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

**→**Power Bi is a business analytics tool by Microsoft that provides interactive visualizations and business intelligence capabilities with an interface simple enough for and users to create their own reports and dashboard .

Excel excels at handling complex computations, large datasets, and offers more advanced statistical functions and offline desktop functionality for in-depth data analysis.

1. **Explain the concept of data modeling in Power BI.**

**→**Data modeling in Power BI is the process of structuring, relating, and defining data within the Power BI tool to create a logical, understandable, and efficient semantic model for analysis and reporting. It involves creating relationships between tables from various data sources, defining constraints, and implementing calculations using [DAX (Data Analysis Expressions)](https://www.google.com/search?sca_esv=7550878c098e0420&cs=0&sxsrf=AE3TifP0f6gevfM8DB5KTk_A_jnt5yjp6A%3A1756992196319&q=DAX+%28Data+Analysis+Expressions%29&sa=X&ved=2ahUKEwiXku79mb-PAxXJ3jgGHYjXHS0QxccNegQIBRAB&mstk=AUtExfDX1G34fwHasupQtZAulH7ZFZDL3gPIZELtD_ssqFByu-CJXaPrHYugepNffzwBH3wmy4ZdqkPrnMR2MouQBXXqEIuyVbx53BWZU2rfAOFKmlF24Y0hON4vXjnyymcyBwjkJ5VWznE4qMn66poEcgEs6uYWr0btv_uJCaVlq1QdlDjUxQLHgOx9BvLCDf7QTOkAMZjPkxhZzK3Vw52Ofiyx7YYKlsb2HjQ89cX9IZCJVF9lnUZ5ej-0z6eqeR00OJAYvIGEwzFiepQgOv4XLH7R&csui=3) to ensure data integrity and support powerful visualizations and insights. The goal is to transform raw data into a usable and insightful "data model" that can be easily analyzed to meet business objectives.

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

→**→** In Power BI, there are several types of data connections available that allow users to import and work with data from various sources. These connections fall into different categories based on how the data is accessed, processed, and stored. Here’s a breakdown of the main types:

**1. Import (Cached Data)**

* **How it works**: Data is imported and stored in Power BI's internal storage (VertiPaq engine).

**2. DirectQuery**

* **How it works**: Data remains in the source; queries are sent in real-time when users interact with reports.

**3. Live Connection**

* **How it works**: Similar to DirectQuery but used for multidimensional models like SSAS (SQL Server Analysis Services).

**4. Composite Models**

* **How it works**: Combines **Import**, **DirectQuery**, and/or **Live connections** in a single model.

**5. Dataflows**

* **How it works**: Power BI Dataflows allow you to create reusable data transformation logic in the Power BI Service (cloud-based).

**6. OData Feeds and Web Connectors**

* **How it works**: Connect to data from web services or APIs.

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

→ Data transformation in Power BI is done using **Power Query Editor**, where you clean, shape, and structure raw data before loading it into the model.

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

1. **Load Data into Power Query**.
2. **Apply Common Transformations**
3. **Combine Data from Multiple Sources**
4. **Handle Data Quality Issues**
5. **Create Custom Columns**
6. **Load Clean Data into Power BI Model**

**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, Power Pivot, and Analysis Services. It is designed for creating **custom calculations** on your data model, such as calculated columns, measures, and tables.

It looks similar to Excel formulas but is **more powerful** because it works with **relational data models** and supports **filter context** and **row context**.

### **Why is DAX Important in Power BI?**

1. **Custom Calculations**
2. **Dynamic Aggregations**
3. **Data Modeling**
4. **Works with Large Datasets**
5. **Business Logic Implementation**

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

### **→ Calculated Columns are** A calculation created using DAX that is evaluated on the fly.

**Measures** -A calculation created using DAX that is evaluated on the fly.

### **Key differences--**

| **Feature** | **Calculated Column** | **Measure** |
| --- | --- | --- |
| **Stored in Model** | Yes (increases model size) | No (calculated on demand) |
| **Context Used** | Row context | Filter context |
| **Performance** | Slower (more storage) | Faster, optimized |
| **Use Case** | Row-level calculations | Aggregations, KPIs |
| **Visibility in Filters** | Can be used in slicers/filters | Cannot be directly used as slicers |

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

## **→ How to Handle Relationships Between Tables in Power BI**

### **1. Understanding Relationships**

### **2. Types of Relationships**

### **3. Cardinality in Power BI**

### **4. Steps to Handle Relationships**

### **5. Best Practices**

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

**→** A **Power BI Gateway** is a **bridge** that connects on-premises data sources (databases, files, servers) to Power BI cloud services (PowerBI.com).

It ensures that reports and dashboards in Power BI Service can stay **up to date** with data stored in your local/on-premises environment, without manually uploading files every time.

### **Key Purposes / Uses**

1. **Secure Data Transfer**
2. **Scheduled Refresh**
3. **Live / Direct Query Connections**
4. **Hybrid Connectivity**
5. **Shared Across Services**

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

## **→ 1. Publish Your Report**

* First, build your report in **Power BI Desktop**.
* Publish it to **Power BI Service** (app.powerbi.com).

