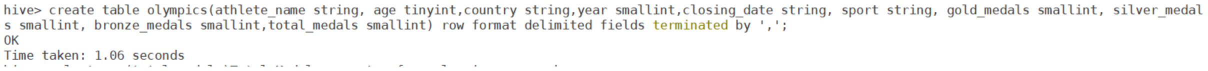
Assignment 9.1

Advance Hive

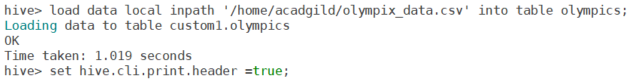
Task 1 1. Write a Hive program to find the number of medals won by each country in swimming.

We have created olympics table with all required fields as columns and as we are using csv file ‘olympix_data’ to load data into this table, we are using ‘,’ as delimiter to separate column values:



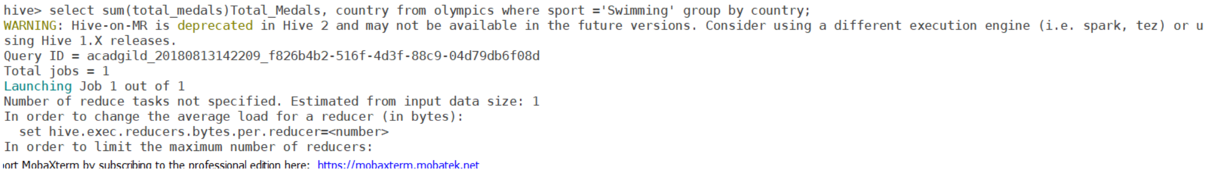
```
hive> create table olympics(athlete_name string, age tinyint,country string,year smallint,closing_date string, sport string, gold_medals smallint, silver_medal
s smallint, bronze_medals smallint,total_medals smallint) row format delimited fields terminated by ',';
OK
Time taken: 1.06 seconds
```

Then we have loaded data from ‘olympix_data.csv’ file present in local system as shown in below screenshot.

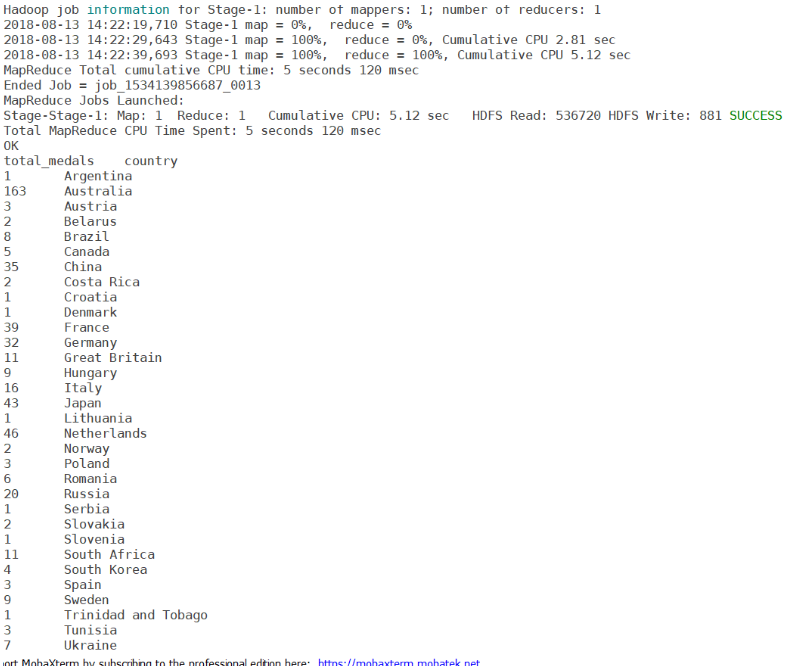


```
hive> load data local inpath '/home/acadgild/olympix_data.csv' into table olympics;
Loading data to table custom1.olympics
OK
Time taken: 1.019 seconds
hive> set hive.cli.print.header =true;
```

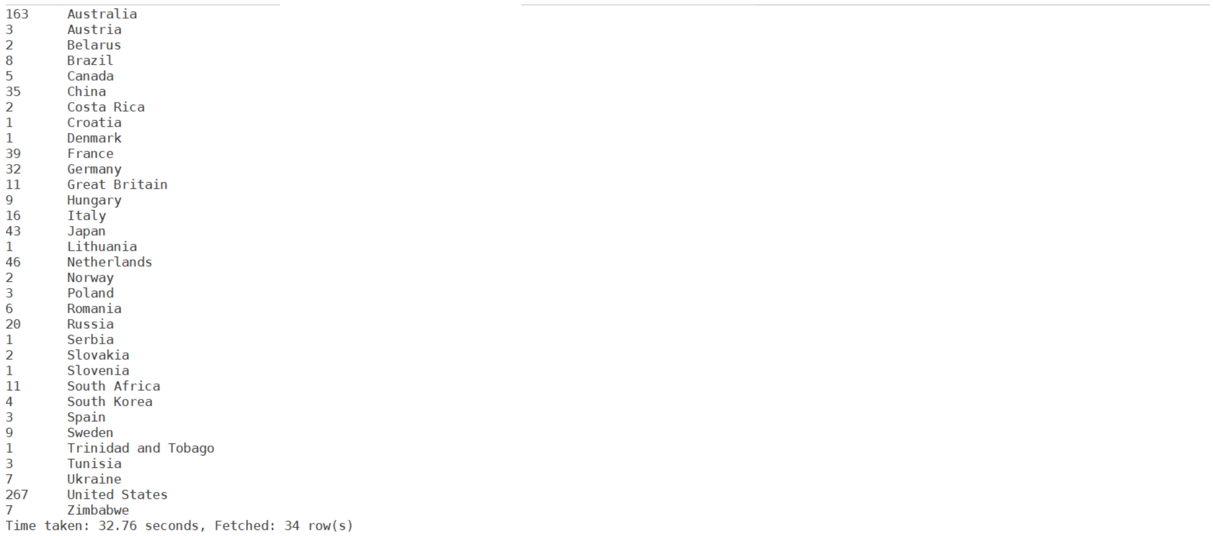
Hive Query 1: Here we are using GROUP BY clause to group all the records by using column ‘country’ and sum function to calculate total number of medals for a country in Swimming sport.



```
hive> select sum(total_medals)Total_Medals, country from olympics where sport ='Swimming' group by country;
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 u
sing Hive 1.X releases.
Query ID = acadgild_20180813142209_f826b4b2-516f-4d3f-88c9-04d79db6f08d
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
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:
ort MobaXterm by subscribing to the professional edition here:  https://mobaxterm.mobatek.net
```



