Power BI Interview Questions

Question 1: What license do you have right now in your project?
Answer: Yes I am using Power BI Pro license.

Question 2: Are you missing any feature that is in premium?

Answer:
1. You can consume reports and dashboards on your organization's portal and mobile apps.
2. It enables data analysis and visualization with AI
3. Reusable data workflows
4. Paginated report creation
5. You can access a Power BI Report Server and the latest embedded analytics capabilities.

Question 3: Have you created any **Paginated Reports** in the past? Tell me the use case where we can apply **Paginated Reports**?

Answer: If you have worked on **Paginated Reports** then u can tell your experience. But if you haven't worked on it then you can refer below answer.

Paginated Reports: Paginated reports are designed for printing purpose or sharing the files in required format. It will display all the data in table format.

Use case -
1. Invoices and Statements: Generating customer invoices or other transactional documents that need to be printed or distributed in a standardized format
2. Financial Reports - Creating detailed balance sheets, income statements, and cash flow statements
3. Exporting to PDF - Creating reports that need to be exported to PDF with consistent layout and formatting for sharing.

Any knowledge on embedded power bi?

Question 4: There is a Power BI APP in Power BI service which have 5 reports. There are two user from different department who will consume the App data, now because of confidentiality of data the first person should only see the first two report and the second person should able to see only remaining last three reports. How you will configure this scenario while creating Power BI App?

Answer: We can create multiple views in Power BI apps in order to keep the report hide which are not supposed to see by others.

While creating an app after "Setup" tab and "Content" tab, we go to "Audience" tab their we can hide the report. In this scenario we will create two new Audience for first user and second user then we will hide the report which are not supposed to see by other user.

Question 5: Suppose your data is placed, on premise **Sql Server Database**. Now tell me the steps from connecting to **Power BI Desktop** till refreshing the dataset in **Power BI Workspace**?

Answer:
1. Click on Get data —> Sql Server
2. Enter server and database names and select data connection as import, click on ok
3. Enter credentials and click on connect
4. Now create you report
5. Save the report and go to home and click on Publish
6. Go to workspace
7. Click on setting icon
8. Click on Power BI Setting option
9. Go to semantic model tab, select data set and expand gateway and cloud connection
10. In mostly organization there are separate team who will responsible for gateway settings, or someone from client side who will have admin rights. You need to ask them to setup the gateway
11. Now in the same window expand schedule refresh and schedule it as per the requirement. Also you can run on demand refresh

Question 6: Did you ever Edit Power BI dataset through the XMLA endpoint using Tabular Editor? If yes, How do you push your change from tabular editor to your dataset?

Answer: Note – This question is mostly asked to experienced Power BI developers those who have around more than 2 to 3 years of exp. Else Interviewer might ask you if you tell them that you have used Tabular editor and XMLA endpoint.

How to connect Power BI data through the XMLA endpoint using Tabular Editor:
1. Go to you Power BI dataset in workspace and click on three dots
2. Click on setting and go to Connection String and expand it
3. Copy the link and go to Tabular editor
4. Go to file in Tabular editor and click on open and select from DB
5. Provide the link which copied from connection string
6. Click on Ok

How to Publish the
1. Once you are ready to publish your changes to the Power BI service, perform a deployment through Tabular Editor (**Model > Deploy**), thus creating a new or overwriting an existing dataset in a Power BI workspace.

Question 7: Difference between **Live Connection** and **DirectQuery**?

Answer:

Live Connection:
1. Connects report to pre-existing semantic model.
2. No data duplication; relies on external model.
3. Supports Power BI service models, Azure Analysis Services, and on-premises SQL Server Analysis Services.
4. Suitable for smaller datasets and non-real-time data needs.

DirectQuery:
1. Dynamically queries data from connected source.
2. No data duplication; real-time access to data.
3. Supports various data sources like SQL Server databases.
4. Ideal for large datasets and 'near real-time' data requirements.

Question 8: Have ever tested how long the visual is taking?

