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

Natural Language Queries is a powerful feature in Power BI that allows users to ask questions about their data in natural language rather than using complex queries or formulas. Some of the key advantages of using natural queries in Power BI are:

Easy to use: Natural language queries are very intuitive and easy to use, even for users who are not familiar with data analytics or SQL queries. Users can simply type in their questions or speak to their device and get the results they need.

Quick insights: Natural queries provide users with quick insights into their data, without the need for complex data modeling or analysis. Users can get answers to their questions in seconds, rather than hours or days.

Interactive: Natural language queries are interactive, allowing users to refine their questions and explore their data in real-time. Users can drill down into their data, filter it, and get more specific answers to their questions.

Versatile: Natural language queries can be used to analyze a wide range of data sources, including Excel spreadsheets, cloud-based data sources, and on-premises data sources.

Here's an example of how natural queries can be used in Power BI:

Suppose you have a sales report that includes information about products, salespeople, and sales regions. You want to find out which product had the highest sales in the last quarter. Instead of writing a complex query or creating a pivot table, you can simply ask Power BI, "What was the product with the highest sales in Q4?" Power BI will then analyze your data and provide you with the answer, along with any related insights or visualizations. You can then refine your question, filter the data, or drill down into specific regions or salespeople to get more detailed information.

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

In the Power BI Service Architecture, the Web Front End (WFE) cluster is responsible for handling incoming requests from users and directing those requests to the appropriate backend services. The WFE cluster acts as a gateway for users to access the Power BI service through their web browsers.

The WFE cluster is made up of multiple servers that work together to handle incoming requests and ensure high availability and scalability. These servers are typically load-balanced to distribute traffic evenly across them and prevent any single server from becoming overloaded.

When a user accesses the Power BI service through their web browser, their request is routed to one of the servers in the WFE cluster. The WFE server then checks the user's credentials to ensure they have access to the requested content and forwards the request to the appropriate backend service. This could be the report rendering service, the dataset service, or another service depending on the user's request.

The WFE cluster also handles other important tasks such as caching, session management, and monitoring. Caching allows frequently accessed data to be stored in memory, reducing the amount of time it takes to retrieve data from the backend services. Session management ensures that each user's session is handled correctly and securely. Monitoring enables the WFE cluster to detect and respond to any issues or failures that may occur.

Overall, the Web Front End (WFE) cluster is a critical component of the Power BI Service Architecture that enables users to access and interact with their data in a secure and scalable way.

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

In the Power BI Service Architecture, the back-end cluster is responsible for providing the core functionality of the Power BI service. The back-end cluster consists of several services that work together to provide a complete end-to-end solution for data analysis and visualization.

The main services that make up the back-end cluster in Power BI include:

Dataset Service: This service is responsible for managing and maintaining the data that is used in reports and dashboards. It allows users to connect to a wide range of data sources, including on-premises data sources and cloud-based data sources, and transform the data into a format that can be used in Power BI.

Report Rendering Service: This service is responsible for rendering reports and visualizations based on the data that is stored in the Dataset Service. It uses advanced rendering techniques to provide high-quality, interactive reports and visualizations that can be viewed on a wide range of devices.

Data Management Gateway: This service is responsible for connecting on-premises data sources to the Power BI service. It provides a secure and reliable connection between the Power BI service and on-premises data sources, allowing users to access their data in real-time.

Security and Authentication Service: This service is responsible for managing security and authentication in the Power BI service. It ensures that only authorized users can access data and reports, and provides a range of security features to protect against unauthorized access or data breaches.

Data Refresh Service: This service is responsible for refreshing data in reports and visualizations. It allows users to schedule data refreshes and ensures that the data in reports is always up-to-date.

All these services work together to provide a powerful and comprehensive solution for data analysis and visualization in Power BI. The back-end cluster is designed to be highly scalable and resilient, ensuring that the service can handle large volumes of data and traffic, while maintaining high levels of performance and availability.

