

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

→ Power BI is a business intelligence (BI) and data visualization tool developed by Microsoft that enables users to transform raw data into interactive, visually appealing reports and dashboards. It connects to various data sources, processes large datasets, and provides real-time insights for data-driven decision-making. Power BI is part of the Microsoft Power Platform and is designed for business analysts and organizations to analyse, visualize, and share data efficiently.

Excel is a spreadsheet application, also by Microsoft, primarily used for data entry, calculations, and basic data analysis. It's part of Microsoft Office and is widely used for tasks like budgeting, data organization, and basic reporting.

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

→ Data modeling in Power BI is the process of structuring and organizing data from various sources to create a coherent, efficient framework for analysis and visualization. It involves defining relationships between data tables, creating calculated measures and columns, and optimizing data for performance and usability in reports and dashboards. The goal is to transform raw data into a format that supports meaningful insights and efficient querying.

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

→ data connections refer to the methods used to connect to and retrieve data from various sources for analysis and visualization. Power BI supports multiple connection types, each suited to different data sources and use cases. Below is an overview of the main types of connections available in Power BI.

### **1. Import**

- Data is fully loaded into Power BI's in-memory engine (VertiPaq) and stored in the Power BI file (.pbix). All data processing and queries occur locally within Power BI.

### **2. Direct Query**

- Power BI queries the data source directly in real-time without importing data. Only metadata and schema are stored in Power BI.

### **3. Live Connection**

- Connects directly to a pre-built data model (e.g., in Analysis Services or Power BI datasets in the Power BI Service) without importing data. Power BI acts as a visualization layer.

### **4. Composite Models**

- Combines Import and Direct Query/Live Connection in a single model, allowing flexibility to mix local and live data.

### **5. Push Datasets**

- Data is pushed to Power BI via an API or streaming service, typically for real-time dashboards.

## **6. Web and API Connections**

- Connects to web-based data sources or APIs to retrieve data, typically in JSON or XML format.

## **7. Cloud-Based Connections**

- Connects to cloud platforms and services, often with native connectors for seamless integration.
- Power BI offers a range of connection types—Import, Direct Query, Live Connection, Composite Models, Push Datasets, ODBC/OLE DB, Web/API, and Cloud-Based—to suit different data sources, performance needs, and real-time requirements. Choosing the right connection depends on the data size, refresh frequency, and whether real-time access or local processing is prioritized.

## **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. It is primarily handled using Power Query, a powerful ETL (Extract, Transform, Load) tool integrated into Power BI. Power Query allows users to connect to various data sources, apply transformations, and create a structured dataset ready for modeling and reporting.

### **Steps for Data Transformation in Power BI–**

Connect to Data Sources, Open Power Query Editor, Apply Transformations( Cleaning Data, Shaping Data, Data Type Adjustments, Merging and Appending, Creating Calculated Columns, Grouping and Aggregating, Conditional Logic, Handling Missing Data), Review Applied Steps, Optimize Transformations and then Load Data to POWER BI.

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

→ DAX (Data Analysis Expressions) is a formula language used in Power BI, Excel Power Pivot, and SQL Server Analysis Services (SSAS) to create custom calculations and aggregations for data analysis. It is designed to work with relational data models and enables users to define complex calculations, measures, and relationships that enhance data analysis and reporting in Power BI.

Why is DAX Important in Power BI?

DAX is a critical component of Power BI because it empowers users to extract meaningful insights from data by enabling advanced calculations, dynamic aggregations, and complex data modelling

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

→ In Power BI, calculated columns and measures are both created using DAX (Data Analysis Expressions) to enhance data models, but they serve different purposes, operate in different

contexts, and have distinct behaviours. Below is a detailed explanation of the differences between calculated columns and measures, including their use cases, characteristics, and examples.