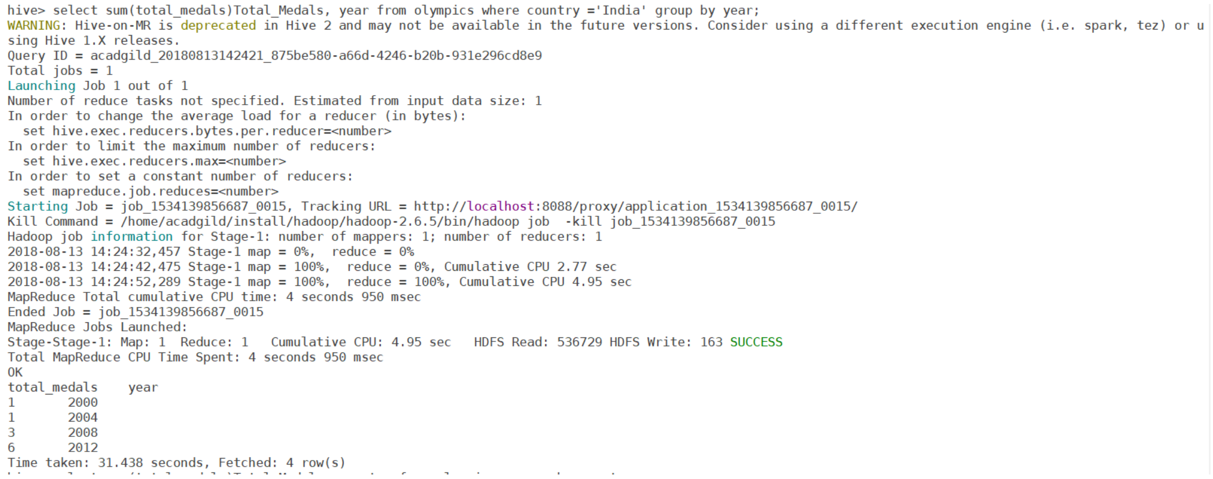
```
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-13 14:22:19,710 Stage-1 map = 0%,  reduce = 0%
2018-08-13 14:22:29,643 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.81 sec
2018-08-13 14:22:39,693 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.12 sec
MapReduce Total cumulative CPU time: 5 seconds 120 msec
Ended Job = job_1534139856687_0013
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.12 sec   HDFS Read: 536720 HDFS Write: 881 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 120 msec
OK
total_medals    country
1       Argentina
163     Australia
3       Austria
2       Belarus
8       Brazil
5       Canada
35      China
2       Costa Rica
1       Croatia
1       Denmark
39      France
32      Germany
11      Great Britain
9       Hungary
16      Italy
43      Japan
1       Lithuania
46      Netherlands
2       Norway
3       Poland
6       Romania
20      Russia
1       Serbia
2       Slovakia
1       Slovenia
11      South Africa
4       South Korea
3       Spain
9       Sweden
1       Trinidad and Tobago
3       Tunisia
7       Ukraine
ort MobaXterm by subscribing to the professional edition here:  https://mobaxterm.mobatek.net
```



```
163     Australia
3       Austria
2       Belarus
8       Brazil
5       Canada
35      China
2       Costa Rica
1       Croatia
1       Denmark
39      France
32      Germany
11      Great Britain
9       Hungary
16      Italy
43      Japan
1       Lithuania
46      Netherlands
2       Norway
3       Poland
6       Romania
20      Russia
1       Serbia
2       Slovakia
1       Slovenia
11      South Africa
4       South Korea
3       Spain
9       Sweden
1       Trinidad and Tobago
3       Tunisia
7       Ukraine
267     United States
7       Zimbabwe
Time taken: 32.76 seconds, Fetched: 34 row(s)
```

2. Write a Hive program to find the number of medals that India won year wise. Here we are using GROUP BY clause to group all the records by using column ‘year’ and sum function to calculate total number of medals for a year in India country.

Hive Query 2: select sum(total_medals) Total_Medals ,year from olympics where country = 'India' group by year;



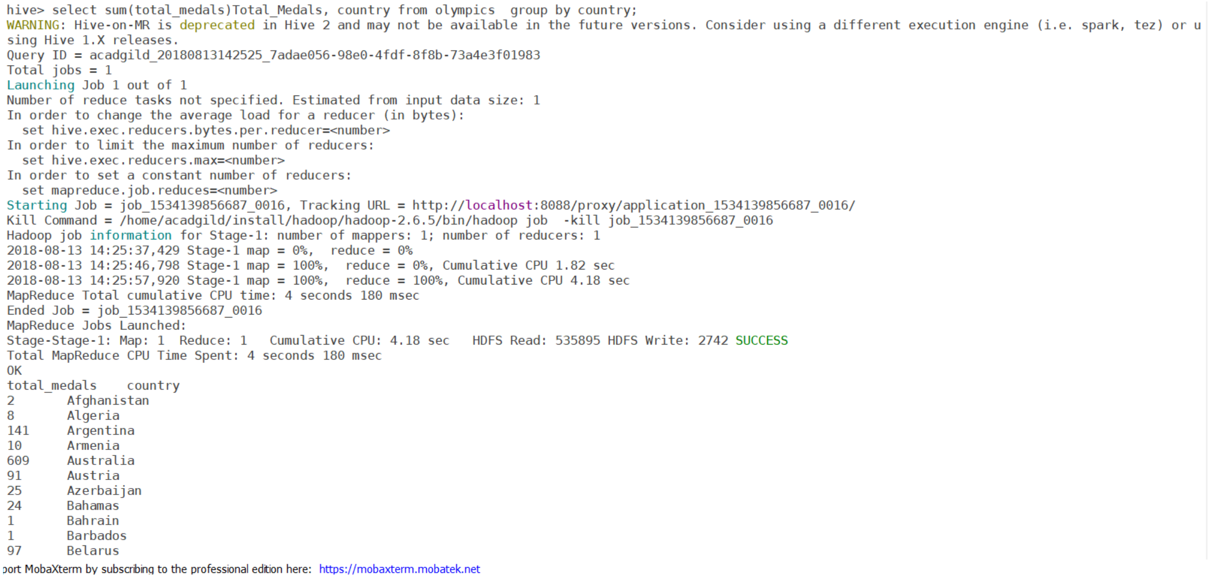
```
hive> select sum(total_medals)Total_Medals, year from olympics where country ='India' group by year;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or u
sing Hive 1.X releases.
Query ID = acadgild_20180813142421_875be580-a66d-4246-b20b-931e296cd8e9
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
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_1534139856687_0015, Tracking URL = http://localhost:8088/proxy/application_1534139856687_0015/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534139856687_0015
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-13 14:24:32,457 Stage-1 map = 0%,  reduce = 0%
2018-08-13 14:24:42,475 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.77 sec
2018-08-13 14:24:52,289 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.95 sec
MapReduce Total cumulative CPU time: 4 seconds 950 msec
Ended Job = job_1534139856687_0015
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.95 sec   HDFS Read: 536729 HDFS Write: 163 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 950 msec
OK
total_medals    year
1       2000
1       2004
3       2008
6       2012
Time taken: 31.438 seconds, Fetched: 4 row(s)
```