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

In the Power BI Service Architecture, ASP.NET is a core component that provides the web application framework for the Power BI service. ASP.NET is a server-side web application framework that allows developers to build dynamic web applications using a variety of programming languages such as C#, VB.NET, and F#.

The ASP.NET component in Power BI is responsible for handling incoming requests from users and providing dynamic content in response to those requests. ASP.NET provides a range of features that make it easier to build secure and scalable web applications, including:

Session management: ASP.NET provides built-in support for managing user sessions, allowing users to maintain their state across multiple requests.

Authentication and authorization: ASP.NET provides a range of authentication and authorization mechanisms that enable developers to build secure applications that protect against unauthorized access.

Caching: ASP.NET provides caching mechanisms that allow frequently accessed data to be stored in memory, reducing the amount of time it takes to retrieve data from the backend services.

Error handling: ASP.NET provides robust error handling mechanisms that help developers identify and resolve errors that occur during application execution.

Extensibility: ASP.NET provides a range of extensibility points that enable developers to customize the behavior of the web application and integrate it with other systems.

Overall, the ASP.NET component in Power BI is a critical part of the Power BI Service Architecture that provides the web application framework for building and delivering dynamic, interactive reports and visualizations to users.

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

Data import

Data transformation

Modeling

Reporting

Server Deployment

Convert Models

Cost

|  |  |  |
| --- | --- | --- |
| Comparison Points | Excel | PowerBi Desktop |
| Data import | Excel allows users to import data from a wide range of sources, including text files, databases, web pages, and other Excel workbooks. | Power BI Desktop also supports data import from a variety of sources, including Excel workbooks, databases, web services, and cloud-based data sources such as Azure SQL Database and SharePoint Online. |
| Data transformation | Excel provides basic data transformation capabilities such as filtering, sorting, and grouping data. It also offers some built-in functions for data cleaning and manipulation. | Power BI Desktop provides more advanced data transformation capabilities, including data modeling, data shaping, and data cleansing. It has a powerful query editor that allows users to perform complex transformations and data cleaning tasks. |
| Modeling | Excel provides basic modeling capabilities such as creating PivotTables and PivotCharts to summarize and analyze data. | Power BI Desktop provides advanced data modeling capabilities, including the ability to create relationships between tables, define hierarchies, and create calculated measures and columns. |
| Reporting | Excel provides basic reporting capabilities such as creating charts, tables, and graphs to visualize data. | Power BI Desktop provides advanced reporting capabilities, including interactive dashboards, custom visuals, and the ability to create sophisticated reports and visualizations. |
| Server Deployment | Excel workbooks can be saved and shared through various channels such as email, OneDrive, SharePoint, and Teams. However, collaboration and real-time updates are limited. | Power BI Desktop allows users to publish their reports to the Power BI service, which enables collaboration and real-time updates. Reports can be accessed and shared with other users through various channels such as email, web, and mobile devices. |
| Convert Models | Excel models cannot be directly converted into Power BI models, but data can be exported from Excel and imported into Power BI Desktop for modeling and analysis. | Power BI Desktop models can be exported and saved as PBIX files, which can be opened in Power BI service for sharing and collaboration. |
| Cost | Excel is part of the Microsoft Office suite, and the cost varies depending on the subscription plan. | Power BI Desktop is a free desktop application, but the cost of using the Power BI service for sharing and collaboration depends on the subscription plan. |

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

1. Excel workbooks
2. CSV files
3. Text files
4. SQL Server databases
5. Oracle databases
6. MySQL databases
7. PostgreSQL databases
8. IBM DB2 databases
9. Amazon Redshift
10. Google BigQuery
11. Salesforce
12. Dynamics 365
13. SharePoint
14. OneDrive
15. Azure Data Lake Storage
16. Azure Cosmos DB
17. Hadoop Distributed File System (HDFS)
18. JSON files
19. Web APIs
20. OData feeds