Hive interview Questions

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

Ans:- Hive is a data warehouse open-source framework built on the top of hive. It is used to analyze structured data. It provides a SQL like interface(HQL) to query data stored in various databases and file systems. Hive 3.13 is the latest and stable version of hive.

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

Ans:- No, hive is not suitable to be used for OLTP system, because hive is not made to be used for frequent update and delete purpose. Hive framework is made to store once and read many times.

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

Ans:- Hive supports OLAP system while the RDBMS is an OLTP system. Hive is made to write once and read many times. RDBMS deals with fixed schema while hive supports both fixed and dynamic schema.

Acid transactions are not supported in hive because hive is used for OLAP system and only support insert/Delete command but not the update command.

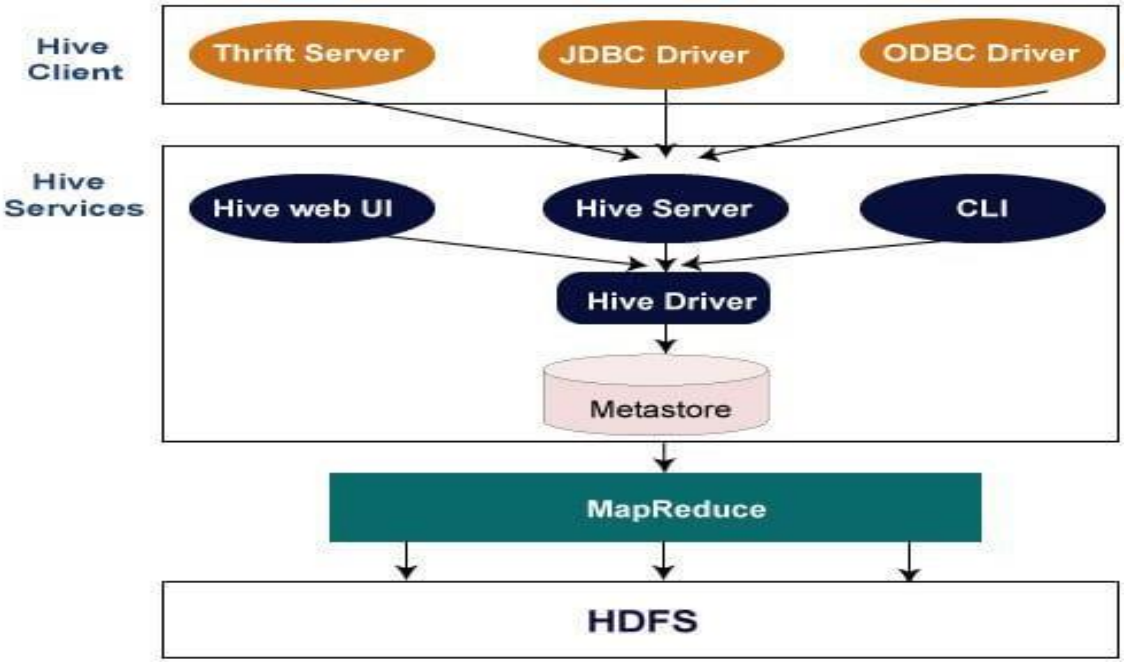
However, from hive version 0.14, we can create a transactional table that stores the ORC file format and this will support the ACID transaction.(we need to manually enable the acid property).

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

Ans:-

Hive client:- Hive allows writing applications in various languages(Java, Python, Scala).

It supports different types of clients such as:-Trift server, JDBC Driver and ODBC Driver. This enables all the requests from various clients.



Hive Services:-

1. Hive CLI/web UI - The Hive CLI (Command Line Interface) is a shell where we can execute Hive queries and commands. Hive web UI is a web UI similar to hive CLI. We can also use 3rd party platform like putty in place of CLI.

2. Hive Metastore:- it stores the the information of location, metadata, partitions , serialization and deserialization for various tables In warehouse.

3. Apache server:- It is referred to as Apache Thrift Server. It accepts the request from different clients and provides it to Hive Driver.

4. Hive Driver - It receives queries from different sources like web UI, CLI, Thrift, and JDBC/ODBC driver. It transfers the queries to the compiler.

5. Hive Compiler - It performs the query and sementic checks, It converts HiveQL statements into MapReduce jobs.

6. Hive Execution Engine- It generates the logical plan and execute it in the form of DAG of map-reduce tasks and HDFS tasks.

