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

-Hive is a data warehouse system built on top of Hadoop for querying and analyzing large datasets.The present version of Hive is 4.0.0

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

-Hive is not suitable for OLTP systems because it is designed for batch processing of large datasets rather than real-time transaction processing.

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

-Hive is different from RDBMS because it is a schema-on-read system, while RDBMS is a schema-on-write system. It does not support ACID property beacuase it cannot guarantee that all changes to the data will be atomic, consistent, isolated, and durable.

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

- mainly there are three component

a.Hive client - to connect with JDBC or ODBC connection

b.Hive services - it compiles,optimese and using metastore convert the query into comaptible languaage

c.Excecution engine - it excecute the code

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

-it performs several functions, including parsing HiveQL queries, generating a logical plan, optimizing the plan. components are Parser, Semantic Analyzer, Query Planner .

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

Local Mode: where Hive runs in a single JVM on the local machine.

MapReduce Mode: where Hive uses MapReduce to execute queries on a Hadoop cluster.

Spark Mode: where Hive uses Apache Spark to execute queries on a Spark cluster.

**7. Features and Limitations of Hive.**

-Features of Hive include support for SQL-like queries, scalability, fault-tolerance, extensibility, and integration with Hadoop.

-Limitations of Hive include its lack of support for real-time processing

**8. How to create a Database in HIVE?**

-CREATE DATABASE mydatabase;

**9. How to create a table in HIVE?**

-CREATE TABLE mytable (

column1 INT,

column2 STRING,

column3 DOUBLE

);

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

-DESCRIBE database\_name: returns information about the specified database.

-DESCRIBE table\_name: returns information about the specified table.

-DESCRIBE FORMATTED table\_name: returns detailed information about the specified table, including file format, partitioning scheme, and storage properties.

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

-CREATE TABLE my\_table (

col1 STRING,

col2 INT

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

TBLPROPERTIES ('skip.header.line.count'='1');

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

-Relational Operators: Used for querying data, such as SELECT, JOIN, UNION, etc.

-Logical Operators: Used to define the logical flow of operations, such as AND, OR, NOT, etc.

-Arithmetic Operators: Used for mathematical calculations, such as +, -, \*, /, etc.

-Conditional Operators: Used for conditional operations, such as IF, CASE, WHEN, etc.

-Transformation Operators: Used for transforming data, such as CAST, COALESCE, CONCAT, etc.

-Aggregate Operators: Used for aggregating data, such as SUM, COUNT, AVG, etc.

**13.Explain about the Hive Built-In Functions**

These functions can be used in Hive queries to manipulate and transform data. Hive provides a wide range of built-in functions, including string functions, mathematical functions, date functions, conditional functions, aggregate functions, and more.

**14. Write hive DDL and DML commands.**

-DDL commands:

CREATE DATABASE: Creates a new database in Hive.

CREATE TABLE: Creates a new table in Hive.

ALTER TABLE: Modifies the structure of an existing table.

DROP TABLE: Deletes a table from Hive.

-DML commands:

SELECT: Retrieves data from one or more tables.

INSERT INTO: Inserts data into a table.

UPDATE: Updates existing records in a table.

DELETE: Deletes records from a table.

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

*-SORT BY*: Orders the rows of the result based on the specified column(s). It does not guarantee total order across reducers.

*-ORDER BY*: Orders the rows of the result based on the specified column(s). It provides a total order across reducers.

*-DISTRIBUTE BY:* Determines how the rows are distributed to different reducers based on the specified column(s).

*- CLUSTER BY* keyword is used to organize data in a table into clusters based on the values of one or more columns.

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

*-Internal Table:* An internal table in Hive manages its data and metadata within Hive's warehouse directory. Hive assumes full control over the table's data, including its lifecycle and storage.

If an internal table is dropped, both the metadata and data associated with the table are deleted. Internal tables are the default table type in Hive.

*-External Table:* An external table in Hive refers to data that is stored outside of Hive's warehouse directory. The data remains in its original location, and Hive only manages the metadata associated with the table. If an external table is dropped, only the metadata is deleted, while the data remains intact.

External tables provide more flexibility in terms of data management and can be shared across multiple Hive instances or other tools.

When to choose:

*-Internal Table:* Choose an internal table when you want Hive to have full control over the table's data, and the data is solely managed by Hive. This is suitable when you want Hive to handle the lifecycle and storage management of the data.

*-External Table:* Choose an external table when the data already exists outside of Hive, and you want Hive to manage only the metadata.

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

-The data of a Hive table gets stored in Hadoop Distributed File System (HDFS) by default.

