BIG DATA
DEVELOPMENT

ACADGILD

Session 08: Advanced Hive

Assignment 3

Problem Statement

Followed the steps given on Acadgild blogs: -

Link: https://acadgild.com/blog/transactions-in-hive/

Note: Hive 0.14 should be installed to implement the hive transaction property.

What is ACID?

ACID stands for Atomicity, Consistency, Isolation, and Durability.

Atomicity means, a transaction should complete successfully or else it should fail completely i.e. it should not be left partially. Consistency ensures that any transaction will bring the database from one valid state to another state. Isolation states that every transaction should be independent of each other i.e. one transaction should not affect another. And Durability states that if a transaction is completed, it should be preserved in the database even if the machine state is lost or a system failure might occur.

These ACID properties are essential for a transaction and every transaction should ensure that these properties are met.

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

There are numerous limitations with the present transactions available in Hive 0.14. ORC is the file format supported by Hive transaction. It is now essential to have ORC file format for performing transactions in Hive. The table needs to be bucketed in order to support transactions.

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:

hive> set hive.support.concurrency=true;

hive> set hive.enforce.bucketing=true;

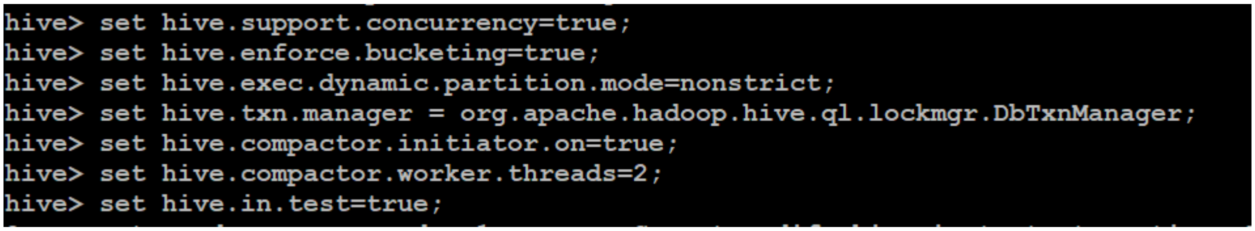
hive> set hive.exec.dynamic.partition.mode=nonstrict;

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

hive> set hive.compactor.initiator.on=true;

hive> set hive.compactor.worker.threads=2;

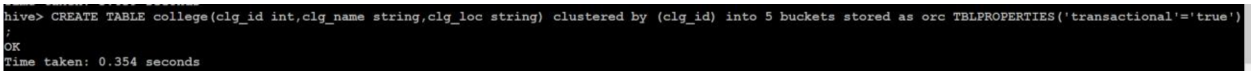
hive> set hive.in.test=true;



```
hive> set hive.support.concurrency=true;
hive> set hive.enforce.bucketing=true;
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> set hive.txn.manager = org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;
hive> set hive.compactor.initiator.on=true;
hive> set hive.compactor.worker.threads=2;
hive> set hive.in.test=true;
```

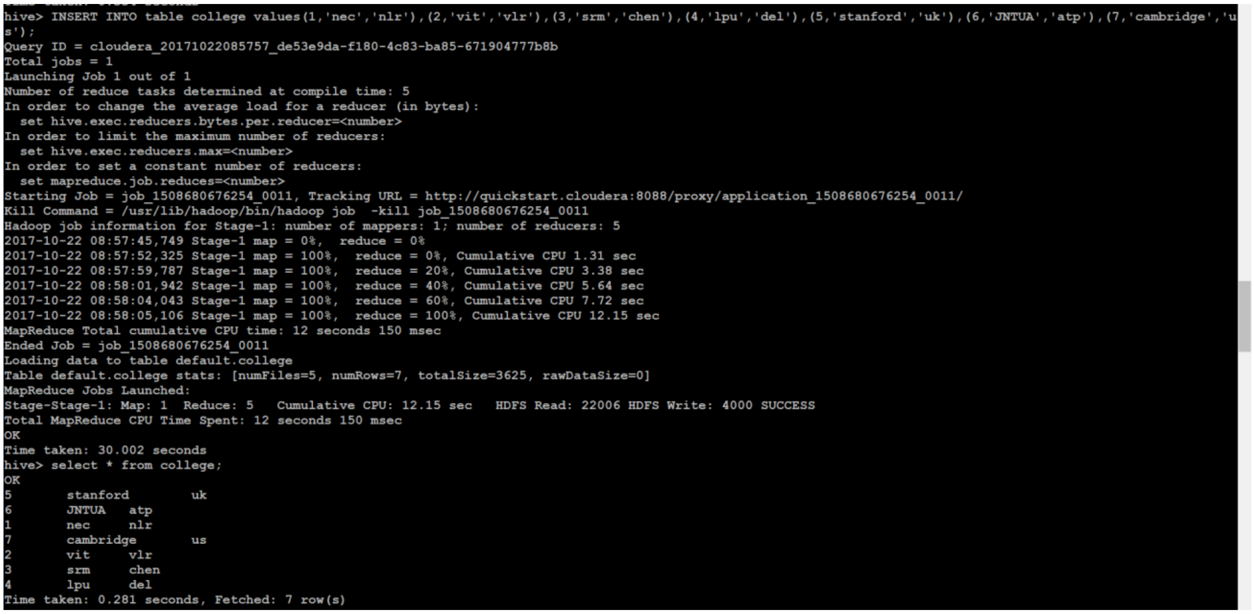
If the above properties are not set properly, the ‘Insert’ operation will work but ‘Update’ and ‘Delete’ will not work and you will receive the following error:
FAILED: SemanticException [Error 10294]: Attempt to do update or delete usingtransaction manager thatdoes not support these operations.

