Session 9: Advance Hive

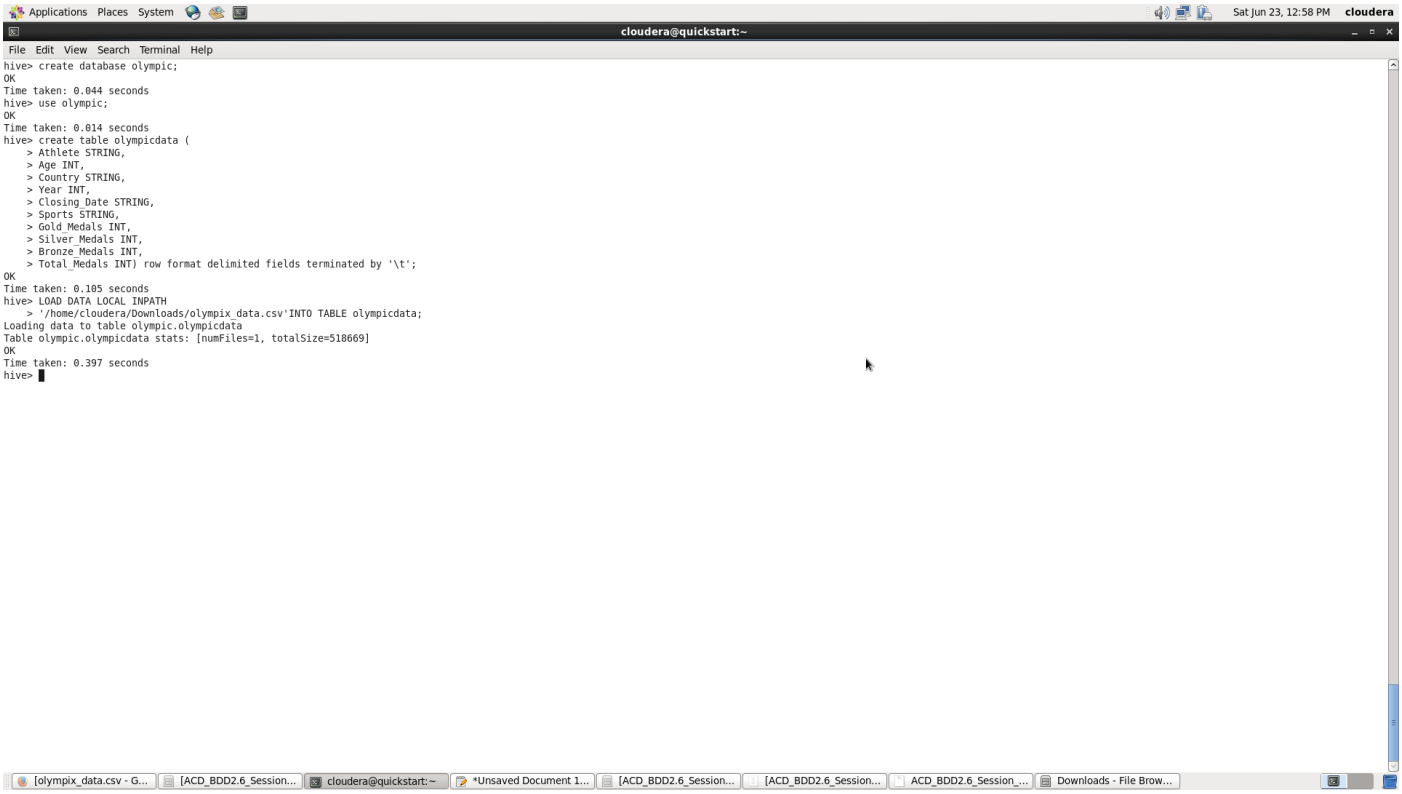
DATE SET DESCRIPTION The data set consists of the following fields.

