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

Ans = Power BI is a business analytics service by Microsoft that provides interactive visualizations and business intelligence capabilities with an interface simple enough for end users to create their own reports and dashboards. Think of it as a powerful tool designed specifically for analyzing and visualizing large and complex datasets to gain business insights.Excel, on the other hand, is primarily a spreadsheet software used for organizing, calculating, and analyzing smaller sets of data. While Excel has charting capabilities, Power BI excels in:

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

Ans = Data modeling in Power BI is the process of structuring and shaping your data to optimize it for analysis and visualization. It involves creating relationships between different tables of data, defining how they connect and interact. A well-designed data model ensures data accuracy, consistency, and efficient querying, ultimately leading to more meaningful insights. Key aspects of data modeling in Power BI include:

**Identifying tables and their relationships:** Determining the different entities (e.g., customers, products, sales) in your data and how they relate to each other (e.g., one customer can have multiple sales orders).

**Creating relationships:** Defining the type of relationship between tables (e.g., one-to-many, one-to-one, many-to-many) and the columns used to link them.

**Creating calculated columns:** Adding new columns to your tables based on existing data using DAX formulas. These can be used for calculations or segmentations.

**Creating measures:** Defining aggregations (e.g., sum, average, count) that are calculated dynamically based on the context of your analysis. Measures are the heart of analytical insights in Power BI.

**Optimizing the model:** Structuring the data in a way that is efficient for querying and reporting, often involving techniques like creating dimension tables and fact table.

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

Ans = Power BI offers a wide variety of data connectors to bring data from various sources. These connections can be broadly categorized into: files Connecting to local files like Excel (.xlsx, .xlsb), CSV files (.csv), XML files (.xml), text files (.txt), and JSON files workbooks (.json).Database : Connecting to various database systems, including:

Microsoft databases: SQL Server, Access database, Analysis Services.

Other databases: Oracle Database, MySQL, PostgreSQL, IBM Db2, Teradata, and many more.

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

1. Ans = Data transformation in Power BI is primarily done using **Power Query Editor**. This powerful tool allows you to clean, shape, and transform your data before loading it into the Power BI data model. Common data transformation steps include:
   * **Connecting to data sources:** Importing data into the Power Query Editor.
   * **Promoting headers:** Making the first row of your data the column headers.
   * **Changing data types:** Ensuring that columns have the correct data type (e.g., text, number, date).
   * **Filtering rows:** Removing unwanted rows based on specific criteria.
   * **Removing or keeping columns:** Selecting only the necessary columns.
   * **Sorting data:** Ordering rows based on column values.
   * **Splitting columns:** Dividing a single column into multiple columns based on a delimiter.
   * **Merging columns:** Combining multiple columns into a single column.
   * **Adding custom columns:** Creating new columns based on formulas using the Power Query M language.
   * **Replacing values:** Finding and replacing specific values within columns.
   * **Appending queries:** Combining rows from multiple tables with the same structure.
   * **Merging queries:** Combining columns from two or more tables based on related columns (similar to SQL JOIN).
   * **Unpivoting columns:** Transforming multiple columns into rows.
   * **Pivoting columns:** Transforming unique values in a column into new columns.

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

Ans = DAX is a formula language used in Power BI to create calculated columns, measures, and calculated tables. It's a powerful language designed to work with relational data and perform calculations based on the context of your analysis.DAX is crucial in Power BI for several reasons:

* + **Creating custom calculations:** DAX allows you to perform calculations that are not directly available in the raw data, such as calculating sales growth, profit margins, running totals, and more complex business metrics.
  + **Enriching the data model:** Calculated columns add new attributes to your tables, allowing for more granular filtering and grouping. Measures provide dynamic aggregations that respond to the context of your visualizations.
  + **Enabling powerful analysis:** DAX formulas can leverage the relationships in your data model to perform calculations across related tables, providing deeper insights.
  + **Creating dynamic visualizations:** Measures calculated with DAX drive the values displayed in your visuals, making them interactive and responsive to user selections and filters.
  + **Implementing business logic:** DAX allows you to embed complex business rules and logic directly into your Power BI model.

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

Ans = Calculated columns and measures are both created using DAX in Power BI, but they serve different purposes and are evaluated differently.

