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

Power BI is a business analytics tool designed to help users visualize data, share insights, and make data-driven decisions. It allows you to connect to various data sources, transform data, create interactive dashboards and reports, and publish them for others to use.

**Power BI differ from Excel:**

|  |  |  |
| --- | --- | --- |
| **Feature / Aspect** | **Power BI** | **Excel** |
| **Primary Purpose** | Business intelligence, dashboards, and data reporting. | General-purpose spreadsheet and data analysis tool. |
| **Visualizations** | Highly interactive, advanced visuals and dashboards. | Basic charts and pivot tables, less interactive. |
| **Data Modeling** | Robust with relationships between tables. | Limited; mostly flat or simple relational data. |
| **Data Volume** | Handles large datasets more efficiently. | Limited by spreadsheet size. |
| **Automation & Refresh** | Scheduled refresh from data sources. | Manual or limited refresh option. |
|  |  |  |
|  |  |  |

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

**Tables and Relationships:**

* Power BI allows you to import data from various sources (Excel, SQL Server, etc.).
* Once data is imported, you can create relationships between tables—similar to a relational database.

**Data Types and Formatting:**

* Assigning correct data types (e.g., Date, Text, Number) ensures accurate calculations and filtering.
* You can also create calculated columns and measures using DAX**.**

**DAX (Data Analysis Expressions):**

* A formula language used to create custom calculations and aggregations.

**Model Optimization:**

Remove unnecessary columns.

Reduce cardinality (unique values).

Use measures instead of calculated columns where possible.

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

**1. Import Mode :-**

* Description: Data is imported and stored in Power BI’s in-memory engine.
* Performance: Fastest query performance due to in-memory processing.
* Use Case: Best for medium to large datasets that don’t need real-time updates.

**2. Direct Query Mode :-**

* Description: Power BI queries the data source in real time; data is not stored in Power BI.
* Performance: Slower than Import Mode due to live queries to the source.
* Use Case: When working with very large datasets or real-time data.

**3. Live Connection :-**

* Description: Connects directly to a live data model such as SQL Server Analysis Services, Azure Analysis Services, or Power BI datasets.
* Performance: Similar to Direct Query but works with pre-built models.
* Use Case: When using centralized semantic models managed outside Power BI Desktop.

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

**Steps to Handle Data Transformation in Power BI:**

**1. Open Power Query Editor:**

* In Power BI Desktop, click on **“Transform Data”** to open the Power Query Editor.

**2. Common Data Transformation Tasks:**

**🔹 Remove Unnecessary Columns**

**🔹 Rename Columns and Tables**

**🔹 Change Data Types**

**🔹 Split Columns**

**🔹 Merge or Append Queries**

* **Merge:** Join two tables based on a key.
* **Append:** Stack rows from multiple tables with similar columns.

**🔹 Remove Duplicates**

**🔹 Add Custom Columns**

**3. Applied Steps Panel:**

**4. Load Transformed Data:**

* After transformations are complete, click **“Close & Apply”** to load the clean data into Power BI’s data model.

**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 Analysis Services to define custom calculations and expressions. It’s similar in syntax to Excel formulas but is far more powerful and designed specifically for data modeling and analysis on large datasets.

**Important of DAX in Power BI:**

* Connects to Multiple Data Sources :- Excel, SQL Server, SharePoint, Salesforce, Google Analytics, Azure, and more.

* Easy-to-Use Visualizations :- Drag and drop interface for building interactive charts, graphs, maps, and custom visuals.

* Powerful Data Modeling and Transformation :-Power Query Editor allows users to clean, transform, and prepare data from multiple sources. Use of DAX enables complex calculations and KPIs.

* Build Advanced Time Intelligence :-Time intelligence in Power BI allows you to perform dynamic, date-based calculations — such as Year-to-Date ,Month-to-Date, Previous Year, Running Totals, and more.

* Data-Driven Decision Making :- Power BI helps users make informed business decisions by presenting data in a clear, visual format.

