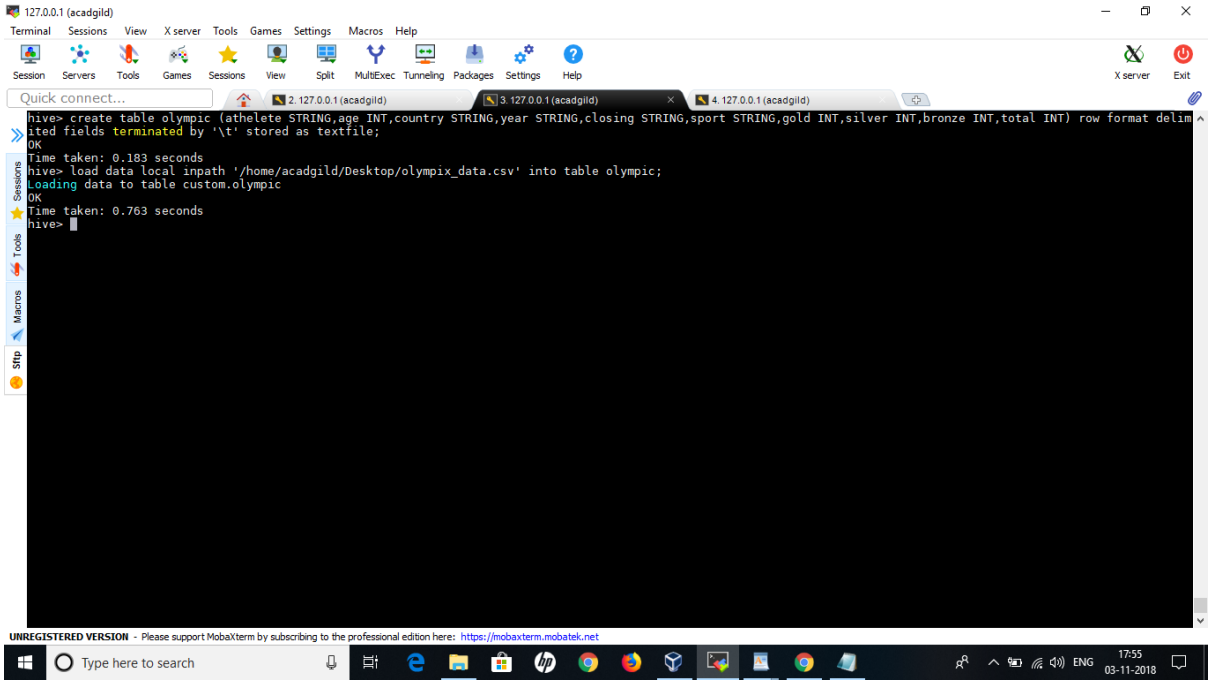
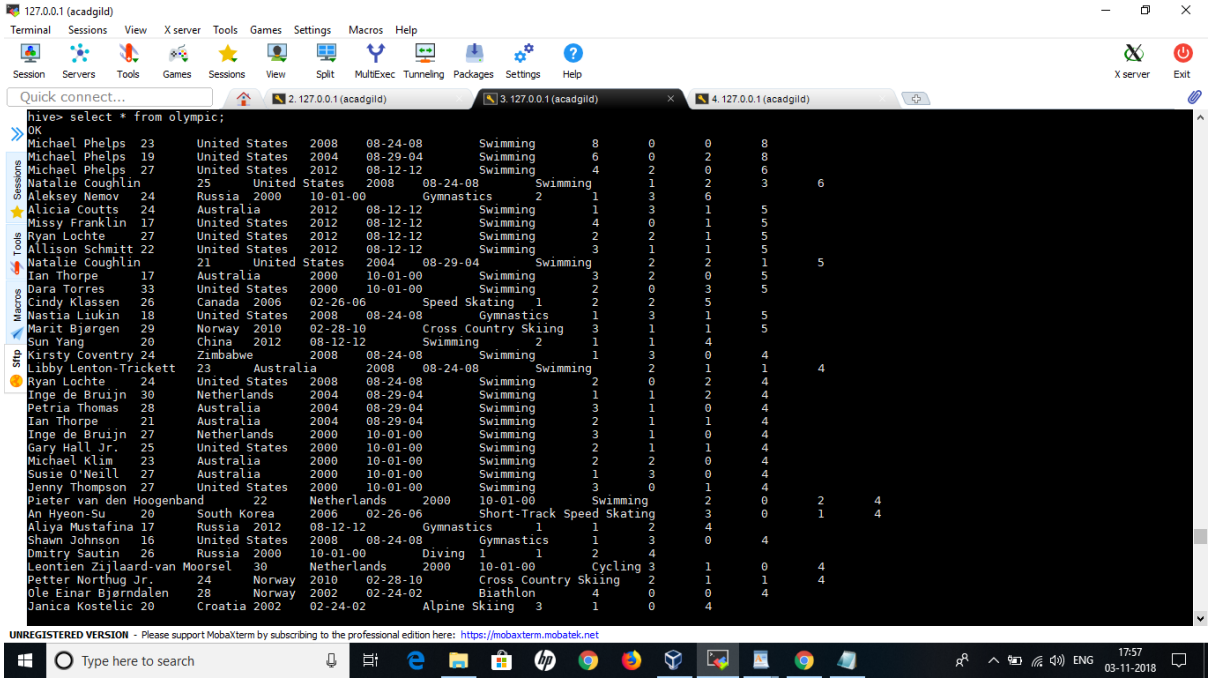
Assignment 9.1 Advance Hive.