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

Ans:- Hive query processor convert graph of MapReduce jobs with the execution time framework. So that the jobs can be executed in the order of dependencies

Following are the components of a Hive Query Processor:-

- Parse and Semantic Analysis (ql/parse)
- Metadata Layer (ql/metadata)
- Type Interfaces (ql/typeinfo)
- Sessions (ql/session)
- Map/Reduce Execution Engine (ql/exec)
- Plan Components (ql/plan)
- Hive Function Framework (ql/udf)
- Tools (ql/tools)
- Optimizer (ql/optimizer)

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

Ans:- Standalone Mode. Pseudo-distributed Mode. Fully-Distributed Mode.

7. Features and Limitations of Hive.

Ans:- Features:-

1. Open Source
2. Support HQL
3. Support multiple users query execution
4. Partitioning and Bucketing
5. Support File format like:- ORC, Avro, Parquet etc.
6. Support Olap warehouse,
7. Support HDFS.

Limitation:-

1. Does not support OLTP.
2. Don’t support real time queries
3. Limited subqueries support
4. Hive execution is slow and with high latency,

8. How to create a Database in HIVE?

Ans:- log in to hive cli and write the command:-

create database if not exists demo;

9. How to create a table in HIVE?

Ans:-

CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.] table_name

[(col_name data_type [COMMENT col_comment], ...)]
[COMMENT table_comment]
[ROW FORMAT row_format]
[STORED AS file_format]

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

Ans:-

Describe Formatted:-

It displays weather table is internal or external, when it was created, the file format, the location of the data in HDFS, FORMATTED is specified, it show/displays the metadata in a tabular format.

Describe:-

The DESCRIBE output for a database includes the location and the comment.

Extended:-

If the EXTENDED is specified, it show/displays all the metadata for the specified table.

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

Ans:- tblproperties ("skip.header.line.count"="1")

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

Ans:-

There are four types of operators in Hive:

- Relational Operators
- Arithmetic Operators

- Logical Operators
- Complex Operators

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

Ans:- To perform some specific mathematical and arithmetic operations, Hive have some built-in functions. These built-in functions extract data from tables in hive and process the calculations.

Eg:- Mathematical function, Collection function, Type conversion function, Date function, Conditional function, and String function.

14. Write hive DDL and DML commands.

Ans:- Hive DDL commands are:

1. CREATE
2. SHOW
3. DESCRIBE
4. USE
5. DROP
6. ALTER
7. TRUNCATE

The various Hive DML commands are:

1. LOAD
2. SELECT
3. INSERT
4. DELETE
5. UPDATE
6. EXPORT
7. IMPORT

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

Ans:- sort by:-

- It can use multiple reducers for final output.
- Only guarantees ordering of rows within a reducer, ie. It orders the data per reducer locally.

Order by:-

- Uses single reducer to guarantee total order in output. ie. Even if we increase the reducer it will only use a single reducer.
- It sort the data per a single reducer globally.
- LIMIT can be used to minimize sort time.

Distribute by:-

- It distribute the input rows among reducers. It ensures that all rows for the same key columns are going to the same reducer.
- It does not sort the data globally.
- No overlapping of data.

Cluster by:-

- CLUSTER BY clause is equivalent to the output of DISTRIBUTE BY + SORT BY clauses.
- The CLUSTER BY clause distributes the data based on the key column and then sorts the output data by putting
- The same key column values adjacent to each other.
- The output of the CLUSTER BY clause is sorted at the reducer level**(Locally).**

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

Ans:- Internal Table:-

- It is a default table in hive.
- Whenenver we create a hive table without mentioning the keyword external, a internal(managed) table gets created.
- The default location of the internal table is /user/hive/warehouse
- We can change the default location of the internal table by specifying into location keyword.
- If we delete the internal table table data and the metadata associated with that table will be deleted from the HDFS.

External Table:-

Hive does not manage the data of the External table.

- We create an external table for external use as when we want to use the data outside the Hive.
- We can create the external table by specifying the EXTERNAL keyword.
- They can access data stored in sources such as remote HDFS locations.
- Whenever we drop the external table, then only the metadata associated with the table will get deleted, the table data remains untouched by Hive.

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

Ans:- /user/hive/warehouse

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

Ans:- yes, we can specify the new location by specifying the directory into the location clause – LOCATION '<hdfs_path>'.