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

**1. Calculated Columns:**

* **Definition**: A calculated column is a new column added to a table using a DAX formula. It is calculated row by row.
* **Storage**: The values are stored in the data model, increasing file size.
* **Evaluation Context**: Uses row context — evaluated for each row in the table.
* **Use Case**: When you need to categorize data or perform row-wise calculations.

**2. Measures:**

* **Definition**: A measure is a calculation used in aggregations, computed **on the fly** based on the filter context.
* **Storage**: The result is not stored; it is computed dynamically, so it’s more efficient.
* **Evaluation Context**: Uses filter context — depends on filters and slicers in the report.
* **Use Case**: When you need to do aggregations like sums, averages, or more complex metrics.

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

A **relationship** connects columns from two tables, enabling you to analyze data across both tables.

**Steps to Handle Relationships in Power BI:**

**1. Auto-detect Relationships (Optional)**

* Power BI can **automatically detect** relationships based on column names and data.
* This happens when you load tables into the model.

**Note**: Always **review and validate** auto-detected relationships.

**2. Manually Create Relationships**

* Go to **Model View**.
* Click **"Manage Relationships"** → **"New"**.
* Select:
  + **Table A and column**
  + **Table B and column**
  + Choose **cardinality** and **cross filter direction**

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

A Power BI Gateway acts as a bridge between your on-premises data sources (like SQL Server, Excel files, etc. stored locally) and Power BI service in the cloud. It allows Power BI to securely access your internal data for scheduled refreshes and live queries.

**Key Purposes of a Power BI Gateway:**

**1. Secure Data Transfer**

* Allows Power BI Service to **safely connect** to data that’s not in the cloud (e.g., on your local servers or network).

**2. Scheduled Data Refresh**

* Enables **automatic refreshes** of your datasets by connecting to your on-prem data sources on a defined schedule.

**3. Live/DirectQuery Support**

* Supports **real-time dashboards** using DirectQuery or Live Connection to sources like SQL Server, SAP HANA, etc.

**4. Shared Use**

* One gateway can be used by **multiple users** and for **multiple data sources**—centralized and efficient.

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

**Schedule a Data Refresh in Power BI Service**

1. **Publish Your Report to Power BI Service:**
   * First, publish your Power BI .pbix file from Power BI Desktop to the Power BI Service.
2. **Go to the Dataset Settings:**
   * In the Power BI Service:
     + Navigate to Workspace → Click “Datasets + dataflows”
     + Find your dataset, click the More options (⋯) → Choose “Settings”
3. **Configure Data Source Credentials:**
   * Under Data source credentials, ensure credentials are set up correctly for all data sources
   * If using a gateway, it must be installed and configured correctly.
4. **Enable Scheduled Refresh:**
   * Scroll down to Scheduled refresh
   * Toggle “Keep your data up to date” to On
   * Set the refresh frequency:
     + Daily or Weekly
     + Choose time slots (up to 8 times per day for Pro, 48 for Premium)

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

**Row-Level Security (RLS) in Power BI** is a feature that allows you to control access to data at the row level, meaning different users can see different subsets of data based on defined filters.

**Concepts:**

1. **Purpose:**
   * To restrict data access for specific users based on their roles.
   * Ensures sensitive data is only accessible to authorized users.
2. **How it Works:**
   * You define roles and rules (DAX filters) within Power BI Desktop.
   * These filters are applied to tables to limit the rows that are visible to users assigned to those roles.
3. **Types of RLS:**
   * **Static RLS:** Rules are hardcoded and apply the same way to all users within a role.
   * **Dynamic RLS:** Uses user-specific data (e.g., username or email) to filter rows dynamically.
4. **Setting Up RLS:**
   * In **Power BI Desktop**:
     + Go to **Modeling > Manage Roles**.
     + Create roles and define filters.
   * In **Power BI Service**:
     + Publish the report.
     + Assign users to the roles under the dataset's **Security** settings.
5. **Benefits:**
   * Centralized control of data visibility.
   * Enhanced data security and compliance.
   * Personalized reporting experience for users.

