Hive interview Questions

1. What is the definition of Hive? What is the present version of Hive?

Hive is an SQL Based tool that builds over Hadoop to process the data for analytical purpose. Present version: 4.0. 0-alpha-2

2. Is Hive suitable to be used for OLTP systems? Why?

Hive doesn't support OLTP. Hive supports Online Analytical Processing (OLAP) only because Hive does not provide insert and update option at row level.

3. How is HIVE different from RDBMS? Does hive support ACID transactions. If not then give the proper reason.

Hive is a data warehouse and not relational database.
1. **Data Model**: Hive follows a schema-on-read approach, while RDBMS follows a schema-on-write approach.
Hive stores data in HDFS and uses the data schema only when read operation has to execute (to read from HDFS) whereas RDBMS has to use data schema to store the data before storing.

Hive does not natively support ACID (Atomicity, Consistency, Isolation, Durability) transactions because it was designed for batch processing and large-scale data analysis, and its underlying storage layer, HDFS, is not optimized for transactional operations.

However, ACID capabilities can be achieved in Hive through the use of ACID tables with alternative storage engines like HBase.

4. Explain the hive architecture and the different components of a Hive architecture?

1. User Interface (UI): Provides an interface for users to interact with Hive, such as Hive CLI (Command Line Interface) or Hive Web UI. Users can submit queries, manage tables, and access query results through the UI.
2. Metastore: Stores metadata about tables, partitions, columns, and other objects in Hive. It contains information about the schema, data types, and storage location of the tables. The Metastore can use various databases like MySQL, Derby, or PostgreSQL to store metadata.
3. Query Compiler and Optimizer: Translates HiveQL (Hive Query Language) queries into a series of MapReduce or Tez jobs that can be executed on a Hadoop cluster. It applies query optimization techniques to improve

performance by generating efficient execution plans.
4. Execution Engine: Executes the query plan generated by the Query Compiler and Optimizer. Hive supports multiple execution engines, including MapReduce, Tez, and Spark. These engines distribute the query tasks across the cluster and handle the actual processing of data.
5. Hadoop Distributed File System (HDFS): The underlying distributed file system used by Hive for storing and accessing data. Hive tables are typically stored as files in HDFS, allowing for scalable and fault-tolerant data storage.
6. SerDe (Serializer/Deserializer): Handles the serialization and deserialization of data between Hive and external data formats. SerDe allows Hive to work with various file formats like CSV, JSON, Parquet, and more. It defines how data is structured and encoded in the tables.
7. Storage Handlers: Provide integration with external storage systems beyond HDFS. Hive supports storage handlers for different databases and data warehouses, enabling access to data stored in other systems like Apache HBase or Amazon S3.

In summary, Hive architecture includes components such as the User Interface, Metastore, Query Compiler and Optimizer, Execution Engine, HDFS, SerDe, and Storage Handlers. These components work together to enable data querying, processing, and management in Hive using a high-level SQL-like language called HiveQL.

5. Mention what Hive query processor does? And Mention what are the components of a Hive query processor?

6. What are the three different modes in which we can operate Hive?

- Embedded Metastore.
- Local Metastore.
- Remote Metastore.

7. Features and Limitations of Hive.

Features of Hive:

1. Data Warehousing Capabilities: Hive is suitable for data warehousing scenarios, as it provides features like partitioning, bucketing, and indexing to optimize data storage and query performance.
2. Integration with Hadoop Ecosystem: Hive integrates well with other components of the Hadoop ecosystem, such as HDFS (Hadoop Distributed File System), YARN (Yet Another Resource Negotiator), and

Apache Spark, enabling seamless data processing and analysis.
3. High-level SQL-like Language

Limitations of Hive:

1. Batch Processing: Hive is primarily designed for batch processing and may not provide real-time or interactive query response times, as it relies on MapReduce or other execution engines that have inherent latency.
2. ACID Transactions: Hive does not provide built-in support for ACID (Atomicity, Consistency, Isolation, Durability) transactions, which makes it less suitable for use cases that require strict data consistency and transactional guarantees.
3. Data Latency: Due to the underlying distributed processing framework, Hive may have higher data latency compared to traditional relational databases.

8. How to create a Database in HIVE?

"CREATE DATABASE mydatabase;"

First, create a data schema as per the data requirement. Then, load the information into it by usind "LOAD INTO" command in Hive Terminal.

