Answers

**1. When Hive is best suited and when is it not?**

Ans: Hive is most suited for data warehouse applications, where relatively static data is analyzed, fast response times are not required where a large data set is maintained and mined for insights, reports, etc and when the data is not changing rapidly.

Hive it is not suitable for OLTP system. It does not provide insert and update at row level.

**2. When should one use Hive over MapReduce?**

**Ans:** MapReduce use java code which may not be known by all**.** Hive is a SQL type language with some good features like partitioning and bucketing to improve reading performance on structure data. Hive enforces schema on read. Syntax of hive queries are easy to learn and implement.

**3. What is Hive metastore?**

Ans: Hive metastore is a database that stores metadata about your Hive tables (eg. table name, column names and types, table location, storage handler being used, number of buckets in the table, sorting columns if any, partition columns if any, etc.). When you create a table, this metastore gets updated with the information related to the new table which gets queried when you issue queries on that table.

**4. How can Hive improve performance with orc file format tables?**

Ans: ORC File Format Full Form is Optimized Row Columnar File Format.ORC File format provides very efficient way to store relational data then RC file,By using ORC File format we can reduce the size of original data up to 75%.Comparing to Text,Sequence,Rc file formats ORC is better . Using ORC files improves performance when Hive is reading, writing, and processing data. Again Comparing to RC and ORC File formats always ORC is better.ORC takes less time to access the data comparing to RC File Format and ORC takes Less space  space to store data.

**5. What is thrift server and client, jdbc and odbc driver importance in hive?**

Ans: Hive Thrift allows access to Hive over a single port. Thrift is a software framework for scalable cross-language services development. Thrift allows clients using languages including Java, C++, Ruby, and many others, to programmatically access Hive remotely.

Hive JDBC and ODBC drivers that let you connect to popular Business Intelligence (BI) tools to query, analyze and visualize data stored.

**6. What is the importance of partition in hive?**

**Ans:**  Partition is used for distributing load horizontally, moving data physically closer to its most

frequent users, and other purposes. Its have important performance benefits, and they can help organize data in a logical fashion, such as hierarchically.

It is a way of dividing a table into related parts based on the values of partitioned columns such as date, city, and department. Using partition, it is easy to query a portion of the data..

CREATE TABLE employees (

name STRING,

salary FLOAT,

address STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>

)

PARTITIONED BY (state STRING);

**7. What is the use of bucketing in hive?**

Ans: The values in a column are hashed into a number of buckets which is defined by user. It is a way to avoid too many partitions or nested partitions while ensuring optimizes query output.Tables or partitions are sub-divided into **buckets,** to provide extra structure to the data that may be used for more efficient querying. Bucketing works based on the value of hash function of some column of a table.

CREATE TABLE user\_info\_bucketed(user\_id BIGINT, firstname STRING, lastname STRING)

COMMENT 'A bucketed copy of user\_info'

PARTITIONED BY(ds STRING)

CLUSTERED BY(user\_id) INTO 256 BUCKETS;

**8. What is the difference between static partitioning and dynamic partitioning in hive?**

Partitions are created when data is inserted into table. Depending on how you load data you would need partitions. Usually when loading files (big files) into Hive tables static partitions are preferred. That saves your time in loading data compared to dynamic partition. You "statically" add a partition in table and move the file into the partition of the table. Since the files are big they are usually generated in HDFS. You can get the partition column value form the filename, day of date etc without reading the whole big file.

Incase of dynamic partition whole big file i.e. every row of the data is read and data is partitioned through a MR job into the destination tables depending on certain field in file.

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

LOAD DATA INPATH '/hdfs path of the file' INTO TABLE t1 PARTITION(country="US")

LOAD DATA INPATH '/hdfs path of the file' INTO TABLE t1 PARTITION(country="UK")

Dynamic partition allow us not to specify partition column value each time.

INSERT INTO TABLE t2 PARTITION(country) SELECT \* from T1;