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');



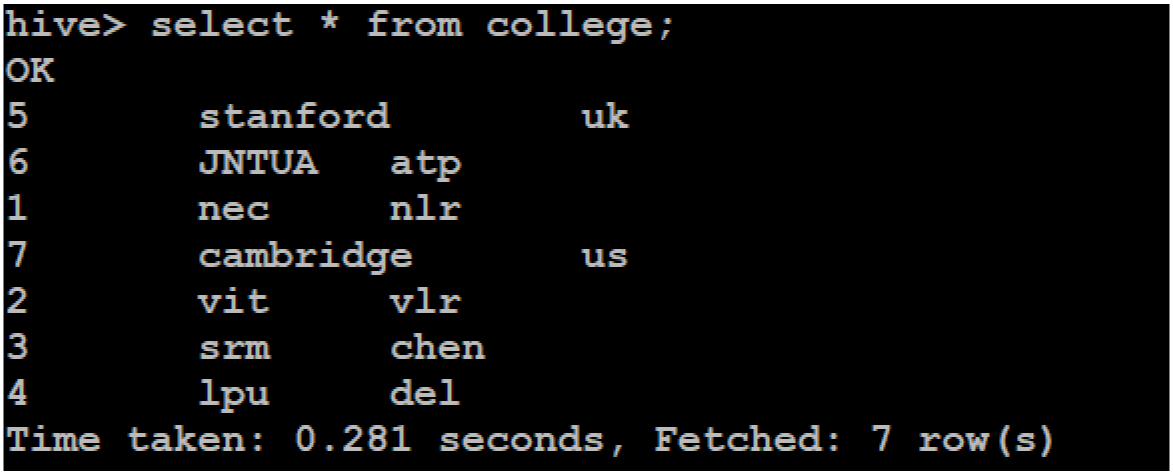
```
hive> CREATE TABLE college(clg_id int,clg_name string,clg_loc string) clustered by (clg_id) into 5 buckets stored as orc TBLPROPERTIES('transactional'='true')
;
OK
Time taken: 0.354 seconds
```

```
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');
```



```
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','u
s');
Query ID = cloudera_20171022085757_de53e9da-f180-4c83-ba85-671904777b8b
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1508680676254_0011, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1508680676254_0011/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1508680676254_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 5
2017-10-22 08:57:45,749 Stage-1 map = 0%,  reduce = 0%
2017-10-22 08:57:52,325 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.31 sec
2017-10-22 08:57:59,787 Stage-1 map = 100%,  reduce = 20%, Cumulative CPU 3.38 sec
2017-10-22 08:58:01,942 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 5.64 sec
2017-10-22 08:58:04,043 Stage-1 map = 100%,  reduce = 60%, Cumulative CPU 7.72 sec
2017-10-22 08:58:05,106 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 12.15 sec
MapReduce Total cumulative CPU time: 12 seconds 150 msec
Ended Job = job_1508680676254_0011
Loading data to table default.college
Table default.college stats: [numFiles=5, numRows=7, totalSize=3625, rawDataSize=0]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 5   Cumulative CPU: 12.15 sec   HDFS Read: 22006 HDFS Write: 4000 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 150 msec
OK
Time taken: 30.002 seconds
hive> select * from college;
OK
5       stanford        uk
6       JNTUA   atp
1       nec     nlr
7       cambridge       us
2       vit     vlr
3       srm     chen
4       lpu     del
Time taken: 0.281 seconds, Fetched: 7 row(s)
```

```
hive> select * from college;

OK

5       stanford        uk

6       JNTUA   atp

1       nec     nlr

7       cambridge       us

2       vit     vlr

3       srm     chen

4       lpu     del
```