Output :

total_medals	year
1	2000
1	2004
3	2008
6	2012

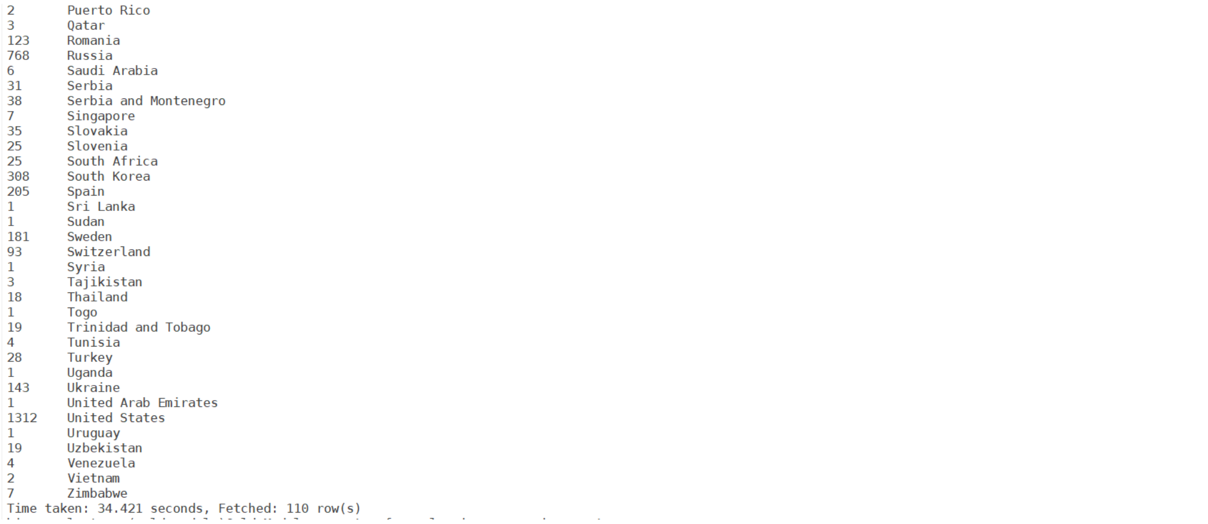
Assignment 9.1 Advance Hive 3. Write a Hive Program to find the total number of medals each country won. Here we are using GROUP BY clause to group all the records by using column ‘country’ and sum function to calculate total number of medals for a country.

Hive Query 3: select sum(total_medals) Total_Medals ,country from olympics group by country;



```
hive> select sum(total_medals)Total_Medals, country from olympics  group by country;
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 u
sing Hive 1.X releases.
Query ID = acadgild_20180813142525_7adae056-98e0-4fdf-8f8b-73a4e3f01983
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
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_1534139856687_0016, Tracking URL = http://localhost:8088/proxy/application_1534139856687_0016/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534139856687_0016
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-13 14:25:37,429 Stage-1 map = 0%,  reduce = 0%
2018-08-13 14:25:46,798 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.82 sec
2018-08-13 14:25:57,920 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.18 sec
MapReduce Total cumulative CPU time: 4 seconds 180 msec
Ended Job = job_1534139856687_0016
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.18 sec   HDFS Read: 535895 HDFS Write: 2742 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 180 msec
OK
total_medals    country
2       Afghanistan
8       Algeria
141     Argentina
10      Armenia
609     Australia
91      Austria
25      Azerbaijan
24      Bahamas
1       Bahrain
1       Barbados
97      Belarus
```

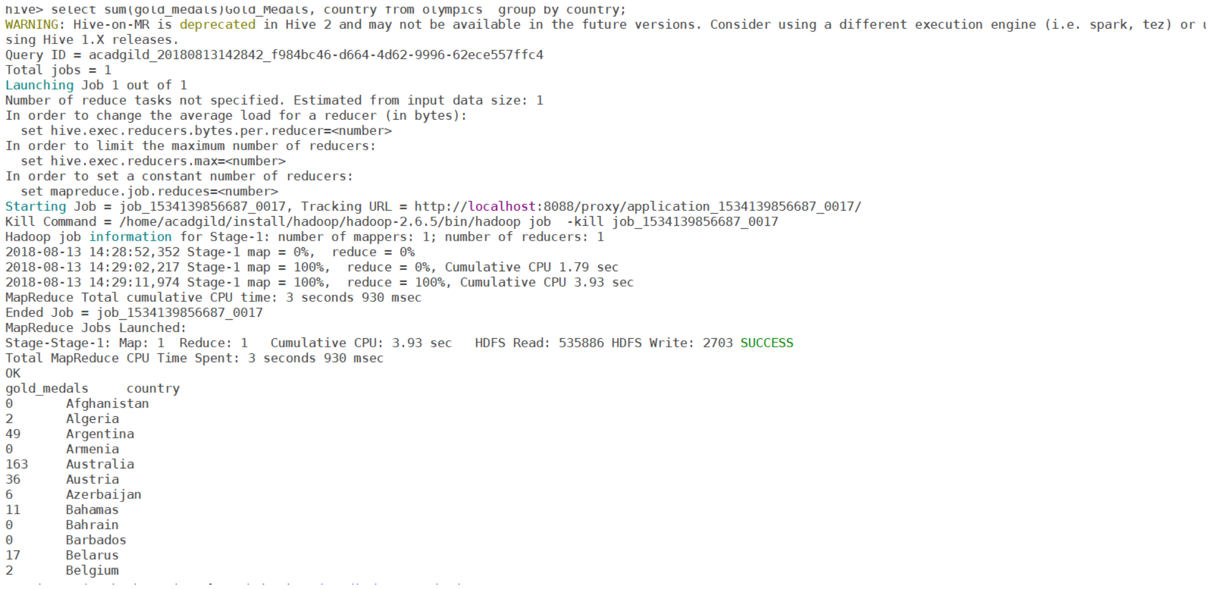
port MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net



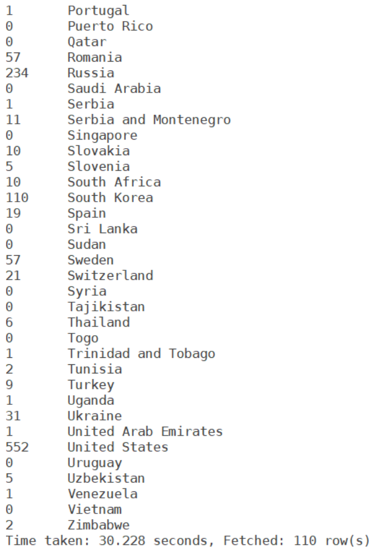
```
2       Puerto Rico
3       Qatar
123     Romania
768     Russia
6       Saudi Arabia
31      Serbia
38      Serbia and Montenegro
7       Singapore
35      Slovakia
25      Slovenia
25      South Africa
308     South Korea
205     Spain
1       Sri Lanka
1       Sudan
181     Sweden
93      Switzerland
1       Syria
3       Tajikistan
18      Thailand
1       Togo
19      Trinidad and Tobago
4       Tunisia
28      Turkey
1       Uganda
143     Ukraine
1       United Arab Emirates
1312    United States
1       Uruguay
19      Uzbekistan
4       Venezuela
2       Vietnam
7       Zimbabwe
Time taken: 34.421 seconds, Fetched: 110 row(s)
```

4. Write a Hive program to find the number of gold medals each country won. Here we are using GROUP BY clause to group all the records by using column ‘country’ and sum function to calculate total number of Gold medals for a country.