Athlete: This field consists of the athlete name Age: This field consists of athlete ages Country: This fields consists of the country names which participated in Olympics Year: This field consists of the year Closing Date: This field consists of the closing date of ceremony Sport: Consists of the sports name Gold Medals: No. of Gold medals Silver Medals: No. of Silver medals Bronze Medals: No. of Bronze medals Total Medals: Consists of total no. of medals

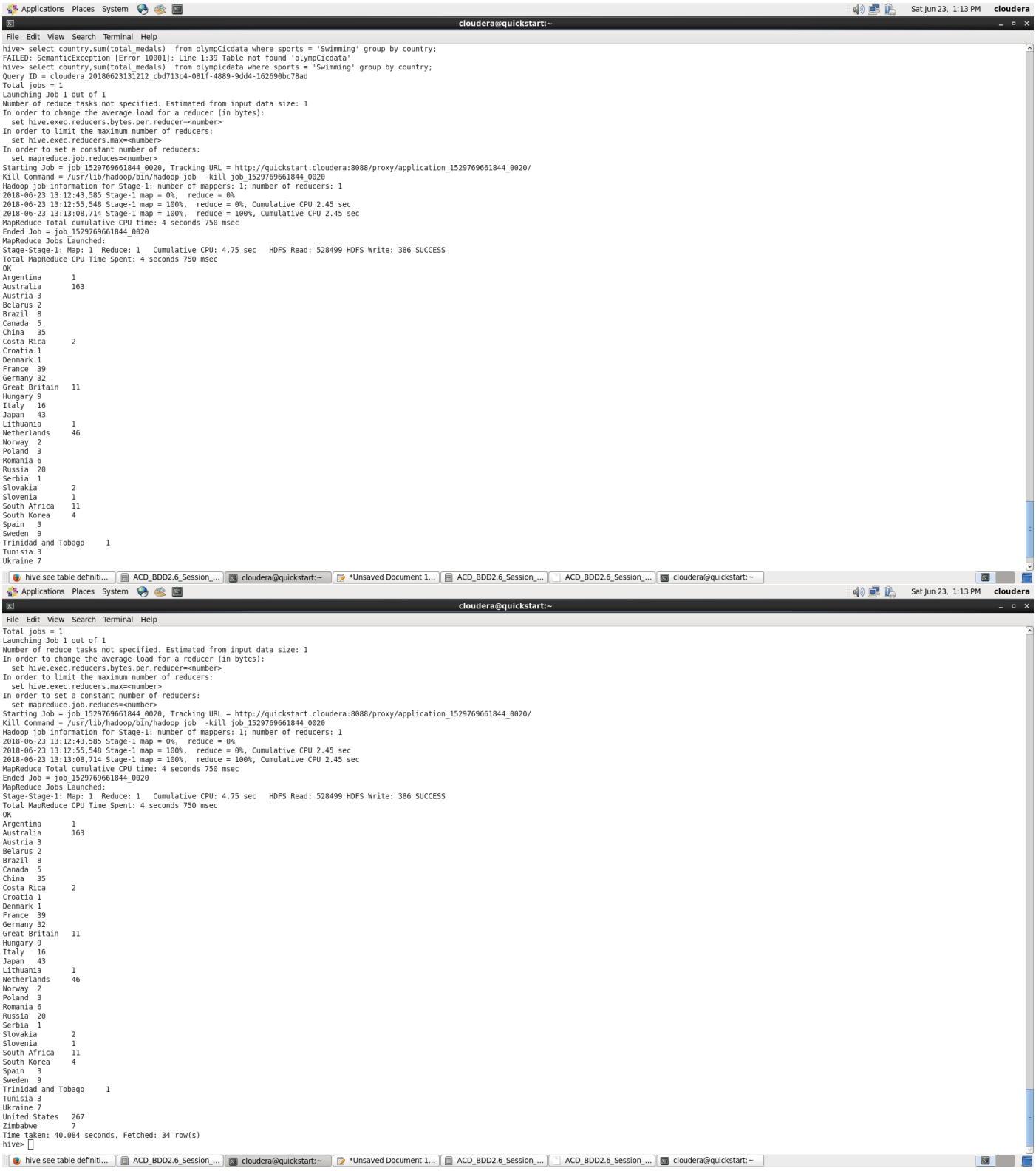
Task 1:

- Created Olympic database & created Olympicdata table
- Inseted data into table
- Processed data as show in below pictures

1. **Write a Hive program to find the number of medals won by each country in swimming.**

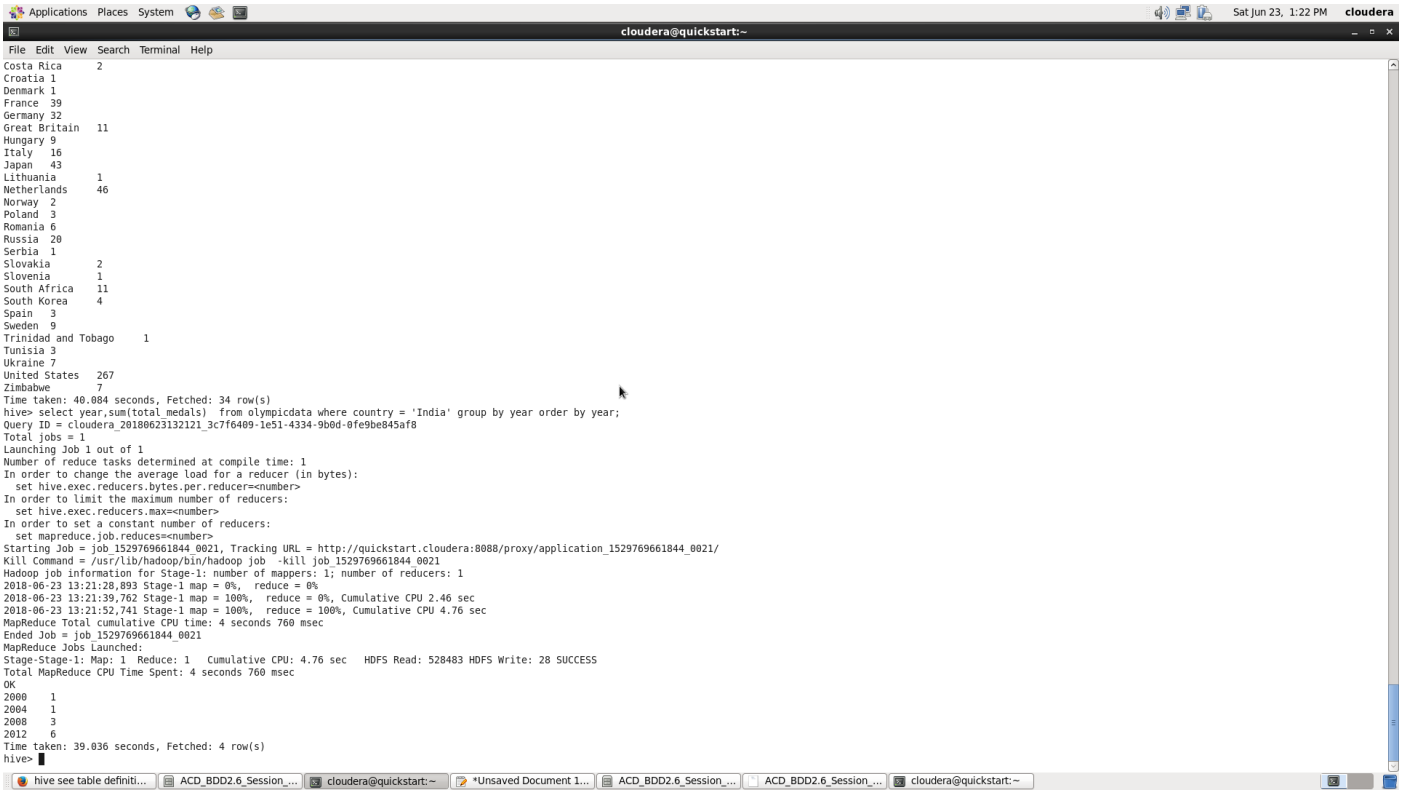


Session 9: Advance Hive

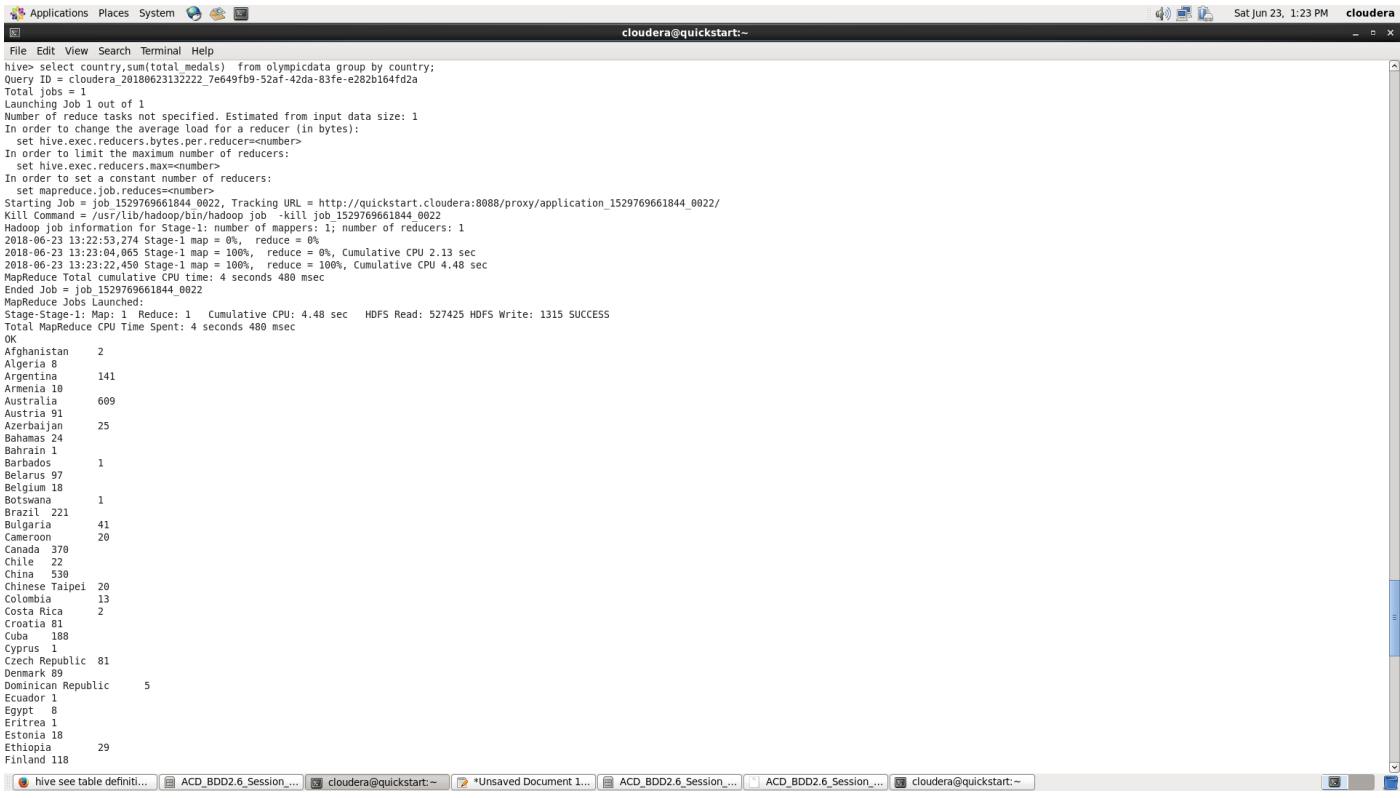


2. **Write a Hive program to find the number of medals that India won year wise.**