### Key Differences Between Calculated Columns and Measures

Aspect	Calculated Columns	Measures
<b>Definition</b>	A new column added to a table, calculated for each row using a DAX formula.	A dynamic calculation that aggregates or computes data across rows, often based on filters or slicers.
<b>Calculation Context</b>	Operates in row context, meaning the formula is evaluated for each row in the table.	Operates in filter context, meaning the calculation depends on report filters, slicers, or visuals.
<b>Storage</b>	Physically stored in the data model, increasing the model size.	Not stored; calculated on-the-fly when used in visuals or reports.
<b>Performance Impact</b>	Consumes memory as values are pre-calculated and stored for every row.	Consumes processing power as calculations occur at query time; no storage impact.
<b>Use Case</b>	Used for row-level calculations that need to be stored and reused (e.g., categorizing rows).	Used for aggregations or dynamic calculations (e.g., totals, averages, KPIs).
<b>Evaluation Timing</b>	Calculated during data refresh or model processing; static until refresh.	Calculated dynamically when a report or visual is rendered.
<b>Visibility</b>	Visible as a column in the table and can be used in slicers, filters, or visuals.	Not visible as a column; exists in the model and is used in visuals or calculations.
<b>Flexibility</b>	Fixed to row-level logic; less flexible for dynamic aggregations.	Highly flexible, adapting to user interactions like slicers or filters.
<b>Example DAX</b>	Profit = Sales[Revenue] - Sales[Cost] (calculated for each row).	Total Profit = SUM(Sales[Profit]) (aggregates profit across filtered data).

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

→ In Power BI, relationships between tables are a core component of data modeling, enabling seamless data analysis across multiple tables by linking them through common fields (keys). Properly managing relationships ensures that data is filtered, aggregated, and visualized accurately in reports and dashboards.

### Steps to Handle Relationships in Power BI-

Prepare Data, Load data into POWER BI, Access the Model View, Create Relationships, Configure Relationship Properties (One-to-One, One-to-Many, Many-to-Many), Manage Relationships, Test Relationships.

Handling relationships in Power BI involves connecting tables via keys, configuring cardinality and filter direction, and optimizing the model for performance and accuracy. By using the Model view, ensuring clean data, and following best practices like star schema design, you can create a robust data model that supports interactive and accurate reports.

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

→ A Power BI Gateway is a software component that enables secure, automated data transfer between on-premises data sources and the Power BI Service (cloud-based platform) for refreshing datasets, enabling real-time or scheduled data updates in reports and dashboards. It acts as a bridge between on-premises data (e.g., databases, files) and Microsoft's cloud services, ensuring that Power BI can access and refresh data securely without requiring direct exposure of on-premises systems to the internet.

### Purpose of a Power BI Gateway:

1. **Secure Data Transfer** – It ensures secure data movement from on-premises systems to the Power BI Service without exposing databases directly to the internet.
2. **Scheduled Refresh** – Enables automatic data refresh for on-premises datasets so that reports and dashboards stay up to date.
3. **Live/Direct Query** – Supports real-time or near real-time queries directly against on-premises databases without moving data to the cloud.
4. **Centralized Management** – Acts as a single point to manage multiple data connections and refresh schedules.
5. **Hybrid Connectivity** – Allows organizations to leverage both cloud-based Power BI and their existing on-premises data infrastructure.
6. **Reusability** – Once installed, multiple users and reports can use the same gateway for different datasets.

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

→ In Power BI Service, you can schedule data refresh to automatically update your datasets so dashboards and reports always show the latest information. Here's how you can do it:

### Steps to Schedule Data Refresh in Power BI Service:

1. **Publish your report**
  - First, publish the Power BI Desktop file (.pbix) to the Power BI Service.
2. **Go to your workspace**
  - In Power BI Service, navigate to the workspace where your dataset is published.
3. **Open Dataset Settings**
  - Locate the dataset you want to refresh.
  - Click on the More options (...) → Settings.
4. **Configure Data Source Credentials**
  - Under Data source credentials, ensure credentials are entered and valid so Power BI can connect to the data source.
5. **Schedule Refresh**
  - Scroll down to the Scheduled refresh section.
  - Turn Keep data updated to On.

