**Power BI Assignment 2**

Q1. Explain the advantages of Natural Queries in PowerBi with an example?

**Ans.** In Power BI, Natural Queries, also known as Q&A (Question and Answer), is a feature that allows users to ask questions in plain, natural language to retrieve insights and generate visualizations from their data. This feature leverages natural language processing (NLP) and machine learning to interpret user queries and generate meaningful responses.

**Here are some advantages of Natural Queries in Power BI:**

1. Ease of Use

2. Faster Insights

3. Increased Accessibility

4. Discover Hidden Insights

5. Customizable Visualizations

**Example:**

Let's say you have a sales dataset in Power BI, and you want to understand the sales performance for a specific product category in the last quarter. With Natural Queries, you can simply type or speak a question like, "What were the sales for electronics in Q3 2023?" Power BI's Natural Query feature would then interpret your question and generate a visualization that shows the sales performance for the electronics category in the third quarter of 2023.

You can further refine your question by asking, "Compare the sales of electronics and clothing in Q3 2023," and Power BI would generate a comparative visualization for these two product categories.

Overall, Natural Queries in Power BI streamline the process of data exploration and analysis, making it more intuitive and accessible for users across different levels of technical expertise.

Q2. Explain Web Front End (WFE) cluster from Power BI Service Architecture?

**Ans.** The Web Front End (WFE) cluster is the first point of contact for users when they connect to the Power BI service. It is responsible for:

* Authenticating users using Azure Active Directory (Azure AD)
* Providing tokens for subsequent user connections to the Power BI service
* Routing user requests to the appropriate back-end cluster
* Serving static content, such as images and JavaScript files, to users

The WFE cluster is made up of a number of web servers that are deployed in Azure. The number of web servers in the cluster is determined by the number of users that are expected to connect to the Power BI service.

The WFE cluster is located in multiple Azure regions around the world. This ensures that users are always connected to the WFE cluster that is closest to them, which improves performance.

The WFE cluster uses Azure Traffic Manager to direct user traffic to the nearest WFE cluster. Traffic Manager uses the DNS record of the client attempting to connect to determine the closest WFE cluster.

The WFE cluster also uses the Azure Content Delivery Network (CDN) to efficiently distribute static content to users. The CDN caches static content in data centers around the world, which reduces the load on the WFE cluster and improves performance for users.

**Here are some of the benefits of using a WFE cluster in the Power BI service architecture:**

* Improved performance
* Scalability
* Reliability

Q3. Explain Back End cluster from Power BI Service Architecture ?

**Ans.** The Back-End cluster in the Power BI service architecture is responsible for managing visualizations, user dashboards, datasets, reports, data storage, data connections, data refresh, and other aspects of interacting with the Power BI service. It is made up of a number of roles, including:

* **Gateway Role:** The Gateway Role acts as a gateway between user requests and the Power BI service. It is responsible for authenticating users, routing requests to the appropriate back-end roles, and managing data access permissions.
* **Presentation Role:** The Presentation Role handles all the visualization related queries like for dashboards and reports. It retrieves the data from the Data Role or Data Movement Role and renders the visualizations.
* **Data Role:** The Data Role manages all the data related queries. It retrieves the data from the underlying data sources and stores it in Azure Blob Storage.
* **Data Movement Role:** The Data Movement Role is responsible for moving data between the underlying data sources and Azure Blob Storage.
* **Metadata Role:** The Metadata Role manages all the metadata related to the Power BI service. This includes things like the names of datasets, reports, and dashboards, as well as the relationships between them.

The Back-End cluster is located in multiple Azure regions around the world. This ensures that users are always connected to the Back-End cluster that is closest to them, which improves performance.

The Back-End cluster uses Azure Traffic Manager to direct user traffic to the nearest Back-End cluster. Traffic Manager uses the DNS record of the client attempting to connect to determine the closest Back-End cluster.

**Here are some of the benefits of using a Back-End cluster in the Power BI service architecture:**

* **Improved performance:** The Back-End cluster is located in multiple Azure regions around the world, which ensures that users are always connected to the Back-End cluster that is closest to them. This improves performance for users.
* **Scalability:** The Back-End cluster can be scaled up or down to meet the needs of the organization. This ensures that the Back-End cluster can handle the increasing number of users and requests.
* **Reliability:** The Back-End cluster is highly available and resilient to failures. This ensures that users can always access the Power BI service.

Q4. What ASP.NET component does in Power BI Service Architecture?

