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

* Natural queries are advantageous because they allow users to ask questions in plain language and receive instant answers in the form of visual reports.
* The fastest way to get an answer from your data is to perform a search over your data using natural language. This feature in Power BI lets you explore your data in your own words by using natural language.
* It makes sense of that unstructured data, making it organized, queryable, and searchable.
* Some experts believe that natural language querying could help drive deeper insights by lowering the expertise required to interact with [BI and analytics tools](https://www.techtarget.com/searchbusinessanalytics/feature/How-to-evaluate-and-select-the-right-BI-analytics-tool).

For example- you could ask “What was our sale revenue last month?’’ and power bi would generate a visual report with answers.

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

in the Power BI service architecture, a web front end (WFE) cluster is a group of servers that handle user requests from the web browser. The WFE cluster is responsible for managing user authentication, routing requests to the appropriate backend server, and serving up the Power BI reports and dashboards to the user. For example, when you log into the Power BI service and request a report, your request is routed to a WFE server. The WFE server then authenticates your user account and forwards the request to the backend server that hosts the report. Once the report is generated, the WFE server sends it back to your web browser for display. The WFE cluster is designed to handle high volumes of user requests and ensure that the Power BI service is always available and responsive. By distributing the workload across multiple servers, the WFE cluster can provide a fast and reliable user experience even during periods of high demand.

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

The backend cluster in Power BI service architecture is a group of servers that are responsible for processing and storing data. The backend cluster handles tasks such as data loading, model processing, and report generation. For example, when you create a report in Power BI Desktop and publish it to the service, the backend cluster is responsible for processing the report and storing the data. The backend cluster also handles data refreshes and updates to ensure that your reports are always up-to-date. The backend cluster is designed to handle large volumes of data and ensure that the Power BI service is always available and responsive. By distributing the workload across multiple servers, the backend cluster can provide a fast and reliable user experience even during periods of high demand.

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

In the Power BI service architecture, ASP.NET is a web application framework that is used to build and host web applications. The Power BI service uses ASP.NET to build and host the web application that powers the Power BI service. ASP.NET provides a number of features that are useful for building web applications, including server-side scripting, database connectivity, and support for web services. It also provides a number of tools for building and deploying web applications, including Visual Studio and Microsoft Azure. In the context of Power BI, ASP.NET is used to build and host the web application that powers the Power BI service. This web application provides the user interface for the Power BI service, as well as the backend services that handle data processing, report generation, and other tasks.

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

Data import

Data transformation

Modeling

Reporting

Server Deployment

Convert Models

Cost

Data imports:

Excel: You can import data from a variety of sources, including CSV, text files, and databases. You can also use Power Query to import data from web pages and other sources.

Power BI Desktop: You can import data from a wide range of sources, including Excel workbooks, databases, and cloud services. Power BI Desktop also includes Power Query for data import.

Data transformation:

Excel: You can use Power Query to transform your data in Excel, including cleaning, shaping, and merging data from different sources.

Power BI Desktop: Power BI Desktop includes Power Query for data transformation, which allows you to clean, shape, and merge data from different sources. Power BI Desktop also includes additional data transformation features, such as calculated columns and measures.

Modeling:

Excel: You can create simple data models in Excel using PivotTables and PivotCharts. Excel also includes Power Pivot for more advanced modeling.

Power BI Desktop: Power BI Desktop includes a powerful data modeling engine that allows you to create complex data models with relationships, hierarchies, and calculations.

Reporting:

Excel: You can create basic reports in Excel using PivotTables and PivotCharts. Excel also includes basic charting and graphing tools.

Power BI Desktop: Power BI Desktop includes a wide range of reporting tools, including interactive visualizations, dashboards, and custom visuals.

Server deployment:

Excel: Excel workbooks can be saved to OneDrive or SharePoint for sharing and collaboration.

Power BI Desktop: Power BI Desktop reports can be published to the Power BI service for sharing and collaboration.

Convert models:

Excel: Excel models can be exported to Power BI Desktop or other BI tools.

Power BI Desktop: Power BI Desktop models can be exported to Excel or other BI tools.

Cost:

Excel: Excel is included with Microsoft Office, which requires a subscription. There are also standalone versions of Excel available for purchase.

Power BI Desktop: Power BI Desktop is a free download. However, to use the Power BI service for sharing and collaboration, a subscription is required.

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

20 data sources that Power BI Desktop supports are:

* 1. Excel workbooks
  2. CSV files
  3. Text files
  4. XML files
  5. JSON file
  6. Folder
  7. SharePoint folders and lists
  8. SQL Server databases
  9. MySQL databases
  10. Oracle databases
  11. PostgreSQL databases
  12. IBM DB2 databases
  13. Teradata databases
  14. Azure SQL Database
  15. Azure Data Lake Storage Gen1 and Gen2
  16. Azure Blob Storage
  17. Azure HDInsight
  18. Amazon Redshift
  19. Google BigQuery
  20. Salesforce