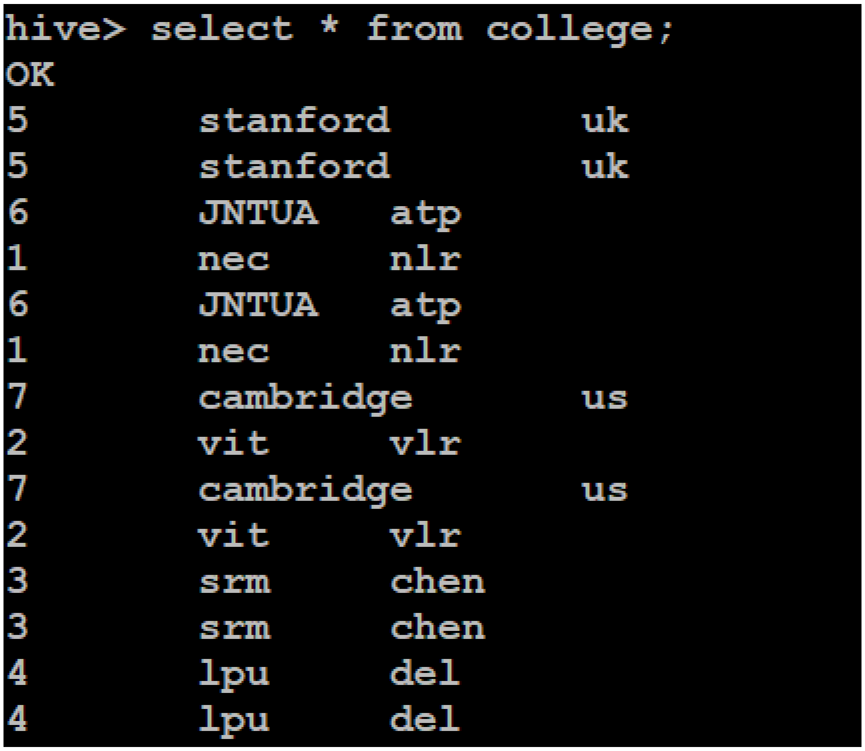


```
hive> select * from college;
OK
5       stanford        uk
6       JNTUA   atp
1       nec     nlr
7       cambridge       us
2       vit     vlr
3       srm     chen
4       lpu     del
Time taken: 0.281 seconds, Fetched: 7 row(s)
```

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');

From the above image, we can see that the data has been inserted successfully into the table.
Now if we try to re-insert the same data again, it will be appended to the previous data as shown below:

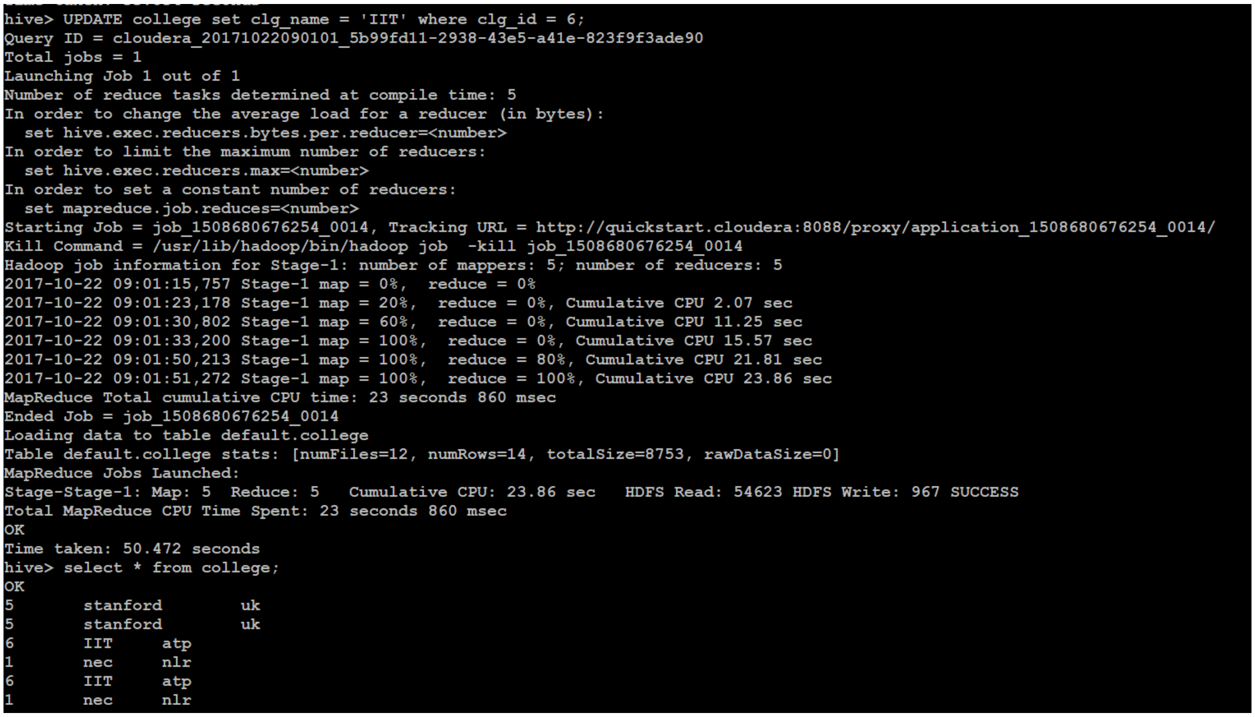
```
hive> select * from college;

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
```



```
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
```