Table Creation:

create table olympic (athelete STRING,age INT,country STRING,year STRING,closing STRING,sport STRING,gold INT,silver INT,bronze INT,total INT) row format delimited fields terminated by '\t' stored as textfile;

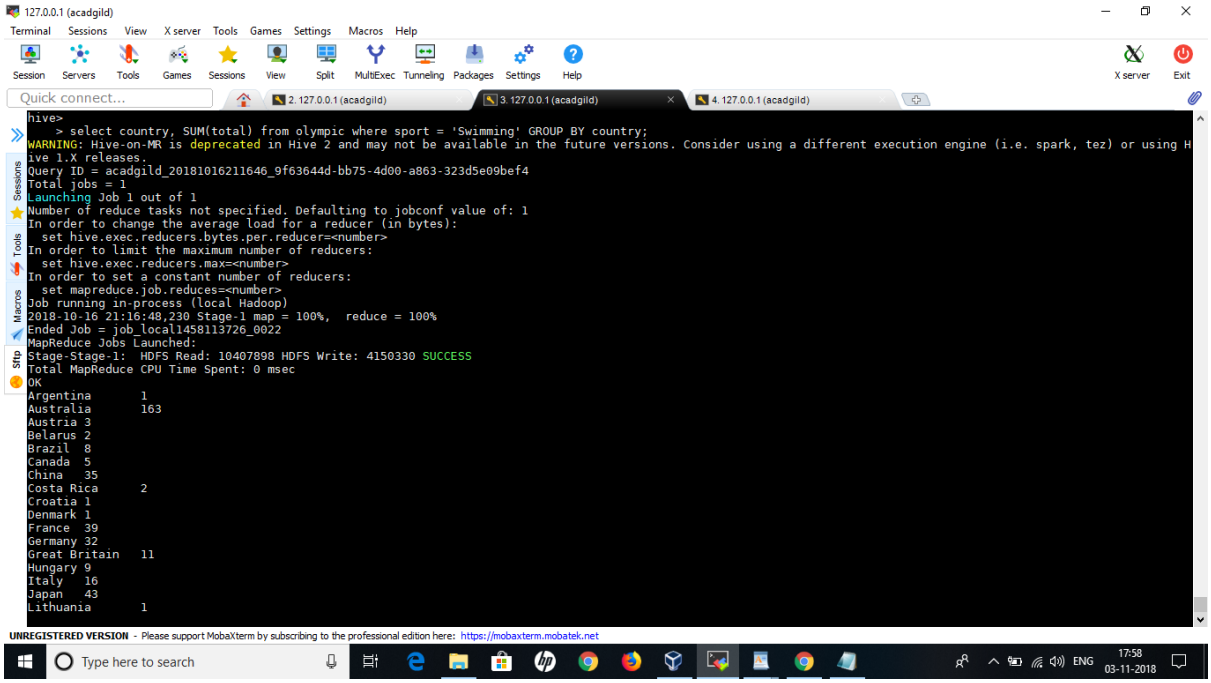




