PowerBI -- ASSIGNMENT NO.2

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

Natural Query Language (NQL) in PowerBI allows users to interact with their data in a more intuitive and human-like way. Instead of writing complex queries in SQL or DAX, users can simply type their questions in plain English.

Some advantages of NQL in PowerBI include:

* Improved User Experience: NQL provides a more user-friendly way for users to interact with their data, allowing them to quickly get the information they need without having to understand complex SQL or DAX syntax.
* Increased Productivity: With NQL, users can complete tasks more quickly, as they don't have to spend time learning or writing complex SQL or DAX queries.
* Better Data Discovery: NQL can help users discover new insights and relationships in their data by allowing them to easily ask questions and explore their data in a more natural and intuitive way.

Example:

Suppose you have a data set of sales data for a retail company and you want to find the total sales for a specific product category. With NQL, you can simply type a question like "What are the total sales for the electronics category?" and PowerBI will automatically generate the appropriate query and display the results. This is a much more natural and intuitive way of getting the information you need, compared to writing a complex SQL or DAX query.

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

In PowerBI Service architecture, a Web Front End (WFE) cluster is a group of servers that are responsible for handling the web requests made by users. The WFE cluster is the main entry point for all PowerBI users, who access the service through a web browser or a mobile app.

The WFE cluster is designed to be highly scalable, allowing it to handle large numbers of concurrent users and requests. It is also designed to provide high availability, so that users can access the PowerBI Service even if one or more servers in the cluster are down.

The WFE cluster is responsible for handling a variety of tasks, including:

* Authentication and authorization: The WFE cluster ensures that only authorized users can access the PowerBI Service and the data they need.
* Request routing: The WFE cluster routes requests to the appropriate backend services, such as the data engine, data modeling, and report rendering.
* Report rendering: The WFE cluster handles the rendering of reports and visualizations, including the processing of DAX calculations and the creation of interactive dashboards.
* Session management: The WFE cluster manages user sessions, ensuring that each user's data and preferences are preserved as they navigate the PowerBI Service.

Overall, the WFE cluster plays a critical role in the PowerBI Service architecture, providing a high-performance, highly scalable, and highly available platform for accessing and exploring business data.

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

In PowerBI Service architecture, the Back End cluster is a group of servers that handle the behind-the-scenes processing of data and reports. The Back End cluster works in conjunction with the Web Front End (WFE) cluster to deliver a high-performance and scalable platform for business intelligence.

The Back End cluster is responsible for handling a variety of tasks, including:

* Data processing: The Back End cluster processes large amounts of data, including data from various data sources, to ensure that it is ready for use in reports and dashboards.
* Data modeling: The Back End cluster handles the creation and management of data models, including the generation of relationships, calculated fields, and other data-related artifacts.
* Report calculation: The Back End cluster performs the calculations required for report rendering, including the processing of DAX formulas and the generation of aggregated data.
* Report rendering: The Back End cluster prepares the data required for report rendering, including the calculation of totals, subtotals, and other report-level metrics.

Overall, the Back End cluster is critical to the performance and scalability of the PowerBI Service, as it handles the heavy lifting of data processing, modeling, and calculation, allowing the WFE cluster to focus on delivering a fast and responsive user experience.

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

In the PowerBI Service architecture, ASP.NET is a key component that provides the web-based framework for delivering the PowerBI Service to users. ASP.NET is a web application framework developed by Microsoft that provides a platform for building dynamic, data-driven web applications.

In the PowerBI Service architecture, ASP.NET is used to handle a variety of tasks, including:

1. User interface: ASP.NET provides the user interface for the PowerBI Service, including the dashboards, reports, and other visualizations that users interact with.
2. Request handling: ASP.NET handles incoming web requests from users, including requests for report data and the rendering of reports and visualizations.
3. Session management: ASP.NET provides the underlying infrastructure for session management, ensuring that each user's data and preferences are preserved as they navigate the PowerBI Service.
4. Security: ASP.NET provides a secure platform for delivering the PowerBI Service, including support for authentication and authorization, encryption, and other security-related features.

Overall, ASP.NET plays an important role in the PowerBI Service architecture, providing the web-based framework that enables users to interact with their data and insights in a fast and intuitive way.

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

**Data import**

**Data transformation**

**Modeling**

**Reporting**

**Server Deployment**

**Convert Models**

**Cost**

Here is a comparison of Microsoft Excel and PowerBI Desktop on the following features:

* + Data import: Both Excel and PowerBI Desktop allow users to import data from a variety of sources, including spreadsheets, databases, and cloud-based data sources. PowerBI Desktop has a more comprehensive set of data import options, including direct connections to cloud-based data sources and support for APIs.
  + Data transformation: Both Excel and PowerBI Desktop provide tools for data transformation, including pivot tables and data cleaning. PowerBI Desktop also has a more advanced data transformation engine, including support for DAX (Data Analysis Expressions) and Power Query.
  + Modeling: Both Excel and PowerBI Desktop provide data modeling capabilities, including the creation of relationships and calculated fields. PowerBI Desktop has a more advanced modeling engine, including support for complex relationships and calculations, and the ability to create multi-table models.
  + Reporting: Both Excel and PowerBI Desktop provide reporting capabilities, including the creation of charts, tables, and dashboards. PowerBI Desktop has a more advanced reporting engine, including support for real-time data visualizations and interactive dashboards.
  + Server Deployment: PowerBI Desktop is designed for use with the PowerBI Service, which provides server-based deployment options for sharing and collaboration. Excel does not have server-based deployment options, but it can be used to create reports and dashboards that can be shared and distributed.
  + Convert Models: PowerBI Desktop provides the ability to convert models to the PowerBI Service, which allows for server-based deployment and collaboration. Excel does not have this capability.
* Cost: Microsoft Excel is typically included as part of the Microsoft Office suite, while PowerBI Desktop requires a separate license. PowerBI Desktop is included with a PowerBI Pro license, which costs $9.99 per user per month, while a PowerBI Premium license, which provides additional capabilities, starts at $4,995 per month.

Overall, while both Excel and PowerBI Desktop provide data analysis and reporting capabilities, PowerBI Desktop is designed for more advanced data analysis and reporting, and is well suited for large-scale data projects, while Excel is well suited for small-scale data projects and ad-hoc reporting.

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

PowerBI Desktop supports a wide range of data sources, including the following:

1. Excel spreadsheets
2. SQL Server
3. Oracle
4. Azure SQL Database
5. Azure Data Lake Storage
6. SharePoint lists
7. Access databases
8. Oracle databases
9. MySQL databases
10. PostgreSQL databases
11. Amazon Redshift databases
12. IBM Db2 databases
13. Snowflake databases
14. Hadoop Hive databases
15. Salesforce
16. Google BigQuery
17. OData feeds
18. PDF files
19. Web content (HTML, XML, etc.)
20. Power BI datasets