Advanced Hive- Assignment 3 : Transactions in Hive

For performing Hive row level transactions, we have used hive in Hortonworks sandbox . We have used Hortonworks Hadoop 2.2 version virtualbox. Hive row-level transactions available in Hive 0.14.So we have used Hive 0.14 for this assignment.

Settings needed for Hive row level transactions:

Before creating a Hive table that supports transactions, the transaction features present in Hive needs to be turned on, as by default they are turned off.

The below properties needs to be set appropriately in hive shell , order-wise to work with transactions in Hive

```
set hive.support.concurrency = true;

set hive.enforce.bucketing = true;

set hive.exec.dynamic.partition.mode = nonstrict;

set hive.txn.manager = org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;

set hive.compactor.initiator.on = true;

set hive.compactor.worker.threads = 1;
```

Creating the table and inserting the data in the table:

The below syntax will create a table with name 'college' and the columns present in the table are 'clg_id, clg_name, clg_loc'. We are bucketing the table by 'clg_id' and the table format is 'orc', also we are enabling the transactions in the table by specifying it inside the TBLPROPERTIES as 'transactional'='true'

```
CREATE TABLE college(clg_id int,clg_name string,clg_loc string) CLUSTERED BY (clg_id) INTO 5 BUCKETS
STORED AS ORC TBLPROPERTIES('transactional'='true');
```

Screenshot of Mobaxterm for creating the table college:

```
hive> CREATE TABLE college(clg_id int,clg_name string,clg_loc string) CLUSTERED BY (clg_id) INTO 5 BUCKETS STORED AS ORC TBLPROPERTIES('transac
tional'='true');
OK
```

We have successfully created a table with name 'college' which supports row-level transactions of Hive.

The below syntax is used to insert data into the college table

```
INSERT INTO table college
values(1,"nec","nlr"),(2,"vit","vlr"),(3,"srm","chen"),(4,"lpu","del"),(5,"stanford","uk"),(6,"JNTUA","atp"),
(7,"cambridge","us");
```

Screenshot of Mobaxterm for Inserting Data into a Hive Table:

```
hive> INSERT INTO table college values(1,"nec","nlr"),(2,"vit","vlr"),(3,"srm","chen"),(4,"lpu","del"),(5,"stanford","uk"),(6,"JNTUA","atp"),(7
,"cambridge","us");
```

Screenshot of Mobaxterm to view the contents of college table:

```
hive> select * from college;
OK
5       stanford        uk
6       JNTUA   atp
1       nec     nlr
8       cambridge       us
2       vit     vlr
3       srm     chen
4       lpu     del
Time taken: 0.104 seconds, Fetched: 7 row(s)
```

From the above image, we can see that the data has been inserted successfully into the table.

If we try to re-insert the same data again, it will be appended to the previous data as shown below:

Screenshot of Mobaxterm for re-insering the same data again

```
hive> select * from college;
OK
5       stanford        uk
5       stanford        uk
6       JNTUA   atp
1       nec     nlr
6       JNTUA   atp
1       nec     nlr
7       cambridge       us
2       vit     vlr
7       cambridge       us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
Time taken: 0.181 seconds, Fetched: 14 row(s)
```

Updating a row in the Hive table:

The below command is used to update a row in Hive table.

```
UPDATE college set clg_id = 8 where clg_id = 7;
```

Screenshot of Mobaxterm for updating clg_id with new clg_id in Hive:

```
hive> UPDATE college set clg_id = 8 where clg_id = 7;
FAILED: SemanticException [Error 10122]: Bucketized tables do not support INSERT INTO: Table: default.college
```

From the above image, we can see that we have received an error message. This means that the Update command is not supported on the columns that are bucketed.

In this table, we have bucketed the **'clg_id'** column and performing the Update operation on the same column, so we have go the error

Now let's perform the update operation on Non bucketed column using the below command.

```
UPDATE college set clg_name = 'IIT' where clg_id = 6;
```

Screenshot of Mobaxterm for updating clg_name with new clg_name in Hive:

```
hive> UPDATE college set clg_name = 'IIT' where clg_id = 6;
Query ID = root_20171111093737_2151ac7e-9620-4c39-8ea8-cf581d6941ab
```

We have successfully updated the data. The updated data can be checked using the command select * from college.

Screenshot of Mobaxterm for the table college after updating a row :

```
hive> select * from college;
OK
5       stanford        uk
5       stanford        uk
6       IIT     atp
1       nec     nlr
6       IIT     atp
1       nec     nlr
7       cambridge       us
2       vit     vlr
7       cambridge       us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
Time taken: 0.098 seconds, Fetched: 14 row(s)
```

Deleting a row in the Hive table:

The below command is used to delete a single row in Hive table.

delete from college where clg_id=5;

Screenshot of Mobaxterm for deleting a row having clg_id=5 in Hive:

```
hive> delete from college where clg_id=5;
Query ID = root_20171111093939_94b22910-cddf-44e7-b179-6672dbecc688
```

We have now successfully deleted a row from the Hive table. This can be checked using the command select * from college.

Screenshot of Mobaxterm for the table college after deleting a row :

```
hive> select * from college;
OK
6       IIT     atp
1       nec     nlr
6       IIT     atp
1       nec     nlr
7       cambridge       us
2       vit     vlr
7       cambridge       us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
Time taken: 0.12 seconds, Fetched: 12 row(s)
```

We can see that there is no row with clg_id =1. This means that we have successfully deleted the row from the Hive table.

This is how the transactions or row-wise operations are performed in Hive.