### **2. Configure Gateway (if needed)**

* If your data source is **on-premises** (e.g., SQL Server, local Excel file), install and configure a **Power BI Gateway**.
* If your data is cloud-based (e.g., Azure SQL, SharePoint Online), no gateway is needed.

### **3. Go to Dataset Settings**

1. In Power BI Service → Navigate to **Workspaces**.
2. Find your **Dataset** (not report).
3. Click on **More options (…) → Settings**.

### **4. Schedule Refresh**

* Under **Dataset settings → Scheduled refresh**:
  + Turn **Keep data up to date** → ON.
  + Set **refresh frequency** → Daily / Weekly.
  + Set **time slots** (up to 8 times/day for Pro, 48 times/day for Premium).
  + Enter **data source credentials** if required

### **5. Advanced Options**

* Enable **Failure Notifications** (get email alerts if refresh fails).
* Control **time zone** for refresh schedules.
* Use **API or Power Automate** for more customized refresh triggers.

### **6. Types of Refresh in Power BI**

1. **Dataset Refresh** → Reloads data from the original sources.
2. **Model/Tile Refresh** → Updates visuals, tiles, and dashboards.
3. **DirectQuery/Live Connection Refresh** → No need for scheduling; queries always pull fresh data.

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

**→ Row-Level Security (RLS)** is a way to **restrict access to specific rows of data** for different users in Power BI.  
It ensures that each user only sees the data they are authorized to see, even though the dataset is the same.

**How RLS Works--**

-You define **roles** with **DAX filters** on tables.

-When a user views the report, Power BI applies those filters dynamically, showing -only the allowed rows.

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

**→ Power Bi Desktop** a free **Windows application** used to connect to data, transform it, build data models, create reports, and design dashboards.

**Power Bi Service** is A **cloud-based SaaS (Software as a Service)** platform hosted on **app.powerbi.com**.

**Key Differences-----**

| **Feature** | **Power BI Desktop** | **Power BI Service** |
| --- | --- | --- |
| **Type** | Windows application (free) | Cloud platform (SaaS) |
| **Primary Use** | Data prep, modeling, report design | Sharing, collaboration, and consumption |
| **Data Sources** | Can connect to many sources directly | Limited direct connections (often needs Gateway) |
| **Saving** | Saves reports as .pbix files locally | Stores reports and dashboards online |
| **Data Refresh** | Manual refresh in Desktop | Scheduled/automatic refresh in Service |
| **Sharing** | Cannot share directly | Can share with others via workspaces & apps |
| **Security** | No RLS enforcement | RLS applied and enforced |
| **Accessibility** | Local machine only | Accessible from anywhere (web/mobile) |

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

**→ Direct Query** is a data connectivity mode in Power BI where the report does **not import data** into Power BI.  
Instead, every time you interact with a report (apply filters, slicers, visuals), Power BI sends a **live query** to the underlying data source and retrieves results in real time. **Direct Query** in Power BI keeps your reports connected directly to the data source without importing, ensuring **real-time insights** but with some performance and feature limitations.

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

**→** A **Power BI Template (.pbit file)** is a reusable blueprint of a Power BI report.  
It contains **Data model structure** (tables, relationships, measures, calculated columns), **Queries & transformations** (Power Query steps), **Report layouts & visuals**, **Parameters** (for dynamic inputs).

**How Templates Are Useful---**

1. **Reusability**
2. **Consistency**.
3. **Collaboration**
4. **Efficiency**
5. **Customization with Parameters**

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

→ Incremental refresh means **loading only new or changed data** instead of reloading the entire dataset every time.  
It helps in:

* Reducing refresh time
* Improving performance
* Handling large datasets efficiently

**Steps to handle incremental data refresh---**

**1. Prepare Your Data (Define Parameters)**

### **2. Enable Incremental Refresh Policy**

### **3. Publish to Power BI Service**

### **4. Use Premium Features (Optional)**

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

**→ Power Query** is the **data transformation and preparation engine** inside Power BI.  
It is used to **connect, clean, reshape, and combine data** before loading it into the Power BI model.

**Roles of Power Query---**

 **Data Connectivity**

 **Data Transformation (ETL)**

 **Data Cleaning**

 **Combining Data**

 **Parameterization & Reusability**

 **M Language (Advanced)**

 **Prepares Data for Modeling**

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

**→ Calculated columns are** a column you create inside an existing table using a **DAX formula**.

**Calculated tables are a** new **entire table** created using a **DAX expression**.

**Key differences----**

| **Feature** | **Calculated Column** | **Calculated Table** |
| --- | --- | --- |
| **What it Creates** | New column in an existing table | Entire new table |
| **Scope** | Row-level calculations | Table-level aggregations/filters |
| **Storage Impact** | Increases model size | Increases model size (even more) |
| **Use Case** | Derive new fields (e.g., Profit) | Create summarized/filtered tables, bridge tables |
| **Based On** | Row context of the existing table | DAX queries on one/multiple tables |

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

**→** There are **two main approaches**:

**1. Use Pre-Built Custom Visuals from AppSource**

* **Steps**:
  1. In Power BI Desktop → Go to **Visualizations pane**.
  2. Click on **Get more visuals (•••)** → **Get more visuals from AppSource**.
  3. Browse the marketplace and choose from hundreds of custom visuals (e.g., Sankey, Bullet Chart, Heatmaps).
  4. Click **Add** → It appears in your Visualizations pane.
  5. Drag and use it like any other built-in visual.

