**ASSIGNMENT-2**

**QUESTION-1:** Explain the advantages of Natural Queries in PowerBi . . with an example?

**ANSWER:**

Natural language queries in Power BI enable users to ask questions using everyday language instead of complex SQL queries or data manipulation techniques. Here are some advantages of natural language queries in Power BI:

**1. Ease of Use :** Natural language queries make it easy for users, including non-technical users, to interact with data and generate insights without requiring expertise in SQL or data analysis tools. Users can simply type or speak their questions in plain language, such as "What were our sales last quarter?" or "Show me the trend of website traffic over the past year."

**2. Increased Accessibility** : Natural language queries democratize data access by making it accessible to a broader audience within an organization. Users who may not be familiar with database structures or query languages can still explore and analyze data effectively using natural language queries.

**3. Faster Insights :** Natural language queries enable users to quickly retrieve the information they need without having to navigate complex user interfaces or build elaborate reports. This can lead to faster decision-making and more agile responses to changing business conditions.

**4. Intuitive Exploration :** Natural language queries encourage users to explore data intuitively by asking follow-up questions or refining their queries based on initial results. For example, a user might start with a broad question like "Show me sales by region" and then drill down by asking more specific questions like "What were sales in the Northeast region last month?"

**5. Conversational Analytics :** Natural language queries can facilitate a conversational interaction between users and their data. Users can ask questions in a conversational manner and receive immediate feedback in the form of visualizations or data summaries. This conversational approach to analytics can make data exploration feel more interactive and engaging.

Example:

Let's say a sales manager wants to analyze sales performance for a specific product category using natural language queries in Power BI. Instead of writing SQL queries or navigating through complex dashboards, the manager can simply type or speak their question into the Power BI interface.

**For example:**

- Manager : "Show me total sales for electronics last quarter."

- Power BI : Power BI generates a visualization showing total sales for the electronics category during the previous quarter.

The manager can then follow up with additional questions or refine the query further based on the initial results, such as:

- Manager : "What were the top-selling products within the electronics category?"

- Power BI : Power BI provides a list of the top-selling products within the electronics category, along with their sales figures.

By using natural language queries, the sales manager can quickly gain insights into sales performance without needing to write complex queries or navigate through multiple reports.

**QUESTION-2:** Explain Web Front End(WFE) cluster from . . . Power BI Service Architecture?

**ANSWER:**

In the context of Power BI Service Architecture, a Web Front End (WFE) cluster plays a crucial role in serving Power BI reports and dashboards to end-users over the web. Here's an explanation of a WFE cluster within the Power BI Service Architecture:

**1. Definition :** A Web Front End (WFE) cluster is a component of the Power BI Service Architecture responsible for handling user requests, rendering Power BI content (such as reports, dashboards, and datasets), and delivering it to end-users through web browsers or mobile devices.

**2. Functionality** : The WFE cluster acts as an intermediary between users and the Power BI service backend. When a user accesses a Power BI report or dashboard through a web browser, their request is routed to the WFE cluster, which then retrieves the requested content from the appropriate backend services.

**3. Load Balancing** : To ensure high availability and scalability, the Power BI Service employs multiple instances of WFEs organized into a cluster. Load balancing techniques are used to distribute incoming user requests evenly across these WFE instances, thereby optimizing performance and preventing any single WFE from becoming overwhelmed with traffic.

**4. Authentication and Authorization** : The WFE cluster handles authentication and authorization for user access to Power BI content. It verifies user credentials, enforces access control policies, and ensures that users only have access to the reports and dashboards that they are authorized to view based on their permissions and role assignments.

**5. Caching and Optimization :** WFEs may implement caching mechanisms to improve performance and reduce latency for frequently accessed Power BI content. Cached data can be served quickly to users without the need to retrieve it from backend services every time, leading to faster load times and a smoother user experience.

**6. Failover and Redundancy :** In case of a WFE instance failure or outage, the Power BI Service Architecture is designed to automatically redirect user requests to healthy WFE instances, ensuring continuity of service and minimizing downtime. Redundant WFE instances are deployed across multiple data centers or regions to further enhance resilience and fault tolerance.