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

**Power BI Desktop** and **Power BI Service** are two core components of the Power BI ecosystem, each serving different purposes in the business intelligence workflow.

**Power BI Desktop:**

Power BI Desktop is a free Windows application used primarily for creating and designing Power BI reports and data models.

**Key Features:**

* Import, clean, and transform data using Power Query.
* Build data models and relationships.
* Create visuals and dashboards.
* Write DAX for calculations and measures.
* Define Row-Level Security.

**Use Case:**  
Used by analysts and developers to prepare and design reports before sharing them.

**Power BI Service :**

Power BI Service is a cloud-based platform used for sharing, collaborating, and consuming Power BI reports and dashboards.

**Key Features:**

* Publish reports from Power BI Desktop.
* Create and share dashboards.
* Schedule data refreshes.
* Collaborate with others via workspaces..

**Use Case:**  
Used by business users and decision-makers to view and interact with reports online and share insights across the organization.

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

**Direct Query** in **Power BI** is a data connectivity mode that allows Power BI to connect directly to a data source without importing the data into Power BI Desktop or Power BI Service.

**Concepts of Direct Query:**

1. **Live Connection to Data Source**:
   * Power BI does not store the data.
   * Queries are sent to the data source in real-time every time a user interacts with a visual or report.
2. **No Data Import**:
   * Unlike the Import mode, data is not loaded into Power BI's in-memory engine.
   * Only metadata is stored.
3. **Real-Time Data**:
   * Users see the most up-to-date data from the source.
   * Useful for reports that require near real-time analytics.
4. **Performance Depends on Source**:
   * Query performance depends on the speed and performance of the underlying data source.
   * Complex transformations or large volumes may lead to slow performance.
5. **Security and Access**:
   * Power BI respects the security rules of the source system.
   * Row-level security (RLS) can be enforced through the data source.
6. **Limitations**:
   * Not all Power BI features are available
   * Some modeling features like calculated tables and some transformations may be restricted or slower.

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

Power BI templates are files with the .pbit extension that contain the structure of a Power BI report—including data model, queries, visuals, relationships, measures, and layout—but without the actual data (or with only minimal sample data).

**Power BI Templates:**

1. **File Type**: .pbit (Power BI Template)
2. **Includes**:
   * Report layout and visuals
   * Data model (tables, relationships, measures, DAX)
   * Power Query transformations
   * Parameters (optional)
3. **Excludes**:
   * Most of the actual data (unless specifically included as small samples)

**Use Cases:**

1. Reusability
2. Standardization
3. Parameterization
4. Efficiency in Deployment
5. Sharing and Collaboration

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

Incremental Data Refresh in Power BI is a feature that enables you to refresh only new or changed data, instead of reloading the entire dataset every time. This improves performance, reduces load on the data source, and speeds up refresh times—especially for large datasets.

**Steps to Set Up Incremental Refresh:**

**1. Enable Parameters:**

* Create two Date/Time parameters in Power BI Desktop:

**2. Filter Your Data:**

* In Power Query, apply a filter to a date column using parameters.

**3. Define Incremental Refresh Policy:**

* In Power BI Desktop, go to:
  + Modeling → Table → Incremental Refresh
* Set the policy:
  + Example: Store data for the last 5 years, refresh only the last 1month.
* You can also enable Detect data changes using a column like LastModifiedDate.

**4. Publish to Power BI Service:**

* Incremental refresh works only after publishing the dataset to Power BI Service.
* The first refresh loads the full data set.
* Future refreshes will load only new or changed records.

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

**Roles of Power Query in Power BI:**

**1. Data Connection (ETL - Extract):** Connects to a wide variety of data sources:

* + Databases (SQL Server, Oracle, MySQL)
  + Files (Excel, CSV, JSON)
  + Online services (SharePoint, Azure, web APIs)

**2. Data Transformation (Transform):** Clean and reshape data using a graphical interface (no coding required):

* + Remove columns
  + Rename fields
  + Filter rows
  + Merge/append tables
  + Pivot/unpivot data
  + Add calculated columns
* All transformations are recorded step-by-step as M language code.