Task 1

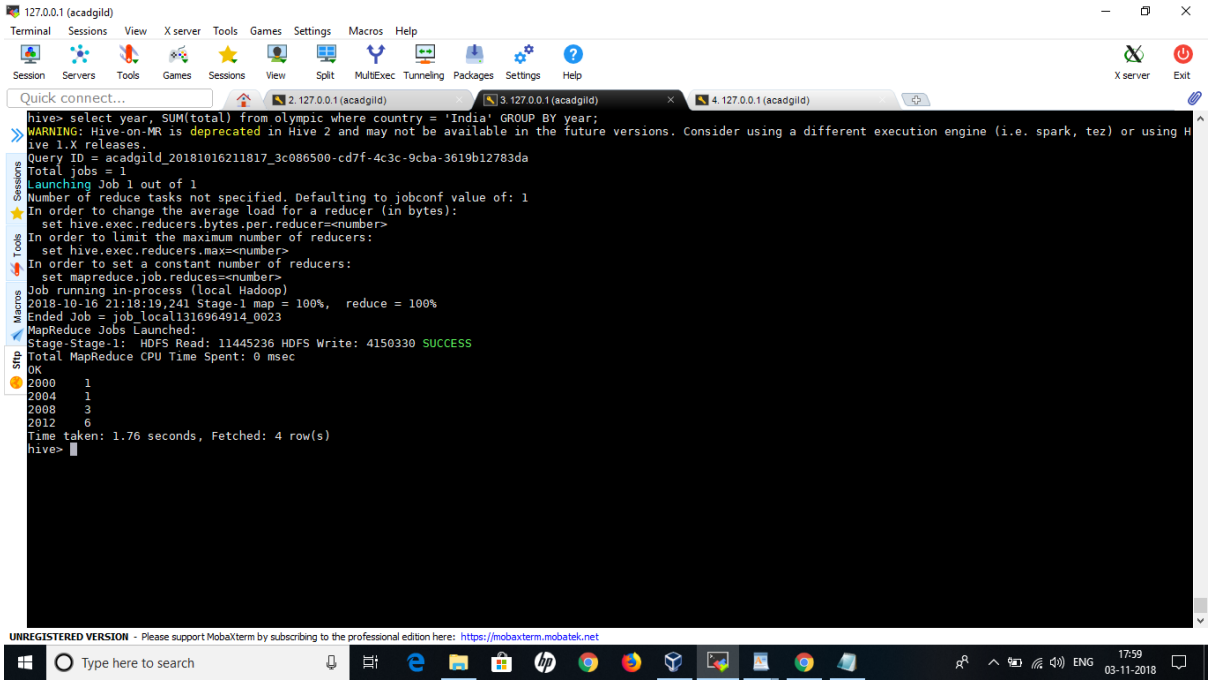
1. Write a Hive program to find the number of medals won by each country in swimming.

Solution: *select country, SUM(total) from olympic where sport = 'Swimming' GROUP BY country;*



