**Name: Bansari Prajapati**

**MODULE-5**

**Assignment**

**1) What is Power BI and how does it differ from Excel?**

-> Power BI is a business intelligence and data visualization tool primarily used for data analysis and reporting from multiple sources, interactive dashboards and visualizations for decision-making, sharing insights across organizations.

* Power BI is designed for business intelligence, creating dashboards, and sharing reports at scale while Excel is a general-purpose spreadsheet tool for calculations, analysis, and smaller-scale reporting.
* Power BI handles large datasets with optimized compression for big data while Excel performance decreases with very large datasets.
* Power BI refreshes dashboards automatically (scheduled or real-time updates). Excel requires manual updates or custom VBA scripting for automation.

**2) Explain the concept of data modeling in Power BI.**

-> Data modeling in Power BI is the process of structuring and organizing data from different sources into a coherent and efficient format that allows for meaningful analysis and visualization. It involves creating relationships between tables, defining calculated fields, and optimizing data to make it useful for building reports and dashboards.

**3) What are the different types of connections available in Power BI?**

->Power BI offers several types of connections to data sources, including Import, DirectQuery, and Live Connection.

**4) How do you handle data transformation in Power BI?**

-> Data transformation in Power BI is the process of cleaning, shaping, and preparing raw data for analysis and visualization. Power BI uses Power Query Editor, a powerful data transformation tool, to handle these tasks.

**5) What is DAX (Data Analysis Expressions) and why is it important in Power BI?**

-> DAX is a formula language used in Power BI, Power Pivot, and SQL Server Analysis Services (SSAS) for creating custom calculations and aggregations on your data. It is a powerful tool for data analysis and enables the creation of dynamic and complex measures, calculated columns, and tables. DAX is a combination of functions, operators, and constants designed to work with relational data models. It shares some similarities with Excel formulas but is far more robust and optimized for data analytics and modeling.

**6) Can you explain the difference between calculated columns and measures in Power BI?**

* Calculated columns work with row context while measures work with filter context. Calculated column.
* Calculated columns store data in the data model, consuming memory while measures do not consume any storage.
* Calculated columns may affect model size and performance for large datasets while measures are more efficient for large datasets (computed dynamically).

**7) How do you handle relationships between tables in Power BI?**

-> Accessing the Relationships View

* Go to the Model View (icon with table and diagram) in Power BI Desktop.
* This view shows all the tables in your model and their relationships.

Creating Relationships

* Relationships can be created manually or detected automatically by Power BI.

Manually Creating a Relationship:

* Drag a column from one table to a corresponding column in another table.
* Example: Drag CustomerID from a Sales table to CustomerID in a Customers table.

Automatically Detect Relationships:

* Use the Manage Relationships option in the Home ribbon.
* Power BI suggests relationships based on matching column names and data types.

Defining Relationship Properties

* When creating or managing a relationship, Power BI prompts you to define these properties:
  + Cardinality:
    - One-to-One (1:1): Each row in Table A corresponds to one row in Table B.
    - One-to-Many (1:\*): One row in Table A corresponds to many rows in Table B.
    - Many-to-Many (*:*): Many rows in Table A correspond to many rows in Table B.
  + Cross-Filter Direction:
    - Single: Filters flow in one direction (from one table to another).
    - Both: Filters flow in both directions. Useful for complex models.
  + Active Relationship:
    - A model can have multiple relationships between two tables, but only one can be active at a time.

Managing and Editing Relationships

* Use Manage Relationships in the ribbon to view, edit, or delete relationships.
* You can change the cardinality or filter direction if needed.

Handling Many-to-Many Relationships

* Introduced in Power BI to handle cases where both tables have duplicate values in the linking columns.
* Power BI automatically creates a bridge table or handles it directly using relationships.

Using Relationship Filtering

* Define how filters propagate between tables by setting cross-filter direction (single or both).
* Example: In a sales model, a filter on the Region table can propagate to the Sales table.

Marking a Table as a Date Table

* For time intelligence calculations, you must mark a table as a Date Table. This enables functions like YTD, QTD, and SamePeriodLastYear.

**8) What is the purpose of a Power BI Gateway?**

-> A Power BI Gateway is a bridge that connects Power BI cloud services to on-premises data sources securely. It allows Power BI to access data that resides in local (on-premises) servers or databases without moving it to the cloud.