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

Ans:- Metastore is nothing but a traditional RDBMS in the hive,
When we create a new Hive table, the information related to the schema (column names, data types, location) is stored in the Hive metastore relational database.
Derby is a default metastore.

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

Ans:- To achieve high performance, speed and low latency as HDFS read/write operations are time consuming processes.

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

Ans:- Dividing the entire big table into some parts based on the values of a single or multiple columns like date, course, city or country.
To restrict the hive to scan the entire table while scan the particular partition increase the query performance time.

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

Ans:- Static Partition:- whenever we have the info. of distinct values of a columns on which the partitioning is being applied or whenever we want make a partition of a table based on few values of column where partition is being applied we use Static partition.

For static partition we need to specify the partition column value in each and every LOAD statement.

Dynamic Partition:- whenever we don't have the idea of the no. of distinct values of the column where the partition is being applied we try to use dynamic partition.

For dynamic partition we need to specify the partition column value in each and every LOAD statement.

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

Ans:-

SHOW PARTITIONS table_name
PARTITION(partitioned_column='partition_value')

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

Ans:-

We can use the ENABLE OFFLINE clause with ALTERTABLE statement.

Syntax:-

ALTER TABLE t1 PARTITION (PARTITION_SPEC)
ENABLE OFFLINE;

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

Ans:- Bucketing in Hive is used to improve performance by eliminating table scans when dealing with a large set of data on a Hadoop file system (HDFS).

Bucketing is a technique to split the data into more manageable files, (By specifying the number of buckets to create). The value of the bucketing column will be hashed by a user-defined number into buckets.

26.In Hive, how can you enable buckets?

Ans:- set.hive.enforce.bucketing=true;

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

Ans:- In bucketing, the partitions can be subdivided into buckets based on the hash function of a column. It gives us a sorted extra structure to the data which can be used for more efficient queries.

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

Ans:-

a. using Tez-Execution Engine in Hive:-

Tez, it is a new application framework built on Hadoop Yarn**.** That executes complex-DAG of general data processing tasks. However, we can consider it to be a much more flexible and powerful successor to the map-reduce framework.
It can work with petabytes of data over thousands of nodes it allows those data access applications.

b. By Using the Suitable File Format in Hive:-

we can use appropriate file format on the basis of data. It will drastically increase our query performance. or increasing your query performance ORC file format is best suitable because it can can store data in an optimized way than the other file formats. ORC reduces the size of the original data up to 75%.

c. Hive Partitioning:-

by Partitioning all the entries for the various columns of the dataset are segregated and stored in their respective partition.
Hence, While we write the query to fetch the values from the table, only the required partitions of the table are queried. Thus it reduces the time taken by the query to yield the result.

d. Hive Bucketing:-

Hive offers Bucketing concept. Basically, that allows the user to divide table data sets into more manageable parts.
Hence, to maintain parts that are more manageable we can use Bucketing. Through it, the user can set the size of the manageable parts or Buckets too.

e. Hive Vectorisation:-

Hive Optimization Techniques, to improve the performance of operations we use Vectorized query execution. Here operations refer to scans, aggregations, filters, and joins.

It significantly improves query execution time, and is easily enabled with two parameters settings:-
set hive.vectorized.execution = true

set hive.vectorized.execution.enabled = true

f. using Parallel execution at a Mapper & Reducer level

We can improve the performance of aggregations, filters, and joins of our hive queries by using vectorized query execution, which means scanning them in batches of 1024 rows at once instead of single row each time.

We should explore the below parameters which will help to bring in more parallelism and which significantly improves query execution time:

set hive.vectorized.execution.enabled=true; set hive.exec.parallel=true;

29. What is the use of Hcatalog?

Ans:-

HCatalog is a tool that allows you to access Hive metastore tables within Pig, Spark SQL, and/or custom MapReduce applications. HCatalog has a REST interface and command line client that allows you to create tables or do other operations. You then write your applications to access the tables using HCatalog libraries.

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

Ans:-

JOIN

JOIN clause is used to combine and retrieve the records from multiple tables. JOIN is same as OUTER JOIN in SQL. A JOIN condition is to be raised using the primary keys and foreign keys of the tables.

