Scenario Based questions:

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

Answer : - Yes, reducer works if we use “limit 1 “ in any HiveQL 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?

Answer:-

The default database which is used for storing the metadata is “Derby”. It has limitation that it can accept only one connection at one time. If multiple user trying to connect at same time then request will be in queue.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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?

Answer:-

Since by default one reducer works thus processing takes time to compute the data, if we increase the reducer and then try the same query, we will leverage the parallel processing and computation will fast.

Step to follow: -

1. We will increase the mapreduce count, here is the query for that (**set mapreduce.job.reduces**)
2. Now we will execute the query, it will be faster than before because of more reducer work

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answer: -

We will create a new static partition table since we know the partition value and while inserting the data from above table we will use where clause in the select command to filter only dec data.

­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­

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?

Answer:-

If we get the mentioned error, we need to set the dynamic partitioning non strict. Here is the code

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?

**Answer: -** Assuming, we have csv file and we need to create a table for that file using built in Serde. Data is available on local machine

Create table :-

Create table sample\_csv

( id int,

first\_name string,

last\_name string

email string,

gender string,

ip\_address string )

row format serde “org.apache.hadoop.hive.serde2.OpenCSVSerde”

WITH SERDEPROTERTIES ( “sepratorChar” = “,” , “quoteChar” = “\””, “escapeChar” = “\\” )

TBLPROPERTIES ( “skip.header.line.count” = “1”);

Load data local inpath “/tmp/sample.csv /user/hive/warehouse/sample\_csv

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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?

Answer

We create a table with CSV, once it is created we will then create a dynamic partition table. Partition column will be country col. We will then insert data from csv and store as orc format.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

LOAD DATA LOCAL INPATH ‘Home/country/state/’

OVERWRITE INTO TABLE address;

The following statement failed to execute. What can be the cause?

Answer: -

Below are causes:-

1. Given path is not correct, we should use root before home. For an example “/home/country/state/
2. Since we are loading data from local we should use file:// before the path. It will indicate that path belongs to local file system, it should be in quote.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Is it possible to add 100 nodes when we already have 100 nodes in Hive? If yes, how?

Yes, it is possible to add nodes. Below are the steps to follow

1. We will take new node and create username and password
2. SSH has to be installed for ssh connection with master node
3. We will create key to sync the system without password ( using key-gen command) it will create two keys 1. Public \_rsa id and private\_rsa id. We will use public rsa
4. Host entry is also required, hostname and ip address
5. We will start data node on new added node
6. Ssh to new data node and start HDFS on that node
7. Use JPS command to check the status of the hadoop component

Hive Practical questions:

Hive Join operations

Create a table named CUSTOMERS(ID | NAME | AGE | ADDRESS | SALARY)

hive> create table customer

> (

> id int,

> name string,

> age int,

> address string,

> salary int)

> row format delimited

> fields terminated by ","

> tblproperties("skip.header.line.count"="1");

hive> load data local inpath 'file:///home/cloudera/hive\_class/customer.csv' into table customer;

**Scenario:-** Create a Second table ORDER(OID | DATE | CUSTOMER\_ID | AMOUNT)

hive> create table order

> (

> oid int,

> date string,

> customer\_id int,

> amount int)

> row format delimited

> fields terminated by ","

> tblproperties("skip.header.line.count"="1");

hive> load data local inpath 'file:///home/cloudera/orderid.csv' into table order;

**Scenario:-**  Now perform different joins operations on top of these tables

(Inner JOIN, LEFT OUTER JOIN ,RIGHT OUTER JOIN ,FULL OUTER JOIN)

**Inner joins: -**

Have increase the mapred and reducer to 3, because default one was taking two much time

select c.id, c.name, c.age, o.oid,o.date, o.amount from customer c join order o on c.id= o.customer\_id;

**Left outer joins: -**

select c.id, c.name, c.age, o.oid,o.date, o.amount from customer c left outer join order o on c.id= o.customer\_id

**Right outer joins**

select c.id, c.name, c.age, o.oid,o.date, o.amount from customer c right outer join order o on c.id= o.customer\_id

**Full outer joins:**

select c.id, c.name, c.age, o.oid,o.date, o.amount from customer c full outer join order o on c.id= o.customer\_id;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BUILD A DATA PIPELINE WITH HIVE

Download a data from the given location -

https://archive.ics.uci.edu/ml/machine-learning-databases/00360/

1. Create a hive table as per given schema in your dataset

**Query:-**

create table air\_quality

> (

> date date,

> time string,

> co\_GT int,

> PT08\_S1\_CO\_ int,

> NMHC\_GT int,

> C6H6\_GT int,

> PT08\_S2\_NMHC\_ int,

> NOx\_GT int,

> PT08\_S3\_NOx int,

> NO2\_GT int,

> PT08\_S4\_NO2 int,

> PT08\_S5\_O3 int,

> T int,

> RH int,

> AH int)

> row format serde "org.apache.hadoop.hive.serde2.OpenCSVSerde"

> with serdeproperties (

> "separatorChar"= "\;")