Hive Query 4 : select sum(gold_medals) Gold_Medals ,country from olympics group by country;



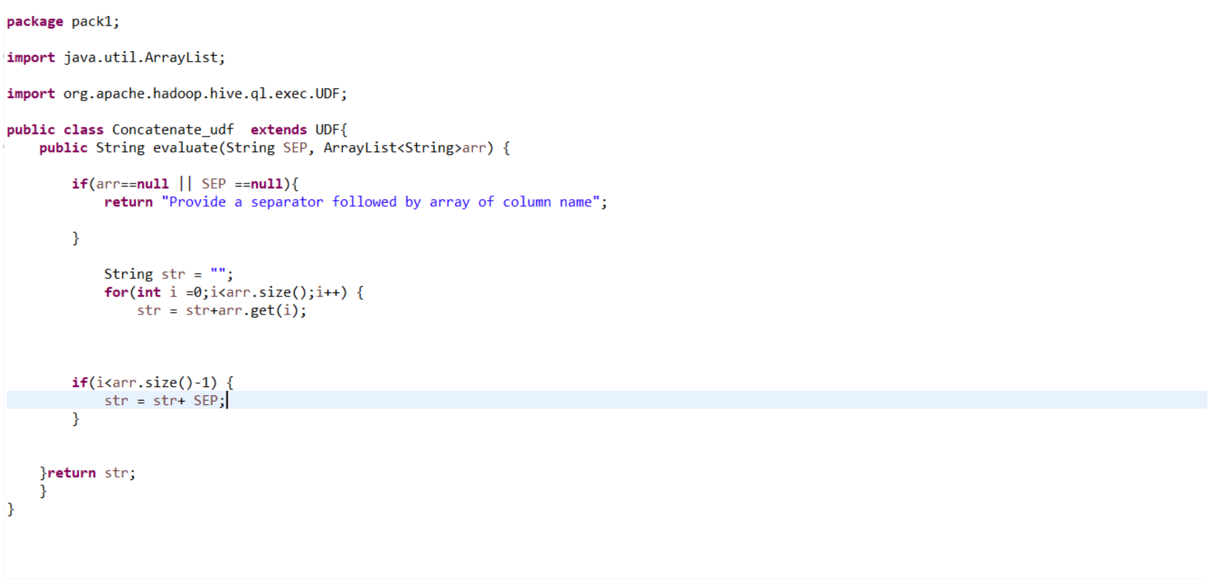
```
hive> select sum(gold_medals)Gold_Medals, country from olympics  group by country;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or u
sing Hive 1.X releases.
Query ID = acadgild_20180813142842_f984bc46-d664-4d62-9996-62ece557ffc4
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
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_1534139856687_0017, Tracking URL = http://localhost:8088/proxy/application_1534139856687_0017/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534139856687_0017
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-13 14:28:52,352 Stage-1 map = 0%,  reduce = 0%
2018-08-13 14:29:02,217 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.79 sec
2018-08-13 14:29:11,974 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.93 sec
MapReduce Total cumulative CPU time: 3 seconds 930 msec
Ended Job = job_1534139856687_0017
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.93 sec   HDFS Read: 535886 HDFS Write: 2703 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 930 msec
OK
gold_medals     country
0       Afghanistan
2       Algeria
49      Argentina
0       Armenia
163     Australia
36      Austria
6       Azerbaijan
11      Bahamas
0       Bahrain
0       Barbados
17      Belarus
2       Belgium
```



```
1       Portugal
0       Puerto Rico
0       Qatar
57      Romania
234     Russia
0       Saudi Arabia
1       Serbia
11      Serbia and Montenegro
0       Singapore
10      Slovakia
5       Slovenia
10      South Africa
110     South Korea
19      Spain
0       Sri Lanka
0       Sudan
57      Sweden
21      Switzerland
0       Syria
0       Tajikistan
6       Thailand
0       Togo
1       Trinidad and Tobago
2       Tunisia
9       Turkey
1       Uganda
31      Ukraine
1       United Arab Emirates
552     United States
0       Uruguay
5       Uzbekistan
1       Venezuela
0       Vietnam
2       Zimbabwe
Time taken: 30.228 seconds, Fetched: 110 row(s)
```

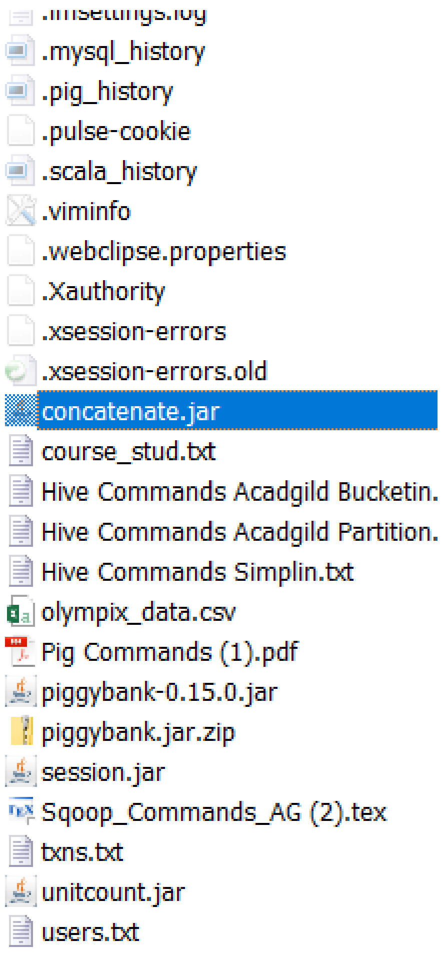
Task 2 : Write a hive UDF that implements functionality of string concat_ws(string SEP, array). This UDF will accept two arguments, one string and one array of string. It will return a single string where all the elements of the array are separated by the SEP

We have written a Java Program `Concatenate_udf.java`



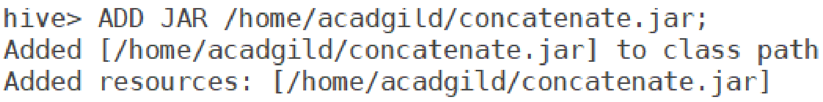
```
package pack1;

import java.util.ArrayList;

import org.apache.hadoop.hive.ql.exec.UDF;

public class Concatenate_udf  extends UDF{
    public String evaluate(String SEP, ArrayList<String>arr) {

        if(arr==null || SEP ==null){
            return "Provide a separator followed by array of column name";

        }

            String str = "";
            for(int i =0;i<arr.size();i++) {
                str = str+arr.get(i);


        if(i<arr.size()-1) {
            str = str+ SEP;
        }


    }return str;
    }
}
```

Then from this java code, we have exported JAR file as ‘concatenate.jar’