The following query executes JOIN on the CUSTOMER and ORDER tables, and retrieves the records:

```
hive> SELECT c.ID, c.NAME, c.AGE, o.AMOUNT
FROM CUSTOMERS c JOIN ORDERS o
ON (c.ID = o.CUSTOMER_ID);
```

On successful execution of the query, you get to see the following response:

```
+----+----------+-----+--------+
```

```
| ID | NAME     | AGE | AMOUNT |
+----+----------+-----+--------+
| 3  | kaushik  | 23  | 3000   |
| 3  | kaushik  | 23  | 1500   |
| 2  | Khilan   | 25  | 1560   |
| 4  | Chaitali | 25  | 2060   |
+----+----------+-----+--------+
```

LEFT OUTER JOIN

The HQL LEFT OUTER JOIN returns all the rows from the left table, even if there are no matches in the right table. This means, if the ON clause matches 0 (zero) records in the right table, the JOIN still returns a row in the result, but with NULL in each column from the right table.A LEFT JOIN returns all the values from the left table, plus the matched values from the right table, or NULL in case of no matching JOIN predicate.

The following query demonstrates LEFT OUTER JOIN between CUSTOMER and ORDER tables:

```
hive>
SELECT c.ID, c.NAME, o.AMOUNT, o.DATE
FROM CUSTOMERS c
LEFT OUTER JOIN ORDERS o
ON (c.ID = o.CUSTOMER_ID);
```

On successful execution of the query, you get to see the following response:-

```
+----+----------+--------+---------------------+
| ID | NAME     | AMOUNT | DATE                |
+----+----------+--------+---------------------+
| 1  | Ramesh   | NULL   | NULL                |
| 2  | Khilan   | 1560   | 2009-11-20 00:00:00 |
| 3  | kaushik  | 3000   | 2009-10-08 00:00:00 |
| 3  | kaushik  | 1500   | 2009-10-08 00:00:00 |
| 4  | Chaitali | 2060   | 2008-05-20 00:00:00 |
| 5  | Hardik   | NULL   | NULL                |
| 6  | Komal    | NULL   | NULL                |
| 7  | Muffy    | NULL   | NULL                |
+----+----------+--------+---------------------+
```

RIGHT OUTER JOIN

The HQL RIGHT OUTER JOIN returns all the rows from the right table, even if there are no matches in the left table. If the ON clause matches 0 (zero) records in the left table, the JOIN still returns a row in the result, but with NULL in each column from the left table.

A RIGHT JOIN returns all the values from the right table, plus the matched values from the left table, or NULL in case of no matching join predicate.

The following query demonstrates RIGHT OUTER JOIN between the CUSTOMER and ORDER tables.

```
hive> SELECT c.ID, c.NAME, o.AMOUNT, o.DATE FROM CUSTOMERS c RIGHT
OUTER JOIN ORDERS o ON (c.ID = o.CUSTOMER_ID);
```

On successful execution of the query, you get to see the following response:

```
+------+----------+--------+---------------------+
| ID   | NAME     | AMOUNT | DATE                |
+------+----------+--------+---------------------+
| 3    | kaushik  | 3000   | 2009-10-08 00:00:00 |
| 3    | kaushik  | 1500   | 2009-10-08 00:00:00 |
| 2    | Khilan   | 1560   | 2009-11-20 00:00:00 |
| 4    | Chaitali | 2060   | 2008-05-20 00:00:00 |
+------+----------+--------+---------------------+
```

FULL OUTER JOIN

The HiveQL FULL OUTER JOIN combines the records of both the left and the right outer tables that fulfil the JOIN condition. The joined table contains either all the records from both the tables, or fills in NULL values for missing matches on either side.

The following query demonstrates FULL OUTER JOIN between CUSTOMER and ORDER tables:

```
hive> SELECT c.ID, c.NAME, o.AMOUNT, o.DATE
FROM CUSTOMERS c
FULL OUTER JOIN ORDERS o
ON (c.ID = o.CUSTOMER_ID);
```

On successful execution of the query, you get to see the following response:

```
+------+----------+--------+---------------------+
| ID   | NAME     | AMOUNT | DATE                |
+------+----------+--------+---------------------+
| 1    | Ramesh   | NULL   | NULL                |
| 2    | Khilan   | 1560   | 2009-11-20 00:00:00 |
| 3    | kaushik  | 1500   | 2009-10-08 00:00:00 |
| 4    | Chaitali | 2060   | 2008-05-20 00:00:00 |
| 5    | Hardik   | NULL   | NULL                |
| 6    | Komal    | NULL   | NULL                |
| 7    | Muffy    | NULL   | NULL                |
| 3    | kaushik  | 1500   | 2009-10-08 00:00:00 |
| 2    | Khilan   | 1560   | 2009-11-20 00:00:00 |
| 4    | Chaitali | 2060   | 2008-05-20 00:00:00 |
+------+----------+--------+---------------------+
```

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

Ans:- Yes it's possible.

table_reference [CROSS] JOIN table_reference join_condition.