### **2. Build Your Own Custom Visuals (Advanced)**

* Used when pre-built visuals don’t meet requirements.
* Requires **TypeScript, Node.js, and Power BI Visuals SDK**.

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

**→ Best Practices for Optimizing Power BI Performance---**

**1. Data Modeling Best Practices**

**2. Data Size Optimization**

**3. DAX & Calculations**

**4. Power Query (ETL)**

**5. Visualization Optimization**

**6. Storage & Connection Modes**

**7. Refresh & Gateway**

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

# **→ Integration of Power BI with Azure and Office 365**

**1.Power bi =+ Azure Integration**

Power BI tightly integrates with many **Azure services** for data, analytics, and AI.

### **\*Azure SQL Database / Data Warehouse**

### **\*Azure Data Lake Storage (ADLS)**

### **\*Azure Data Factory (ADF)**

### **\*Azure Analysis Services (AAS)**

### \* **Azure Machine Learning**

### \* **Azure Stream Analytics**

## **2. Power BI + Office 365 Integration**

Power BI works seamlessly with **Office 365 tools** for collaboration and productivity.

\***Microsoft Teams**

**\*SharePoint Online**

**\*Excel**.

**\*Outlook / PowerPoint**

**\*Power Automate (part of O365)**

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

**→** In Power BI, aggregations refer to the process of summarizing or grouping data to provide meaningful insights, such as totals, averages, counts, or other calculations across a dataset. Aggregations are essential for transforming raw data into a format that’s easier to analyze, visualize, and interpret. They are commonly used in visuals like charts, tables, and matrices to display summarized results rather than individual records.

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

## → **1. In Power Query (Data Preparation Stage)**

Power Query is the **first line of defense** for cleaning and handling errors.

### **Error Handling**

* **Remove Errors**: Right-click → Remove Errors (use when invalid rows can be discarded).
* **Replace Errors**: Replace error values with a default (e.g., 0 or "Unknown").
* **Try-Catch with M functions**:

### **Data Quality Checks**

* **Column Profiling, Quality, Distribution**:
  + In Power Query → View → Data Preview.
  + Shows % of valid, error, empty values.
* **Change Data Types**: Enforce correct types (e.g., Date, Decimal).
* **Remove Duplicates**: Keep only unique rows if needed.
* **Trim & Clean Text**: Remove extra spaces, non-printable characters.
* **Null Handling**: Replace or filter out nulls based on context.

## **2. In Data Modeling (DAX & Relationships)**

* Use **Calculated Columns / Measures** to handle missing or unexpected values.
* SafeSales = IF(ISBLANK(Sales[Amount]), 0, Sales[Amount])
* Use **COALESCE()** function in DAX to replace blanks with a default.
* Validate **relationships** to avoid incorrect joins or orphan records.

## **3. At Visualization Layer**

* Add **data validation visuals** (like row counts, missing value counts) to monitor quality.
* Use **conditional formatting** to highlight outliers or invalid data.
* Provide **error messages / fallback values** in visuals.

## **4. Source-Level Practices**

* Apply **constraints and validation** at the database level (e.g., primary keys, NOT NULL).
* Use **stored procedures or views** to pre-clean data before Power BI loads it.
* Ensure proper **indexes** and data integrity rules.

## **5. Ongoing Monitoring**

* **Dataflows**: Reuse centralized, cleaned data for multiple reports.
* **Scheduled refresh alerts**: Get notified if refresh fails due to bad data.
* **Power Automate Integration**: Trigger alerts/workflows if data quality issues occur.

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

→ Power BI Embedded is an **Azure service (PaaS)** that allows developers to **embed Power BI reports, dashboards, and visuals into custom applications, portals, or websites**.End users can interact with Power BI reports **without having a Power BI license or going to the Power BI Service**.

## **Purpose of Power BI Embedded**

* Provide **analytics inside applications** without forcing users to open Power BI separately.
* Give **customers, partners, or external users** access to interactive reports securely.
* Allow **white-label analytics** → reports look like part of your app (custom branding, no Power BI logos).
* Reduce cost → instead of giving every external user a Power BI Pro license, you pay for capacity (Azure).

## **When to Use Power BI Embedded**

1. **Independent Software Vendors (ISVs)**
   * Build apps for clients (e.g., healthcare, finance, retail) and embed analytics.
   * Example: A hospital app with embedded patient dashboards.
2. **Customer Portals**
   * Companies provide **interactive reports** to their customers through portals.
   * Example: A logistics company shows delivery KPIs to each client.
3. **Internal Apps with External Users**
   * When employees + vendors/partners need access but you don’t want to manage Pro licenses.
4. **White-Labeled BI**
   * Organizations want their own **branding** on dashboards (colors, logo) inside their apps.