* + **Calculated Columns:**
    - **Row context:** Calculated columns are evaluated row by row within a table. The DAX formula operates on the values in the current row.
    - **Stored in the data model:** The results of calculated columns are stored in the data model, increasing its size.
    - **Used for filtering, grouping, and slicing:** Because they are part of the table, calculated columns can be used in slicers, filters, row/column headers in visuals, and for grouping data.
    - **Example:** A calculated column to determine if a sale amount is "High," "Medium," or "Low" based on a threshold.
  + **Measures:**
    - **Filter context:** Measures are evaluated based on the filters and context of the visual they are placed in. They perform aggregations (like SUM, AVERAGE, COUNT) over a set of rows.
    - **Calculated on the fly:** The results of measures are not stored in the data model but are calculated dynamically when a visual is rendered.
    - **Used for aggregations and analysis:** Measures are primarily used to display aggregated values in cards, tables, charts, and other visuals.
    - **Example:** A measure to calculate the "Total Sales" by summing the sales amount for the filtered data.

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

Ans = Relationships between tables in Power BI are crucial for combining and analyzing data from multiple sources. You manage these relationships in the **Model view**. Here's how you handle them.

* + **Automatic Relationship Detection:** Power BI can automatically detect relationships based on matching column names and data types. However, it's always good practice to review and manage these relationships manually.
  + **Creating Relationships Manually:** You can create relationships by dragging a field from one table to a matching field in another table.
  + **Configuring Relationships:** When creating or editing a relationship, you need to define.

8. What is the purpose of a Power BI Gateway?

Ans = A Power BI Gateway acts as a bridge between the Power BI cloud service and on-premises data sources (data sources that are behind a firewall or not directly accessible from the internet). Its primary purpose is to enable Power BI to securely access and refresh data from these on-premises sources without moving the data to the cloud.Key purposes of a Power BI Gateway include.

* + **Secure data access:** It provides a secure channel for Power BI to connect to on-premises data sources like local SQL Server databases, SharePoint on-premises, file shares, etc., without exposing these systems directly to the internet.
  + **Scheduled data refresh:** It allows you to set up scheduled refreshes of your Power BI datasets that rely on on-premises data, ensuring your reports and dashboards are always up-to-date with the latest information.
  + **Live connections:** For some data sources like SQL Server Analysis Services, the gateway enables live connections, meaning Power BI queries the on-premises data source in real-time.

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

Ans = Scheduling data refresh in Power BI Service ensures that your reports and dashboards always display the latest data from your underlying data sources. Here's how you do it:

* 1. **Publish your Power BI Desktop report (.pbix file) to the Power BI Service.**
  2. **Ensure your data sources are supported for scheduled refresh.** Cloud-based data sources generally don't require a gateway. For on-premises data sources, you'll need to have a Power BI Gateway configured and the data source registered with the gateway.
  3. **Navigate to your dataset in the Power BI Service.** You can find your datasets in the "Datasets" section of your workspace.
  4. **Open the settings for your dataset.** Hover over the dataset and click the "..." (More options) menu, then select "Settings."
  5. **Expand the "Scheduled refresh" section.**
  6. **Configure the refresh schedule:**
     + **Turn scheduled refresh on.**
     + **Refresh frequency:** Choose how often you want the data to refresh (e.g., daily, weekly). Depending on your Power BI license and the data source, you might have options for more frequent refreshes.
     + **Time slots:** Specify the time(s) of day when you want the refresh to occur. You can add multiple time slots.
     + **Time zone:** Select the appropriate time zone.
     + **Keep data up to date even if all refresh attempts fail:** This option allows you to receive email notifications if refreshes fail.
  7. **Configure gateway connection (if applicable):** If your dataset connects to on-premises data sources, you'll need to ensure the gateway connection is configured correctly under the "Gateway connection" section. You'll need to map your data sources in Power BI Service to the data sources registered in your gateway.
  8. **Apply your settings.** Click "Apply" to save your refresh schedule.

Power BI will then automatically attempt to refresh your dataset according to the schedule you've defined. You can also perform an on-demand refresh from the dataset settings.