Session 9: Advance Hive



3. Write a Hive Program to find the total number of medals each country won.



Session 9: Advance Hive

4. **Write a Hive program to find the number of gold medals each country won.**

Session 9: Advance Hive

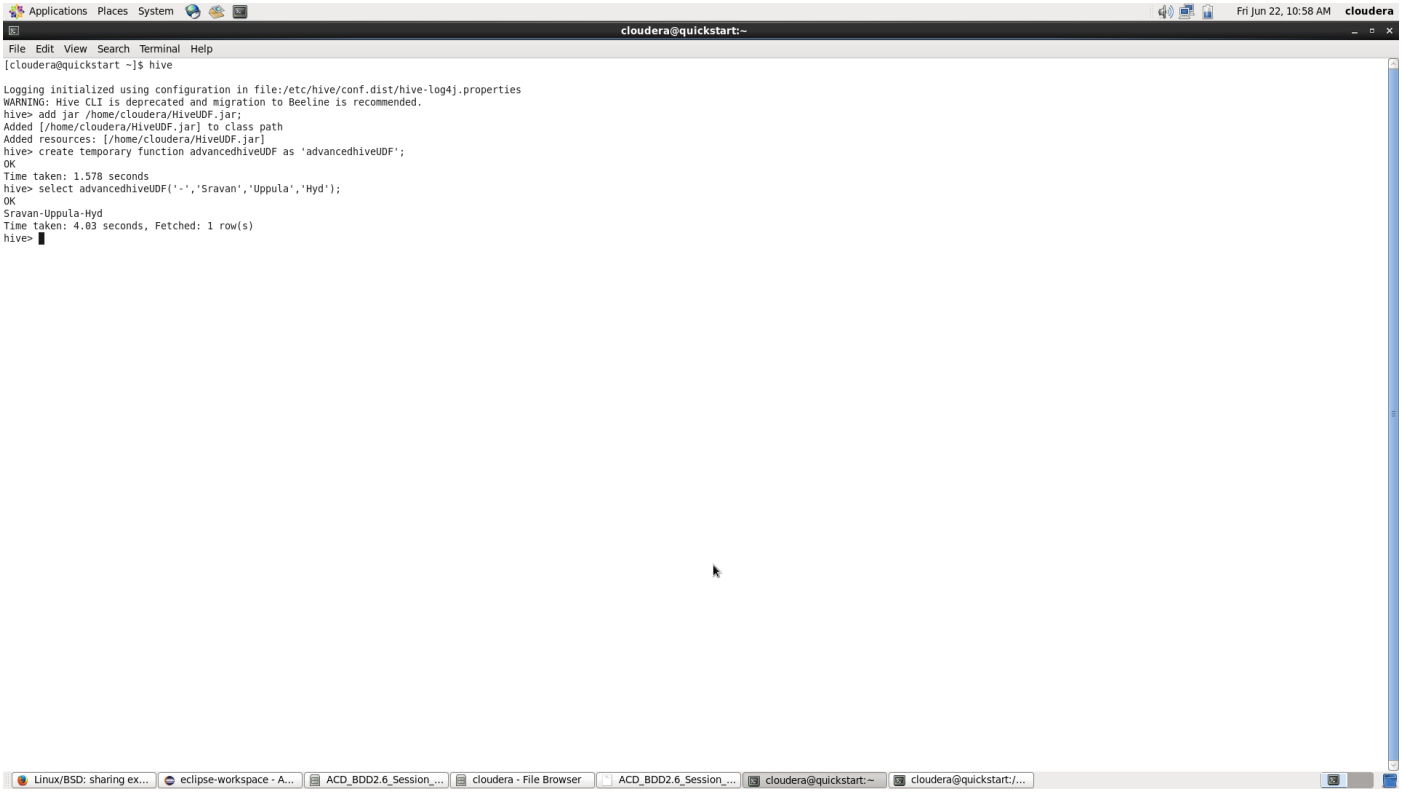
Task 2:

Write a hive UDF that implements functionality of string concat_ws(string SEP, array<string>). This UDF will accept two arguments, one string and one array of string. It will return a single string where all the elements of the array are separated by the SEP.

- Created UDF for concat_ws example, example is given & executed\

Session 9: Advance Hive

- Attached jar & java file for the same in Github



Session 9: Advance Hive

Task 3:

Transactions in Hive

Transactions in Hive are introduced in Hive 0.13, but they only partially fulfill the ACID properties like atomicity, consistency, durability, at the partition level. Here, Isolation can be provided by turning on one of the locking mechanisms available with zookeeper or in memory.