**Ans.** The ASP.NET component in the Power BI service architecture is responsible for handling user requests for reports and dashboards. It is a web application that is hosted on Azure App Service. The ASP.NET component communicates with the other back-end roles to retrieve data and render visualizations.

**Here are some of the specific tasks that the ASP.NET component performs:**

* Handles user authentication and authorization
* Routes user requests to the appropriate back-end roles
* Renders reports and dashboards
* Supports features such as drill-down and slicers
* Manages user session state
* Logs errors and metrics

The ASP.NET component is a critical part of the Power BI service architecture. It provides a user-friendly interface for interacting with reports and dashboards, and it helps to ensure that the service is reliable and scalable.

**Here are some of the benefits of using ASP.NET in the Power BI service architecture:**

* **Robustness:** ASP.NET is a mature and robust platform that has been used to build many successful web applications. This ensures that the ASP.NET component in the Power BI service architecture is reliable and scalable.
* **Security:** ASP.NET supports a variety of security features, such as authentication and authorization, to protect user data. This ensures that user data is safe and secure.
* **Scalability:** ASP.NET can be scaled up or down to meet the needs of the organization. This ensures that the ASP.NET component in the Power BI service architecture can handle the increasing number of users and requests.
* **Maintainability:** ASP.NET is a well-documented platform with a large community of developers. This makes it easy to maintain and extend the ASP.NET component in the Power BI service architecture.

Q5. Compare Microsoft Excel and PowerBi Desktop on the following features:

Data import

Data transformation

Modeling

Reporting

Server Deployment

Convert Models

Cost ?

**Ans.** The comparison of Microsoft Excel and Power BI Desktop on the features you mentioned:

**Data Import**

* **Excel:** Excel can import data from a variety of sources, including text files, CSV files, and Microsoft Access databases. It can also connect to live data sources, such as SQL Server and Oracle databases.
* **Power BI Desktop:** Power BI Desktop can import data from a wider range of sources than Excel, including cloud-based data sources such as Azure SQL Database and Salesforce. It can also connect to live data sources.

**Data transformation**

* **Excel:** Excel has a variety of data transformation tools, such as filters, pivot tables, and macros.
* **Power BI Desktop:** Power BI Desktop has a more powerful set of data transformation tools than Excel. These tools include data cleansing, data modeling, and data mining.

**Modeling**

* **Excel:** Excel does not have a dedicated modeling tool. However, you can use pivot tables and macros to create simple models.
* **Power BI Desktop:** Power BI Desktop has a dedicated modeling tool called the Power BI Model. The Power BI Model can be used to create complex models that can be used for data analysis and reporting.

**Reporting**

* **Excel:** Excel has a variety of reporting tools, such as charts, tables, and pivot tables.
* **Power BI Desktop:** Power BI Desktop has a more powerful set of reporting tools than Excel. These tools include interactive dashboards, custom visuals, and paginated reports.

**Server deployment**

* **Excel:** Excel is a desktop application that cannot be deployed on a server.
* **Power BI Desktop:** Power BI Desktop can be deployed on a server as a Power BI Report Server. Power BI Report Server can be used to share reports and dashboards with a wider audience.

**Convert Models**

* **Excel:** Excel does not have a built-in tool to convert models. However, you can use third-party tools to convert Excel models to other formats, such as Power BI models.
* **Power BI Desktop:** Power BI Desktop has a built-in tool to convert models. This tool can be used to convert models from other formats, such as Excel models, to Power BI models.

**Cost**

* **Excel:** Excel is included with Microsoft Office.
* **Power BI Desktop:** Power BI Desktop is available in two editions: Power BI Desktop and Power BI Pro. Power BI Desktop is free to use. Power BI Pro costs $9.99 per user per month.

In general, Power BI Desktop is a more powerful tool than Excel for data analysis and reporting. However, Excel is a more affordable option and it is easier to learn.

Q6. List 20 data sources supported by Power Bi desktop.

**Ans.**

1. Microsoft Excel
2. Microsoft Access
3. SQL Server
4. Oracle
5. IBM Db2
6. MySQL
7. PostgreSQL
8. SAP HANA
9. Amazon Redshift
10. Google Big Query
11. Snowflake
12. Vertica
13. Teradata
14. Azure SQL Database
15. Azure Synapse Analytics
16. Azure Data Explorer
17. Azure Blob Storage
18. Azure Table Storage
19. Azure Cosmos DB
20. Salesforce
21. Google Analytics
22. Facebook Ads
23. Twitter