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

Ans = Row-level security (RLS) in Power BI allows you to restrict data access for specific users within an organization. It enables you to ensure that users only see the data that is relevant to them, based on their roles or permissions. This is crucial for data privacy, compliance, and ensuring that users are not overwhelmed with irrelevant information.Here's how RLS works in Power BI:

* + **Roles:** You define roles within Power BI Desktop and associate them with filters. These filters determine which rows of data users assigned to that role can see. For example, you might create a "Sales Manager" role that can only see sales data for their specific region.
  + **Filters:** Filters are created using DAX expressions that evaluate to TRUE or FALSE. These filters are applied to specific tables in your data model.
  + **Users and Groups:** Once the roles are defined and the report is published to Power BI Service, you assign users or security groups to these roles.
  + **Enforcement in Power BI Service:** The RLS rules are enforced in the Power BI Service. When a user assigned to a specific role views a report, they will only see the data that passes the filters defined for that role.
  + **Types of RLS:**
    - **Static RLS:** The filters are based on static values or relationships within the data model.
    - **Dynamic RLS:** The filters are based on the username or other attributes of the user viewing the report, often leveraging a separate table that maps users to the data they are allowed to see.

RLS is a powerful feature for creating secure and personalized business intelligence solutions.

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

* + Ans = **Connecting to data sources:** Importing and connecting to various data sources.
  + **Transforming and shaping data:** Using Power Query Editor to clean and prepare data.
  + **Creating the data model:** Defining relationships between tables, creating calculated columns and measures using DAX.
  + **Designing reports and visualizations:** Building interactive charts, graphs, and dashboards.

Think of Power BI Desktop as the development environment where you build your Power BI solutions.**Power BI Service** is a cloud-based platform (SaaS - Software as a Service) that you access through a web browser. It's used for:

* + **Publishing reports and datasets:** Sharing the reports and underlying data models created in Power BI Desktop.
  + **Viewing and interacting with reports and dashboards:** End users consume the reports and dashboards in the service.
  + **Collaboration and sharing:** Sharing reports and dashboards with other users and teams.
  + **Scheduling data refresh:** Automating the process of updating the data in your datasets.
  + **Implementing row-level security:** Managing data access for different users.
  + **Creating and managing dashboards:** Pinning visuals from reports to create summarized views of key metrics.
  + **Administering the Power BI environment:** Managing users, workspaces, and settings.

In essence, **Power BI Desktop is for building and designing, while Power BI Service is for sharing, collaborating, and consuming the insights.** You typically use Power BI Desktop to create your reports and then publish them to the Power BI Service for others to access and interact with.

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

Ans = The query you make in the Power BI interface is **sent directly to the underlying data source**, like a relational database or an Analysis Services model, when you use DirectQuery. Instead of using Power BI’s own engine (SQL Profiler), the data source itself retrieves and processes the data.

13. What are Power BI templates and how are they useful?

Ans = Power BI templates are **pre-built files that contain a set of predefined configurations, data models, queries, and visualizations**. These templates can be used to streamline the process of creating reports and dashboards in Power BI Desktop based on existing templates.

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

Ans = Incremental refresh in Power BI allows you to refresh only the data that has changed, rather than a full refresh of your entire dataset. This is particularly useful for large datasets where a full refresh can be time-consuming and resource-intensive. Here's how it's typically handled:

* **Parameter Definition:** You define two parameters in Power Query: RangeStart and RangeEnd. These parameters filter the data source to include only the relevant period for the incremental refresh.
* **Filtering in Power Query:** You use these parameters in your Power Query M code to filter the data being imported. For example, you might filter a date column in your source table to be within the RangeStart and RangeEnd.
* **Incremental Refresh Policy:** In the Power BI Desktop, you configure the incremental refresh policy for the table. This involves specifying:
  + The date column to be used for filtering.
  + The period to refresh (e.g., refresh data for the last 30 days).
  + The period to store in the archive (e.g., keep data for the last 3 years).
* **First Refresh (Full Refresh):** The first time you refresh the dataset after configuring incremental refresh, a full refresh occurs to load the historical data.
* **Subsequent Refreshes (Incremental):** Subsequent scheduled or manual refreshes will only retrieve the data within the defined refresh period (based on RangeStart and RangeEnd), making the refresh process much faster.
* **Power BI Service Handling:** The Power BI service manages the RangeStart and RangeEnd values automatically during scheduled refreshes, ensuring that only the necessary data is processed.