**9) How can you schedule data refresh in Power BI Service?**

-> Set Up Scheduled Refresh

* Under the Scheduled Refresh section:
  + Toggle the Keep data updated option to On.
  + Set the Refresh Frequency:
    - Daily or Weekly.
    - Specify the days and time(s) for the refresh.
  + Optionally, enable notifications for refresh failures by toggling the appropriate setting.

Save Settings

* Click Apply to save your refresh schedule.

**10) Explain the concept of row-level security in Power BI.**

-> Row-Level Security (RLS) is a feature in Power BI that restricts data access for users based on roles. It ensures that users can only view the data that is relevant to them, even if they are working on the same dataset or report. RLS is commonly used to implement permissions and privacy for different users or groups.

**11) What is the Power BI Desktop and how does it differ from Power BI Service?**

-> Power BI Desktop is a free, Windows-based application used for creating, designing, and developing Power BI reports and data models. It is the primary tool for connecting to data sources, transforming and modeling data, creating visualizations, and building interactive reports. Once a report is built in Power BI Desktop, it can be published to the Power BI Service for sharing and collaboration. It is a desktop application while Power BI Services is a cloud-based platform accessible via a web browser. It is used for creating and developing reports and data models while Power BI Services is used for hosting, sharing, and collaborating on reports and dashboards.

**12) Explain the concept of Direct Query in Power BI.**

-> DirectQuery is a data connectivity mode in Power BI where queries are sent directly to the data source each time a user interacts with a visualization, rather than importing the data into Power BI. This enables real-time access to large datasets without the need to store the data in Power BI.

**13) What are Power BI templates and how are they useful?**

-> Power BI Templates are pre-configured Power BI files with a .PBIT extension that store the structure and design of a Power BI report without the actual data. Templates include the following:

1. Report Layout and Visuals: The design of the report, including charts, tables, slicers, and other visuals.
2. Data Model: Relationships between tables, calculated columns, measures, and DAX formulas.
3. Queries: Data transformation steps defined in the Power Query Editor.
4. Parameters: Dynamic inputs that allow users to customize the template for their own data.

Power BI Templates have significance in reusability, standardization, quick start, easy collaboration, and feasible customisation.

**14) How do you handle incremental data refresh in Power BI?**

-> Incremental data refresh in Power BI can be handled by following steps:

1. Filter Data by Date:
   * Ensure your dataset includes a reliable date column for filtering.
2. Optimize Query Performance:
   * Use indexed columns in the source database to speed up data filtering and retrieval.
3. Validate Parameters:
   * Test the parameters in Power Query to ensure they work as expected.
4. Use Detect Data Changes:
   * Configure the "Detect data changes" option to refresh only when necessary, minimizing unnecessary refreshes.
5. Partition Data Sensibly:
   * Choose appropriate ranges for historical and recent data to balance refresh performance and data retention.

**15) What is the role of Power Query in Power BI?**

-> Power Query allows users to connect to various data sources, clean, transform, and shape the data before loading it into the Power BI model. Power Query is the starting point for data preparation and helps ensure that the data used in reports is clean, consistent, and ready for analysis.

**16) Explain the difference between calculated columns and calculated tables in Power BI.**

| Aspect | Calculated Column | Calculated Table |
| --- | --- | --- |

| Purpose | Adds a new column to an existing table. | Creates a completely new table. |
| --- | --- | --- |

| Row vs. Table Context | Row context: Calculated for each row individually. | Table context: Works with entire datasets or groups. |
| --- | --- | --- |

| Input Source | Operates on data within a single table. | Can combine or aggregate data from multiple tables. |
| --- | --- | --- |

| Output | Single column. | Entire table. |
| --- | --- | --- |

| Performance Impact | Impacts storage and refresh time for the table. | May increase model size significantly. |
| --- | --- | --- |

| Use Case | Row-level calculations or derived fields. | Aggregations, filtering, or summarization. |
| --- | --- | --- |

**17) How do you create custom visuals in Power BI?**

-> In Power BI, we can create custom visuals to meet specific reporting needs that aren't satisfied by the built-in visuals. Custom visuals are developed using TypeScript, JavaScript, and the Power BI Visuals SDK. Once created, they can be imported and used in Power BI reports.