- Set the refresh frequency (Daily or Weekly).
- Choose the time slots (up to 8 times per day for Pro, 48 times per day for Premium).
- Save your settings.

#### 6. (Optional) Gateway Setup

- If your data source is on-premises (like SQL Server, Oracle, Excel on local network), you must configure an On-premises Data Gateway.
- Without the gateway, scheduled refresh will not work for on-prem data sources.

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

→ Row-Level Security (RLS) in Power BI is a feature that restricts data access at the row level based on user roles, ensuring users only see data they are authorized to view. It's commonly used to secure sensitive data in reports and dashboards, tailoring data visibility based on user identity or role without needing separate reports.

#### How it works:

##### 1. Roles and Rules

- You define **roles** in Power BI Desktop (e.g., "Sales Manager", "Sales Rep").
- For each role, you apply **DAX filters** to restrict what data that role can access.  
Example:
- [Region] = "East".

##### 2. Assigning Roles to Users

- After publishing to the **Power BI Service**, you assign users (or groups from Azure Active Directory) to those roles.
- When users view reports, the RLS filters are automatically applied, so they only see their permitted rows.

##### 3. Static vs Dynamic RLS

- **Static RLS:** Fixed filter rules (e.g., Region = "East").
- **Dynamic RLS:** Rules depend on the logged-in user (using the USERPRINCIPALNAME() or USERNAME() DAX function).

## 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 to create, model, and visualize data for business intelligence reports. It's designed for data analysts and report developers to build interactive reports and dashboards locally before publishing them to the cloud.

→ **Power BI Service** is a cloud-based platform (accessible via a web browser at [app.powerbi.com](https://app.powerbi.com)) where reports and dashboards created in Power BI Desktop are published, shared, and consumed by end-users. It's designed for collaboration, sharing, and managing reports at scale.

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

→ Direct Query in Power BI is a data connection mode that allows Power BI to query data directly from the underlying data source in real-time, rather than importing and storing data within the Power BI model. This approach ensures that reports and visuals always reflect the most current data from the source, making it ideal for scenarios requiring up-to-date information or handling large datasets that are impractical to import.

### How it works:

- When you build a report in Direct Query mode, **no data is stored inside Power BI** (except metadata and visuals).
- Each time you interact with a visual (e.g., apply a filter, drill down), Power BI sends a query in real-time to the source system, retrieves the results, and displays them.

### Advantages of Direct Query:

1. **Real-time data access** – Always shows the latest data from the source.
2. **No data duplication** – Data stays in the source system, avoiding storage issues.
3. **Handles large datasets** – Suitable for very large databases that cannot be imported into Power BI.

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

→ Power BI Templates are pre-built files (.pbit) created in Power BI Desktop that contain the structure, data model, queries, and report layouts of a Power BI report but exclude the actual data. They serve as reusable blueprints for creating new reports with consistent design and functionality, allowing users to connect to their own data sources while retaining the template's predefined elements.

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

→ Incremental refresh allows you to **load only new or changed data** instead of refreshing the entire dataset every time. This makes refreshes **faster, more efficient, and scalable** for large datasets.

### Steps to Configure Incremental Refresh:

1. **Prepare the dataset with date filters.**
2. **Define Incremental Refresh Policy** (in Power BI Desktop).
3. **Publish to Power BI Service.**
4. **Scheduled Refresh.**

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

→ Power Query in Power BI is a powerful data transformation and preparation tool used to connect, clean, shape, and transform raw data from various sources into a format suitable for analysis and reporting. It serves as the data ingestion and transformation engine within Power BI Desktop (and partially in Power BI Service via dataflows), enabling users to prepare data before it's loaded into the Power BI data model for visualization and analysis.