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

-Yes, it is possible to change the default location of a managed table in Hive.

**19.What is a metastore in Hive? What is the default database provided byApache Hive for metastore?**

-In Hive, a metastore is a central repository that stores metadata information about Hive tables, partitions, and schemas.

The default metastore database provided by Apache Hive is Derby.

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

-Hive does not store metadata information in HDFS because HDFS is designed for storing large files and is optimized for data storage and retrieval,

while metadata management is better suited for a database system like Apache Derby or Apache MySQL.

**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 a specific column's value.

Partitioning in Hive improves query performance by allowing the system to scan only relevant partitions instead of the entire table.

**22.What is the difference between dynamic partitioning and static partitioning?**

-The main difference between dynamic partitioning and static partitioning in Hive is how the partitions are created. In dynamic partitioning, partitions are automatically created by Hive based on the data being loaded into the table.

-In static partitioning, the partitions need to be explicitly defined and created by the user before loading the data.

**23.How do you check if a particular partition exists?**

-can use the "SHOW PARTITIONS"

**24.How can you stop a partition form being queried?**

-To prevent a specific partition from being queried in Hive, you can use the "ALTER TABLE" statement with the "ARCHIVE PARTITION" command.

This command moves the data of the partition to the HDFS trash directory, making it inaccessible for queries.

**25.Why do we need buckets? How Hive distributes the rows into buckets?**

-Buckets in Hive are used for data organization and faster querying. They divide the data within a partition into smaller, equally sized files based on a hash function applied to a chosen column.

Hive distributes rows into buckets by assigning each row to a specific bucket based on the hash value of the designated column.

**26.In Hive, how can you enable buckets?**

-This can be done using the "CLUSTERED BY" and "SORTED BY" clauses in the "CREATE TABLE" statement.

**27.How does bucketing help in the faster execution of queries?**

-Bucketing helps in faster query execution in Hive by reducing the amount of data that needs to be scanned. When a query is executed on a bucketed table, Hive can determine which specific buckets need to be read based on the filtering conditions.

This eliminates the need to scan the entire table, resulting in improved performance.

**28.How to optimise Hive Performance? Explain in very detail.**

-Partitioning the data appropriately to reduce the amount of data scanned.

-Using bucketing to distribute the data evenly and enable efficient querying.

-Applying appropriate compression techniques to reduce storage and I/O.

-Tuning Hive configuration settings, such as memory allocation and parallelism.

**29. What is the use of Hcatalog?**

HCatalog is a component of Apache Hive that provides a metadata and table management layer.

It allows different tools and frameworks, such as Pig and MapReduce, to interact with Hive's metadata and access data stored in Hive tables

**30. Explain about the different types of join in Hive.**

-*Inner join:* Returns only the matching records from both tables.

-*Left join:* Returns all records from the left table and the matching records from the right table.

-*Right join:* Returns all records from the right table and the matching records from the left table.

-*Full outer join:* Returns all records when there is a match in either the left or right table.

**31.Is it possible to create a Cartesian join between 2 tables, using Hive?**

-No, Hive does not support Cartesian joins directly. Cartesian joins can be achieved by omitting the join condition or by using a cross join.

**32.Explain the SMB Join in Hive?**

-SMB (Sort-Merge-Bucketed) Join in Hive is a type of join where both tables being joined are bucketed and sorted on the join columns.

This type of join can significantly improve query performance by avoiding data shuffling and reducing the amount of data that needs to be processed.

**33.What is the difference between order by and sort by which one we should use?**

-In Hive, the "order by" clause is used to sort the entire result set based on specified columns. The "sort by" clause is used to sort the data within each reducer task. If a single reducer is used or the data is already sorted, both "order by" and "sort by" will produce the same result.

Otherwise, "sort by" is more efficient.

**34.What is the usefulness of the DISTRIBUTED BY clause in Hive?**

-The "DISTRIBUTED BY" clause in Hive is used to determine how the data should be distributed across the nodes in a cluster during a map-reduce job.

It helps in achieving data locality and can improve query performance by minimizing data movement during processing.

**35.How does data transfer happen from HDFS to Hive?**

-Data transfer from HDFS (Hadoop Distributed File System) to Hive typically involves two steps. First, the data files are loaded into HDFS using tools like Hadoop's hdfs command or Hadoop File System APIs. Then, Hive is used to create an external table that points to the data location in HDFS.

Hive uses metadata to access the data stored in HDFS and perform query operations.