9. How to create a table in HIVE?

```
create table my_table
(
roll_no int,
gender varchar(10)
PRIMARY KEY(roll_no)
)
row format delimited
partition by ‘,’
stored as PARQUET;
```

10.What do you mean by describe and describe extended and describe formatted with respect to database and table

1. DESCRIBE database_name:
    - It displays the list of tables within the specified database.

2. DESCRIBE table_name:
    - It shows the column names and their data types of the specified table.

3. DESCRIBE EXTENDED table_name:
    - It provides additional information about the specified table, including column statistics, location, input/output formats, and storage properties.

4. DESCRIBE FORMATTED table_name:
    - It displays a detailed description of the specified table, including the table properties, column statistics, partitions, and storage information.

11.How to skip header rows from a table in Hive?

```
create table my_table
(
roll_no int,
gender varchar(10)
PRIMARY KEY(roll_no)
)
row format delimited
partition by ‘,’
stored as PARQUET;
TBLPROPERTIES ("skip.header.line.count"="1")
```

12.What is a hive operator? What are the different types of hive operators?

In Hive, operators are symbols or keywords used to perform various operations on data stored in tables.

1. Arithmetic Operators:
    - Arithmetic operators perform mathematical calculations on numeric data.
    - Examples: + (addition), - (subtraction), * (multiplication), / (division), % (modulus).
2. Comparison Operators:
    - Comparison operators are used to compare two values and evaluate conditions.
    - Examples: = (equal to), != (not equal to), < (less than), > (greater than), <= (less than or equal to), >= (greater than or equal to).
3. Logical Operators:
    - Logical operators are used to combine conditions and perform logical operations.
    - Examples: AND, OR, NOT

13.Explain about the Hive Built-In Functions

1. String Functions:
    - String functions allow manipulation and analysis of string data. Examples include LENGTH, SUBSTR, CONCAT, UPPER, LOWER, TRIM, REPLACE, SPLIT, and more.
2. Date and Time Functions:
    - Hive supports various functions for handling date and time values, such as CURRENT_DATE, CURRENT_TIMESTAMP, TO_DATE, YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, DATEDIFF, DATE_ADD, DATE_SUB, and more.
3. Conditional Functions:

Hive offers functions to handle conditional expressions, such as CASE, WHEN, IF,

14. Write hive DDL and DML commands.

DDL Commands:

1. CREATE DATABASE database_name;
2. CREATE TABLE table_name (
   column1 datatype,
   column2 datatype,
   ...)
3. Alter Table

DML Commands:

1. Insert Into Table
2. SELECT column1, column2 from Table
3. Update and Delete (require ACID mode)

15.Explain about SORT BY, ORDER BY, DISTRIBUTE BY and CLUSTER BY in Hive.

16.Difference between "Internal Table" and "External Table" and Mention when to choose “Internal Table” and “External Table” in Hive?

Choose internal tables when Hive has full control over the data, and you want Hive to manage the data lifecycle.

- Choose external tables when the data is managed by external processes or tools, and you want to access and analyze the data using Hive.
- External tables are beneficial when the data needs to be shared between multiple systems or when you want to preserve the data even if the table metadata is deleted.

Internal Table:
The data for an internal table is managed by Hive, which means Hive assumes full control over the data and its lifecycle.
When you drop an internal table, both the table metadata and the associated data stored in the warehouse directory are deleted.

External Table:

- An external table in Hive is a table that refers to data stored outside the Hive warehouse directory.
- The data for an external table can be located in various locations such as HDFS, S3, or any other file system accessible by Hive.
- When you drop an external table, only the table metadata is deleted, and the underlying data remains intact in its original location.

17.Where does the data of a Hive table get stored?

If mentioned Internal before the table, then it will get stored in Hive datawarehouse and if mentioned External before the table, then it will get stored in the HDFS location.

18.Is it possible to change the default location of a managed table?

No, it is not possible to change the default location of a managed table in Hive. The default location for managed tables is determined by the Hive configuration property **hive.metastore.warehouse.dir**, which specifies the default root directory for storing table data in the Hive data warehouse. This location is set during the initialization of Hive and is typically not meant to be changed.

19.What is a metastore in Hive? What is the default database provided by Apache Hive for metastore?

The metastore in Hive is a central repository that stores the metadata information for Hive tables, such as the table structure, partition information, column details. The metastore stores the metadata in a relational database management system (RDBMS) such as MySQL, PostgreSQL.

The default database provided by Apache Hive for the metastore is called "**metastore_db**".

20.Why does Hive not store metadata information in HDFS?