32. Explain the SMB Join in Hive?

Ans:- SMB is a join performed on bucket tables that have the same sorted, bucket, and join condition columns. It reads data from both bucket tables and performs common joins (map and reduce triggered) on the bucket tables. We need to enable the following properties to use SMB:

> SET hive.input.format=>
org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;
> SET hive.auto.convert.sortmerge.join=true;
> SET hive.optimize.bucketmapjoin=true;
> SET hive.optimize.bucketmapjoin.sortedmerge=true;
> SET hive.auto.convert.sortmerge.join.noconditionaltask=true;

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

Ans:- sort by:-

- It can use multiple reducers for final output.
- Only guarantees ordering of rows within a reducer, ie. It orders the data per reducer locally.

Order by:-

- Uses single reducer to guarantee total order in output. ie. Even if we increase the reducer it will only use a single reducer.
- It sort the data per a single reducer globally.
- LIMIT can be used to minimize sort time.

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

Ans:- Distribute by:-

- It distribute the input rows among reducers. It ensures that all rows for the same key columns are going to the same reducer.
- It does not sort the data globally and not locally

- No overlapping of data.
- DISTRIBUTE BY clause is used to distribute the input rows among reducers.

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

Ans:- In case of HIVE, there is no transfer of data.
HIVE provides you with the functionality to see your flat file data in tabular form and perform some SQL like queries.

There are 2 types of tables in HIVE, External and Internal.
In both the cases, there is no physical table present.

In case of an external table, the HIVE table is mapped to a directory in HDFS. When you put a file in that directory, the data gets reflected in the table.

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

Ans:- This is because we you use Embedded derby mode by default. To use single metastore_db location. you need to change the metastore properties which are available in configuration file hive-site.xml.

The property of interest here is javax.jdo.option.ConnectionURL. The default value of this property is jdbc:derby:;databaseName=metastore_db;create=true. This value specifies that you will be using embedded derby as your Hive metastore and the location of the metastore is metastore_db. Also the metastore will be created if it doesn't already exist.
So to change the behavior change the location to absolute path, so metastore will be used from that location

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

Ans:-
the number of files that will be generated in the table directory to be not equal to the number of buckets.

38. Can a table be renamed in Hive?

Ans:- ALTER TABLE <table_name> RENAME TO <new_table_name>;

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)?

Ans:-

ALTER TABLE table_name CHANGE COLUMN new_col INT BEFORE x_col;

40. What is serde operation in HIVE?

Ans:- A SerDe allows Hive to read in data from a table, and write it back out to HDFS in any custom format. Anyone can write their own SerDe for their own data formats. Serde converts the objects to bytes for seriliasation and reconverts the same. Remember that the medium use for serializing and deserializing should be same.

41. Explain how Hive Deserializes and serialises the data?

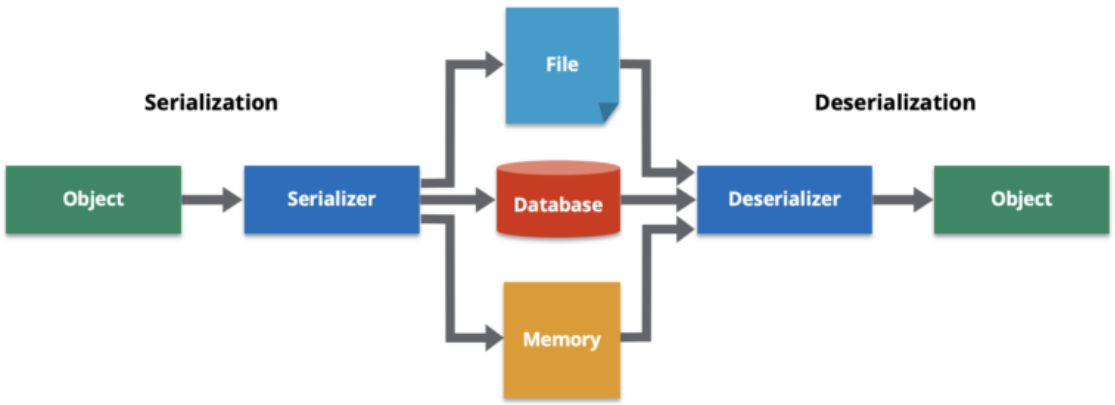
Ans:-

Serialization — Process of converting an object in memory into bytes that can be stored in a file or transmitted over a network.