**3. Data Loading (Load):** After transformation, data can be:

* + Loaded into the Power BI model (in-memory)
  + Used with Direct Query or composite models in some cases

**4. Automation and Reusability**

* Transformation steps are **automated** and can be refreshed regularly.
* Promotes consistency and repeatability in data preparation

**16) Explain the difference between calculated columns and**

**calculated tables in Power BI.**

**1. Calculated Columns**

**Definition:** A calculated column is a new column added to an existing table, created using a DAX expression.

**Use Case:** Used when you need to create new data based on row-by-row operations on existing columns.

**Key Points:**

* Stored in memory like other columns.
* Recalculated when data is refreshed.
* Useful for categorization, segmentation, or combining fields**.**

**2. Calculated Tables**

**Definition:** A calculated table is a new table created from existing tables or expressions using DAX.

**Use Case:** Used when you need to create summary tables**,** filter-based tables, or support for many-to-many relationships.

**Key Points:**

* Can be used to define relationships, build slicers, or perform advanced modeling.
* Also stored in memory and recalculated on refresh.
* Often used for intermediate logic or custom dimensions.

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

**1. Use Existing Custom Visuals from AppSource:**

**Steps:**

1. In Power BI Desktop, go to the Visualizations pane.
2. Click on the three dots (…) > Get more visuals.
3. Browse or search Microsoft AppSource for the visual you need.
4. Click Add to import it into your report.

**2. Create Custom Visuals with Developer Tools:**

**Steps:**

1. Set up the environment
2. Create a new visual project
3. Develop the visual
4. Test locally
5. Package the visual
6. Import into Power BI

**3. Use R or Python for Custom Visuals:**

If you're familiar with R or Python, you can use scripts to create advanced charts.

**Steps:**

* Enable R or Python scripting in Power BI settings.
* Add a Python Visual or R Visual from the visualizations pane.
* Write your plotting code using libraries like matplotlib, seaborn, or ggplot2.

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

**Power BI?**

**Data Modeling Best Practices**

1. Use Star Schema
2. Avoid Bi-Directional Relationships
3. Reduce Cardinality
4. Remove Unnecessary Columns and Tables
5. Use Numeric Keys Instead of Text Keys

**Data Loading & Transformations (Power Query)**

1. Perform Data Transformation at the Source
2. Disable Auto Date/Time for Time Intelligence
3. Use Staging Queries

**DAX and Calculations Optimization**

1. Use Measures Instead of Calculated Columns
2. Avoid Complex or Iterative DAX in Visuals
3. Minimize Use of IF and SWITCH in Visuals

**Visualizations and Reports**

1. Limit the Number of Visuals per Page
2. Avoid Using Too Many Slicers

**19) How can you integrate Power BI with other Microsoft**

**products like Azure and Office 365?**

**1. Power BI + Office 365 Integration:-**

**Excel**

* Import Excel Data
* Analyze in Excel
* Publish to Power BI

**Microsoft Teams**

* Embed Power BI reports into Teams channels and chats.
* Use the Power BI app for Teams to view dashboards, collaborate on data, and receive report notifications.
* @mention reports or dashboards in Teams messages.

**SharePoint Online**

* Embed Power BI reports in SharePoint Online pages using the Power BI web part.
* Enables secure, interactive reporting in SharePoint portals.

**Outlook**

* Power BI reports and dashboards can be shared via links in emails.
* Use Power BI Smart Narrative with Power Automate to generate email alerts with visuals.

**2. Power BI + Azure Integration:-**

**Azure SQL Database & Synapse Analytics**

* Connect directly to Azure SQL or Synapse for real-time or scheduled data refreshes.
* Use Direct Query or Import Mode based on performance needs.

**Azure Analysis Services / Power BI Datasets**

* Power BI can connect to Azure Analysis Services for semantic modeling.
* Power BI Premium now supports migrating models from AAS to Power BI datasets.