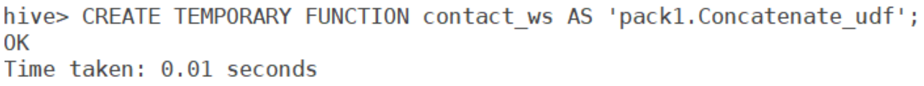
.imsettings.log
.mysql_history
.pig_history
.pulse-cookie
.scala_history
.viminfo
.webclipse.properties
.Xauthority
.xsession-errors
.xsession-errors.old
concatenate.jar
course_stud.txt
Hive Commands Acadgild Bucketin.
Hive Commands Acadgild Partition.
Hive Commands Simplin.txt
olympix_data.csv
Pig Commands (1).pdf
piggybank-0.15.0.jar
piggybank.jar.zip
session.jar
Sqoop_Commands_AG (2).tex
txns.txt
unitcount.jar
users.txt

After this we have added this JAR file into our VM in path location:



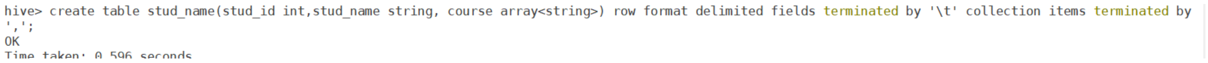
```
hive> ADD JAR /home/acadgild/concatenate.jar;
Added [/home/acadgild/concatenate.jar] to class path
Added resources: [/home/acadgild/concatenate.jar]
```

Then we have created a temporary function concatenate_ws



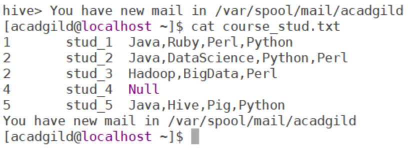
```
hive> CREATE TEMPORARY FUNCTION contact_ws AS 'pack1.Concatenate_udf';
OK
Time taken: 0.01 seconds
```

We have created table stud_name

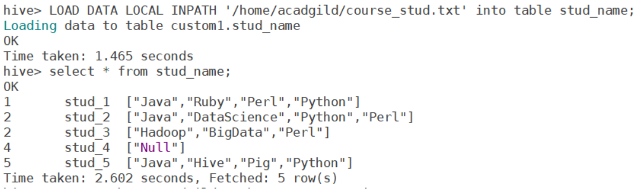


```
hive> create table stud_name(stud_id int,stud_name string, course array<string>) row format delimited fields terminated by '\t' collection items terminated by
',';
OK
Time taken: 0.596 seconds
```

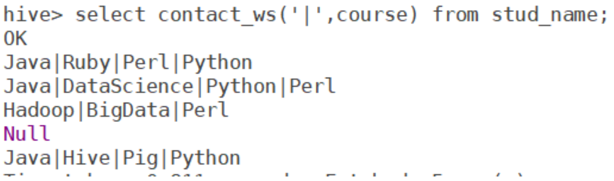
Below is the contents of course_stud which we will load into our table stud_name



```
hive> You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ cat course_stud.txt
1       stud_1  Java,Ruby,Perl,Python
2       stud_2  Java,DataScience,Python,Perl
2       stud_3  Hadoop,BigData,Perl
4       stud_4  Null
5       stud_5  Java,Hive,Pig,Python
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$
```



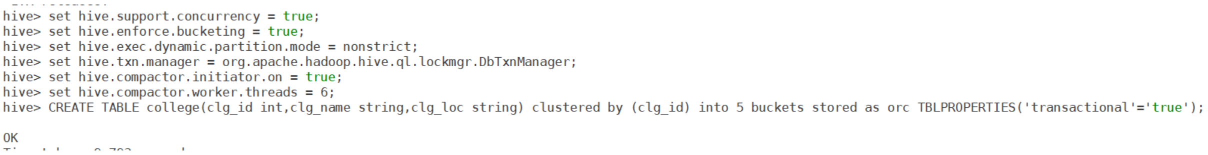
```
hive> LOAD DATA LOCAL INPATH '/home/acadgild/course_stud.txt' into table stud_name;
Loading data to table custom1.stud_name
OK
Time taken: 1.465 seconds
hive> select * from stud_name;
OK
1       stud_1  ["Java","Ruby","Perl","Python"]
2       stud_2  ["Java","DataScience","Python","Perl"]
2       stud_3  ["Hadoop","BigData","Perl"]
4       stud_4  ["Null"]
5       stud_5  ["Java","Hive","Pig","Python"]
Time taken: 2.602 seconds, Fetched: 5 row(s)
```



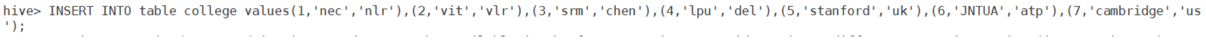
```
hive> select contact_ws('|',course) from stud_name;
OK
Java|Ruby|Perl|Python
Java|DataScience|Python|Perl
Hadoop|BigData|Perl
Null
Java|Hive|Pig|Python
```

Explanation:Displaying stud_name using HIVE UDF ‘CONCAT_WS’ using ‘|’ separator.

Task 3 Link: https://acadgild.com/blog/transactions-in-hive/ Refer the above given link for transactions in Hive and implement the operations given in the blog using your own sample data set and send us the screenshot. We are setting below properties in Hive.Because without setting these properties ‘Update’ and ‘Delete’ will not work and we will receive errors.



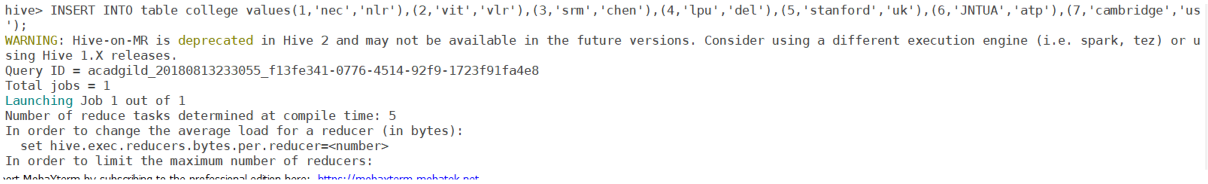
```
hive> set hive.support.concurrency = true;
hive> set hive.enforce.bucketing = true;
hive> set hive.exec.dynamic.partition.mode = nonstrict;
hive> set hive.txn.manager = org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;
hive> set hive.compactor.initiator.on = true;
hive> set hive.compactor.worker.threads = 6;
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
```



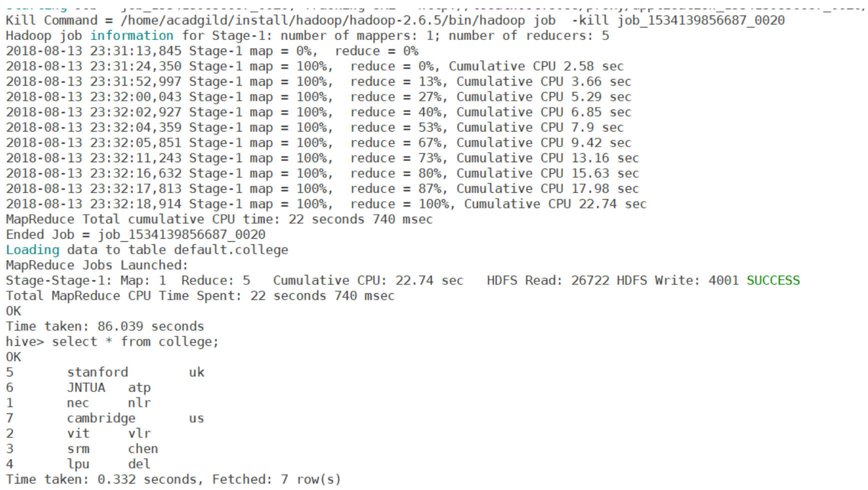
```
hive> INSERT INTO table college values(1,'nec','nlr'),(2,'vit','vlr'),(3,'srm','chen'),(4,'lpu','del'),(5,'stanford','uk'),(6,'JNTUA','atp'),(7,'cambridge','us
');
```

We have created a table with name ‘college’ and its columns are clg_id, clg_name, clg_loc. We are bucketing this table by clg_id column and using ORC file format.

