**Module 5) Creating Dashboard with Visualization Tool**

**Assignment**

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

**Power BI** is a **business analytics tool** from Microsoft that enables users to visualize data, share insights, and make data-driven decisions.  
**Key Differences from Excel**:

* **Purpose**: Power BI is built for data visualization and dashboarding, while Excel is more spreadsheet-oriented.
* **Data Handling**: Power BI can handle large datasets more efficiently.
* **Data Modeling**: Power BI allows relational data modeling with DAX and relationships; Excel has limited capabilities.
* **Refresh & Sharing**: Power BI supports scheduled refreshes and publishing to cloud dashboards; Excel lacks robust sharing features.

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

Data modeling in Power BI involves:

* **Importing multiple tables** (from various sources),
* **Establishing relationships** (like foreign key-primary key),
* **Creating calculated columns/measures** using **DAX**, and
* **Designing a schema** (typically star schema) to support efficient analysis and reporting.

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

Power BI supports two main connection types:

* **Import**: Data is imported into Power BI's internal storage (fast performance).
* **DirectQuery**: Queries run directly on the source database without importing data (real-time updates).
* **Live Connection**: Used mainly with SSAS (SQL Server Analysis Services); similar to DirectQuery but read-only.
* **Composite Models**: Combines Import and DirectQuery in a single model.

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

Data transformation is handled using **Power Query Editor**, where you can:

* Clean and reshape data (remove columns, filter rows, merge/join tables).
* Perform ETL tasks like changing data types, splitting/combining columns.
* Use **M language** for complex transformations.  
  This ensures the data is in the right format for modeling and analysis.

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

DAX is a **formula language** used in Power BI for creating:

* **Calculated columns**
* **Measures**
* **Custom tables**  
  It enables users to perform **advanced calculations**, time intelligence (YTD, MTD), and dynamic filtering logic — making reports more insightful.

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

|  |  |  |
| --- | --- | --- |
| Feature | Calculated Column | Measure |
| Stored in model | Yes (increases data model size) | No (calculated on the fly) |
| Context | Row context | Filter/aggregation context (like SUM, AVG) |
| Use case | Used for filtering/slicing in visuals | Used for values like totals, averages, KPIs |

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

In **Model View**, you:

* Define relationships (one-to-many, many-to-one).
* Set **cardinality** and **cross-filter direction**.
* Power BI uses these relationships to apply filters and join data.  
  Use **Manage Relationships** or drag-and-drop fields to connect tables visually.

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

A **Power BI Gateway**:

* Acts as a **bridge** between on-premises data sources (like SQL Server) and Power BI Service.
* Allows **scheduled refreshes** or **live queries** from cloud to local servers.
* Two types:
  + **Personal Gateway** (for personal use),
  + **Enterprise Gateway** (for shared datasets and scheduled refresh).

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

Steps:

1. Publish your report to Power BI Service.
2. Go to **Dataset > Settings**.
3. Under **Scheduled Refresh**, configure:
   * **Frequency** (daily/hourly),
   * **Time zone** and credentials (OAuth, Windows, etc.).  
     Requires **Power BI Gateway** if data is from on-premises.

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

**Row-Level Security (RLS)** restricts data access for users at the **row level**.

* You define **roles and DAX filters** (e.g., [Region] = "West").
* When users log in, they only see data applicable to their assigned role.  
  Set up RLS in **Power BI Desktop** → publish to **Power BI Service** → assign users to roles.

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

* **Power BI Desktop**: A free **Windows application** used to connect, transform, model data, and build reports locally.
* **Power BI Service**: A **cloud-based platform** used to publish, share, collaborate, and schedule data refreshes.

**Key Differences**:

|  |  |  |
| --- | --- | --- |
| Feature | Power BI Desktop | Power BI Service |
| Usage | Development & report creation | Sharing, collaboration, web access |
| Data Refresh | Manual | Scheduled |
| Accessibility | Windows only | Web browser |

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

**DirectQuery** connects Power BI to a live data source (e.g., SQL Server) without importing data.

* **Data stays in the source**, and queries are sent every time visuals are refreshed.
* Pros: Real-time data, small file size.
* Cons: Slower performance, limited DAX and transformation options.

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

* A **Power BI template (.pbit)** contains the report layout, queries, and model structure, **but no data**.
* Useful for:
  + Reusing standard reports across projects.
  + Sharing report structures without exposing sensitive data.
  + Building white-labeled solutions.

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

**Incremental refresh** loads only new or changed data instead of reloading the entire dataset.  
**Steps**:

1. Define **RangeStart** and **RangeEnd** parameters in Power Query.
2. Filter your data source using these parameters.
3. Enable **incremental refresh** in the table properties (in Power BI Desktop).
4. Publish to Power BI Service.

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

Power Query is used for:

* **Extracting, transforming, and loading (ETL)** data.
* Automating data preparation tasks using a GUI or **M language**.
* Cleaning data before it enters the data model.

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

|  |  |  |
| --- | --- | --- |
| Feature | Calculated Column | Calculated Table |
| Definition | Column added to existing table using DAX | New table created using DAX expression |
| Use Case | Row-level calculations | Creating filtered or summarized tables |
| Storage | Consumes memory | Stored as a separate table in model |

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

1. Use **R** or **Python** for script visuals directly in Power BI.
2. Or develop custom visuals using:
   * **TypeScript + Power BI Visuals SDK**.
   * Package the visual and import it as a .pbiviz file.
3. Alternatively, download from **AppSource (Marketplace)**.

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

* Use **star schema** and avoid snowflake.
* Reduce cardinality in columns (e.g., avoid datetime with time).
* Filter out unused rows and columns.
* Use **Import mode** over DirectQuery for performance.
* Minimize use of complex DAX expressions.
* Aggregate data where possible before import.
* Disable auto date/time if not needed.

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

* **Azure**: Connect to Azure SQL DB, Azure Data Lake, use **Azure Synapse**, or **Azure Machine Learning**.
* **Office 365**: Embed Power BI in **Teams**, export to **Excel**, connect to **SharePoint Lists**.
* **Power Platform**: Integrate with **PowerApps** and **Power Automate** for workflows.

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

* **Aggregations** summarize detailed data to improve performance (e.g., sum of sales by region).
* Power BI uses **aggregation tables** to serve visuals faster.
* You define an aggregated table and **map it to detailed data** via relationships or manage aggregations settings.

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

* Use **Power Query** to:
  + Remove duplicates.
  + Replace or handle null values.
  + Use “Keep errors” or “Remove errors” options.
  + Create **custom error logic** with conditional columns.
* Monitor data quality using **Data Profiling tools** in Power Query (column stats, value distribution).

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

**Power BI Embedded** allows developers to embed Power BI reports and dashboards into **custom applications** (web/mobile).

* Use Cases:
  + ISVs embedding analytics for their customers.
  + Enterprises building internal apps with interactive reports.
* Requires **Azure Power BI Embedded capacity**.
* Enables full control over look, feel, and user access.