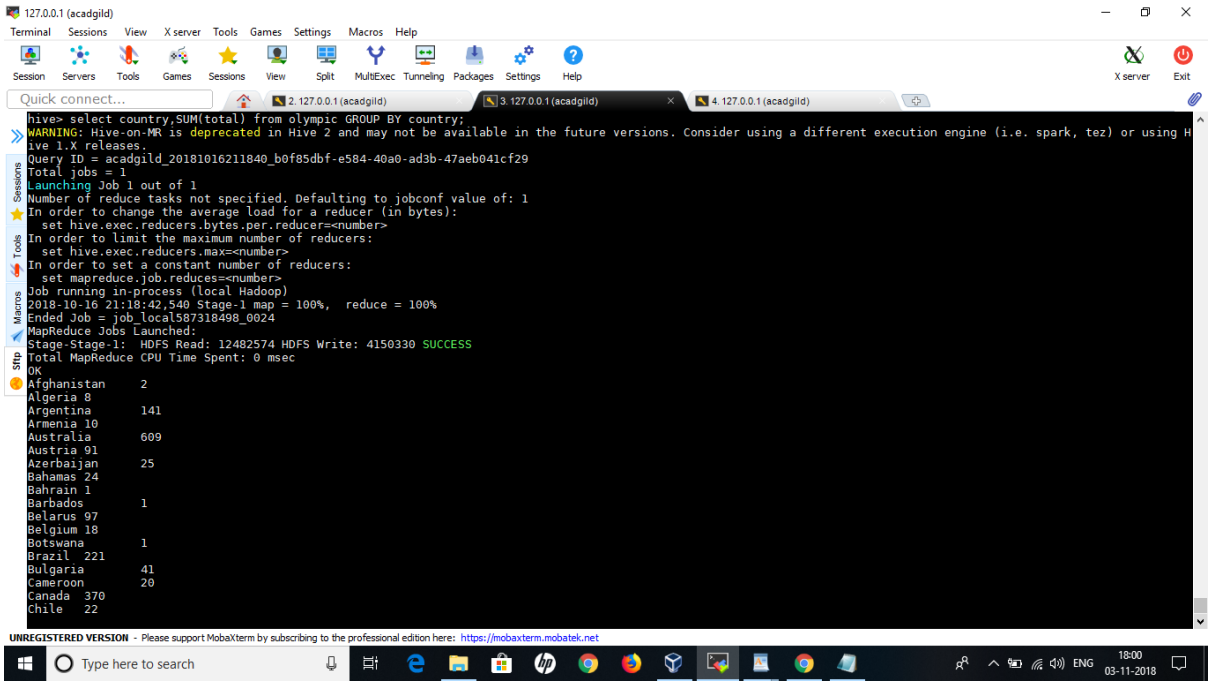
2. Write a Hive program to find the number of medals that India won year wise.

Solution: *select year, SUM(total) from olympic where country = 'India' GROUP BY year;*



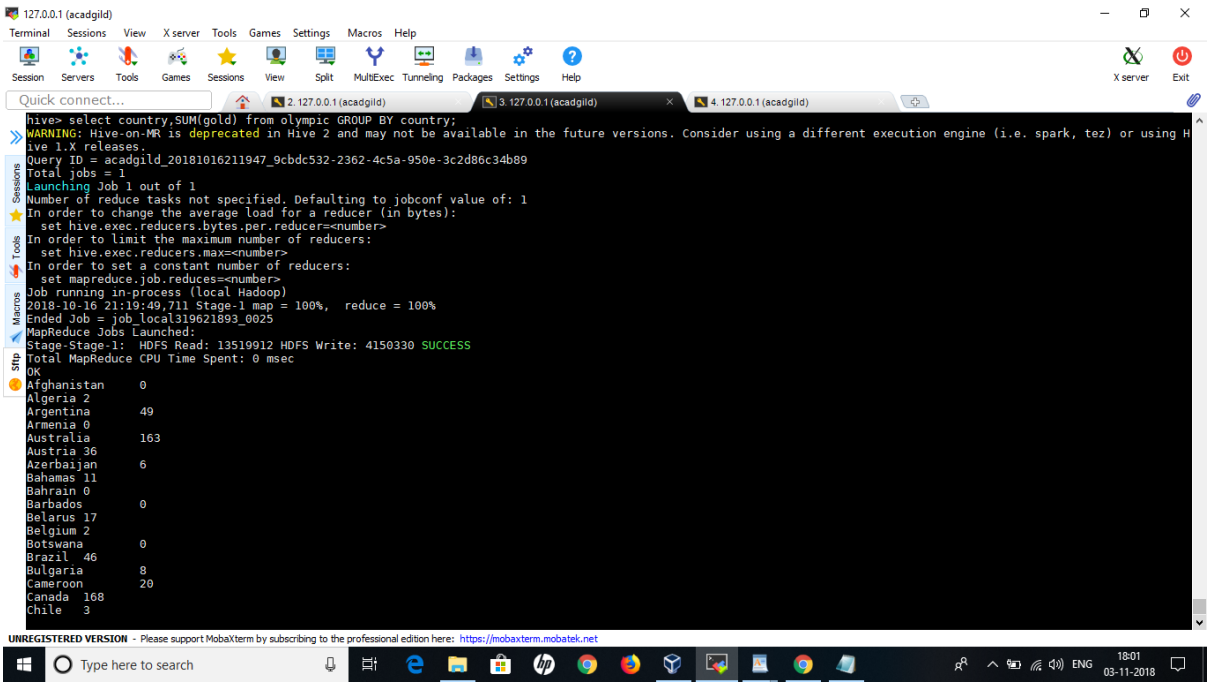
3. Write a Hive Program to find the total number of medals each country won.

