1. What kind of applications is supported by Apache Hive?

Hive supports all those client applications that are written in:

- Java
- PHP
- Python
- C++
- Ruby

by exposing its Thrift server.

2. Where does the data of a Hive table gets stored?

By default, the Hive table is stored in an HDFS directory – /user/hive/warehouse. One can change it by specifying the desired directory in *hive.metastore.warehouse.dir* configuration parameter present in the hive-site.xml.

3. What is a metastore in Hive?

Metastore in Hive stores the meta data information using RDBMS and an open source ORM (Object Relational Model) layer called Data Nucleus which converts the object representation into relational schema and vice versa.

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

Hive stores metadata information in the metastore using RDBMS instead of HDFS. The reason for choosing RDBMS is to achieve low latency as HDFS read/write operations are time consuming processes.

5. What is the difference between local and remote metastore?

Local Metastore:

In local metastore configuration, the metastore service runs in the same JVM in which the Hive service is running and connects to a database running in a separate JVM, either on the same machine or on a remote machine.

Remote Metastore:

In the remote metastore configuration, the metastore service runs on its own separate JVM and not in the Hive service JVM. Other processes communicate with the metastore server using Thrift Network APIs. You can have one or more metastore servers in this case to provide more availability.

6. What is the default database provided by Apache Hive for metastore?

By default, Hive provides an embedded Derby database instance backed by the local disk for the metastore. This is called the embedded metastore configuration.

7. What is the difference between external table and managed table?

Here is the key difference between an external table and managed table:

- In case of managed table, If one drops a managed table, the metadata information along with the table data is deleted from the Hive warehouse directory.
- On the contrary, in case of an external table, Hive just deletes the metadata information regarding the table and leaves the table data present in HDFS untouched.

8. 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. It can be achieved by using the clause – LOCATION '<hdfs_path>'.

9. When should we use SORT BY instead of ORDER BY?

We should use SORT BY instead of ORDER BY when we have to sort huge datasets because SORT BY clause sorts the data using multiple reducers whereas ORDER BY sorts all of the data together using a single reducer. Therefore, using ORDER BY against a large number of inputs will take a lot of time to execute.

10. What is a partition in Hive?

Hive organizes tables into partitions for grouping similar type of data together based on a column or partition key. Each Table can have one or more partition keys to identify a particular partition. Physically, a partition is nothing but a sub-directory in the table directory.

11. Why do we perform partitioning in Hive?

Partitioning provides granularity in a Hive table and therefore, reduces the query latency by scanning only **relevant** partitioned data instead of the whole data set.

For example, we can partition a transaction log of an e – commerce website based on month like Jan, February, etc. So, any analytics regarding a particular month, say Jan, will have to scan the Jan partition (sub – directory) only instead of the whole table data.

12. What is dynamic partitioning and when is it used?

In dynamic partitioning values for partition columns are known in the runtime, i.e. It is known during loading of the data into a Hive table.

One may use dynamic partition in following two cases:

- Loading data from an existing non-partitioned table to improve the sampling and therefore, decrease the query latency.
- When one does not know all the values of the partitions before hand and therefore, finding these partition values manually from a huge data sets is a tedious task.

13. What is the default maximum dynamic partition that can be created by a mapper/reducer? How can you change it?

By default the number of maximum partition that can be created by a mapper or reducer is set to 100. One can change it by issuing the following command:

SET hive.exec.max.dynamic.partitions.pernode = <value>

Note: You can set the total number of dynamic partitions that can be created by one statement by using: SET hive.exec.max.dynamic.partitions = <value>

14. Why do we need buckets?

Ans. Basically, for performing bucketing to a partition there are two main reasons:

- A map side join requires the data belonging to a unique join key to be present in the same partition.
- It allows us to decrease the query time. Also, makes the sampling process more efficient.

15. How Hive distributes the rows into buckets?

Ans. By using the formula: hash_function (bucketing_column) modulo (num_of_buckets) Hive determines the bucket number for a row. Basically, hash_function depends on the column data type. Although, hash_function for integer data type will be:

hash_function (int_type_column)= value of int_type_column

16. What is indexing and why do we need it?

Ans. Hive Index is a Hive query optimization techniques. Basically, we use it to speed up the access of a column or set of columns in a Hive database. Since, the database system does not need to read all rows in the table to find the data with the use of the index, especially that one has selected.

17. Where is table data stored in Apache Hive by default?

Ans. hdfs: //namenode_server/user/hive/warehouse

18. How can you configure remote metastore mode in Hive?

Ans. Basically, hive-site.xml file has to be configured with the below property, to configure metastore in Hive –

hive.metastore.uris

thrift: //node1 (or IP Address):9083

IP address and port of the metastore host

19. Is it possible to change the default location of Managed Tables in Hive, if so how?

Ans. Yes, by using the LOCATION keyword while creating the managed table, we can change the default location of Managed tables. But the one condition is, the user has to specify the storage path of the managed table as the value of the LOCATION keyword.

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

Ans. Basically, the user need not LOAD DATA that moves the files to the /user/hive/warehouse/. But only if data is already present in HDFS. Hence, using the keyword external that creates the table definition in the hive metastore the user just has to define the table.

Create external table table_name (

id int,

myfields string) //location '/my/location/in/hdfs';

21. What are the different components of a Hive architecture?

Ans. There are several components of Hive Architecture. Such as –

1. User Interface – Basically, it calls the execute interface to the driver. Further, driver creates a session handle to the query. Then sends the query to the compiler to generate an execution plan for it.
2. Metastore – It is used to Send the metadata to the compiler. Basically, for the execution of the query on receiving the send MetaData request.
3. Compiler- It generates the execution plan. Especially, that is a DAG of stages where each stage is either a metadata operation, a map or reduce job or an operation on HDFS.
4. Execute Engine- Basically, by managing the dependencies for submitting each of these stages to the relevant components we use Execute engine.

22. Is it possible to use the same metastore by multiple users, in case of the embedded hive?

Ans. No, we cannot use metastore in sharing mode. It is possible to use it in standalone "real" database. Such as MySQL or PostGresSQL.

23. Usage of Hive.

Ans.

– We use Hive for Schema flexibility as well as evolution.

– Moreover, it is possible to portion and bucket, tables in Apache Hive.

– Also, we can use JDBC/ODBC drivers, since they are available in Hive.

24. Features and Limitations of Hive.

Ans. Features of Hive

1. The best feature is it offers data summarization, query, and analysis in much easier manner.
2. To process data without actually storing in HDFS, Hive supports external tables.
3. Moreover, it fits the low-level interface requirement of Hadoop perfectly.

- **Limitation of Hive**

1. We can not perform real-time queries with Hive. Also, it does not offer row-level updates.
2. Moreover, for interactive data browsing Hive offers acceptable latency.
3. Also, we can say Hive is not the right choice for online transaction processing.