX is used for data analysing purposes, which means using different DAX functions we can analyse data more effectively.

**Q. 9: How do you know in which scenario we need to use M or DAX?**

Both languages are valued based on their usages. For example, if we want to create one flag column based on some existing columns values, then it is preferable to create in Power Query Editor using M. Here we are transforming the data. Let’s consider another example where we want to provide year on year or month on month analysis. In this case, DAX is ideal because it has time intelligence functions for implement this. So after analysing the requirements, we can decide the best way to implement them. We need to learn both languages.

**Q. 10: What is DAX Variable?**

In DAX calculation, we can use variables to make the calculations easier to understand.When you are writing any complex or nested expression using DAX functions, variables can help to break these complex calculations into smaller, more useful sections.

**Q. 11: Why do we need the DAX variable?**

There are main 4 reasons to use DAX variables.

* Reduce Complexity
* Easy to Debug
* Improve Readability
* Improve Performance

**Q. 12: Tell me about the context concept in DAX. How many types of contexts are there?**

In DAX, context is the layer of filtering and is applied to calculations to produce a result related to every value of a visual or pivot table including rows and columns totals. By definition, context is an important concept for building high-performing, dynamic analyses and for troubleshooting problems in formulas.

**Q. 13: How many types of contexts are there?**

There are 3 types of contexts.

* Filter Context
* Row Context
* Query Context

**Q. 14: What do you mean by Filter context in DAX? How many types of filter contexts are there?**

When we are applying filters on the set of values of columns or tables using DAX calculations, that is known as Filter Context. There are two types, implicit and explicit filter context.

**Q. 15: How does Filter context work in a DAX expression?**

Before executing the core expression, the filter context will finalize. Once it is decided, filter rules (inside context) are applied across the data and calculations execute on the remaining data.

**Q. 16: In a DAX expression, which one will execute first Filter context or Row Context?**

Filter context applies on top of other contexts, such as row context or query context.

**Q. 17: Difference between implicit and explicit filter contexts.**

When in the DAX calculation, you have not declared filters explicitly, so the value will be different based on any dimension field selection. This is the effect of the DAX implicit filter context. Explicit filter context means in calculations where specifically adds or removes column filter rules to and from the filter context.

**Q. 18: What do you mean by Row Context?**

Row Context is related to current rows. If you create a calculation using the calculated column, the row context involves the values of all columns from the current row. If that table has a relationship with the other table, then it includes all the related values from the other table for that row.

**Q. 19: What is Query Context?**

The combination of row and filters create the final query for DAX. You can define this is as query context. Users explicitly mention row and filter context for DAX, and DAX implicitly creates the query context from that filter and row context.

**Q. 20: Will DAX relationships work for many to many relationships?**

No, DAX relationships work only for the one to one and one to many relationships. Self-joins that means a table joins back with itself, cannot be possible.

**Q. 21: For the DAX relationship, can you work with the composite key?**

No, in this situation where multiple columns are responsible for data uniqueness, we need to create a new column with the combination of multiple columns and use it for relationships.

**Q. 22: How DAX will work without a standard relationship? or**

**How will you join two tables without any standard relationship between them?**

We can use different DAX functions to solve this problem.

* CROSSJOIN → More than two tables can be joined. The final output will have cartesian effects
* GENERATE → Only Two tables. If you want to add FILTER, then it is the better option.
* NATURALINNERJOIN and NATURALLEFTOUTERJOIN → These will behave like inner join, left outer join.
* Union, Except, and Intersect → These functions help to merge or combine tables.

**Q. 23: What is the purpose of summarization in any Data Analysis or BI project?**

For any industry, millions of transaction data have been captured. From this large table, we can prepare a summary table using aggregation logic and display a summarised visual.

Performance of visual will be faster if data is fetching from the summary table instead of the detail table which has raw data. You will get more optimize reports for a better experience.

**Q. 24: Which functions are you going to use for summarization?**

There are mainly 3 functions,

* SUMMARIZE
* SUMMARIZECOLUMNS
* GROUP BY

**Q. 25: Do you know what is difference between SUMMARIZE and SUMMARIZECOLUMNS?**

SUMMARIZE function has both the context filter and row. Whereas SUMMARIZECOLUMNS has one only filter context. In comparison with SUMMARIZE, the SUMMARIZECOLUMNS function produces an optimized query plan. It means it has to perform lesser steps when the engine run the queries.

**Q. 26: If you observe, GROUP BY and SUMMARIZE, both have the same syntax. Then what is the difference between them?**

GROUP BY differs from SUMMARIZE and SUMMARIZECOLUMNS in the <expression> section arguments. GROUP BY only works with DAX iterator functions, so it uses SUMX instead of SUM, AVERAGEX rather than AVERAGE. For specific scenarios, it works.