Solution: *select country,SUM(total) from olympic GROUP BY country;*



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

Solution: *select country,SUM(gold) from olympic GROUP BY country;*



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.

Solution:

```
package com.udfconcatws;
import java.util.ArrayList;
import org.apache.hadoop.hive.ql.exec.UDF;
public class concatws extends UDF{
public String evaluate (String separator, ArrayList<String> array) {
StringBuffer sBuffer;
if (array == null) {
return null;
}
sBuffer = new StringBuffer();
sBuffer.append(array.get(0));
for (int i=1; i < array.size(); i++) {
sBuffer.append(separator);
sBuffer.append(array.get(i));
}
return sBuffer.toString();
}
}
```

eclipse-workspace - HiveUDF/src/com/udfconcatws/concatws.java - Eclipse

File Edit Source Refactor Refactor Navigate Search Project Run Window Help

Quick Access

Package Expl

- HbaseApi
  - JRE System Library [Jav
  - src
    - com.hbase.api
  - Referenced Libraries
- HiveUDF
  - JRE System Library [Jav
  - src
    - com.udfconcatws
      - concatws.java
  - Referenced Libraries

concatws.java

```
1  package com.udfconcatws;
2
3  import java.util.ArrayList;
4  import org.apache.hadoop.hive.ql.exec.UDF;
5
6  public class concatws extends UDF{
7  public String evaluate (String separator, ArrayList<String> array) {
8  StringBuffer sBuffer;
9  if (array == null) {
10 return null;
11 }
12 sBuffer = new StringBuffer();
13 sBuffer.append(array.get(0));
14 for (int i=1; i < array.size(); i++) {
15 sBuffer.append(separator);
16 sBuffer.append(array.get(i));
17 }
18
19 return sBuffer.toString();
20
21 }
22 }
```

Problems Declaration Console

No consoles to display at this time.

