Scenario Based questions:-

Will the reducer work or not if you use “Limit 1” in any HiveQL query?

Ans:- if our query is simple then no reducer would be called.
If we use tez engine, then also most probably the reducer would not be called because, Tez tries to execute the query mostly into the map phase only using CBO technique, condition is that our dataset should be small and pertaining that there are no joins.we can always use the explain key word to see what map phases and reduce phase are involved in the query.

Suppose I have installed Apache Hive on top of my Hadoop cluster using default metastore configuration. Then, what will happen if we have multiple clients trying to access Hive at the same time?

Ans:- No, multiple users cannot access the hive metastore at the same time because by default default metastore configuration allows only one Hive session to be opened at a time for accessing the metastore. If the multiple users try to access the metastore at a single time, they will face an error.

We need to configure a standalone metastore , to enable the multiusers access to hive metastore at a single point of the time.

Suppose, I create a table that contains details of all the transactions done by the customers: CREATE TABLE transaction_details (cust_id INT, amount FLOAT, month STRING, country STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘,’ ; Now, after inserting 50,000 records in this table, I want to know the total revenue generated for each month. But, Hive is taking too much time in processing this query. How will you solve this problem and list the steps that I will be taking in order to do so?

Ans:- we can solve this issue by creating partiotions of the tables Month wise so that we do not need to scan the whole table.

steps to solve the problem:-

1. Create a partitioned table, say partitioned_transaction:

CREATE TABLE partitioned_transaction (cust_id INT, amount FLOAT, country STRING) PARTITIONED BY (month STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘,’ ;

2. Enable dynamic partitioning in Hive:-

```
SET hive.exec.dynamic.partition = true;

SET hive.exec.dynamic.partition.mode = nonstrict;
```

3. Transfer the data from the non – partitioned table into the newly created partitioned table:

```
INSERT OVERWRITE TABLE partitioned_transaction PARTITION (month)
SELECT cust_id, amount, country, month FROM transaction_details;
```

How can you add a new partition for the month December in the above partitioned table?

Ans:- ALTER TABLE partitioned_transaction ADD PARTITION (month='Dec')
LOCATION '/partitioned_transaction';

I am inserting data into a table based on partitions dynamically. But, I received an error – FAILED ERROR IN SEMANTIC ANALYSIS: Dynamic partition strict mode requires at least one static partition column. How will you remove this error?

Ans:- we need to execute the following command:-

```
SET hive.exec.dynamic.partition = true;

SET hive.exec.dynamic.partition.mode = nonstrict;
```

Suppose, I have a CSV file – 'sample.csv' present in '/temp' directory with the following entries:

id first_name last_name email gender ip_address

How will you consume this CSV file into the Hive warehouse using built-in SerDe?

Ans:-

we can use the serde properties by hive to work with the csv files, so we will create a table and feed the data to store the data into the warehouse.

```
hdfs dfs -put sample.csv /tmp/serdes/


drop table if exists sample;

create external table sample(id int,first_name string,last_name string,email string,gender string,ip_address string)

row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

stored as textfile
```

```
location '/tmp/serdes/';
```

Suppose, I have a lot of small CSV files present in the input directory in HDFS and I want to create a single Hive table corresponding to these files. The data in these files are in the format: {id, name, e-mail, country}. Now, as we know, Hadoop performance degrades when we use lots of small files. So, how will you solve this problem where we want to create a single Hive table for lots of small files without degrading the performance of the system?

Ans:- we can use the SequenceFile format which will group these small files together to form a single sequence file.

Create a temporary table:-

```
CREATE TABLE temp_table (id INT, name STRING, e-mail STRING, country STRING)

ROW FORMAT FIELDS DELIMITED TERMINATED BY ‘,’ STORED AS TEXTFILE;
```

Load the data into temp_table:-

```
LOAD DATA INPATH ‘/input’ INTO TABLE temp_table;
```

Create a table that will store data in SequenceFile format:-

```
CREATE TABLE sample_seqfile (id INT, name STRING, e-mail STRING, country STRING)

ROW FORMAT FIELDS DELIMITED TERMINATED BY ‘,’ STORED AS SEQUENCEFILE;
```

Transfer the data from the temporary table into the sample_seqfile table:-

```
INSERT OVERWRITE TABLE sample SELECT * FROM temp_table;
```