Apache Hive

Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis. It is an open-source data warehouse software project that facilitates easy data management and querying of large datasets residing in distributed storage. Hive uses a language called HiveQL, which is similar to SQL, to express queries.

Here are some key features and components of Apache Hive:

HiveQL: Hive Query Language is similar to SQL and allows users to express queries using SQL-like syntax. It abstracts the complexity of Hadoop MapReduce programming, making it easier for users familiar with SQL to work with big data.

Metastore: The Hive Metastore is a database that stores metadata about Hive tables, partitions, and their associated schemas. It helps in managing and organizing metadata to optimize query performance.

Hive Server: The Hive Server is responsible for handling client requests and executing queries. It supports multiple clients and provides a Thrift interface for communication.

Hadoop Distributed File System (HDFS): Hive stores its data on HDFS, making it scalable and fault-tolerant. It leverages Hadoop's distributed storage capabilities.

MapReduce: Although Hive abstracts MapReduce programming for users, it internally translates HiveQL queries into a series of MapReduce jobs that are executed on the Hadoop cluster.

Partitions and Buckets: Hive allows users to partition data based on one or more columns, which can significantly improve query performance. Additionally, data can be further organized into buckets, which are subsets of partitions.

Extensibility: Hive supports the addition of custom User-Defined Functions (UDFs), allowing users to incorporate their own logic or processing capabilities into Hive queries.

Integration with Hadoop Ecosystem: Hive seamlessly integrates with other Hadoop ecosystem tools such as HBase, Spark, and more, making it a versatile component in the big data processing pipeline.

Task 1: Create a Hive table and upload the u.data from HDFS into it. Find the number of records present in the dataset.

Task 2: Fetch the count of each film rating in the dataset and sort them in descending order.

Task 3: Find the min, max and avg rating for each film_id

Task 4: Find the users who rated more than 20 number of films

Task 5: Join u.user with u.data and find the number of user based on their occupation