com.udfconcatws.concatws.java - HiveUDF/src

acadgild@localhost:~

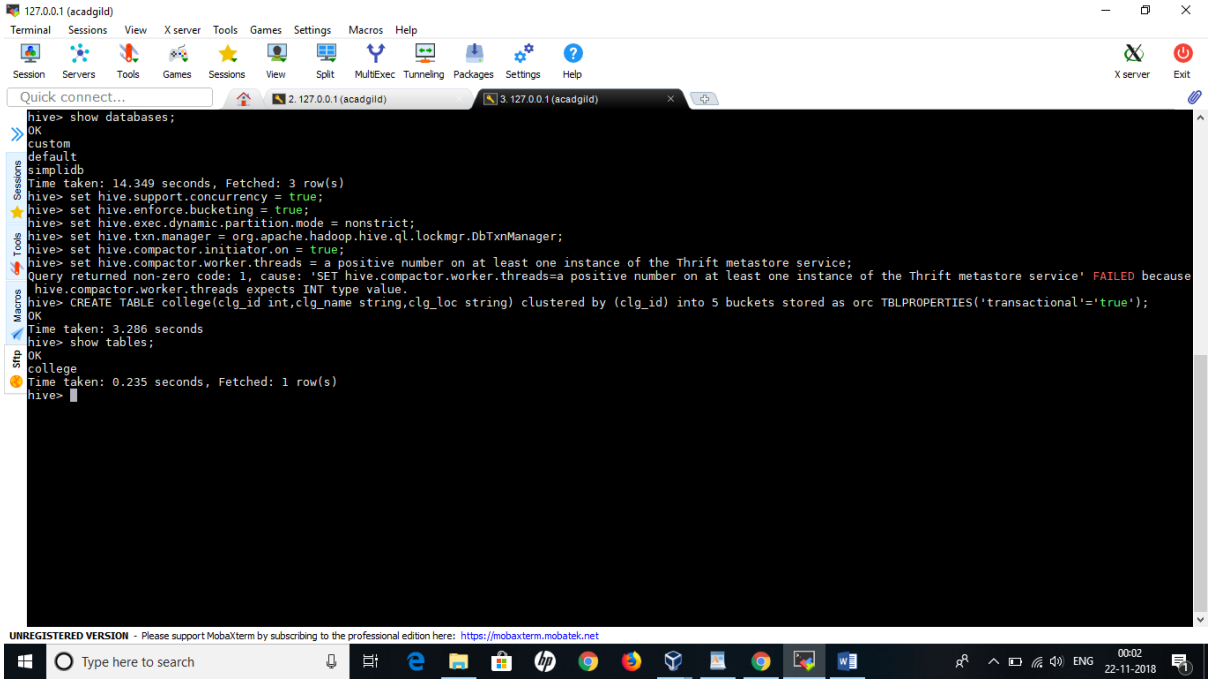
File Edit View Search Terminal Help

```
hive> add jar /home/acadgild/Hiveudfconcat_ws.jar;
Added [/home/acadgild/Hiveudfconcat_ws.jar] to class path
Added resources: [/home/acadgild/Hiveudfconcat_ws.jar]
hive> create temporary function concat_ws_extended as 'com.udfconcatws.concatws';
OK
Time taken: 0.043 seconds
hive>
```

Task 3: ACID Properties in Hive:

Solution: Creating a Table That Supports ACID Hive Transactions

CMD: *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 the Hive Table:

CMD: *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','us');
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using H
ive 1.X releases.
Query ID = acadgild_20181119043438_b055b3b9-9467-4133-936c-03b4180d4e30
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>
Job running in-process (local Hadoop)
2018-11-19 04:34:48,585 Stage-1 map = 0%,  reduce = 0%
2018-11-19 04:34:49,668 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local364725743_0001
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 492 HDFS Write: 12595 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 14.176 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: 0.666 seconds, Fetched: 7 row(s)
hive>
```

Re-insert the Same Value again:

```
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');
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using H
ive 1.X releases.
Query ID = acadgild_20181119043556_5ff02460-8c61-46ae-97bc-e48ded56a43f
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>
Job running in-process (local Hadoop)
2018-11-19 04:35:58,781 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local772997562_0002
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 29172 HDFS Write: 37093 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 5.141 seconds
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
```

Update the Data in Hive Table:

CMD: *UPDATE college set clg_name = 'IIT' where clg_id = 6;*

```
hive> UPDATE college set clg_name = 'IIT' where clg_id = 6;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using H
ive 1.X releases.
Query ID = acadgild_20181119043839_94f066ed-5dc1-408d-a2b8-19ffec23f6b7
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>
Job running in-process (local Hadoop)
2018-11-19 04:38:42,142 Stage-1 map = 100%,  reduce = 0%
2018-11-19 04:38:43,172 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local197971906_0003
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 207461 HDFS Write: 85301 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 5.936 seconds
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
```

Delete the Row from Hive Table:

CMD: *delete from college where clg_id=5;*

```
hive> delete from college where clg_id=5;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using H
ive 1.X releases.
Query ID = acadgild_20181119044014_b320c06d-cf7c-4f81-9854-c1f1b34178c8
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>
Job running in-process (local Hadoop)
2018-11-19 04:40:16,853 Stage-1 map = 100%,  reduce = 0%
2018-11-19 04:40:17,874 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local651447377_0004
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 397249 HDFS Write: 94655 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 4.374 seconds
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: 0.446 seconds, Fetched: 12 row(s)
```