Then we have inserted data into this college table with below insert command:



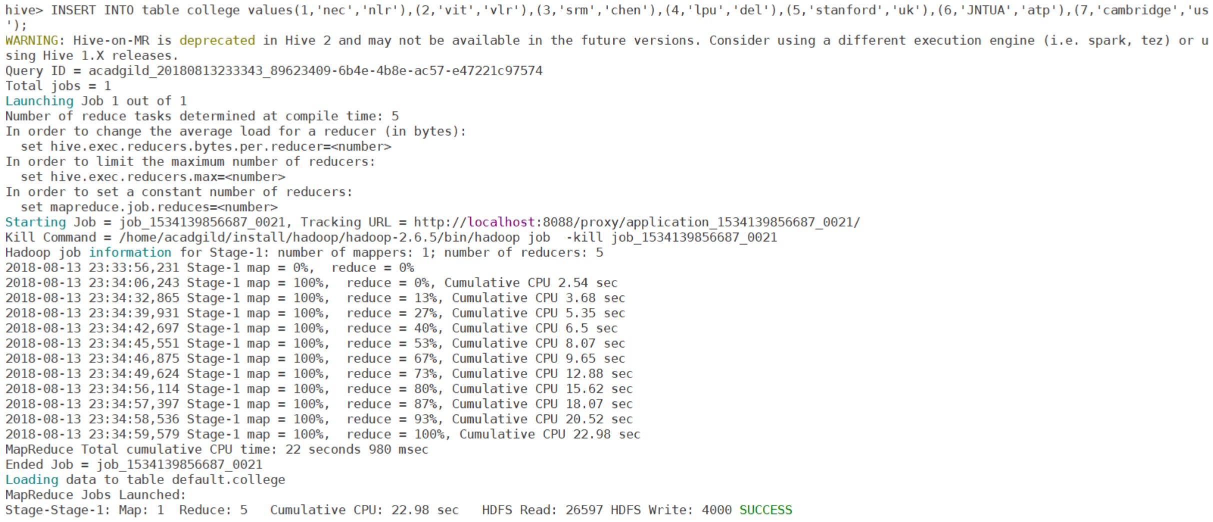
```
hive> INSERT INTO table college values(1,'nec','nlr'),(2,'vit','vlr'),(3,'srm','chen'),(4,'lpu','del'),(5,'stanford','uk'),(6,'JNTUA','atp'),(7,'cambridge','us
');
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or u
sing Hive 1.X releases.
Query ID = acadgild_20180813233055_f13fe341-0776-4514-92f9-1723f91fa4e8
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:
```



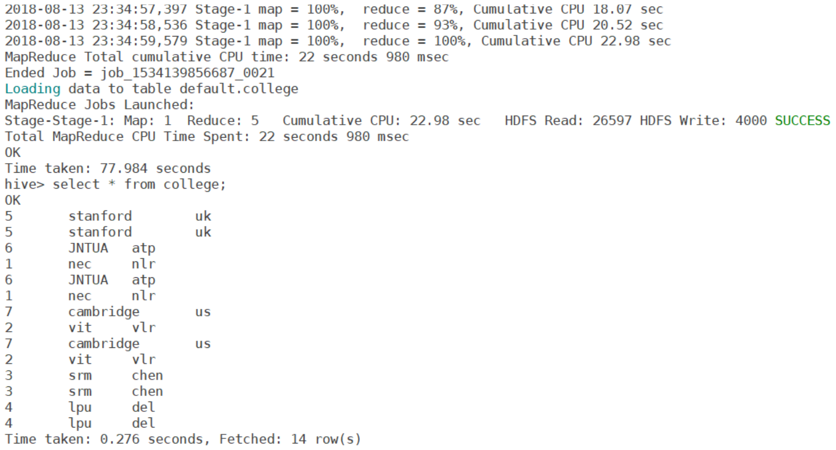
```
Starting Job = job_1534139856687_0020, Tracking URL = http://localhost:8088/proxy/application_1534139856687_0020/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534139856687_0020
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 5
2018-08-13 23:31:13,845 Stage-1 map = 0%,  reduce = 0%
2018-08-13 23:31:24,350 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.58 sec
2018-08-13 23:31:52,997 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 3.66 sec
2018-08-13 23:32:00,043 Stage-1 map = 100%,  reduce = 27%, Cumulative CPU 5.29 sec
2018-08-13 23:32:02,927 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 6.85 sec
2018-08-13 23:32:04,359 Stage-1 map = 100%,  reduce = 53%, Cumulative CPU 7.9 sec
2018-08-13 23:32:05,851 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 9.42 sec
2018-08-13 23:32:11,243 Stage-1 map = 100%,  reduce = 73%, Cumulative CPU 13.16 sec
2018-08-13 23:32:16,632 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 15.63 sec
2018-08-13 23:32:17,813 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 17.98 sec
2018-08-13 23:32:18,914 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 22.74 sec
MapReduce Total cumulative CPU time: 22 seconds 740 msec
Ended Job = job_1534139856687_0020
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 5   Cumulative CPU: 22.74 sec   HDFS Read: 26722 HDFS Write: 4001 SUCCESS
Total MapReduce CPU Time Spent: 22 seconds 740 msec
OK
Time taken: 86.039 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: 0.332 seconds, Fetched: 7 row(s)
```

We could see below that 7 rows have been inserted into college table successfully.

Now we are inserting same records again into college table and these rows will be appended.