Deserialization — Process of converting the bytes back into an object in memory.

Java understands objects and hence object is a deserialized state of data. When you use the same concept, Hive understands "columns" and hence if given a "row" of data, the task of converting that data into columns is the Deserialization part of Hive SerDe. In short

"A select statement creates deserialized data(columns) that is understood by Hive. An insert statement creates serialized data(files) that can be stored into an external storage like HDFS".



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

Ans:- The Hive SerDe library is in org.apache.hadoop.hive.serde2.

43. What is the need of custom Serde?

Ans:-

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

Ans:- Struct, MAP , and Array .

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

Ans:- Yes

source /path/to/file/file_with_query.hql

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

Ans:- The default record delimiter is − \n

The filed delimiters are − \001,\002,\003

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

Ans:- SHOW DATABASES LIKE 's%';

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

Ans:- We use LIKE to search for string with similar text.

It is used to search the advanced Regular expression pattern on the columns. If the given pattern matches with any substring of the column, the function returns TRUE

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

Ans:- ALTER TABLE table_name CHANGE column_name column_name new_datatype;

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

Ans:- select cast (’51.2’ as float)

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

Ans:- Null

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;

Ans:-

It inserts the values or if something is available already it overwrites values in table employees .

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

Ans:-

```
CREATE TABLE NEW_TABLE AS
  SELECT ID, SALARY
  FROM OLD_TABLE:
```

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

Ans:-

By default, the columns metadata for Hive does not specify a maximum data length for STRING columns.The driver has the parameter DefaultStringColumnLength, default is 255 maximum value.

Hive supports two more primitive data types, BOOLEAN and BINARY.BINARY is an array of Bytes. BINARY columns are stored within the record, not separately like BLOBs. We can include arbitrary bytes in BINARY column and these bytes are not parsed by Hive as numbers or strings

55. What File Formats and Applications Does Hive Support?

Ans:- Sequence files, ORC files, Avro data files, and Parquet file formats.

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

Ans:- ORC file format is best suitable because it can can store data in an optimized way than the other file formats. ORC reduces the size of the original data up to 75%. This reduced file size increases the query performance.

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

Ans:-

hive.fetch.task.conversion property can (FETCH task) minimize latency of mapreduce overhead.

When queried SELECT, FILTER, LIMIT queries, this property skip mapreduce and using FETCH task.

As a result Hive can execute query without run mapreduce task.

58. What is view and indexing in hive?

Ans:-

View:-

Basically, Apache Hive View is similar to Hive tables, that are generated on the basis of requirements.

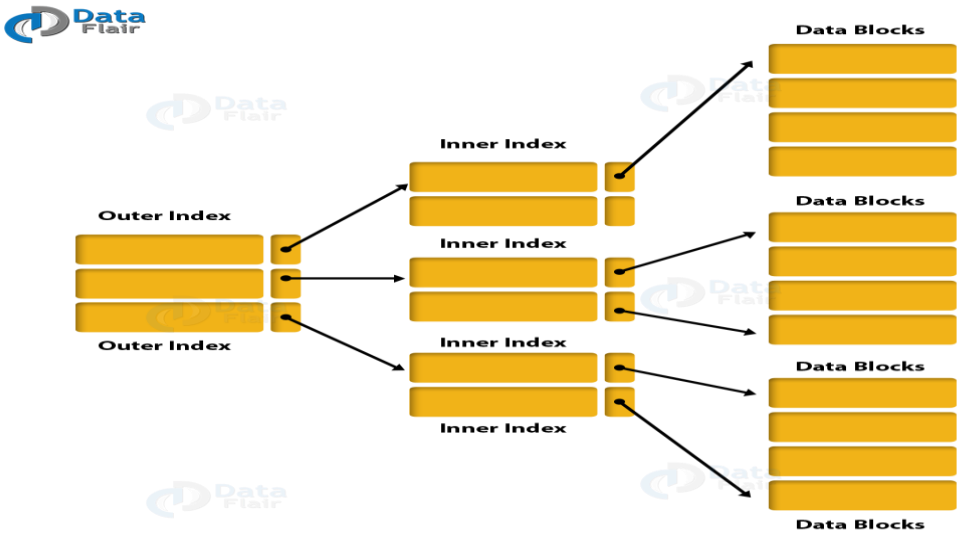
Apache Hive View is a searchable object in a database which we can define by the query. However, we can not store data in the view. Still, some refer to as a view as "virtual tables".

