1. **What is HIVE? Explain the features of HIVE.**

Hive is a Data Warehousing tool. Hive is used to query structured data built on top of Hadoop. Facebook created Hive component to manage their ever-growing volumes of log data.

**Hive makes use of the following:**

1. HDFS for Storage.

2. MapReduce for execution.

3. RDBMS for storing metadata.

Hive provides HQL (Hive Query Language) which is similar to SQL. Hive compiles SQL queries into Reduce jobs and then runs the job in the Hadoop Cluster. Hive provides extensive data type functions and formats for data summarization and analysis. Hive is suitable for Data warehousing applications. Process batch jobs on huge data that is immutable.

**Hive features:**

1. **It is similar to SQL.**

Hive uses a language called Hive Query Language (HQL), which is similar to SQL (Structured Query Language). This makes it easier for users who are familiar with SQL to transition to Hive without a steep learning curve.

1. **HQL is easy to code.**

HQL is designed to be user-friendly and easy to write, especially for those who are already familiar with SQL syntax. This feature helps users quickly adapt to and start using Hive for querying and analyzing data stored in Hadoop.

1. **Hive supports rich data types such as structs, lists, and maps.**

Hive supports a variety of complex data types, including structs, lists, and maps. This allows users to work with and manipulate data in a more flexible and structured way, beyond simple scalar types.

1. **Hive supports SQL filters, Group-By, and Order-By Clauses:**

Hive supports many SQL-like features, including filtering data using the WHERE clause, grouping data with GROUP BY, and ordering results with ORDER BY. This enables users to perform complex data analysis and reporting tasks using familiar SQL syntax.

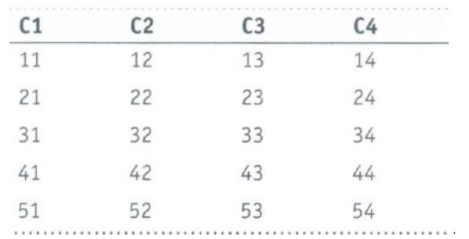
1. **Custom Types, Custom Functions can be defined.**

Hive allows users to define custom data types and functions. This is beneficial when dealing with specific data formats or when users need to perform operations that are not covered by the built-in functions. Custom types and functions enhance the extensibility and flexibility of Hive for handling diverse data scenarios.

1. **Explain RC file format used in HIVE.**

RCFile (Record Columnar File) is a columnar file format that stores data column-wise rather than row-wise, which can lead to better compression and query performance for certain workloads.

For example, consider a table which contains four columns as shown below:

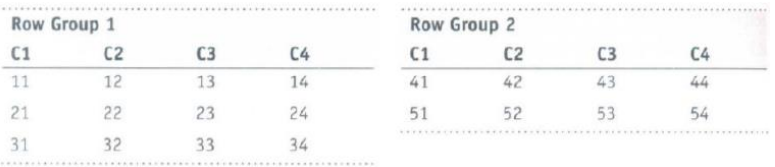


Instead of only partitioning the table horizontally like the row-oriented DBM5 (row-store),

RCFile partitions this table first horizontally and then vertically to serialize the data. Based

on the user-specified value, first the table is partitioned into multiple row groups horizontally

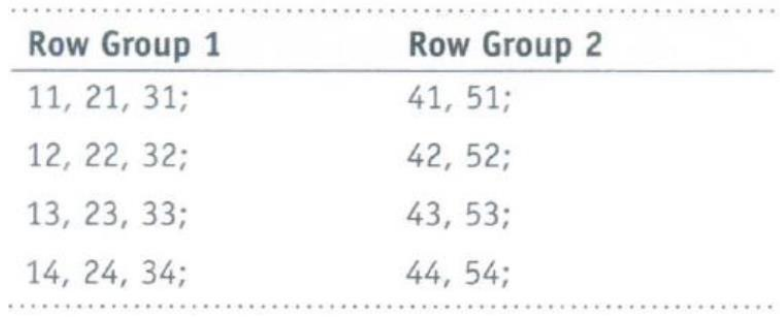
as shown below.



The first table is partitioned into two row groups by considering three rows as the size of each

row group. Next, in every row group RCFile partitions the data vertically like column-store.