But in Hive 0.14, new API's have been added to completely fulfill the ACID properties while performing any transaction.

Transactions are provided at the row-level in Hive 0.14. The different row-level transactions available in Hive 0.14 are as follows:

1. Insert
2. Delete
3. Update

Row-level Transactions Available in Hive 0.14

Let's perform some row-level transactions available in Hive 0.14. 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:

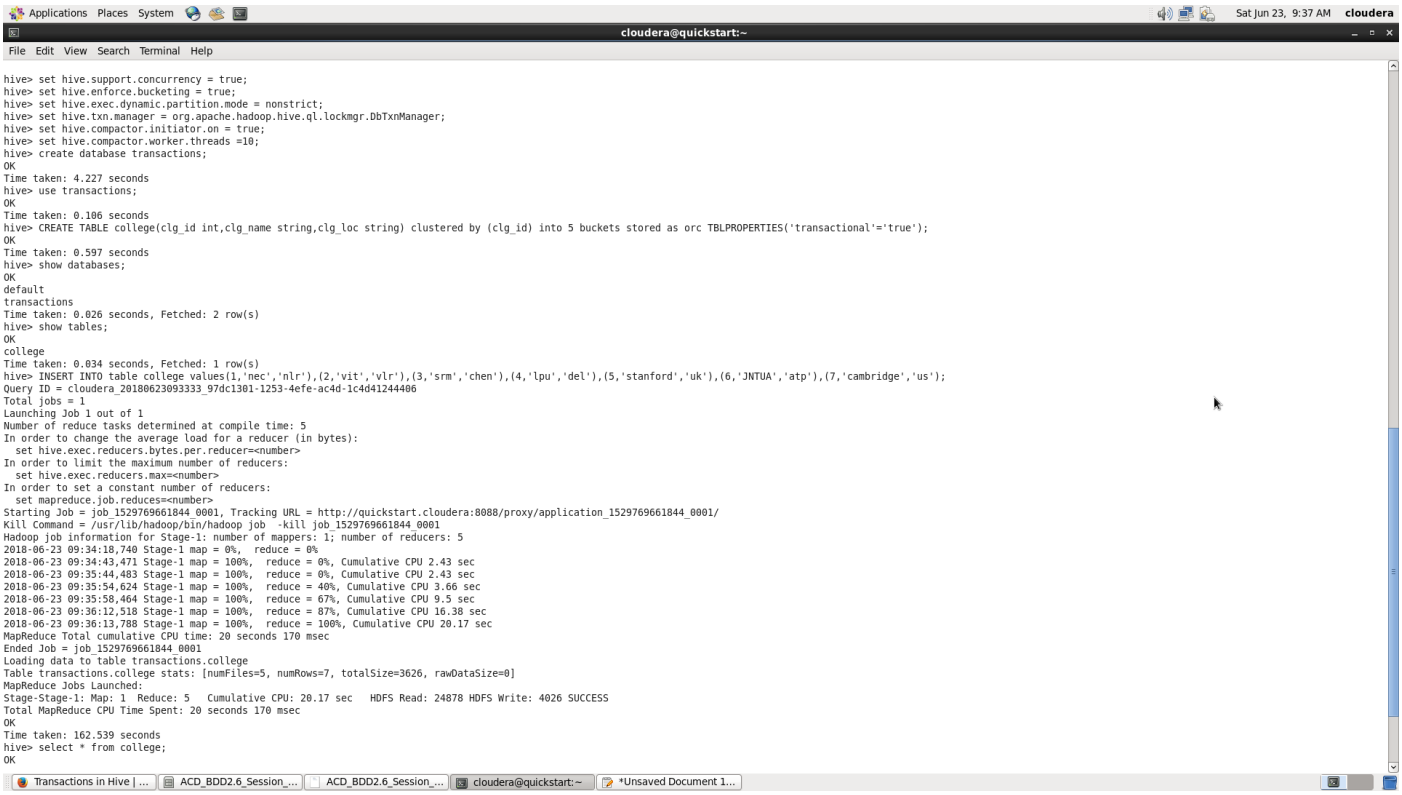
Session 9: Advance Hive

Creating a Table That Supports Hive Transactions

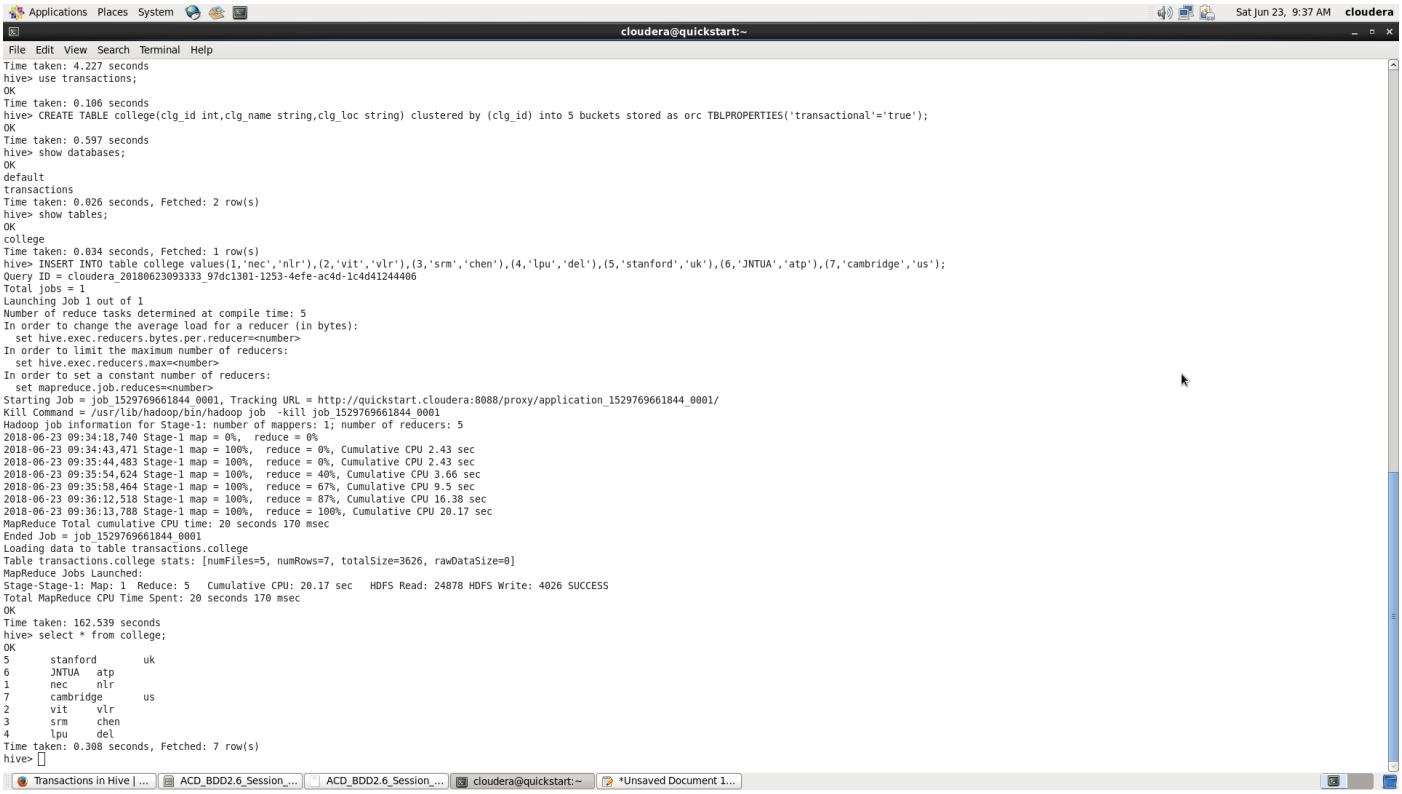
```
CREATE TABLE college(clg_id int,clg_name string,clg_loc string) clustered by (clg_id)
into 5 buckets stored as orc TBLPROPERTIES('transactional'='true');
```