Hence, we can query a view like we can a table. Moreover, by using joins it is possible to combine data from or more table. Also, it contains a subset of information.

Syntax:- **Create VIEW < VIEWNAME> AS SELECT**

Indexing in hive:-

- On defining indexing in Hive we can say these are pointers to particular column name of a table
- However, the user has to manually define the Hive index
- Basically, we are creating the pointer to particular column name of the table, wherever we are creating Hive index.
- By using the Hive index value created on the column name, any Changes made to the column present in tables are stored.



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

Ans:- No, use different names for a table and it's view.

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

Ans:- A processing cost in arranging the values of the column on which index is created since Indexes occupies

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

Ans:- *SHOW INDEX ON table_name;*

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

Ans:-

```
hive> Set mapred.input.dir.recursive=true;

hive> Set hive.mapred.supports.subdirectories=true;
```

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

Ans:- Hive fetches the whole data from file as a FetchTask rather than a mapreduce task which just dumps the data as it is without doing anything on it.

64.What are the uses of Hive Explode?

Ans:- Explode is a **User Defined Table generating Function**(UDTF) in Hive. It takes an array (or a map) as an input and outputs the elements of the array (or a map) as separate rows.

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

Ans:-

- Thrift Client
- ODBC Driver
- JDBC Driver

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

Ans:- Yes. Just define the custom location with the location clause while creating a table.

67. What is the Hive ObjectInspector function?

Ans:- it is used to analyze the internal structure of the row object and also the structure of the individual columns.

ObjectInspector provides a uniform way to access complex objects that can be stored in multiple formats in the memory, including: Instance of a Java class (Thrift or native Java).

68. What is UDF in Hive?

Ans:-

UDF stands for User Defined Functions.

Hive is a powerful tool that allows us to provision sql queries on top of stored data for basic querying and/or analysis, and on top of an already rich set of built-in functions, it allows us to extend its functionality by writing custom functions of our own.

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

Ans:- load data inpath '<hdfs location>' into table <hive table name>;

70. What is TextInputFormat and SequenceFileInputFormat in hive.

Ans:-

1. TextInputFormat

It is the default InputFormat. This InputFormat treats each line of each input file as a separate record. It performs no parsing. TextInputFormat is useful for unformatted data or line-based records like log files. Hence,

- **Key –** It is the byte offset of the beginning of the line within the file (not whole file one split). So it will be unique if combined with the file name.
- **Value –** It is the contents of the line. It excludes line terminators.
-

2. SequenceFileInputFormat

It is an InputFormat which reads sequence files. Sequence files are binary files. These files also store sequences of binary key-value pairs. These are block-compressed and provide direct serialization and deserialization of several arbitrary data. Hence,

Key & Value both are user-defined.

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

Ans:- by setting the MapReduce jobs to execute in strict mode

set hive.mapred.mode=strict;

72.When do we use explode in Hive?

Ans:- Explode is a **User Defined Table generating Function**(UDTF) in Hive. It takes an array (or a map) as an input and outputs the elements of the array (or a map) as separate rows.

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

- Ans:- No, hive can only process the structured data, because the database used by hive by default is Derby (RDBMS).
- Hive supports four file formats those are **TEXTFILE, SEQUENCEFILE, ORC and RCFILE** (Record Columnar File).

Hadoop File Formats and its Types:-

Types of Hadoop File Formats

Hive and Impala table in HDFS can be created using four different Hadoop file formats:

- Text files
- Sequence File
- Avro data files
- Parquet file format

Let's learn about each Hadoop file formats in detail.

1. Text files

A text file is the most basic and a human-readable file. It can be read or written in any programming language and is mostly delimited by comma or tab.