hive> UPDATE college set clg_name = 'IIT' where clg_id = 6;

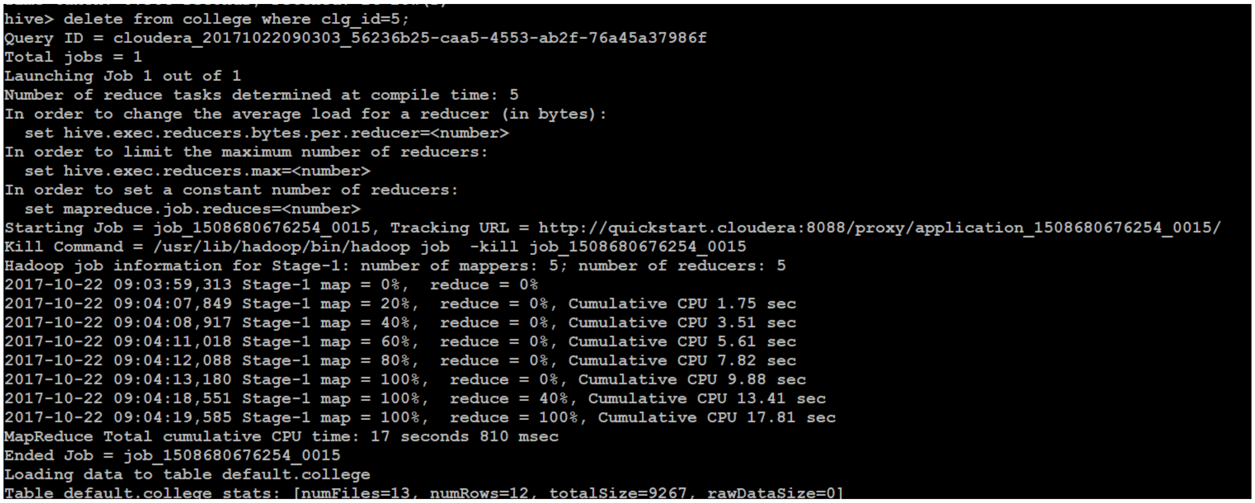
The above command is used to update a row in Hive table.



```
hive> UPDATE college set clg_name = 'IIT' where clg_id = 6;
Query ID = cloudera_20171022090101_5b99fd11-2938-43e5-a41e-823f9f3ade90
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1508680676254_0014, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1508680676254_0014/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1508680676254_0014
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2017-10-22 09:01:15,757 Stage-1 map = 0%,  reduce = 0%
2017-10-22 09:01:23,178 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 2.07 sec
2017-10-22 09:01:30,802 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 11.25 sec
2017-10-22 09:01:33,200 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 15.57 sec
2017-10-22 09:01:50,213 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 21.81 sec
2017-10-22 09:01:51,272 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 23.86 sec
MapReduce Total cumulative CPU time: 23 seconds 860 msec
Ended Job = job_1508680676254_0014
Loading data to table default.college
Table default.college stats: [numFiles=12, numRows=14, totalSize=8753, rawDataSize=0]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 23.86 sec   HDFS Read: 54623 HDFS Write: 967 SUCCESS
Total MapReduce CPU Time Spent: 23 seconds 860 msec
OK
Time taken: 50.472 seconds
hive> select * from college;
OK
5       stanford        uk
5       stanford        uk
6       IIT     atp
1       nec     nlr
6       IIT     atp
1       nec     nlr
```

hive> delete from college where clg_id=5;