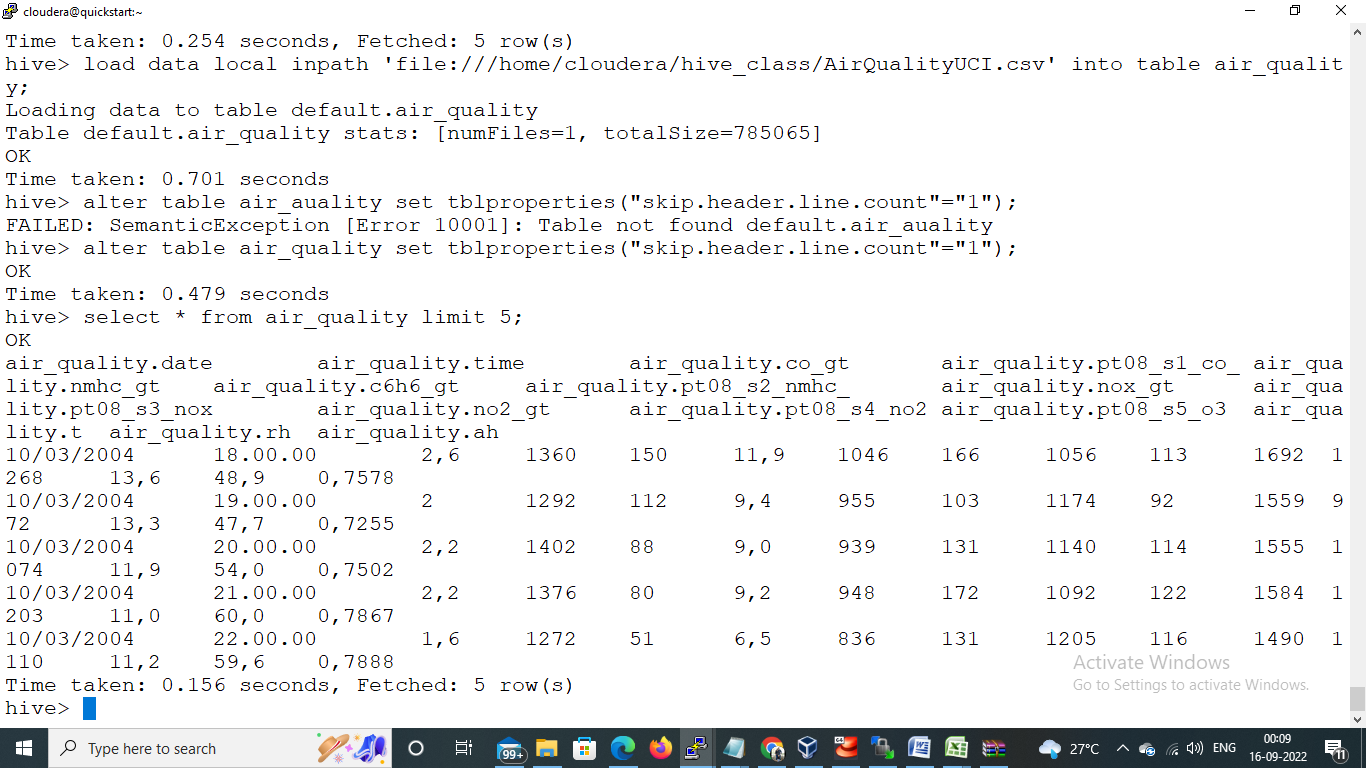
Tblproperties(“skip.header.line.count”=”1”);

2. try to place a data into table location

**Query:-** load data local inpath 'file:///home/cloudera/hive\_class/AirQualityUCI.csv' into table air\_quality;

3. Perform a select operation.

Query:-



4. Fetch the result of the select operation in your local as a csv file .

**Query:-** insert overwrite local directory 'file:///home/cloudera/save\_data/output.csv' select \* from air\_quality;

5. Perform group by operation.

Query:- grouping data by date;

select date, count(\*) from air\_quality group by date;

6. Perform filter operation at least 5 kinds of filter examples.

Query

1. select \* from air\_quality where date ='01/01/2005';
2. select date, time, nox\_gt from air\_quality where nox\_gt = 166';
3. select \* from air\_quality where time ='10.00.00';
4. select count(\*) from air\_quality where nmhc\_gt =150;
5. select count(\*) from air\_quality where nmhc\_gt =150 and pt08\_s4\_no2 =1692;

7. show and example of regex operation

Query: -

8. alter table operation

Query: - alter table air\_quality change date date1 date;

9 . drop table operation

Query drop table sales\_order\_orc;

10 . order by operation .

Query: - select \* from air\_quality order by date1;

11 . where clause operations you have to perform .

Query: - select date, time, nox\_gt from air\_quality where nox\_gt = 166'

12 . sorting operation you have to perform .

Query: - select \* from air\_quality sort by date1;

13 . distinct operation you have to perform .

Query: -select distinct(date1), time, pt08\_s3\_nox from air\_quality;

14 . like an operation you have to perform .

Query: - select \* from air\_quality where pt08\_s3\_nox like “%10%”;

15 . union operation you have to perform .

16 . table view operation you have to perform .

Query:- create view motorcycle\_sales as select \* from sales\_order\_orc where productline = ‘Motorcycles’;

hive operation with python

Create a python application that connects to the Hive database for extracting data, creating sub tables for data processing, drops temporary tables.fetch rows to python itself into a list of tuples and mimic the join or filter operations