The text file format consumes more space when a numeric value needs to be stored as a string. It is also difficult to represent binary data such as an image.

2. Sequence File

The sequencefile format can be used to store an image in the binary format. They store key-value pairs in a binary container format and are more efficient than a text file. However, sequence files are not human- readable.

3. Avro Data Files

The Avro file format has efficient storage due to optimized binary encoding. It is widely supported both inside and outside the Hadoop ecosystem.

The Avro file format is ideal for long-term storage of important data. It can read from and write in many languages like Java, Scala and so on.Schema metadata can be embedded in the file to ensure that it will always be readable. Schema evolution can accommodate changes. The Avro file format is considered the best choice for general-purpose storage in Hadoop.

4. Parquet File Format

Parquet is a columnar format developed by Cloudera and Twitter. It is supported in Spark, MapReduce, Hive, Pig, Impala, Crunch, and so on. Like Avro, schema metadata is embedded in the file.

Parquet file format uses advanced optimizations described in Google's Dremel paper. These optimizations reduce the storage space and increase performance. This Parquet file format is considered the most efficient for adding multiple records

at a time. Some optimizations rely on identifying repeated patterns. We will look into what data serialization is in the next section.

74. Whenever we run a Hive query, a new metastore_db is created. Why?

Ans:- This is because we you use Embedded derby mode by default. To use single metastore_db location. you need to change the metastore properties which are available in configuration file hive-site.xml.

The property of interest here is javax.jdo.option.ConnectionURL. The default value of this property is jdbc:derby:;databaseName=metastore_db;create=true. This value specifies that you will be using embedded derby as your Hive metastore and the location of the metastore is metastore_db. Also the metastore will be created if it doesn't already exist.

So to change the behavior change the location to absolute path, so metastore will be used from that location.

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

Ans:- ALTER TABLE table_name CHANGE column_name column_name new_datatype;

76. While loading data into a hive table using the LOAD DATA clause, howdo you specify it is a hdfs file and not a local file ?

Ans:- LOAD DATA INPATH '' OVERWRITE INTO TABLE ;

Just don’t mention the local keyword before inpath, it will be directed towards the hdfs path then,

77. What is the precedence order in Hive configuration?

Ans:- In Hive we can use following precedence order to set the configurable properties.

1. SET Command in HIVE
2. The command line –hiveconf option
3. Hive-site.XML
4. Hive-default.xml
5. Hadoop-site.xml
6. Hadoop-default.xm

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

Ans:-

Web interface that can be used for Hive commands is WebHCat, a REST API (not a GUI). With WebHCat, applications can make HTTP requests to access the Hive metastore (HCatalog DDL) or to create and queue Hive queries and commands, Pig jobs, and MapReduce or YARN jobs (either standard or streaming).

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

Ans:- yes, we can gzip our data files and then put them into the hdfs location in the *.gz format.

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

Ans:-

Local Metastore

Here metastore service still runs in the same JVM as Hive but it connects to a database running in a separate process either on same machine or on a remote machine.

Remote Metastore

Metastore runs in its own separate JVM not on hive service JVM.

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

Ans:- As we know that the table data in Hadoop is stored in hdfs, and due to the architecture of Hadoop, the number of files in the filesystem directly affects the memory consumption in the namenode. memory usage may hit the limits of accessible memory on a single machine when there are >50-100 million files. In such situations, it is advantageous to have as few files as possible.

The use of Hadoop Archives is one approach to reducing the number of files in partitions. Hive has built-in support to convert files in existing partitions to a Hadoop Archive (HAR) so that a partition that may once have consisted of 100's of files can occupy just ~3 files (depending on settings).

However, the trade-off is that queries may be slower due to the additional overhead in reading from the HAR.

82. What is DBPROPERTY in Hive?

Ans:-

DBPROPERTIES takes multiple arguments in the form of a key-value pair.

Syntax:-

ALTER (DATABASE|SCHEMA) <database_name> SET DBPROPERTIES ('<property_name>'='<property_value>',..);

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

Ans:-

MapReduce Mode

- Hive script is executed on Hadoop cluster.
- The Hive scripts are converted into MapReduce jobs and then executed on Hadoop cluster (hdfs)

Local Mode

- Hive script runs on a Single machine without the need of Hadoop cluster or hdfs.
- Local mode is used for development purpose to see how the script would behave in an actual environment.