**36.Wherever (Different Directory) I run the hive query, it creates a new metastore\_db, please explain the reason for it?**

-Each time you run a Hive query from a different directory, a new metastore\_db is created in that directory to maintain separate metadata for that specific instance or environment.

This ensures that the metadata remains isolated and independent for different instances of Hive.

**37.What will happen in case you have not issued the command: ‘SET hive.enforce.bucketing=true;’ before bucketing a table in Hive?**

-the bucketing feature will not be enforced.

This means that the table will still be created, but the data will not be properly organized into buckets based on the specified criteria.

**38.Can a table be renamed in Hive?**

-Yes, a table can be renamed in Hive using the RENAME TABLE command.

**39.Write a query to insert a new column(new\_col INT) into a hive table at aposition before an existing column (x\_col)**

-ALTER TABLE table\_name ADD COLUMNS (new\_col INT) BEFORE x\_col;

**40.What is serde operation in HIVE?**

-SerDe (Serializer/Deserializer) is a key component in Hive that facilitates the serialization and deserialization of data between Hive tables and the underlying storage formats.

It defines how the data is structured and how it is converted between its binary representation and a format that can be queried and processed.

**41.Explain how Hive Deserializes and serialises the data?**

-Hive deserializes data by reading the serialized data from the storage format and converting it into a structured format that can be queried. The deserialization process involves interpreting the metadata and schema of the data.

When serializing data, Hive converts the structured data into a binary format that can be stored efficiently in the chosen storage format.

**42.Write the name of the built-in serde in hive.**

-The built-in SerDe in Hive is called "LazySimpleSerDe."

**43.What is the need of custom Serde?**

-The need for a custom SerDe arises when the built-in SerDes in Hive do not meet specific requirements for data serialization and deserialization.

**44.Can you write the name of a complex data type(collection data types) inHive?**

-Array, Struct, Map, and Union are complex data types in Hive

**45.Can hive queries be executed from script files? How?**

-Yes, hive queries can be executed from script files using the command "hive -f script\_file\_name".

**46.What are the default record and field delimiter used for hive text files?**

-The default record delimiter used for Hive text files is '\n' (newline), and the default field delimiter is '\t' (tab).

**47.How do you list all databases in Hive whose name starts with s?**

-"SHOW DATABASES LIKE 's%'".

**48.What is the difference between LIKE and RLIKE operators in Hive?**

-The LIKE operator in Hive is used for pattern matching using wildcard characters ('%', '\_'),

while the RLIKE operator is used for pattern matching using regular expressions.

**49.How to change the column data type in Hive?**

-ALTER TABLE table\_name CHANGE COLUMN column\_name new\_column\_name new\_data\_type;

**50.How will you convert the string ’51.2’ to a float value in the particular column?**

-SELECT CAST('51.2' AS FLOAT) AS float\_column;

**51.What will be the result when you cast ‘abc’ (string) as INT?**

-the result will be NULL. Casting a non-numeric string to an integer will result in a null value in Hive

**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;

--The query performs an overwrite insert into the "employees" table, partitioned by "country" and "state",

selecting specific columns and data from the "staged\_employees" table.

**53.Write a query where you can overwrite data in a new table from the existing table.**

-INSERT OVERWRITE TABLE new\_table

SELECT \* FROM existing\_table;

**54.What is the maximum size of a string data type supported by Hive? Explain how Hive supports binary formats.**

-The maximum size of a string data type supported by Hive is 2^31 - 1 (2,147,483,647) characters. Hive uses the Java String data type, which has a similar maximum size limitation.

**55. What File Formats and Applications Does Hive Support?**

-File Formats: Hive supports popular file formats like TextFile, SequenceFile, RCFile, ORC (Optimized Row Columnar), and Parquet.

-Applications: Hive is commonly used alongside Apache Hadoop and Apache Spark. It integrates with Hadoop Distributed File System (HDFS) and can process data stored in HDFS or other compatible file systems.

Hive can also interact with external systems like Apache Kafka for data ingestion or Apache Tez for optimized query execution.

**56.How do ORC format tables help Hive to enhance its performance?**

-ORC format tables in Hive enhance performance by providing optimized storage and compression, column pruning, predicate pushdown, and efficient data encoding,

which reduce I/O and improve query execution speed.

**57.How can Hive avoid mapreduce while processing the query?**

-Hive can avoid MapReduce while processing a query by utilizing Tez or Spark as the execution engine. These engines provide in-memory processing, data caching, and more efficient query optimization,

resulting in faster query execution compared to MapReduce.

**58.What is view and indexing in hive?**

-In Hive, a view is a logical table that is based on the result of a query. It provides a way to create a virtual table with a predefined query, simplifying complex queries or providing data security.