### Role of Power Query in Power BI-

- **Data Connection**
- **Data Transformation**
- **Data Modeling Preparation**
- **Query Reusability**
- **Automation of ETL Processes**
- **Incremental Data Refresh Support**
- **User-Friendly Interface**

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

→ **Calculated Columns** and **Calculated Tables** are two distinct features used to enhance data models by creating new data elements using **Data Analysis Expressions (DAX)**. They serve different purposes and are used in different scenarios within Power BI Desktop and Power BI Service. Below is a detailed explanation of their differences, including their definitions, use cases, and key characteristics.

#### Key Differences Between Calculated Columns and Calculated Tables

Aspect	Calculated Columns	Calculated Tables
Definition	Adds a new column to an existing table.	Creates an entirely new table in the data model.
Scope	Row-level calculations within a single table.	Table-level operations, creating new rows/columns.
DAX Context	Uses row context for calculations.	Uses table-level DAX functions (e.g., FILTER, SUMMARIZE).
Output	A single column with a value for each row.	A new table with multiple columns and rows.
Use Case	Deriving per-row values (e.g., profit, categories).	Summarizing, filtering, or combining tables.
Storage Impact	Increases dataset size by one column's data.	Can significantly increase size, depending on table size.
Performance	Minimal impact unless complex calculations are used.	Higher impact, especially for large or complex tables.
Creation Process	Created via <b>Modeling &gt; New Column</b> or Fields pane.	Created via <b>Modeling &gt; New Table</b> or DAX editor.
Dependencies	Depends on columns in the same table.	Can depend on multiple tables or data sources.

Aspect	Calculated Columns	Calculated Tables
Flexibility	Limited to enhancing a single table.	Can create new datasets or reshape data entirely.

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

→ Creating custom visuals in Power BI involves developing tailored visualizations that extend beyond the built-in visuals provided by Power BI to meet specific business or design requirements. Power BI supports custom visuals through its Custom Visuals SDK, which allows developers to create interactive and unique visuals using web technologies like TypeScript, JavaScript, and D3.js. These visuals can be used in Power BI Desktop and Power BI Service, shared with others, or published to the Microsoft AppSource for broader use.

### Ways to Create Custom Visuals

1. Import from AppSource (Easiest Method).
2. Import from a.pbiviz File.
3. Develop Your Own Custom Visual (Advanced Method).

### When to Use Custom Visuals

- When built-in visuals don't meet business requirements.
- To add advanced charts (e.g., Sankey diagrams, Radar charts, Heatmaps).
- To match corporate branding/design requirements.

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

→ Optimizing performance in Power BI is critical to ensure fast report rendering, efficient data refreshes, and a smooth user experience, especially when dealing with large datasets or complex reports. Performance optimization involves improving data model efficiency, query execution, and report design. Below are best practices for optimizing performance in Power BI, categorized by key areas: data modeling, data refresh, DAX calculations, visuals, and general considerations.

### Best Practices for Optimizing Performance in Power BI

1. Data Modeling Optimization
2. Data Reduction
3. DAX Optimization
4. Query Performance
5. Visual & Report Design



6. Power Query Optimization

7. Service-Level Optimization.

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

→ Integrating Power BI with other Microsoft products like Azure and Microsoft 365 (formerly Office 365) enables organizations to leverage a cohesive ecosystem for data analytics, visualization, and collaboration. Power BI's native integration with these platforms allows seamless data connectivity, enhanced security, and improved productivity.

### **Integrating Power BI with Azure and Office 365**

#### **1. Integration with Azure**

- **Azure SQL Database / Azure Synapse**
  - Power BI connects directly to Azure SQL Database or Synapse Analytics for live queries or imported data.
- **Azure Data Lake / Blob Storage**
  - Store big data in Azure Data Lake or Blob and connect Power BI for analysis.
- **Azure Analysis Services**
  - Use as a semantic layer; Power BI reports consume data models from Azure Analysis Services.
- **Azure Machine Learning**
  - Integrate ML models into Power BI for predictive analytics.
- **Azure Active Directory (AAD)**
  - Provides secure authentication and row-level security via role-based access.