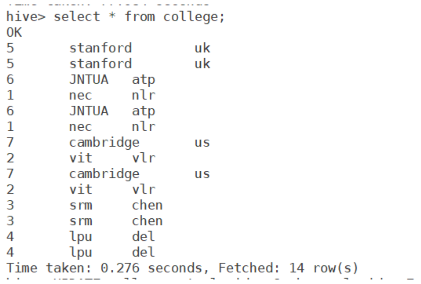
```
hive> INSERT INTO table college values(1,'nec','nlr'),(2,'vit','vlr'),(3,'srm','chen'),(4,'lpu','del'),(5,'stanford','uk'),(6,'JNTUA','atp'),(7,'cambridge','us
');
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or u
sing Hive 1.X releases.
Query ID = acadgild_20180813233343_89623409-6b4e-4b8e-ac57-e47221c97574
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_1534139856687_0021, Tracking URL = http://localhost:8088/proxy/application_1534139856687_0021/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534139856687_0021
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 5
2018-08-13 23:33:56,231 Stage-1 map = 0%,  reduce = 0%
2018-08-13 23:34:06,243 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.54 sec
2018-08-13 23:34:32,865 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 3.68 sec
2018-08-13 23:34:39,931 Stage-1 map = 100%,  reduce = 27%, Cumulative CPU 5.35 sec
2018-08-13 23:34:42,697 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 6.5 sec
2018-08-13 23:34:45,551 Stage-1 map = 100%,  reduce = 53%, Cumulative CPU 8.07 sec
2018-08-13 23:34:46,875 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 9.65 sec
2018-08-13 23:34:49,624 Stage-1 map = 100%,  reduce = 73%, Cumulative CPU 12.88 sec
2018-08-13 23:34:56,114 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 15.62 sec
2018-08-13 23:34:57,397 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 18.07 sec
2018-08-13 23:34:58,536 Stage-1 map = 100%,  reduce = 93%, Cumulative CPU 20.52 sec
2018-08-13 23:34:59,579 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 22.98 sec
MapReduce Total cumulative CPU time: 22 seconds 980 msec
Ended Job = job_1534139856687_0021
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 5   Cumulative CPU: 22.98 sec   HDFS Read: 26597 HDFS Write: 4000 SUCCESS
```



```
2018-08-13 23:34:57,397 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 18.07 sec
2018-08-13 23:34:58,536 Stage-1 map = 100%,  reduce = 93%, Cumulative CPU 20.52 sec
2018-08-13 23:34:59,579 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 22.98 sec
MapReduce Total cumulative CPU time: 22 seconds 980 msec
Ended Job = job_1534139856687_0021
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 5   Cumulative CPU: 22.98 sec   HDFS Read: 26597 HDFS Write: 4000 SUCCESS
```

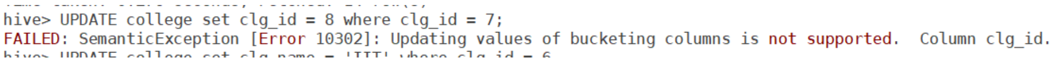
```
Total MapReduce CPU Time Spent: 22 seconds 980 msec
OK
Time taken: 77.984 seconds
hive> select * from college;
OK
5       stanford        uk
5       stanford        uk
6       JNTUA   atp
1       nec     nlr
6       JNTUA   atp
1       nec     nlr
7       cambridge       us
2       vit     vlr
7       cambridge       us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
Time taken: 0.276 seconds, Fetched: 14 row(s)
```

We could see data in college table below :



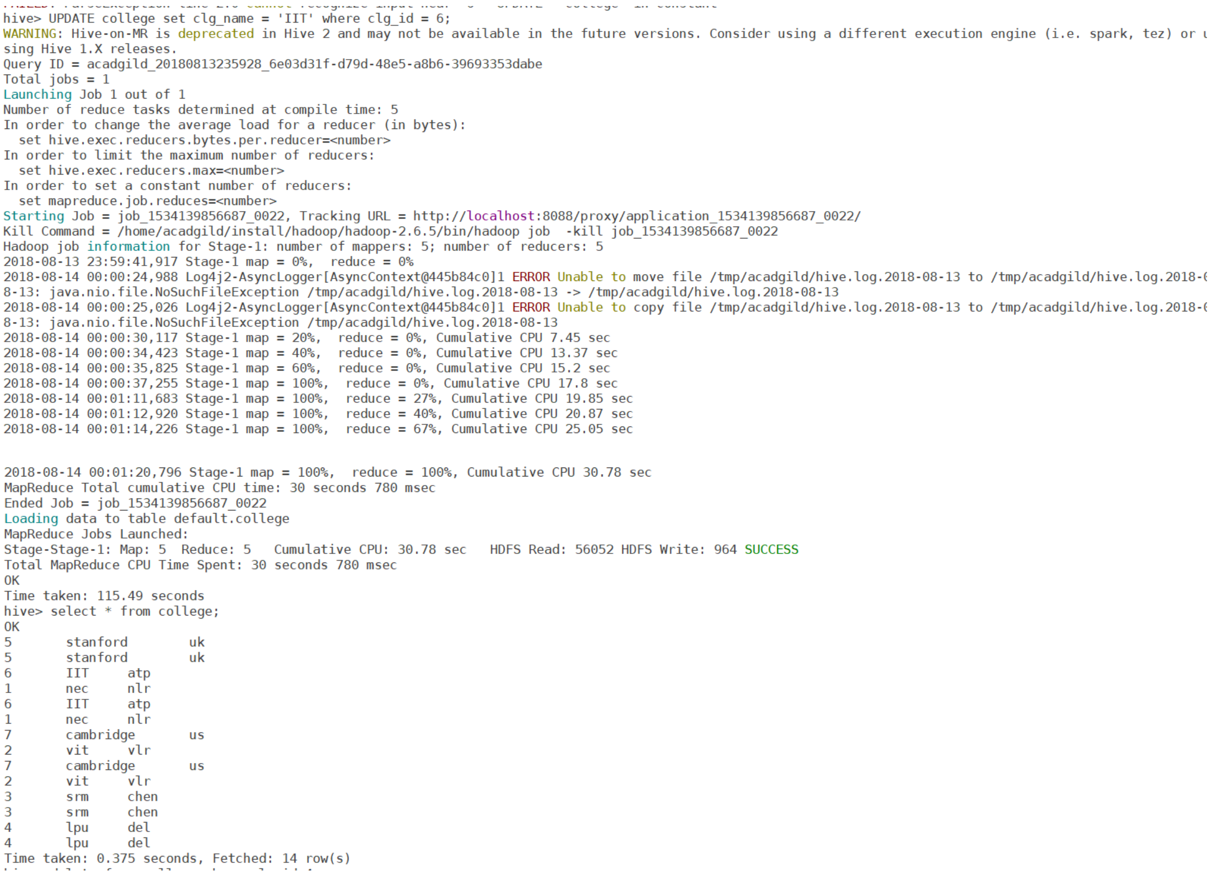
```
hive> select * from college;
OK
5       stanford        uk
5       stanford        uk
6       JNTUA   atp
1       nec     nlr
6       JNTUA   atp
1       nec     nlr
7       cambridge       us
2       vit     vlr
7       cambridge       us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
Time taken: 0.276 seconds, Fetched: 14 row(s)
```

Below we are trying to update bucketed column ‘clg_id’. But we have received error. So it means that we cannot update bucketed column.



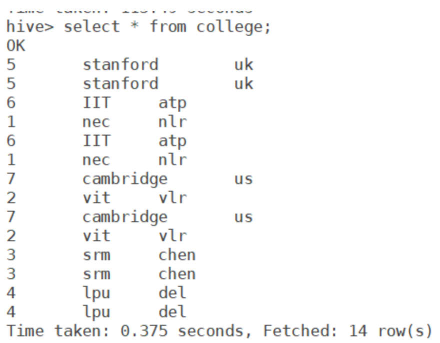
```
hive> UPDATE college set clg_id = 8 where clg_id = 7;
FAILED: SemanticException [Error 10302]: Updating values of bucketing columns is not supported.  Column clg_id.
```

Below we have performed update on non-bucketed column ‘clg_name’ and it has been updated successfully. This means that we can update non-bucketed column.