-Indexing in Hive is a mechanism to improve query performance by creating an index on specific columns of a table, allowing faster data retrieval.

**59.Can the name of a view be the same as the name of a hive table?**

-Yes, the name of a view can be the same as the name of a Hive table.

**60.What types of costs are associated in creating indexes on hive tables?**

-Creating indexes on Hive tables involves two main costs: the initial cost of creating the index, which includes scanning and indexing the data, and the ongoing maintenance cost,

which includes updating the index whenever the underlying table is modified.

**61.Give the command to see the indexes on a table.**

-SHOW INDEXES ON table\_name;

**62. Explain the process to access subdirectories recursively in Hive queries.**

-SELECT \* FROM table\_name WHERE file\_path LIKE '/data/%';

**63.If you run a select \* query in Hive, why doesn't it run MapReduce?**

-it doesn't necessarily run MapReduce because Hive can leverage its metadata to retrieve the data directly from the underlying storage (e.g., HDFS)

without the need for MapReduce processing.

**64.What are the uses of Hive Explode?**

-Hive's Explode function is used to transform a column with an array or map type into multiple rows, with each element of the array or map occupying a separate row.

It is useful for unnesting complex data structures and performing further analysis or processing on the expanded data.

**65. What is the available mechanism for connecting applications when we run Hive as a server?**

-Hive provides Thrift and JDBC/ODBC as mechanisms for connecting applications when running Hive as a server.

**66.Can the default location of a managed table be changed in Hive?**

-Yes, the default location of a managed table can be changed in Hive by altering the table and specifying the new location using the LOCATION clause.

**67.What is the Hive ObjectInspector function?**

-The Hive ObjectInspector function is a utility in Hive that helps in inspecting and manipulating the internal representation of data objects.

It provides methods to access and interpret the data stored in different formats.

**68.What is UDF in Hive?**

-UDF in Hive stands for User-Defined Function. It is a mechanism provided by Hive to allow users to define their own functions to process data.

UDFs can be used to perform custom operations and transformations on data within Hive queries.

**69.Write a query to extract data from hdfs to hive.**

-LOAD DATA INPATH 'hdfs://<hdfs\_path>/input\_data.csv' INTO TABLE hive\_table;

**70.What is TextInputFormat and SequenceFileInputFormat in hive.**

--TextInputFormat is the default input format in Hive and it treats each line of a file as a separate record.

--SequenceFileInputFormat is an optimized input format in Hive that stores data in a binary format called SequenceFile.

It is suitable for large-scale data processing and provides efficient compression and serialization.

**71.How can you prevent a large job from running for a long time in a hive?**

-Adjust the number of reducers using hive.exec.reducers.max property to limit the parallelism of the job.

-Enable dynamic partitioning to optimize the query execution time by setting hive.exec.dynamic.partition.mode to nonstrict.

Use bucketing to distribute the data evenly across different files and improve performance.

-Tune the query parameters such as memory allocation, parallelism, and input/output formats based on the characteristics of the job.

**72.When do we use explode in Hive?**

- the explode function is used to split a column that contains an array or map into multiple rows,

with each element of the array or map occupying a separate row.

**73. Can Hive process any type of data formats? Why? Explain in very detail**

**-** Hive's architecture is designed to be flexible and extensible, enabling it to process a wide range of data formats. This adaptability is achieved through its support for built-in data formats, custom SerDes, input/output formats, storage handlers, and its inherent data serialization and deserialization capabilities. However, it's important to note that while Hive can process different data formats, the performance and efficiency of processing may vary based on the format and the specific use case.

*Built-in Data Formats:*Hive supports various built-in data formats such as TEXT, CSV, JSON, Parquet, ORC (Optimized Row Columnar), Avro, and more. These formats are designed to efficiently store and manage data in different ways, considering factors like compression, columnar storage, and schema evolution.

*Custom SerDes:*Hive allows you to define and use custom SerDes to process non-standard data formats. A SerDe is responsible for converting data between the internal binary representation used by Hive and the external format of the data. For instance, if you have data in a specific format not natively supported by Hive, you can develop a custom SerDe to read and write that data.

*Input and Output Formats:*Hive supports different Input and Output Formats, which determine how data is read from or written to external storage systems like HDFS. These formats play a role in processing data in various formats. For example, the TextInputFormat and TextOutputFormat are used for plain text files, while ParquetInputFormat and ParquetOutputFormat are used for the Parquet columnar format.