Answer: By using Performance Analyzer in Power BI, This feature allows you to analyze the time taken by each visual, DAX measure, or data query.

Here's how you can do it:
In Power BI Desktop, go to the "View" tab in the top menu.
Click on "Performance Analyzer" in the "Show" group. This will open the Performance Analyzer pane.

Here's a step-by-step guide on how to use the Performance Analyzer:
1. Click on "Start Recording" in the Performance Analyzer pane. This will start recording the performance metrics.
2. Interact with your report as you normally would. Click on different visuals, change slicer selections, or apply filters. The Performance Analyzer will track the time taken for each action.
3. After you've performed the actions you want to analyze, click on "Stop Recording" in the Performance Analyzer pane.
4. The Performance Analyzer will display a list of events along with their corresponding durations. You can see how long each visual took to render, as well as the time taken for data queries and DAX calculations.
5. You can click on each event to get more details, such as the breakdown of time spent on data retrieval, visual rendering, and DAX calculation.

Question 9: Difference between formula engine & storage engine in dax studio?

Answer:
Function: The Formula Engine executes the DAX calculations and expressions, while the Storage Engine retrieves and prepares the data.
Processing: Formula Engine processes DAX expressions and filter context to calculate results, while Storage Engine handles data retrieval, aggregation, and storage optimization.
Aggregations: Formula Engine performs aggregations at a granular level, while Storage Engine pre-aggregates and optimizes data storage to improve query performance.
Responsibilities: Formula Engine focuses on dynamic calculations based on the current context, while Storage Engine focuses on optimizing data storage and preparing data for calculations.

Question 10: Any issue which you faced in live connection or any limitation you can tell me about live connection?

Answer:
Limited Data Source Support: Not all data sources are supported for live connections. If your desired data source is not supported, you won't be able to use a live connection.

Limited to Report Layout: When connected to certain data sources like SSAS, you might be limited to the report layout and unable to create calculated columns or measures directly within Power BI.

Question 11: What is the difference between **SUMMARIZE** and **GROUPBY**?

Answer:
SUMMARIZE:
1. Syntax: SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, <name>, <expression>]...)
2. Requires referring to actual table names within expressions.
3. Cannot nest SUMMARIZE functions with references to the inner function's columns in the outer function.

GROUPBY:
1. Syntax: GROUPBY(<table>, [<groupBy_columnName1>], [<name>, <expression>]...)
2. Uses CURRENTGROUP function within expressions.
3. Allows nesting of GROUPBY functions with references to the inner function's columns in the outer function.

Question 12: What is the difference between VALUE and VALUES in DAX?

Answer: Often interviewer ask this question to confuse the candidate. Please refer below answer for this question:

VALUE:
1. Converts a text string that represents a number to a number
2. Syntax --> VALUE(<text>)
3. If text is not in one of below formats, an error is returned
a. Supporting format(constant, number, date, or time)

Example : The following formula converts the typed string, "3", into the numeric value 3.
= VALUE("3")

VALUES:
1. Returns a table with distinct values from a specified column. It will also balnk value
2. Syntax --> VALUES(<TableNameOrColumnName>)
3. Example: The following formula counts the number of unique invoices (sales orders), and produces the following results when used in a report that includes the Product Category Names:
= COUNTROWS(VALUES('InternetSales_USD'[SalesOrderNumber]))

Question 13: What is the difference between RELATED and RELATEDTABLE in DAX? Use case where you have used RELATED and RELATEDTABLE

Answer:
RELATED:
1. Related function requires active relationship between two tables

2. Related function works only in Many-to-one relationship
3. Related function returns single values as an output

RELATEDTABLE:
1. RelatedTable function also requires active relationship between two tables
2. RelatedTable function works in Many-to-Many, Many-to-one and One-to-One relationships
3. RelatedTable function returns table as an output
4. RelatedTable function works with aggregator functions like COUNTROWS, SUMX & AVERAGEX
5. RelatedTable function is same as CalculateTable function except it doesn't have logical expression, it also transform from row context to filter context.