#### **2. Integration with Office 365**

- **Excel**
  - Import Excel files into Power BI.
  - Use **Analyse in Excel** to connect Power BI datasets directly in Excel.
- **SharePoint Online**
  - Publish reports/dashboards to SharePoint pages via **Power BI web part**.
- **Microsoft Teams**
  - Embed Power BI reports inside Teams channels or tabs.
  - Share dashboards in Teams chat.
- **Outlook & PowerPoint**
  - Export visuals from Power BI into PowerPoint presentations.
  - Share dashboards or snapshots via Outlook.
- **OneDrive.**
  - Keep Power BI reports in sync with Excel/CSV files stored on OneDrive or SharePoint.

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

→ Aggregations in Power BI are a performance optimization technique used to improve query speed and efficiency when working with large datasets, particularly in Direct Query or Import mode with big data scenarios. Aggregations involve creating pre-summarized tables that store aggregated data (e.g., sums, counts, averages) at a higher granularity level, allowing Power BI to query these smaller, optimized tables instead of the full, detailed dataset for certain queries. This reduces processing time and resource usage, especially for high-level reports or dashboards.

### How Aggregations Work

1. You have a **large fact table** (e.g., billions of sales transactions).
2. You create a **smaller aggregation table** with summarized data (e.g., Sales by Region, by Month).
3. In Power BI, you define the aggregation behaviour (e.g., SUM, COUNT, MIN, MAX).
4. When a report query matches the aggregation level (e.g., Sales by Region), Power BI automatically uses the aggregation table.
  - If the query asks for more detail (e.g., transaction-level data), Power BI falls back to the detailed fact table.

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

→ Handling error handling and data quality in Power BI is critical to ensure accurate, reliable, and trustworthy reports. Poor data quality or unhandled errors can lead to misleading insights, slow performance, or report failures. Power BI provides tools and techniques across Power Query, DAX, and Power BI Service to manage errors and maintain high data quality.

### 1. Power Query (Data Preparation Stage)

- Remove Errors
- Replace Errors
- Data Type Checks
- Null Handling
- Applied Steps Review.

### 2. DAX (Modeling Stage)

- Error-safe DAX functions
- Validation Columns

### 3. Data Quality Checks

- Profiling Tools in Power Query.
- Data Validation Reports
- Source System Fixes

### 4. Refresh & Gateway Error Handling

- Scheduled Refresh Errors.

- **Gateway Connection Errors**

## 5. Best Practices

- Document assumptions about missing or replaced values.
- Apply transformations as early as possible (ideally at the source).
- Use consistent data cleaning rules across all queries.
- Continuously monitor data quality with alerts/dashboards.

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

→ Power BI Embedded is a Microsoft Azure service that allows developers to embed Power BI reports, dashboards, and visuals into custom applications, websites, or portals, enabling seamless integration of analytics for end-users without requiring them to access the Power BI Service directly. It is designed for scenarios where organizations want to provide interactive, data-driven insights to internal users, external customers, or partners within their own applications, while maintaining control over branding, user experience, and security.

### When to Use Power BI Embedded

- 1. Independent Software Vendors (ISVs)**
  - SaaS product companies that want to provide analytics features directly within their product (e.g., a CRM or ERP tool showing embedded Power BI reports).
- 2. Customer-Facing Applications**
  - When you want **external users** (customers, partners, vendors) to view reports without giving them direct access to Power BI Service.
- 3. Custom Enterprise Portals**
  - Embedding analytics into intranet portals or business apps for internal employees.
- 4. White-Label Analytics**
  - To provide a seamless brand experience (embed reports without Power BI logo or UI).

