POWER BI – ASSIGNMENT 2

Q.1 **Explain the advantages of Natural Queries in Power Bi with an example?**

* Natural queries in Power BI refer to the ability to ask questions about your data using natural language rather than writing complex queries or formulas. There are several advantages in natural queries:

1. **Accessibility**: Natural queries make it easier for non-technical users to interact with data. Instead of needing expertise in writing SQL queries or understanding complex data structures, users can simply ask questions in plain language.
2. **Time-saving**: Natural queries can significantly reduce the time it takes to obtain insights from data. Rather than spending time writing and debugging queries, users can quickly ask questions and receive immediate answers.
3. **Ease of Use**: Natural queries simplify the process of data exploration. Users can ask questions in a way that feels intuitive to them, without needing to worry about the syntax or structure of a query language.
4. **Empowering Business Users**: Natural queries empower business users to perform their own analysis without relying on data analysts or IT professionals. This promotes self-service BI and allows users to make data-driven decisions more independently.

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

* In the architecture of the Power BI Service, the Web Front End (WFE) cluster plays a crucial role in handling user interactions with the Power BI service through their web browsers. Here's an explanation:

1. **User Interaction Point**: The WFE cluster serves as the primary interaction point for users accessing Power BI reports, dashboards, and other artifacts through their web browsers.
2. **Load Balancing**: To ensure scalability and reliability, the WFE cluster typically consists of multiple servers or instances distributed across different physical or virtual machines. A load balancer sits in front of this cluster, distributing incoming user requests evenly across these servers. This ensures that no single server gets overloaded with requests, improving overall performance and responsiveness.
3. **Processing User Requests**: Each server within the WFE cluster is responsible for handling user requests, such as rendering reports, executing queries, or updating dashboard visuals in real-time. These servers communicate with other components of the Power BI Service, such as the backend data processing engines and storage systems, to retrieve and manipulate data as requested by users.

Q.3 **Explain Back End cluster from Power BI Service Architecture?**

* In the Power BI Service architecture, the Back-End cluster forms the core of the platform, handling data processing, storage, and various backend functionalities.

1. **Data Processing**: The Back-End cluster is responsible for processing and transforming data from various sources into formats suitable for visualization in Power BI reports and dashboards. This includes data ingestion, transformation, modeling, and aggregation operations.
2. **Storage Management**: The Back-End cluster manages the storage of data and metadata required by Power BI artifacts such as reports, datasets, and dashboards. This includes both structured data stored in relational databases and unstructured data stored in distributed file systems or cloud storage solutions.
3. **Query Execution**: When users interact with Power BI reports and dashboards, their queries are executed by the Back-End cluster against the underlying data sources. The cluster optimizes query execution to minimize latency and maximize throughput, leveraging techniques such as query caching, parallel processing, and query optimization algorithms.
4. **Metadata Management**: The Back-End cluster maintains metadata about Power BI artifacts, including data schemas, relationships, security permissions, and usage statistics. This metadata is essential for ensuring data integrity, access control, and auditability within the Power BI Service.
5. **Security and Compliance**: Security features such as authentication, authorization, encryption, and data masking are implemented and enforced by the Back-End cluster to protect sensitive data and ensure compliance with regulatory requirements.

Q.4 **What ASP.NET component does in Power BI Service Architecture?**

* In the Power BI Service architecture, ASP.NET components play a significant role in facilitating the web-based functionalities of the platform.

1. **Web Application Framework:** ASP.NET provides the framework for building and hosting web applications, including the Power BI web portal where users interact with reports, dashboards, and other artifacts. It enables the creation of dynamic, interactive web pages that respond to user input and display data visualizations generated by Power BI.
2. **User Interface**: ASP.NET components are responsible for rendering the user interface elements of the Power BI web portal, such as menus, navigation bars, buttons, and interactive controls. These components generate HTML, CSS, and JavaScript code that is rendered in users' web browsers, allowing them to interact with Power BI content seamlessly.
3. **Request Handling**: ASP.NET handles incoming HTTP requests from users accessing the Power BI web portal and routes them to the appropriate components within the Power BI Service architecture.
4. **Integration with Backend Services**: ASP.NET components facilitate communication with backend services and components within the Power BI Service architecture, such as the Back-End cluster responsible for data processing and storage. They initiate requests to retrieve data, execute queries, and perform other operations necessary to fulfill user requests and render Power BI visualizations dynamically.
5. **Customization and Extension Points**: ASP.NET provides extensibility mechanisms that allow developers to customize and extend the functionality of the Power BI web portal.

