Question 1: Difference between a report & a dashboard.

In Power BI, reports and dashboards are distinct components used for data visualization and analysis. They serve different purposes and offer varying levels of interactivity and detail.

Report

- Definition: A report in Power BI is a multi-page document that provides a detailed view of your data. It can contain a variety of visualizations, such as charts, tables, and maps, and allows for complex data exploration and analysis.

- Content: Reports can have multiple pages and provide in-depth details about data. They can include a wide range of visuals and offer interactive elements like slicers and filters to drill down into specific aspects of the data.

- Interactivity: Reports are highly interactive, allowing users to explore data in depth. Users can interact with visuals, apply filters, and drill through data to gain insights.

- Creation: Reports are typically created in Power BI Desktop and published to the Power BI Service. They are designed to be comprehensive and detailed, catering to various analytical needs.

- Usage: Reports are used for detailed data analysis and exploration. They are suitable for scenarios where users need to analyze data across multiple dimensions and view extensive data details.

Dashboard

- Definition: A dashboard in Power BI is a single-page, interactive view that aggregates key metrics and visualizations from various reports and datasets. It provides a high-level overview of important business metrics and KPIs.

- Content: Dashboards consist of tiles that can display visualizations, reports, and real-time data. Each tile can link to detailed reports or other dashboards, providing a snapshot of key information.

- Interactivity: Dashboards are interactive but are designed to provide a summary view. Users can click on tiles to navigate to detailed reports or filter data at a high level. However, dashboards themselves are generally less detailed than reports.

- Creation: Dashboards are created and customized within the Power BI Service. They allow users to pin visuals from different reports and datasets to a single page for quick access to essential information.

- Usage: Dashboards are used for monitoring and visualizing high-level metrics and KPIs. They are ideal for executive summaries and for users who need quick access to important information without delving into detailed reports.

Summary

In summary, reports are detailed, multi-page documents offering in-depth data analysis and interactive exploration, while dashboards are single-page summaries that aggregate key metrics and provide a high-level overview. Reports are suited for detailed analysis, whereas dashboards are designed for quick, high-level monitoring of business performance.

Question 2: Explain different types of Power BI Licences.

https://www.microsoft.com/en-in/power-platform/products/power-bi/pricing#tabs-pill-bar-ocbbe94_tab0

Question 3: What is DAX, explain the difference between measures & calculated columns.

DAX (Data Analysis Expressions) is a powerful formula language used in Power BI, Power Pivot, and SQL Server Analysis Services (SSAS) to define custom calculations and create complex data models. DAX allows users to create calculations, aggregations, and data manipulations that enhance reporting and analytical capabilities.

Measures vs. Calculated Columns

Both measures and calculated columns are used in DAX to perform calculations, but they serve different purposes and are used in different contexts. Here's a detailed explanation of each:

Measures

- Definition: Measures are calculations that are performed on data dynamically based on the context in which they are used. They are typically used to aggregate data and produce results that can change depending on the filters or slicers applied to a report.

- Characteristics:
  - Context-Aware: Measures are calculated based on the current context, which includes filters and slicers applied to the data. This makes them flexible for dynamic analysis.
  - Storage: Measures do not store data. Instead, they are computed on the fly when the data is queried.
  - Usage: Measures are often used for aggregations such as sums, averages, counts, and more complex calculations that depend on user interactions or report filters.

- Example:

```
  Total Sales = SUM(Sales[SalesAmount])
```

In this example, `Total Sales` is a measure that calculates the sum of the `SalesAmount` column from the `Sales` table. Its value changes based on the filters applied in the report.

Calculated Columns

- Definition: Calculated columns are columns added to a table that are computed using a DAX formula. These columns are calculated row by row when the data is loaded or refreshed and are stored in the data model.

- Characteristics:
  - Row Context: Calculated columns are evaluated for each row individually and do not change based on the context in which they are used. They provide static results for each row.
  - Storage: Calculated columns are stored in the data model and consume additional memory. Their values are precomputed and available for querying.
  - Usage: Calculated columns are useful for adding new data to your model based on existing columns, such as creating a new field that combines other fields or performs row-level calculations.

- Example:

```
  Full Name = [FirstName] & " " & [LastName]
```

In this example, `Full Name` is a calculated column that concatenates the `FirstName` and `LastName` columns for each row in the table. The result is stored in the table and can be used like any other column in the data model.

Summary

DAX is a versatile language for creating custom calculations in Power BI. Measures are dynamic calculations that respond to the context of filters and slicers in reports, providing aggregated results. They are calculated on-the-fly and do not store data. Calculated columns, on the other hand, perform row-level calculations and store results in the data model, providing static values for each row. Measures are typically used for aggregations and analysis, while calculated columns are used for adding new data fields to your model.

Question 4: What is the difference between SUM & SUMX functions? Explain calculate & Summarize function in DAX, explain using 2 examples each.

In DAX (Data Analysis Expressions), functions like SUM and SUMX are used for aggregating data, but they have distinct uses and behaviors. CALCULATE and SUMMARIZE are also important functions for manipulating and summarizing data in Power BI. Here's a detailed explanation of each, along with examples.

Difference Between SUM and SUMX

SUM

- Definition: The `SUM` function calculates the total of a single column in a table. It performs a straightforward aggregation of numeric values.

- Usage: `SUM` is used when you need to total the values in a single column without any additional calculations or conditions.

- Syntax:

```
 SUM(<column>)
```

- Example:

```
  Total Sales = SUM(Sales[SalesAmount])
```

  This example calculates the total sales amount by summing all values in the `SalesAmount` column of the `Sales` table.