**Q. 27: What is the difference between CALCULATETABLE and CALCULATE?**

In terms of functionality, CALCULATETABLE is the same as CALCULATE function, but the difference is in their output. CALCULATETABLE returns a table whereas CALCULATE returns a single value like an integer or a string.

**Q. 28: CALCULATETABLE and FILTER, both functions return a table, then what is the difference between them?**

Both have the same syntax and will generate the same number of rows.

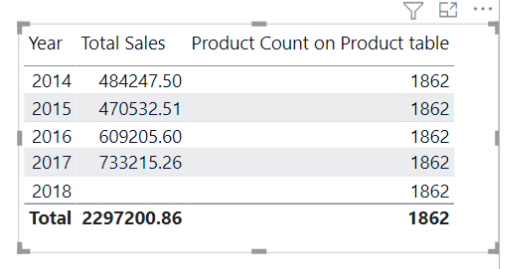
But there is a difference between the CALCULATETABLE and FILTER functions in terms of formula interpretation.

CALCULATETABLE first work on filter context, then evaluate the expression.

On the other hand, FILTER iterates the first expression results and it is not working on the filter context.

**Q. 29: In spite of having a relationship between two tables, still we are using CROSSFILTER. Why is it so?**

CROSSFILTER function is used to implement a specific cross-filtering behaviour in a calculation where a relationship exists between two columns. For example, we have two tables Fact-Sales and Dim-Product. They are joined with product\_key. Now we create a table with year, sales and product count (considering products from product dimension). Here is the output



But, this is wrong, the product count is repeated for each year. To solve this, if we are going to change the relationship between these two tables as both cross filter direction, then all other measures will be impacted. But we want to use both cross filter direction relationships only for this measure. We can use the CROSSFILTER function and recreate the measure.

CALCULATE(DISTINCTCOUNT(<ProductID from Dim-Product>,CROSSFILTER(<Sales\_Fact-ProductID>,<Dim\_Product\_ProductID>Both))

**Q. 30: What is the purpose of the USERELATIONSHIP function?**

The USERELATIONSHIP function does not return anything. It specifies between two columns to define the relationship for a calculation. It uses as an argument for other functions like CALCULATE, CALCULATETABLE etc.

**Q. 31: What is the difference between RELATED and RELATEDTABLE?**

The RELATED function returns a column from another related table. It requires a relationship between the current table and the related table. It follows many to one relationship. Without mentioning a relationship, it will not work.

The RELATEDTABLE returns a table of values. It considers an existing table as a parameter. It follows one to many relationships, which means you need to create the measure in that table that is holding one information.

**Q. 32:Why do we need a date table in Power BI?**

In any data set which is set for a data analytics project, we can find date information. After importing the data set, the date column is automatically recognized as a date and created a hierarchy. But sometimes we need a separate date table. If we want to use the Time intelligence features of Power BI, the main prerequisite is, we need one date table in the data model.

To create a proper star-schema data model, we need different dimension tables and a date table is among them.

**Q. 33: What is the difference between CALENDER and CALENDERAUTO functions.**

For the CALENDER() function, you need to provide the start date and end date.

Whereas, for CALENDERAUTO(), Power BI automatically finds the start year and end year from the data model where some date fields exist.

**Q. 34: Could you please tell me some common date-time DAX functions which you used in the project?**

There are 23 date and time DAX functions. When we create a date table, some functions are very common to use. For example, DATE(),YEAR(), MONTH(), WEEKNUM(), WEEKDAY(), TODAY(), DATEDIFF() etc.

**Q. 35: How to handle multiple relationships to the Date Table?**

In any data set, if we have two date fields, for example, order date and ship date, then there are two ways to handle it. Creating duplicate fields (calculated measures) using active and inactive relationships between the fact table and single date table.

Instead of creating duplicate fields for each measure, we can create different date tables, for example, one for order date, another for ship date. From a maintenance point of view, it is a better solution, but it is complex to display ordered sales and shipped sales in the same report.

**Q. 36: If you want to use the Time Intelligence function in your project, what are the prerequisite to start with?**

To start with Time Intelligence functions, we need one Date table in the data model. As well as the Date table should meet the following requirements.

* A Date column with date/time data type
* Unique values for date column
* No missing values or blanks
* The date column should cover full years

**Q. 37: How many types of categorizations are available in DAX time intelligence functions?**

There are two main categorizations for DAX time intelligence functions, one is summarizations over time and another is comparisons over time.

**Q. 38: Give some examples of summarizations over time functions.**

These functions are related to year-to-date, quarter-to-date, and month-to-date.

With CALCULATE function, we can use DATESYTD, DATESQTD, DATESMTD functions.

Without CALCULATE function, we can use simplified DAX functions TOTALYTD, TOTALQTD, and TOTALMTD.