**QUESTION-3:** Explain Back End cluster from Power BI Service . . . . Architecture?

**ANSWER:**

In the Power BI Service Architecture, the Back End cluster is a crucial component responsible for managing and processing the data that powers Power BI reports, dashboards, and datasets. Here's an explanation of the Back End cluster:

1. Definition : The Back End cluster is the backend infrastructure of the Power BI Service Architecture, comprising various services and components that handle data storage, processing, analysis, and management.

2. Data Storage : The Back End cluster stores the data that is used to create Power BI reports and dashboards. This includes structured data from sources such as databases, data warehouses, and data lakes, as well as semi-structured and unstructured data from sources like files, APIs, and streaming data sources.

3. Data Processing and Analysis : The Back End cluster processes and analyzes the stored data to generate insights and visualizations that are presented in Power BI reports and dashboards. This involves tasks such as data cleansing, transformation, aggregation, modeling, and calculation of key performance indicators (KPIs).

4. Data Management : The Back End cluster provides capabilities for managing and organizing data within the Power BI Service. This includes features such as data modeling, relationships between data tables, data lineage tracking, data governance, and data security.

5. Query Execution : When users interact with Power BI reports and dashboards, their queries are executed by the Back End cluster. This involves retrieving the relevant data from storage, applying any required transformations and calculations, and returning the results to the user interface for display.

6. Integration with Data Sources : The Back End cluster integrates with various data sources to ingest data into the Power BI Service. This may involve direct connections to on-premises or cloud-based data sources, as well as data import, refresh, and incremental loading mechanisms to keep the data up to date.

7. Scalability and Performance : The Back End cluster is designed to scale horizontally and vertically to accommodate increasing data volumes, user loads, and computational demands. It employs distributed computing techniques and parallel processing to achieve high performance and responsiveness.

8. Reliability and Availability : The Back End cluster is built with redundancy, fault tolerance, and failover mechanisms to ensure reliability and availability of the Power BI Service. This includes data replication, disaster recovery, and automatic failover to backup instances in case of hardware failures or service disruptions.

**QUESTION-4:** What ASP.NET component does in Power BI

. Service Architecture?

**ANSWER:**

In the Power BI Service Architecture, ASP.NET components play a crucial role in handling the web-based aspects of the service, including user authentication, authorization, and interaction with Power BI reports and dashboards through the web interface. Here's an explanation of what ASP.NET components do in the Power BI Service Architecture:

1. Web Front End (WFE) : ASP.NET components are primarily responsible for serving the web-based front end of the Power BI Service to users. This includes rendering the Power BI web portal, report viewer, dashboard viewer, and other user interface elements within a web browser.

2. User Authentication and Authorization : ASP.NET components handle user authentication and authorization processes within the Power BI Service. This includes validating user credentials, enforcing access control policies, and verifying permissions to access and interact with specific reports, dashboards, datasets, and features based on user roles and privileges.

3. Session Management : ASP.NET components manage user sessions and maintain state information as users navigate through the Power BI web interface. This involves managing session cookies, session timeouts, and session persistence across multiple requests to ensure a seamless and secure user experience.

4. Request Handling and Routing : ASP.NET components handle incoming HTTP requests from users' web browsers and route them to the appropriate backend services and components within the Power BI Service. This includes parsing and processing request parameters, URL routing, and dispatching requests to the corresponding handlers.

5. Client-Side Interactions : ASP.NET components facilitate client-side interactions between users' web browsers and the Power BI Service. This includes handling user inputs, button clicks, mouse events, and other interactions within the web interface, and triggering corresponding actions such as refreshing data, applying filters, navigating between pages, and interacting with visualizations.

6. Integration with Power BI Embedded : For applications that embed Power BI reports and dashboards using Power BI Embedded, ASP.NET components provide the necessary infrastructure and APIs to embed Power BI content within custom web applications. This involves managing authentication tokens, embedding URLs, and handling interactions between the host application and embedded Power BI content.