15. What is the role of Power Query in Power BI?

Ans = Power Query is the data transformation and preparation engine within Power BI. Think of it as the backstage area where you clean, shape, and combine your data before it's loaded into the Power BI model for visualization and analysis. Its key roles include:

* **Data Connection:** Power Query allows you to connect to a wide variety of data sources, including databases, files (Excel, CSV, etc.), web services, and cloud platforms.
* **Data Transformation:** This is its core function. Power Query provides a user-friendly interface (the Power Query Editor) with a rich set of tools to:
  + **Filter rows:** Select only the relevant data.
  + **Remove or keep columns:** Focus on the necessary fields.
  + **Rename columns:** Make them more descriptive.
  + **Change data types:** Ensure data is in the correct format (e.g., text to number, date to date).
  + **Split columns:** Separate data into multiple columns.
  + **Merge columns:** Combine data from different columns.
  + **Add custom columns:** Create new columns based on existing data using formulas (M language).
  + **Unpivot/Pivot data:** Reshape data for better analysis and visualization.
  + **Handle errors:** Clean or replace erroneous data.
* **Data Combination:** Power Query enables you to combine data from multiple sources or tables using operations like:
  + **Merging queries:** Joining tables based on common columns (similar to SQL JOIN).
  + **Appending queries:** Stacking tables with similar structures together (similar to SQL UNION).
* **Data Profiling:** Power Query provides insights into the quality and distribution of your data, helping you identify potential issues.
* **Reusable Transformations:** The steps you apply in Power Query are recorded as a series of M language statements. This allows you to easily refresh your data with the same transformations applied automatically.

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

Ans = Both calculated columns and calculated tables use DAX (Data Analysis Expressions) to create new data within your Power BI model, but they operate at different levels:

* **Calculated Columns:**
  + **Row Context:** Calculated columns are evaluated row by row within an existing table. The DAX expression you write has access to the values of other columns in the *same row*.
  + **Purpose:** They are primarily used to perform calculations or create new attributes based on the data already present in a table. Examples include calculating a total sales amount (quantity \* price), extracting the year from a date column, or categorizing products based on their properties.
  + **Storage:** Calculated columns add data to the existing table, increasing its size in memory.
* **Calculated Tables:**
  + **Table Context:** Calculated tables create an entirely *new table* based on DAX expressions. These expressions can reference other tables and can perform aggregations, filtering, and combinations to produce a new set of rows and columns.
  + **Purpose:** They are used to create summarized tables, derived tables, or tables with a specific structure needed for analysis or relationships. Examples include creating a calendar table, a table of top-selling products, or a table summarizing sales by region.
  + **Storage:** Calculated tables are stored as separate tables in the Power BI model, consuming memory.

17. How do you create custom visuals in Power BI?

Ans= Power BI offers a rich set of built-in visuals, but sometimes you might need a specific visualization that isn't available. You can create custom visuals in Power BI using the following methods:

* **AppSource:** The easiest way to extend Power BI's visualization capabilities is by downloading pre-built custom visuals from the Power BI Visuals marketplace (AppSource) directly within Power BI Desktop or the Power BI service. These visuals are created by Microsoft and the community and offer a wide range of specialized charts and graphs.
* **Power BI Visuals SDK:** For more advanced and highly tailored visuals, you can use the Power BI Visuals SDK (Software Development Kit). This allows developers to create their own custom visuals using web technologies like TypeScript, JavaScript, HTML, and CSS. The general process involves.
* **R and Python Visuals:** If you are familiar with R or Python, Power BI allows you to create visuals using scripts in these languages. Power BI integrates with R and Python environments, and you can use popular visualization libraries like ggplot2 (for R) and Matplotlib or Seaborn (for Python) to generate charts within your reports. However, these visuals have some limitations and require the corresponding runtime environments to be installed.