For Usecase please refer Youtube video link - https://lnkd.in/dveFSjXi

Question 14: Difference between PARALLELPERIOD vs SAMEPERIODLASTYEAR

Answer:
SAMEPERIODLASTYEAR :
1. Syntax - SAMEPERIODLASTYEAR(<dates>)
2. The SAMEPERIODLASTYEAR function is dynamic, adjusting with the filter context's period.
3. It simply goes one year back.

PARALLELPERIOD :
1. Syntax - PARALLELPERIOD(<dates>,<number_of_intervals>,<interval>)
2. PARALLELPERIOD is working STATICALLY based on the interval selected in the parameter.
3. It can navigate more than one interval back and forward.
4. PARALLELPERIOD only operates on month, quarter, and year interval.
For example: For Q4 of 2006 SAMEPERIODLASTYEAR will return Q4 of 2005, but PARALLELPERIOD will return all 2005.

Question 15: Why is **drillthrough** used in Power BI? How to Set up the **drillthrough** target page in Power BI?

Answer: With **drillthrough** in Power BI reports, you can create a destination target page in your report that focuses on a specific entity such as supplier, customer or manufacturer. When you right click on any specific data point in other source report page and click on **drillthrough** Power BI will take you to target report page with detailed information.

Setup Drillthrough in Power BI:
1. Create a new target page for the **drillthrough** details.
2. Add visualizations to the target **drillthrough** page that display the detailed data, then add columns and KPI to visualization
3. Go to **drillthrough** filter in target page, add columns to **drillthrough** filter to the page
4. Test the **drillthrough** functionality by right-clicking on a data point in the source report page and selecting "**Drill through**" to navigate to the target page with filtered information.

Question 16: What is the purpose of TREATAS function in DAX?

Answer: TREATAS is a DAX function in Power BI that allows you to create temporary relationships between tables. It's often used to enhance relationships in scenarios where you want to filter data in one table based on the values in another table without creating a permanent relationship.

Let's say you have two tables: Expense and Calendar. The Expense table has the granularity till month, and the Calendar table also has the granularity till Date. Since granularity is different hence making a relationship between them will affect you model. However, there might be situations where you want to filter the Expense table based on a different set of month without having the permanent relationship.

Here's where TREATAS comes in. You can use it to create a virtual relationship between the tables for a specific calculation. For example, you might want to calculate Expense for a specific range of month and year, even if there's no direct relationship in the model.

Youtube link - https://lnkd.in/d_jAEhBG

Question 17: If I don't have any option I only need to use import mode then what do you think what will optimize my report?

Answer :

Data Modeling: Properly design your data model with relationships, hierarchies, and calculated columns to reduce the size and complexity of your datasets.

Remove Unnecessary Columns: Only import columns that are needed for your analysis to reduce the size of your data model.

Data Cleansing: Clean and transform your data in Power Query Editor to remove nulls, duplicates, and irrelevant columns.

Use Query Folding: When possible, use query folding in Power Query to push data transformation steps back to the data source. This reduces the amount of data brought into Power BI.

Filter Data Early: Apply filters and transformations early in your data loading process to reduce the amount of data imported.

Pre-Aggregate Data: Create summary tables with pre-aggregated values for large datasets. This can speed up calculations and reduce the amount of data needed to be loaded.

Limit Visuals: Avoid creating too many visuals on a single page. Each visual requires calculations and can impact performance.

Question 18: Assume you have product and sales table? Write a DAX formula to Rank the product by highest to lowest sales?

Answer :

Product Rank =

RANKX(
ALL('Product'[Product]), -- All products to rank across all products

```
CALCULATE(
SUM('Sales'[SalesAmount]),  -- Calculate the total sales amount for each product
ALLEXCEPT('Product', 'Product'[Product])  -- Remove all filters except for the product ),
 , DESC, Dense  -- Sort in descending order, assign dense ranks (no gaps between ranks)
)
```