The above command will delete a single row in the Hive table.



```
hive> delete from college where clg_id=5;
Query ID = cloudera_20171022090303_56236b25-caa5-4553-ab2f-76a45a37986f
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1508680676254_0015, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1508680676254_0015/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1508680676254_0015
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2017-10-22 09:03:59,313 Stage-1 map = 0%,  reduce = 0%
2017-10-22 09:04:07,849 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 1.75 sec
2017-10-22 09:04:08,917 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 3.51 sec
2017-10-22 09:04:11,018 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 5.61 sec
2017-10-22 09:04:12,088 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 7.82 sec
2017-10-22 09:04:13,180 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.88 sec
2017-10-22 09:04:18,551 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 13.41 sec
2017-10-22 09:04:19,585 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 17.81 sec
MapReduce Total cumulative CPU time: 17 seconds 810 msec
Ended Job = job_1508680676254_0015
Loading data to table default.college
Table default.college stats: [numFiles=13, numRows=12, totalSize=9267, rawDataSize=0]
```

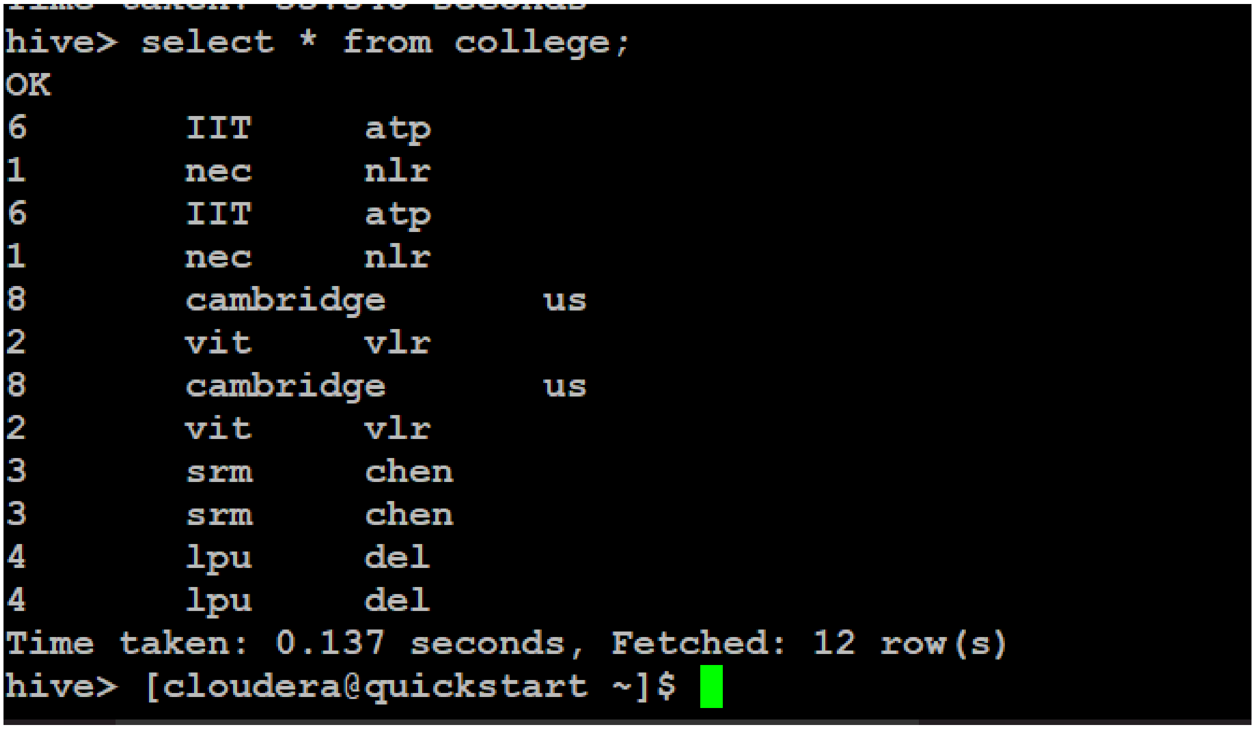
hive> select * from college;

```
OK

6	IIT	atp

1	nec	nlr

6	IIT	atp

1	nec	nlr

8	cambridge	us

2	vit	vlr

8	cambridge	us

2	vit	vlr

3	srm	chen
```

3 srm chen

4 lpu del

4 lpu del



```
hive> select * from college;
OK
6       IIT     atp
1       nec     nlr
6       IIT     atp
1       nec     nlr
8       cambridge       us
2       vit     vlr
8       cambridge       us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
Time taken: 0.137 seconds, Fetched: 12 row(s)
hive> [cloudera@quickstart ~]$
```