*Storage Handlers:*Storage Handlers extend Hive's capabilities by allowing it to interact with external storage systems. These handlers enable Hive to process data residing in different storage systems while providing a unified SQL-like interface. Examples include the HBaseStorageHandler for Apache HBase and the JdbcStorageHandler for relational databases.

*Data Serialization and Deserialization:*SerDes are crucial components for handling data serialization (conversion of data into a format suitable for storage) and deserialization (conversion of stored data back into its original format). Hive's SerDes enable it to accommodate different data formats by defining how data is transformed between its internal structure and the external format.

*Schema Evolution:*Some data formats, like Parquet and ORC, support schema evolution, allowing you to modify the schema of your data over time without requiring a full rewrite of the data. Hive leverages these capabilities to accommodate changes in data structure over time.

**74. Whenever we run a Hive query, a new metastore\_db is created. Why?**

**-** The metastore\_db in Hive is a persistent database that stores metadata related to tables, partitions, columns, and other objects needed for query optimization, execution, and management. It is not created anew with each query execution; rather, it is established during the Hive installation and maintained across different sessions and queries to ensure data consistency and efficient query processing.

**75. Can we change the data type of a column in a hive table? Write a complete query.**

**-** Yes we can change the data type of a column in a Hive table using the ALTER TABLE statement along with the CHANGE clause

-- Assume you have a table named "my\_table" with a column "old\_column" of type INT.

-- You want to change the data type of "old\_column" from INT to STRING.

-- Step 1: Create a backup of the original table, if needed (recommended for safety).

-- This step is optional but recommended to ensure data integrity in case something goes wrong.

CREATE TABLE my\_table\_backup AS SELECT \* FROM my\_table;

-- Step 2: Alter the table to change the data type of the column.

-- In this example, we are changing the data type of "old\_column" from INT to STRING.

-- Make sure to update the column name, original data type, and new data type accordingly.

ALTER TABLE my\_table

CHANGE COLUMN old\_column new\_column STRING;

-- Step 3: Update the new column with the data from the original column (optional).

-- This step is necessary only if you want to preserve the data from the original column.

-- If not needed, you can skip this step.

INSERT OVERWRITE TABLE my\_table

SELECT \*, CAST(new\_column AS INT) AS old\_column FROM my\_table;

-- Step 4: Drop the backup table (optional).

-- If you have created a backup in Step 1 and no longer need it, you can drop it.

DROP TABLE my\_table\_backup;

**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 ?**

**-**

When using the LOAD DATA statement in Hive to load data into a table, the location of the data source (whether it's a local file or an HDFS file) is determined by the path we provide in the statement. To specify that the source file is located in HDFS, we need to provide the full HDFS path in the LOAD DATA statement.

*LOAD DATA [LOCAL] INPATH 'hdfs\_path' [OVERWRITE] INTO TABLE table\_name [PARTITION (partition\_spec)];*

*LOAD DATA INPATH '/user/hadoop/input/datafile.txt' INTO TABLE my\_table;*

**77. What is the precedence order in Hive configuration?**

**-** In Hive, configuration settings are specified using Hive configuration properties. These properties determine the behavior and settings of the Hive environment. When multiple configuration properties conflict or overlap, Hive follows a precedence order to determine which value to use. The precedence order helps ensure that configurations are applied in a predictable manner.

The precedence order for Hive configuration is as follows:

*-Session and CLI-Level Properties*

*-HiveServer2 (HS2) Level Properties*

*-System-Level Properties*

*-Hive-Site.xml*

*-Hadoop Configuration*

*-Built-In Defaults*

**78.Which interface is used for accessing the Hive metastore?**

-The interface used for accessing the Hive metastore is called the Hive Metastore Thrift Service. It allows clients to communicate with the metastore using the Thrift protocol.

**79.Is it possible to compress json in the Hive external table ?**

-es, it is possible to compress JSON data in Hive external tables

**80.What is the difference between local and remote metastores?**

-In a local metastore, the metadata is stored on the same machine where Hive is installed.

-In a remote metastore, the metadata is stored on a separate machine, allowing multiple instances of Hive to share the same metastore.

**81.What is the purpose of archiving tables in Hive?**

-The purpose of archiving tables in Hive is to preserve the data and metadata of a table before performing any destructive operations like dropping or altering the table.

It allows users to revert to a previous state of the table if needed.

**83.Differentiate between local mode and MapReduce mode in Hive.**

-In Hive, local mode refers to running Hive queries on the local machine where Hive is installed,

using the local file system for data storage. MapReduce mode, on the other hand, utilizes the Hadoop MapReduce framework to distribute the query processing across a cluster of machines, providing scalability and fault tolerance.