7. Customization and Extensibility : ASP.NET components enable customization and extensibility of the Power BI web interface through customizations such as custom visuals, custom themes, custom navigation menus, and custom branding. This allows organizations to tailor the Power BI experience to their specific requirements and branding guidelines.

**QUESTION-5:** Compare Microsoft Excel and PowerBi Desktop on the following features:

Data import

Data transformation

Modeling

Reporting

Server Deployment

Convert Models

Cost

**ANSWER**

Here's a comparison between Microsoft Excel and Power BI Desktop across various features:

1. Data Import:

- Microsoft Excel: Excel allows users to import data from a variety of sources such as databases, files (CSV, TXT, etc.), web, and other Excel files. It offers features like Get Data and Power Query to facilitate data import.

- Power BI Desktop: Power BI Desktop offers robust data import capabilities similar to Excel but with more advanced options. Users can import data from a wide range of sources including databases, files, online services, and cloud platforms. Power Query is integrated directly into Power BI Desktop for data import and transformation.

2. Data Transformation:

- Microsoft Excel: Excel provides basic data transformation capabilities through functions, formulas, and PivotTables. Power Query add-in enhances data transformation capabilities but is separate from Excel.

- Power BI Desktop: Power BI Desktop excels in data transformation with its built-in Power Query Editor, offering a rich set of data transformation features such as data cleansing, shaping, merging, and appending. Users can perform complex data transformations and create reusable queries.

3. Modeling:

- Microsoft Excel: Excel provides basic data modeling capabilities through PivotTables, PivotCharts, and formulas. It supports creating relationships between tables but lacks advanced modeling features.

- Power BI Desktop: Power BI Desktop offers advanced data modeling capabilities including creating relationships, defining calculated columns, measures, and tables, creating hierarchies, and managing data security. Users can build robust data models to analyze complex data relationships.

4. Reporting:

- Microsoft Excel: Excel offers basic reporting capabilities through PivotTables, charts, and conditional formatting. Users can create static reports that are limited in interactivity and visualization options.

- Power BI Desktop: Power BI Desktop provides advanced reporting capabilities with interactive and visually appealing reports. Users can create dynamic reports with various visualizations, drill-down capabilities, cross-filtering, and interactive elements such as slicers, filters, and bookmarks.

5. Server Deployment:

- Microsoft Excel: Excel files can be shared through email, shared drives, or cloud storage services like OneDrive or SharePoint. Collaborative editing is limited, and version control can be challenging.

- Power BI Desktop: Power BI Desktop reports can be published to the Power BI Service for centralized sharing and collaboration. Users can share reports with colleagues, control access permissions, schedule data refresh, and receive automatic updates.

6. Convert Models:

- Microsoft Excel: Excel models can be converted into Power BI Desktop models by importing Excel data into Power BI Desktop and recreating the model structure.

- Power BI Desktop: Power BI Desktop models can be converted into Excel files by exporting data from Power BI Desktop to Excel or using Analyze in Excel feature to connect Excel to Power BI datasets.

7. Cost:

- Microsoft Excel: Excel is a part of Microsoft Office suite and is typically licensed through Office 365 subscriptions or perpetual licenses. The cost varies based on the Office 365 plan or the version of Office purchased.

- Power BI Desktop: Power BI Desktop is a free standalone application available for download from the Microsoft website. However, to publish and share reports, users need a Power BI Pro or Premium license, which comes with a subscription cost.

**QUESTION-6 :** List 20 data sources supported by Power Bi desktop.

Answer:

There are 20 data sources supported by Power BI Desktop:

1. Excel workbook

2. SQL Server Database

3. Azure SQL Database

4. SQL Server Analysis Services (SSAS)

5. Azure Analysis Services

6. Oracle Database

7. MySQL Database

8. PostgreSQL Database

9. IBM Db2 Database

10. Teradata Database

11. Google BigQuery

12. Amazon Redshift

13. Salesforce

14. Google Analytics

15. Adobe Analytics

16. Web (HTML, XML, JSON)

17. SharePoint Online List

18. OData Feed

19. Hadoop File (HDFS)

20. Folder (for importing multiple files from a folder)