So, the table will be serialized as shown below:



**Uses:**

* It is efficient for read-heavy workloads where you only need specific columns from a table, as it allows for column pruning.
* It is especially suitable for data warehousing and analytics use cases.
* It is a data placement structure that determines how to store relational tables on computer clusters.
* It provides a more space-efficient and query-optimized way to store structured data compared to traditional row-based storage formats.

**RCFILE IMPLEMENTATION:**

**Objective: To work with RC file implementation**

CREATE TABLE STUDENT\_RC (collno int, name string, gpa float) STORED AS RCFILE;

INSERT OVERWRITE table STUDENT\_RC SELECT \* FROM STUDENT;

SELECT SUM (gpa) FROM STUDENT\_RC;

1. **Explain the following in Hive:** **(a) SERDE (b) UDF**
2. **SERDE**

SERDE(Serializer/De-serializer) is a fundamental component that enables you to define how data is serialized (converted to a binary format) for storage and how it is deserialized (converted back to its original format) when retrieved from storage.

SERDEs are used for data serialization and deserialization within Hive tables, making it possible to work with different data formats and structures. Contains the logic to convert unstructured data into records. It is Implemented using Java.

De-serializer interface takes a binary representation or string of a record, converts it into a java

object that Hive can then manipulate. Serializer takes a java object that Hive has been

working with and translates it into something that Hive can write to HDFS.

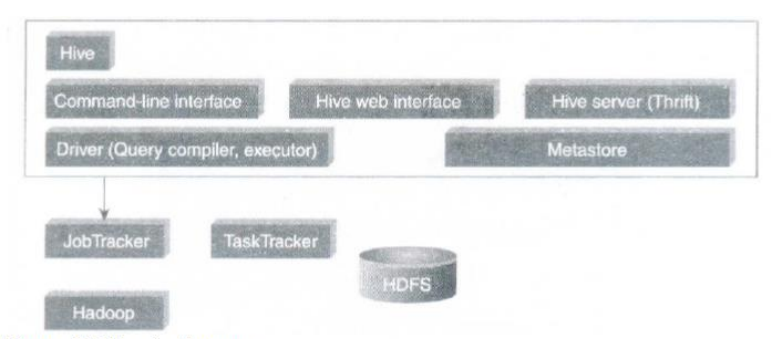
1. **UDF (User-Defined Function):**

User Defined Functions (UDFs) in Hive allow you to extend the functionality of Hive by defining your own custom functions.

These functions can be written in various programming languages, such as Java, Python, and others, and can be used to perform custom operations on the data within Hive queries.

UDFs are particularly useful when you need to apply complex or specialized logic to your data that is not achievable with Hive's built-in functions.

1. **Explain Hive architecture.**

****

1. **Hive Command-Line Interface (Hive CLI):** The most commonly used interface to interact with Hive.
2. **Hive Web Interface:** It is a simple Graphic User Interface to interact with Hive and to execute query.
3. **Hive Server:** This is an optional server. This can be used to submit Hive Jobs from a remote client.
4. **JDBC-ODBC:** Jobs can be submitted from a JDBC Client. One can write a Java code to connect to Hive and submit jobs on it.
5. **Driver:** Hive queries are sent to the driver for compilation, optimization and execution.
6. **Metastore:** Hive table definitions and mappings to the data are stored in a Metastore. A Metastore consists of the following:

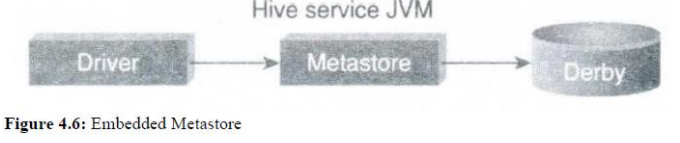
* **Metastore service**: Offers interface to the Hive.
* **Database:** Stores data definitions, mappings to the data and others.

The metadata which is stored in the metastore includes IDs of Database, IDs of Tables, IDs of Indexes, etc. The time of creation of a Table, the Input and Output Format used for a Table, etc. The metastore is updated whenever a table is created or deleted from Hive.