**18) What are the best practices for optimizing performance in Power BI?**

-> Optimizing Power BI performance is essential for ensuring fast report loading, smooth interaction, and efficient data processing. Poorly optimized reports can lead to long refresh times, laggy dashboards, and increased memory usage. Here are some ways:

1. Data Model Optimization: remove unnecessary columns and tables
2. DAX Optimization: use SUMX Instead of CALCULATE for Row Context
3. Query Optimization: use Query Folding (Push to Source)
4. Report and Visualization Optimization: limit the Number of Visuals on a Page
5. Data Refresh Optimization: use Incremental Refresh

**19) How can you integrate Power BI with other Microsoft products like Azure and Office 365?**

->**Product**: Azure SQL Database

**Integration type**: DirectQuery, Import

**Use cases**: Real-time querying

**Product**: Azure Data Lake

**Integration type**: Import, DirectQuery

**Use cases**: Big data storage

**20) Explain the concept of aggregations in Power BI.**

Aggregations in Power BI are a way to improve performance by pre-calculating and storing summary data at a higher level of granularity. Instead of performing calculations on detailed data every time a user interacts with a report, Power BI can query the pre-aggregated data, significantly reducing the amount of data processed and improving query response times.

| Aggregation Type | Description | Example |
| --- | --- | --- |
| Sum | Returns the total value of a column. | SUM(Sales[Amount]) |
| Average | Returns the mean of a column’s values. | AVERAGE(Sales[Amount]) |
| Min | Returns the smallest value in a column. | MIN(Sales[Amount]) |
| Max | Returns the largest value in a column. | MAX(Sales[Amount]) |
| Count | Returns the number of rows or distinct values. | COUNT(Sales[OrderID]) |
| Distinct Count | Returns the number of unique values in a column. | DISTINCTCOUNT(Sales[CustomerID]) |
| Median | Returns the middle value in a sorted column. | MEDIAN(Sales[Amount]) |
| Standard Deviation | Measures the spread of values around the mean. | STDEV(Sales[Amount]) |

**21) How do you handle error handling and data quality in Power BI?**

->1. Error Handling in Power Query

* Remove/Keep Errors – Remove or isolate rows with errors.
* Replace Errors – Replace errors with a default value using ReplaceErrors().
* Conditional Handling – Use IF to handle specific errors.
* Remove Null/Blank Values – Remove or fill missing values using Fill Down/Up.
* Data Profiling – Identify data quality issues using column statistics.

### 2. Error Handling in DAX

* IFERROR() – Replace errors with a fallback value.
  + ProfitMargin = IFERROR(DIVIDE(SUM(Profit), SUM(Revenue)), 0)
* COALESCE() – Handle null values with a default value.
  + TotalSales = COALESCE(SUM(Sales[Amount]), 0)
* ISBLANK() – Check for missing values and handle them.
* TRY/ERROR – Catch and handle calculation errors.

### 3. Data Quality Techniques

* Remove Duplicates – Use Table.Distinct() to eliminate duplicate rows.
* Fill Missing Data – Use Fill Down or Fill Up.
* Format Data – Convert text, dates, and numbers to consistent formats.
* Validate Data Types – Ensure correct types (e.g., dates, integers).

### 4. Handling Load and Refresh Errors

* Incremental Refresh – Load only new data to reduce processing time.
* Gateway Logs – Monitor and fix refresh failures.
* Set Alerts – Set up email or Teams alerts for failed refreshes.

### 5. Handling Visual Errors

* "Can't Display Visual" Error – Reduce data size, simplify visuals.
* Blank Values – Use IFBLANK() or COALESCE() to handle missing data.
* Friendly Error Messages – Provide user-friendly fallback text.

**22) What is the purpose of Power BI Embedded and when would you use it?**

-> Power BI Embedded is a service that allows developers to embed interactive Power BI reports and dashboards into their custom applications, websites, or portals. It enables seamless data visualization and analytics without requiring users to log into the Power BI service separately.

### When to Use It

1. Custom Applications – Embed Power BI reports into web or mobile applications to provide users with real-time insights.
2. Client-Facing Solutions – Provide clients with access to interactive reports without needing a Power BI license.
3. Internal Portals – Embed dashboards into internal company platforms like SharePoint or Teams.
4. White-Labeling – Offer analytics as part of a branded application experience.
5. Scalable Reporting – Handle high traffic and complex reporting needs through Azure infrastructure.

### Key Features

* No need for users to have a Power BI account.
* Scalable and secure using Azure infrastructure.
* Customizable with control over layout and branding.
* Supports DirectQuery, Import, and Live Connection modes.
* Uses row-level security to control data access.

### Example Use Case

* A company creates a customer portal where users can log in and view personalized sales reports embedded using Power BI Embedded.
* The portal is customized to match the company’s branding and provides real-time insights based on the user’s data access level.

**Practice link: (Charts)**