SUMX

- Definition: The `SUMX` function is an iterator that performs row-by-row calculations before aggregating the results. It allows you to perform calculations on each row and then sum the results.

- Usage: `SUMX` is used when you need to calculate an expression for each row and then sum those results, especially when the calculation involves multiple columns or complex expressions.

- Syntax:

```
  SUMX(<table>, <expression>)
```

- Example:

```
  Total Profit = SUMX(Sales, Sales[Quantity] * Sales[ProfitMargin])
```

  In this example, `SUMX` calculates the profit for each row by multiplying `Quantity` by `ProfitMargin` and then sums up these values to get the total profit.

CALCULATE Function

- Definition: The `CALCULATE` function evaluates an expression in a modified filter context. It allows you to apply filters and modify the context in which a calculation is performed.

- Usage: `CALCULATE` is used to change the context in which a measure or calculation is evaluated, allowing for more dynamic and conditional calculations.

- Syntax:

```
 CALCULATE(<expression>, <filter1>, <filter2>, ...)
```

- Example 1:

```
 Total Sales 2024 = CALCULATE(SUM(Sales[SalesAmount]), Sales[Year] = 2024)
```

This example calculates the total sales amount for the year 2024 by applying a filter that restricts the data to 2024.

- Example 2:

```
 Total Sales High Margin = CALCULATE(SUM(Sales[SalesAmount]), Sales[ProfitMargin] > 0.3)
```

In this example, `CALCULATE` computes the total sales amount where the profit margin is greater than 30%.

SUMMARIZE Function

- Definition: The `SUMMARIZE` function groups data by one or more columns and performs aggregations on these groups. It returns a table with the aggregated results.

- Usage: `SUMMARIZE` is used for creating summary tables that group data based on specific columns and calculate aggregations like sums, averages, or counts.

- Syntax:

```
  SUMMARIZE(<table>, <groupBy_column1>, <groupBy_column2>, ..., <name1>, <expression1>, <name2>, <expression2>, ...)
```

- Example 1:

```
  Sales Summary = SUMMARIZE(Sales, Sales[ProductCategory], "Total Sales", SUM(Sales[SalesAmount]))
```

This example creates a summary table grouped by `ProductCategory` and calculates the total sales amount for each category.

- Example 2:

```
  Employee Summary = SUMMARIZE(Employees, Employees[Department], "Average Salary", AVERAGE(Employees[Salary]))
```

Here, `SUMMARIZE` groups employees by `Department` and calculates the average salary within each department.

Summary

- SUM and SUMX: `SUM` totals values in a single column, while `SUMX` iterates over rows, calculates an expression for each row, and then sums the results.

- CALCULATE: Changes the context of a calculation, allowing for dynamic and conditional evaluations based on applied filters.

- SUMMARIZE: Groups data by specified columns and performs aggregations, creating a summary table based on the groupings and calculations.

These functions are fundamental for creating dynamic and detailed analyses in Power BI and other DAX-enabled tools.

Question 5: What are time intelligence functions in DAX, give an example to illustrate the use of any two-time intelligence functions.

Time Intelligence functions in DAX are designed to perform calculations based on dates and time periods. These functions help in analyzing data over different time periods, such as months, quarters, and years, and are useful for creating reports that require time-based calculations like year-to-date, month-to-date, and comparisons between different periods.

Common Time Intelligence Functions in DAX

1. TOTALYTD: Calculates the Year-To-Date value for a given expression.

2. SAMEPERIODLASTYEAR: Returns a table that contains a column of dates shifted by one year.

3. DATEADD: Returns a table that contains a column of dates shifted by a specified number of intervals (e.g., days, months).

4. DATESYTD: Returns a table with dates from the start of the year to the specified date.

5. DATESMTD: Returns a table with dates from the start of the month to the specified date.

6. PARALLELPERIOD: Returns a table that contains a column of dates shifted by a specified number of intervals.

7. QUARTER: Provides the quarter number of the year for a given date.

Examples Illustrating Time Intelligence Functions

Example 1: `TOTALYTD`

Objective: Calculate the total sales year-to-date.

- Syntax:

```
 TOTALYTD(<expression>, <dates>, [<filter>])
```

- Example:

```
 Total Sales YTD = TOTALYTD(SUM(Sales[SalesAmount]), Dates[Date])
```

In this example, `TOTALYTD` calculates the total sales amount from the beginning of the year up to the current date based on the `Dates[Date]` column. If you have a date column `Dates[Date]` in your model, this function will sum the `SalesAmount` for the current year up to the date specified in the report's filter context.

Example 2: `SAMEPERIODLASTYEAR`

Objective: Compare sales with the same period last year.

- Syntax:

```
 SAMEPERIODLASTYEAR(<dates>)
```

- Example:

Sales Last Year = CALCULATE(SUM(Sales[SalesAmount]),
SAMEPERIODLASTYEAR(Dates[Date]))

In this example, `SAMEPERIODLASTYEAR` is used within the `CALCULATE` function to compute the total sales amount for the same period in the previous year. This is useful for comparing current year sales with those of the previous year. If the report shows data for January 2024, this calculation will retrieve sales for January 2023.

Summary

Time Intelligence functions in DAX, such as `TOTALYTD` and `SAMEPERIODLASTYEAR`, provide powerful tools for analyzing data over time. `TOTALYTD` helps in aggregating values from the start of the year up to a specified date, while `SAMEPERIODLASTYEAR` facilitates comparisons with data from the same period in the previous year. These functions are essential for performing time-based analysis and creating meaningful reports in Power BI and other DAX-supported platforms.