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

* Ans = **Data Modeling:**
  + **Star Schema:** Design your data model using a star schema (or snowflake schema where appropriate) with fact tables and dimension tables. This generally leads to better query performance.
  + **Relationships:** Create efficient relationships between tables. Ensure the cardinality and cross-filter direction are set correctly.
  + **Avoid Bi-directional Filtering:** Use bi-directional filtering only when absolutely necessary, as it can impact performance and lead to ambiguity.
  + **Calculated Columns vs. Measures:** Use calculated columns sparingly, as they are computed during data refresh and increase model size. Prefer measures for aggregations and calculations that change based on context.
  + **Data Types:** Ensure columns have the most appropriate data types to minimize memory usage.
  + **Remove Unnecessary Columns:** Only import the columns you need for your analysis and visualizations.
* **DAX Optimization:**
  + **Efficient Measures:** Write efficient DAX measures. Avoid iterating through entire tables unnecessarily (e.g., using SUMX or AVERAGEX when a simple SUM or AVERAGE on a column would suffice).
  + **CALCULATE Usage:** Understand how CALCULATE works and use it effectively for context manipulation.
  + **Variables:** Use variables within your DAX expressions to improve readability and potentially performance by storing intermediate results.
  + **Avoid Complex Iterators:** Minimize the use of complex iterator functions like FILTER within iterators if possible.
  + **Performance Analyzer:** Use Power BI Desktop's Performance Analyzer to identify slow-performing visuals and DAX queries.
* **Power Query Optimization:**
  + **Filter Early:** Apply filters as early as possible in your Power Query steps to reduce the amount of data loaded.
  + **Remove Unnecessary Steps:** Eliminate any Power Query steps that are not required.
  + **Native Queries:** When connecting to databases, consider using native queries (SQL) if you have complex filtering or transformations that can be performed more efficiently at the source. Be mindful of query folding.
  + **Query Folding:** Ensure that as many Power Query steps as possible are "folded" back to the data source. This means the data source performs the transformations, which is usually much faster than Power BI's engine.
  + **Data Reduction Techniques:** If dealing with very large datasets, consider techniques like sampling or aggregation in Power Query to reduce the data volume.
* **Visualizations:**
  + **Limit Visuals on a Page:** Avoid having too many visuals on a single report page, as each visual generates queries.
  + **Efficient Visual Types:** Choose visual types that are appropriate for your data and analysis. Some complex visuals can be more resource-intensive.
  + **Minimize Interactions:** Excessive use of interactive features like drill-throughs and tooltips can impact performance.
  + **Optimize Images:** Optimize the size and resolution of any images used in your reports.
* **Power BI Service:**
  + **Dedicated Capacity:** For large-scale deployments and performance-critical reports, consider using Power BI Premium capacity, which provides dedicated resources.
  + **Scheduled Refresh:** Schedule refreshes during off-peak hours to minimize impact on users.
  + **Gateway Optimization:** Ensure your data gateway is properly configured and has sufficient resources if you are connecting to on-premises data sources.

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

Ans = Power BI seamlessly integrates with various Microsoft products, enhancing its capabilities and providing a unified data and analytics experience. Here's how:

* **Azure:**
  + **Azure Data Services:** Power BI can connect to a wide range of Azure data services, including Azure SQL Database, Azure Synapse Analytics, Azure Data Lake Storage, Azure Cosmos DB, Azure Analysis Services, and more. This allows you to build reports and dashboards on data stored and processed in Azure.
  + **Azure Active Directory (Azure AD):** Power BI leverages Azure AD for user authentication and authorization, ensuring secure access to reports and dashboards.
  + **Power BI Embedded:** Azure provides Power BI Embedded, an Azure service that allows you to embed Power BI reports and dashboards into your own applications without requiring users to have a Power BI license.
  + **Azure Functions and Logic Apps:** You can use Azure Functions or Logic Apps to automate data refresh processes or trigger actions based on Power BI alerts.
  + **Azure Machine Learning:** Integrate insights from Azure Machine Learning models into Power BI reports to visualize predictions and analytical results.
  + **Azure DevOps:** Use Azure DevOps for version control and deployment of Power BI reports and datasets.
* **Office 365:**
  + **Excel:** Power BI Desktop can import data from Excel workbooks. You can also publish Power BI reports to Power BI from within Excel. "Analyze in Excel" allows users to connect to Power BI datasets from Excel for further analysis.
  + **SharePoint:** Power BI reports can be embedded in SharePoint Online pages, making them easily accessible to team members. You can also store and manage Power BI files in SharePoint.
  + **Teams:** Power BI reports and dashboards can be easily shared and viewed directly within Microsoft Teams channels and chats, fostering collaboration around data. You can also embed entire Power BI apps as tabs in Teams.
  + **Power Automate (formerly Microsoft Flow):** Integrate Power BI with Power Automate to create automated workflows triggered by Power BI data alerts or to perform actions based on insights from your reports (e.g., send an email when sales drop below a certain threshold).
  + **OneDrive for Business:** Power BI Desktop can connect to data stored in OneDrive for Business.