Hive does not store metadata information in HDFS (Hadoop Distributed File System) because HDFS is designed for storing and managing large amounts of

data in a distributed manner, optimized for high-throughput data access. Metadata, on the other hand, typically consists of smaller pieces of information about the data, such as table structures, column names, data types, and partition information.

Storing metadata in HDFS would result in unnecessary duplication and replication of metadata across the distributed file system, which can impact performance and increase storage requirements. Additionally, HDFS is optimized for data storage and retrieval, not for efficient metadata management and querying.

21.What is a partition in Hive? And Why do we perform partitioning in Hive?

In Hive, a partition is a way to divide a table into smaller, more manageable parts based on one or more columns.
Each partition represents a subset of the data with a specific value or range of values in the partitioning column(s).

reasons why we perform partitioning in Hive:

- Improved Query Performance
- Enhanced Data Filtering
- Simplified Data Management

22.What is the difference between dynamic partitioning and static partitioning?
23.How do you check if a particular partition exists?
24.How can you stop a partition form being queried?
25.Why do we need buckets? How Hive distributes the rows into buckets?
26.In Hive, how can you enable buckets?
27.How does bucketing help in the faster execution of queries?
28.How to optimise Hive Performance? Explain in very detail.
29. What is the use of Hcatalog?
30. Explain about the different types of join in Hive.
31.Is it possible to create a Cartesian join between 2 tables, using Hive?
32.Explain the SMB Join in Hive?
33.What is the difference between order by and sort by which one we should use?
34.What is the usefulness of the DISTRIBUTED BY clause in Hive?
35.How does data transfer happen from HDFS to Hive?
36.Wherever (Different Directory) I run the hive query, it creates a new metastore_db, please explain the reason for it?
37.What will happen in case you have not issued the command: ‘SET hive.enforce.bucketing=true;’ before bucketing a table in Hive?
38.Can a table be renamed in Hive?

39.Write a query to insert a new column(new_col INT) into a hive table at a position before an existing column (x_col)
40.What is serde operation in HIVE?
41.Explain how Hive Deserializes and serialises the data?
42.Write the name of the built-in serde in hive.
43.What is the need of custom Serde?
44.Can you write the name of a complex data type(collection data types) in Hive?
45.Can hive queries be executed from script files? How?
46.What are the default record and field delimiter used for hive text files?
47.How do you list all databases in Hive whose name starts with s?
48.What is the difference between LIKE and RLIKE operators in Hive?
49.How to change the column data type in Hive?
50.How will you convert the string '51.2' to a float value in the particular column?
51.What will be the result when you cast 'abc' (string) as INT?
52.What does the following query do?
a. INSERT OVERWRITE TABLE employees
b. PARTITION (country, state)
c. SELECT ..., se.cnty, se.st
d. FROM staged_employees se;
53.Write a query where you can overwrite data in a new table from the existing table.
54.What is the maximum size of a string data type supported by Hive? Explain how Hive supports binary formats.
55. What File Formats and Applications Does Hive Support?
56.How do ORC format tables help Hive to enhance its performance?
57.How can Hive avoid mapreduce while processing the query?
58.What is view and indexing in hive?
59.Can the name of a view be the same as the name of a hive table?
60.What types of costs are associated in creating indexes on hive tables?
61.Give the command to see the indexes on a table.
62. Explain the process to access subdirectories recursively in Hive queries.
63.If you run a select * query in Hive, why doesn't it run MapReduce?
64.What are the uses of Hive Explode?
65. What is the available mechanism for connecting applications when we run Hive as a server?
66.Can the default location of a managed table be changed in Hive?
67.What is the Hive ObjectInspector function?
68.What is UDF in Hive?
69.Write a query to extract data from hdfs to hive.
70.What is TextInputFormat and SequenceFileInputFormat in hive.
71.How can you prevent a large job from running for a long time in a hive?
72.When do we use explode in Hive?

73.Can Hive process any type of data formats? Why? Explain in very detail
74.Whenever we run a Hive query, a new metastore_db is created. Why?
75.Can we change the data type of a column in a hive table? Write a complete query.
76.While loading data into a hive table using the LOAD DATA clause, how do you specify it is a hdfs file and not a local file ?
77.What is the precedence order in Hive configuration?
78.Which interface is used for accessing the Hive metastore?
79.Is it possible to compress json in the Hive external table ?
80.What is the difference between local and remote metastores?
81.What is the purpose of archiving tables in Hive?
82.What is DBPROPERTY in Hive?
83.Differentiate between local mode and MapReduce mode in Hive.