Inserting Data into a Hive Table

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



Session 9: Advance Hive



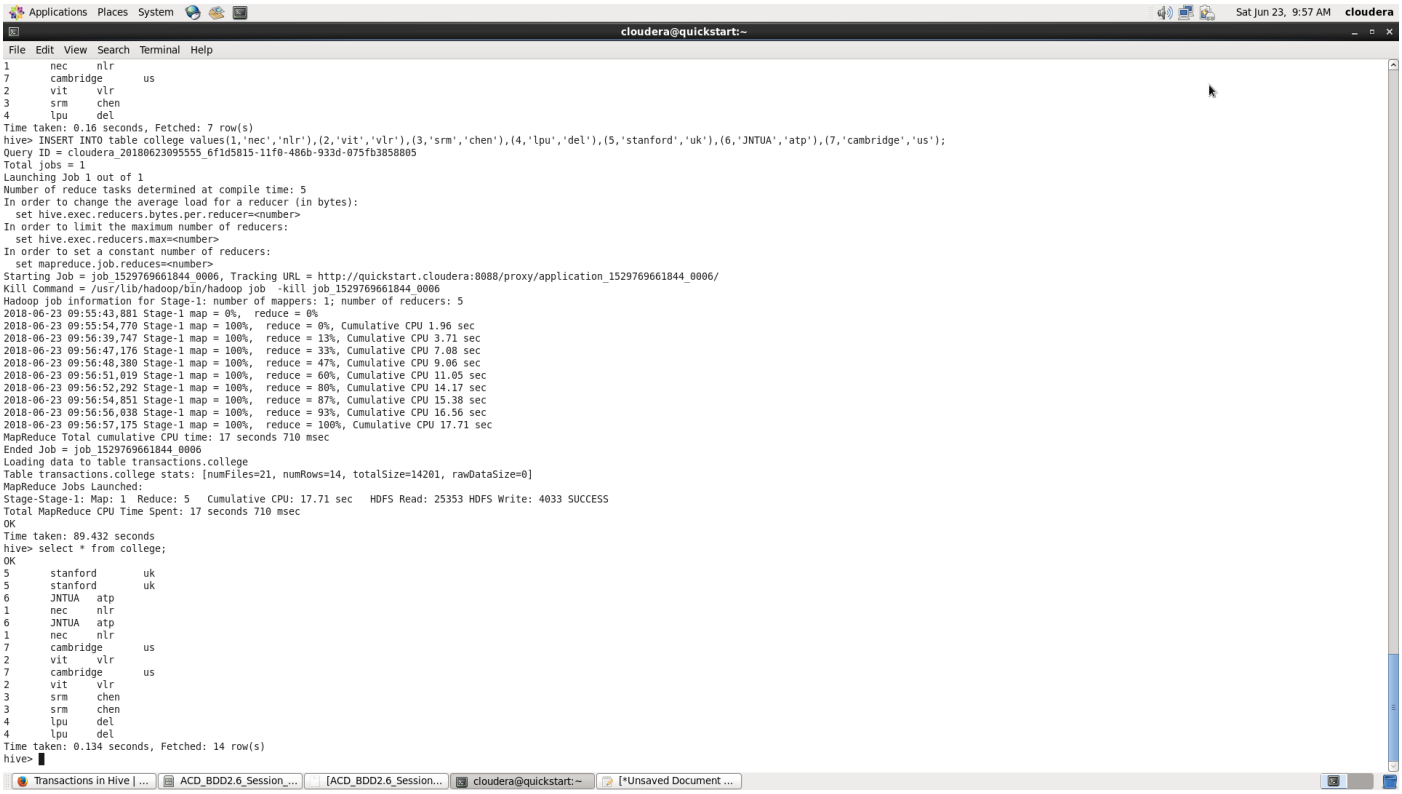
Updating the Data in Hive Table

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

Now let's perform the update operation on Non bucketed column

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

Session 9: Advance Hive



Deleting a Row from Hive Table

```
delete from college where clg_id=5;
```

Session 9: Advance Hive