**There are three kinds of metastore.**

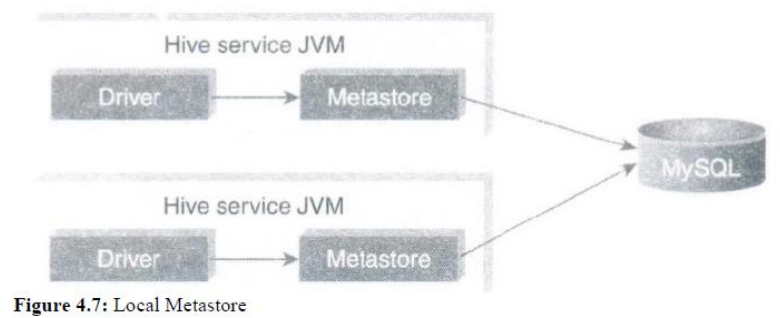
1. **Embedded Metastore:**

This metastore is mainly used for unit tests. Here, only one process is allowed to connect to the metastore at a time. This is the default metastore for Hive. It is Apache Derby Database. In this metastore, both the database and the metastore service runs, embedded in the main Hive Server process.



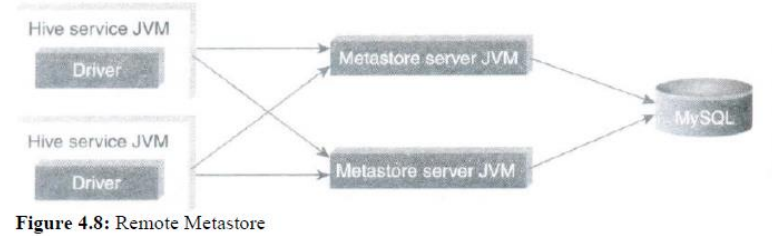
1. **Local Metastore:**

Metadata can be stored in any RDBMS component like MySQL. Local metastore allows multiple connections at a time. In this mode, the Hive metastore service runs in the main Hive Server process, but the metastore database runs in a separate process, and can be on a separate host.



1. **Remote Metastore:**

In this, the Hive driver and the metastore interface run on different JVMs (which can run on different machines as well). This way the database can be fire-walled from the Hive user and also database credentials are completely isolated from the users of Hive.



1. **What is Pig? Explain the features Pig.**

Pig is an open-source high-level platform for processing and analyzing large datasets in the Hadoop ecosystem. Apache Pig is a platform for data analysis. It is an alternative to Map Reduce Programming. Pig was developed as a research project at Yahoo.

The main components of Pig are as follows:

* Data flow language (Pig Latin).
* Interactive shell where you can type Pig Latin statements (Grunt).
* Pig interpreter and execution engine.

**Key Features of PIG**

1. **It provides an engine for executing data flows (how your data should flow).**

Pig runs on Hadoop and it processes data in parallel on the Hadoop cluster. Pig uses both Hadoop Distributed File System and MapReduce Programming. By default, Pig reads input files from HDFS. Pig stores the intermediate data (data produced by MapReduce jobs) and the output in HDFS. Pig can also read input from and place output to other sources.

1. **It provides a language called "Pig Latin" to express data flows.**

Pig Latin statement is an operator. Pig Latin statements include schemas and expressions to process data. Pig Latin statements should end with a semi-colon.

Pig Latin Statements are generally ordered as follows:

* LOAD statement that reads data from the file system.
* Series of statements or perform transformations.
* DUMP or STORE to display/store result.

1. **Operators:**

Latin contains operators for many of the traditional data operations such as join, filter, sort, etc.

**Join** is used to two or more relations based on values in the common filed. It always performs inner join.

**Filter** operator is used to select tuples from a relation based on specified conditions.

**Order by** is used to sort relation based on the specific value.

1. **User Defined Functions:**

Pig supports the development of User-Defined Functions (UDFs) in Java, Python, and other languages, allowing users to create custom functions to process and analyze data.

1. **Interactive Mode:**

Pig offers an interactive mode, known as Grunt shell, where users can write and execute Pig Latin scripts interactively, making it suitable for ad-hoc data analysis and exploration.

1. **Simplicity:**

Pig Latin is a simple and easy-to-understand scripting language that abstracts the complexities of MapReduce programming.

1. **Multi-language Support:**

Pig supports multiple scripting languages, enabling developers and data analysts to write scripts in their language of choice. Pig Latin is the primary language, but other languages like Python and JavaScript can also be used.