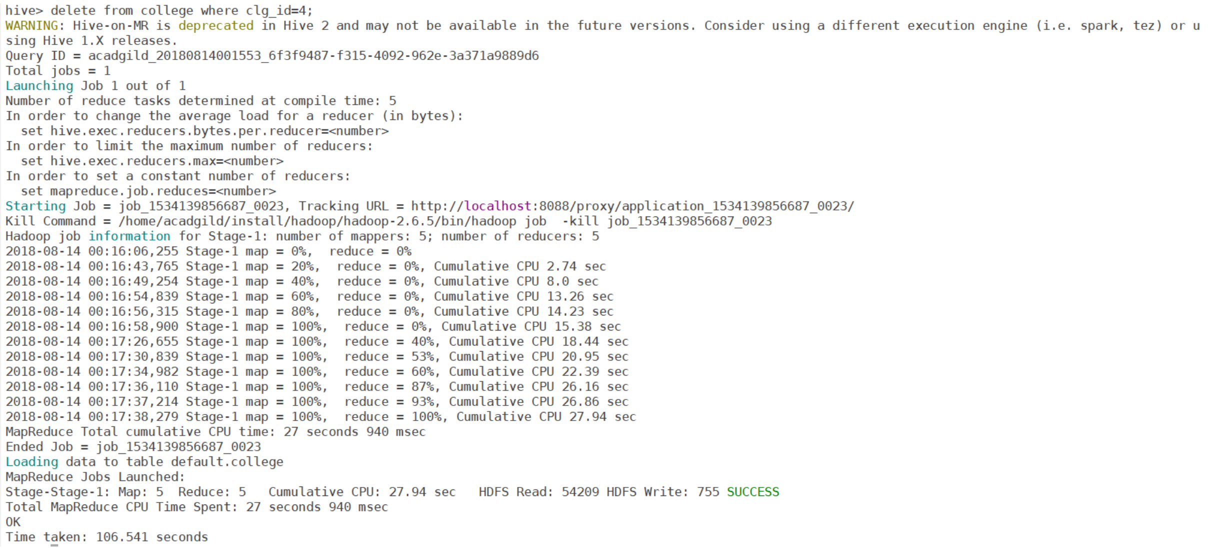
```
hive> UPDATE college set clg_name = 'IIT' where clg_id = 6;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or u
sing Hive 1.X releases.
Query ID = acadgild_20180813235928_6e03d31f-d79d-48e5-a8b6-39693353dabe
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_1534139856687_0022, Tracking URL = http://localhost:8088/proxy/application_1534139856687_0022/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534139856687_0022
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-08-13 23:59:41,917 Stage-1 map = 0%,  reduce = 0%
2018-08-14 00:00:24,988 Log4j2-AsyncLogger[AsyncContext@445b84c0]1 ERROR Unable to move file /tmp/acadgild/hive.log.2018-08-13 to /tmp/acadgild/hive.log.2018-0
8-13: java.nio.file.NoSuchFileException /tmp/acadgild/hive.log.2018-08-13 -> /tmp/acadgild/hive.log.2018-08-13
2018-08-14 00:00:25,026 Log4j2-AsyncLogger[AsyncContext@445b84c0]1 ERROR Unable to copy file /tmp/acadgild/hive.log.2018-08-13 to /tmp/acadgild/hive.log.2018-0
8-13: java.nio.file.NoSuchFileException /tmp/acadgild/hive.log.2018-08-13
2018-08-14 00:00:30,117 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 7.45 sec
2018-08-14 00:00:34,423 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 13.37 sec
2018-08-14 00:00:35,825 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 15.2 sec
2018-08-14 00:00:37,255 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 17.8 sec
2018-08-14 00:01:11,683 Stage-1 map = 100%,  reduce = 27%, Cumulative CPU 19.85 sec
2018-08-14 00:01:12,920 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 20.87 sec
2018-08-14 00:01:14,226 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 25.05 sec


2018-08-14 00:01:20,796 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 30.78 sec
MapReduce Total cumulative CPU time: 30 seconds 780 msec
Ended Job = job_1534139856687_0022
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 30.78 sec   HDFS Read: 56052 HDFS Write: 964 SUCCESS
Total MapReduce CPU Time Spent: 30 seconds 780 msec
OK
Time taken: 115.49 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
4       lpu     del
Time taken: 0.375 seconds, Fetched: 14 row(s)
```

Below you could see that clg_name has been changed to IIT for clg_id =6



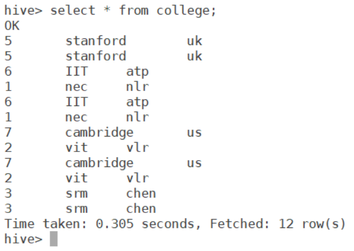
```
hive> select * from college;
OK
5       stanford        uk
5       stanford        uk
6       IIT     atp
1       nec     nlr
6       IIT     atp
1       nec     nlr
7       cambridge       us
2       vit     vlr
7       cambridge       us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
Time taken: 0.375 seconds, Fetched: 14 row(s)
```

Below we have deleted data having clg_id = 4



```
hive> delete from college where clg_id=4;
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 u
sing Hive 1.X releases.
Query ID = acadgild_20180814001553_6f3f9487-f315-4092-962e-3a371a9889d6
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_1534139856687_0023, Tracking URL = http://localhost:8088/proxy/application_1534139856687_0023/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534139856687_0023
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-08-14 00:16:06,255 Stage-1 map = 0%,  reduce = 0%
2018-08-14 00:16:43,765 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 2.74 sec
2018-08-14 00:16:49,254 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 8.0 sec
2018-08-14 00:16:54,839 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 13.26 sec
2018-08-14 00:16:56,315 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 14.23 sec
2018-08-14 00:16:58,900 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 15.38 sec
2018-08-14 00:17:26,655 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 18.44 sec
2018-08-14 00:17:30,839 Stage-1 map = 100%,  reduce = 53%, Cumulative CPU 20.95 sec
2018-08-14 00:17:34,982 Stage-1 map = 100%,  reduce = 60%, Cumulative CPU 22.39 sec
2018-08-14 00:17:36,110 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 26.16 sec
2018-08-14 00:17:37,214 Stage-1 map = 100%,  reduce = 93%, Cumulative CPU 26.86 sec
2018-08-14 00:17:38,279 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 27.94 sec
MapReduce Total cumulative CPU time: 27 seconds 940 msec
Ended Job = job_1534139856687_0023
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 27.94 sec   HDFS Read: 54209 HDFS Write: 755 SUCCESS
Total MapReduce CPU Time Spent: 27 seconds 940 msec
OK
Time taken: 106.541 seconds
```

We could see that clg_id having value as 4 has been deleted successfully from college table.



```
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
Time taken: 0.305 seconds, Fetched: 12 row(s)
hive>
```