Q.5 **Customization and Extension Points: ASP.NET provides extensibility mechanisms that allow developers to customize and extend the functionality of the Power BI web portal.**

**Compare Microsoft Excel and Power-Bi Desktop on the following features:**

**Data import**

**Data transformation**

**Modeling**

**Reporting**

**Server Deployment**

**Convert Models**

**Cost**

1. **Data Import:**

* **Excel**: Excel allows users to import data from various sources such as databases, text files, web pages, etc. It offers built-in connectors for common data sources.
* **Power BI Desktop**: Power BI Desktop also supports importing data from various sources including databases, files, online services, etc. It provides a wide range of connectors and allows for more advanced data import and transformation capabilities compared to Excel.

1. **Data Transformation:**

* **Excel**: Excel provides basic data transformation functionalities like sorting, filtering, and basic formulas for data manipulation.
* **Power BI Desktop**: Power BI Desktop offers more advanced data transformation capabilities through its Power Query Editor. Users can perform complex data shaping, cleansing, and transformation operations using a user-friendly interface.

1. **Modeling:**

* **Excel**: Excel supports basic data modeling through pivot tables, charts, and formulas. Users can create relationships between data tables but with limitations compared to Power BI.
* **Power BI Desktop**: Power BI Desktop provides robust data modeling capabilities with its Data Model feature. Users can create relationships, calculated columns, measures, and hierarchies to build complex data models for analysis and visualization.

1. **Reporting:**

* **Excel**: Excel allows users to create reports using pivot tables, charts, and other graphical elements. It provides basic reporting capabilities suitable for simple analysis.
* **Power BI Desktop**: Power BI Desktop offers more advanced reporting capabilities with interactive and dynamic visualizations. Users can create interactive reports, dashboards, and explore data in more depth compared to Excel.

1. **Server Deployment:**

* **Excel**: Excel files can be shared and stored on network drives, SharePoint, or cloud storage services like OneDrive. However, real-time collaboration and centralized management are limited.
* **Power BI Desktop**: Power BI Desktop reports can be published to the Power BI Service for centralized management, sharing, and collaboration. It allows for real-time collaboration, access control, and data refresh scheduling.

1. **Convert Models**:

* **Excel**: Excel models can be converted to Power BI Desktop models, although the process might involve some manual effort depending on the complexity of the model.
* **Power BI Desktop**: Power BI Desktop models can be easily published to the Power BI Service for sharing and collaboration. Additionally, Power BI Desktop supports importing Excel models for further enhancement and visualization.

Q.6 **List 20 data sources supported by Power Bi desktop**

1. **Excel**: Import data from Excel workbooks (.xlsx, .xls).
2. **CSV**: Import data from Comma Separated Values (CSV) files.
3. **SQL Server Database**: Connect to on-premises or cloud-hosted SQL Server databases.
4. **Azure SQL Database**: Connect to Microsoft Azure SQL databases.
5. **SQL Server Analysis Services**: Connect to on-premises or cloud-hosted Analysis Services databases.
6. **Azure Analysis Services:** Connect to Microsoft Azure Analysis Services.
7. **Oracle Database**: Connect to Oracle databases.
8. **MySQL Database**: Connect to MySQL databases.
9. **PostgreSQL**: Connect to PostgreSQL databases.
10. **IBM DB2**: Connect to IBM DB2 databases.
11. **Sybase Database**: Connect to Sybase databases.
12. **Teradata Database**: Connect to Teradata databases.
13. **Google Big-Query**: Connect to Google Big-Query data warehouse.
14. **Amazon Redshift**: Connect to Amazon Redshift data warehouse.
15. **Web**: Directly import data from web pages or web services using APIs.
16. **Microsoft SharePoint**: Connect to SharePoint lists and libraries.
17. **Microsoft Exchange**: Connect to Exchange Online mailboxes.
18. **Salesforce**: Connect to Salesforce CRM data.
19. **Dynamics 365**: Connect to Microsoft Dynamics 365 data.
20. **JSON**: Import data from JSON files or web APIs.