**Azure Machine Learning**

* Integrate ML models directly into Power BI for scoring and predictions.
* Use Azure ML Web Services within Power BI dataflows.

**Azure Active Directory (AAD)**

* Used for secure SSO and user authentication.
* Role-based access control using AAD security groups.

**3. Power BI + Power Platform:-**

**Power Automate**

* Automate alerts, emails, approvals, and workflows triggered by Power BI events (e.g., data threshold met).
* Use the Power BI connector in Power Automate flows.

**Power Apps**

* Embed Power Apps into Power BI reports for write-back capabilities.
* Embed Power BI tiles into Power Apps as visuals.

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

Aggregations in Power BI are a performance optimization technique used to speed up queries on large datasets by summarizing detailed data into higher-level pre-aggregated tables.

**Use of Aggregations**

When dealing with large datasets (millions or billions of rows), direct querying of the detail table for every report visual can slow down performance. Aggregations allow Power BI to retrieve pre-summarized data quickly, reducing load times and improving efficiency.

**Types of Aggregations**

1. **Imported Aggregations**
   * Stored in memory for fast performance.
   * Power BI uses these when users don’t need detailed data.
2. **Direct Query Aggregations**
   * Data stays in the source (e.g., SQL Server, Synapse).
   * Aggregation tables are still faster than querying detail data.

**How It Works (Aggregation Table Strategy)**

1. **Create an Aggregation Table**
   * Example: Instead of storing individual sales transactions, create a table with total sales per product per month.

1. **Define Relationships**
   * Ensure the aggregation table is related to your dimension tables (e.g., Date, Product).
2. **Map Aggregations in Model**
   * In Power BI Desktop, go to:  
     Model view > Manage Aggregations
   * Specify how the aggregation table matches the detailed table:
3. **Automatic Switching**
   * Power BI checks if a query can be answered using the aggregation table.
   * If yes, it uses the faster aggregated data.
   * If not, it falls back to the detailed table (usually via Direct Query).

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

**1. Error Handling in Power BI (Power Query Editor):**

a) Use try...otherwise

b) Remove Errors

c) Replace Errors

d) Use Conditional Logic

**2. Data Quality Checks in Power BI:**

a) Profile Data in Power Query

b) Check for Duplicates

c) Validate Data Types

d) Detect and Fix Missing Values

**3. Handling Data Issues at the Model Level:**

**Using DAX for Quality Flags**

* Create calculated columns or measures to flag potential data issues.
* Use visuals or conditional formatting to highlight bad data.

**4. Automation & Monitoring:**

**Use Power Automate**

* Trigger alerts when data quality thresholds are breached (e.g., too many nulls).

**Data Alerts**

* Set alerts on dashboard tiles (e.g., if error count > threshold).

**5. Data Source-Level Validation:**

* Perform heavy data cleansing at the data source or ETL level (SQL, Azure Data Factory, etc.) when possible.
* This reduces complexity inside Power BI.

**22) What is the purpose of Power BI Embedded and when**

**would you use it?**

**Purpose of Power BI Embedded:**

1. **Embed analytics into custom apps**: Enables businesses to integrate reports and dashboards into their own software products or internal apps.
2. **White-labeling**: You can present reports under your own brand, with no Power BI branding visible to end users.
3. **User access control**: Offers control over who sees what data often used with applications that have their own authentication systems.
4. **Scalability**: Designed for scenarios with many users, providing scalable licensing and usage-based pricing.
5. **No need for Power BI Pro licenses for each user**: The application uses a master (service) account with appropriate licensing, so end users don’t need individual Power BI licenses.

**When to Use Power BI Embedded:**

* You are a software developer or (Independent Software Vendor) who wants to integrate analytics into a SaaS application.
* You want to provide data insights to customers or external users who are not part of your organization.
* You need custom UI/UX beyond what standard Power BI dashboards offer.
* You want to offer a seamless user experience without redirecting users to the Power BI service.
* Your users don’t have or don’t need Power BI accounts, and you manage access on their behalf.