1. Explain Architecture of Spark ?

Spark architecture consists of a driver program coordinating tasks and communicating with a cluster manager, which allocates resources across worker nodes. Worker nodes run executors responsible for executing tasks. RDDs form the core data structure, transformed through operations forming a DAG.

2. Difference between Hadoop n spark.

Hadoop MapReduce operates on disk-based data, reading and writing data between each Map and Reduce stage. Spark keeps intermediate data in memory whenever possible, reducing disk I/O and speeding up processing.

Spark generally performs faster than Hadoop MapReduce, especially for iterative algorithms and interactive data analysis, due to its in-memory processing capability. Hadoop MapReduce may be more suitable for batch processing of large, static datasets.

Hadoop MapReduce is well-suited for batch processing of large datasets, particularly when fault tolerance and scalability are essential. Spark is preferred for iterative processing, real-time stream processing, and interactive analytics due to its faster processing speedl.

3. Difference between RDD, Dataframe, Dataset.

RDDs provide low-level control and fault tolerance but lack optimization.
DataFrames offer a high-level, optimized API for working with structured data.
Datasets combine the benefits of RDDs and DataFrames, providing type safety and optimization, primarily in Scala and Java.

4. Explain the similarities in all API of Spark.

Spark APIs share distributed processing, fault tolerance, lazy evaluation, optimizations, and support for multiple languages, ensuring seamless interoperability and integration with external systems for efficient large-scale data processing.

5. What is Transformation? Explain in detail.

A transformation in Apache Spark refers to an operation applied to a distributed dataset to produce another dataset. Transformations are lazy operations, meaning they don't compute their results immediately, instead, they build up a directed acyclic graph (DAG) representing the sequence of operations to be executed. When an action is invoked, Spark uses this DAG to optimize and execute the computations in parallel across the cluster. transformations in Spark are immutable, meaning they do not modify the original dataset, instead, they create a new dataset with the applied transformation. This immutability ensures fault tolerance and allows for easy recovery in case of failures.

6. What is Actions in spark? Explain in detail.

Actions in Apache Spark are operations that trigger the execution of the directed acyclic graph of transformations defined on a distributed dataset. Unlike transformations, which are lazy and do not compute results immediately, actions are eager and initiate the computation process by instructing Spark to execute the transformations and produce a result.

7. What is the Wide Transformation ?, explain with example.

Wide transformations in Apache Spark are operations that require data to be shuffled across partitions, often involving a stage boundary. These transformations typically result in a redistribution of data across the cluster and may involve network communication and disk I/O. Wide transformations are necessary when the operation requires data from multiple partitions to be combined or grouped together.

Example: groupByKey Transformation

Suppose we have a dataset of key-value pairs representing sales data, where the key is the product category and the value is the sales amount. We want to calculate the total sales amount for each product category.

8. What is Narrow Transformation? Explain with example.

Narrow transformations in Apache Spark are operations where each input partition contributes to only one output partition. These transformations can be computed in parallel without shuffling data across partitions, making them more efficient in terms of execution.

Example: map Transformation

Suppose we have an RDD containing a list of numbers, and we want to double each number in the list.

9. Write down the query of wide n narrow transformation with example.

we start with an RDD sales_data containing key-value pairs representing sales data, where the key is the product category and the value is the sales amount.

```
mapped_sales = sales_data.map(lambda x: (x[0], x[1]))

grouped_sales = mapped_sales.groupByKey()

total_sales_per_category = grouped_sales.mapValues(lambda sales: sum(sales))
```

10. Explain Kerberos Architecture.

In Kerberos Authentication server and database is used for client authentication. Kerberos runs as a third-party trusted server known as the Key Distribution Center (KDC).

The main components of Kerberos are:

Authentication Server (AS): The Authentication Server performs the initial authentication and ticket for Ticket Granting Service.

Database: The Authentication Server verifies the access rights of users in the database.

Ticket Granting Server (TGS): The Ticket Granting Server issues the ticket for the Server.