20 . Explain the concept of aggregations in Power BI.

Ans = Aggregations in Power BI are a performance optimization technique used to speed up queries on large datasets. They work by pre-calculating and storing summarized data at a higher level of granularity than the detailed data. When a user queries the data at this higher level, Power BI can retrieve the pre-calculated results instead of scanning the entire detailed dataset.

Here's a breakdown of the concept:

* **Levels of Detail:** Imagine you have a sales table with millions of rows, each representing a single transaction with details like date, product, customer, and sales amount. This is the detailed level. Aggregations allow you to create summarized versions of this data, for example:
  + Total sales by month and product category.
  + Average sales by region and year.
  + Count of customers by country.
* **Aggregation Tables:** You create new, smaller tables in your Power BI model that contain these pre-calculated aggregations. These tables have fewer rows and often fewer columns than the detailed tables.
* **Storage Modes:** Aggregation tables can be stored in Import mode (like regular tables), but they often benefit from being in Dual storage mode. Dual mode allows Power BI to use the in-memory cache for aggregations when possible and fall back to DirectQuery for detailed data if needed.
* **Automatic Query Redirection:** When a user queries a visual that can be satisfied by the aggregated data, Power BI's storage engine intelligently detects this and automatically redirects the query to the appropriate aggregation table, resulting in much faster query execution. If the query requires more detail than what's available in the aggregations, Power BI will retrieve the necessary data from the detailed tables.
* **User Experience:** Users typically don't need to know that aggregations are in place. The process is transparent, and they interact with the same fields and visuals, but the underlying query performance is significantly improved.

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

Ans = Ensuring data quality and handling errors gracefully are essential for building reliable and trustworthy Power BI reports. Here's how you can approach this in Power BI:

* **Power Query for Data Quality and Error Handling:**
  + **Identify and Filter Errors:** In the Power Query Editor, you can identify rows with errors (they will be marked). You can then filter these rows out for investigation or removal.
  + **Replace Errors:** Power Query allows you to replace error values with a specific value (e.g., 0, null, or a custom text). You can do this for individual columns or across the entire dataset.
  + **Try...Otherwise:** The M language in Power Query provides try...otherwise blocks to handle potential errors during data transformations. This allows you to define alternative actions if a step fails, preventing the entire query from breaking.
  + **Data Profiling:** Use Power Query's data profiling features (column quality, column distribution, column profile) to identify inconsistencies, missing values, and other data quality issues.
  + **Data Type Conversion:** Ensure data types are correctly applied to avoid errors during calculations and comparisons. Power Query can help you identify and fix data type mismatches.
  + **Conditional Logic:** Use conditional columns to categorize or flag data based on certain quality criteria.
* **DAX for Error Handling and Data Quality:**
  + **IFERROR() Function:** In DAX measures and calculated columns, you can use the IFERROR() function to handle errors that might occur during calculations (e.g., division by zero). This function allows you to return an alternative value if an error is encountered.
  + **DIVIDE() Function:** For division operations, it's best practice to use the DIVIDE() function, which automatically handles division by zero errors by returning a blank or a specified alternative value.
  + **Error Detection Measures:** You can create DAX measures to count or flag records that violate certain data quality rules. These measures can then be used in visuals to highlight potential issues.

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

Ans = Power BI Premium is capacity geared toward enterprises who want a complete BI solution that provides a single view of its organization, partners, customers, and suppliers. Power BI Premium helps your organization make decisions. Power BI Premium is a SaaS product that allows users to consume content through mobile apps, internally developed apps, or at the Power BI portal.

Power BI Embedded is for ISVs who want to embed visuals into their applications. Power BI Embedded helps your customers make decisions because Power BI Embedded is for application developers, customers of that application can consume content stored on Power BI Embedded capacity, including anyone inside or outside the organization. You can't share Power BI Embedded capacity content through one-click publish to Web or one-click publish to SharePoint.