

Session 9 - ADVANCED HIVE

Assignment 1

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

Create the above table 'olympics' with the above structure.

```
Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.jar!/hive-log4j2.properties Async: true
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 Hive 1.X releases.
hive> create table olympics(athlete 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;
OK
Time taken: 10.462 seconds
hive> load data local inpath '/home/acadgild/user acadgild/assignments/Hive/olympic data.csv' into table olympics;
Loading data to table default.olympics
OK
Time taken: 2.559 seconds
hive> select * from olympics limit 10;
OK
Michael Phelps 23 United States 2008 08-24-08 Swimming 8 0 0 8
Michael Phelps 19 United States 2004 08-29-04 Swimming 6 0 2 8
Michael Phelps 27 United States 2012 08-12-12 Swimming 4 2 0 6
Natalie Coughlin 25 United States 2008 08-24-08 Swimming 1 2 3 6
Aleksey Nemov 24 Russia 2000 10-01-00 Gymnastics 2 1 3 6
Alicia Coutts 24 Australia 2012 08-12-12 Swimming 1 3 1 5
Missy Franklin 17 United States 2012 08-12-12 Swimming 4 0 1 5
Ryan Lochte 27 United States 2012 08-12-12 Swimming 2 2 1 5
Allison Schmitt 22 United States 2012 08-12-12 Swimming 3 1 1 5
Natalie Coughlin 21 United States 2004 08-29-04 Swimming 2 2 1 5
Time taken: 4.52 seconds, Fetched: 10 row(s)
hive>
```

With reference to the screenshot above,

1 : create a table using the following syntax.

create table olympics(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;

2 : load the data from the text file.

Load data local inpath

‘/home/acadgild/user_acadgild/assignments/Hive/olympic_data.csv’ into table olympics;

3 : displaying the contents of the table.

TASK 1:

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

```
acadgild@localhost:~$ hive
hive> select country,sum(total) 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 using Hive 1.X releases.
Query ID = acadgild_20180426035210_c5682205-fe50-489e-b0ed-15024dd2c10d
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_1524630371965_0015, Tracking URL = http://localhost:8088/proxy/application_1524630371965_0015/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1524630371965_0015
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-04-26 03:52:23,272 Stage-1 map = 0%, reduce = 0%
2018-04-26 03:52:34,926 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.82 sec
2018-04-26 03:52:46,760 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 6.7 sec
MapReduce Total cumulative CPU time: 6 seconds 700 msec
Ended Job = job_1524630371965_0015
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 6.7 sec HDFS Read: 528565 HDFS Write: 881 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 700 msec
OK
```

select country,sum(total) from olympics where sport = 'Swimming' group by country;

```
acadgild@localhost:~$ hive
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 6.7 sec HDFS Read: 528565 HDFS Write: 881 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 700 msec
OK
country          total
Argentina         1
Australia        163
Austria           3
Belarus           2
Brazil            8
Canada            5
China             35
Costa Rica        2
Croatia           1
Denmark           1
France            39
Germany           32
Great Britain     11
Hungary           9
Italy             16
Japan             43
Lithuania         1
Netherlands       46
Norway            2
Poland            3
Romania           6
Russia            20
Serbia            1
Slovakia          2
Slovenia          1
South Africa     11
South Korea       4
Spain             3
Sweden            9
Trinidad and Tobago 1
Tunisia           3
Ukraine           7
United States     267
Zimbabwe          7
Time taken: 38.933 seconds, Fetched: 34 row(s)
hive>
```

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

select year,sum(total) from olympics where country = 'India' group by year;

```
hive> select year,sum(total) 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 using Hive 1.X releases.
Query ID = acadgild_20180426040015_57331c67-98df-4def-96d9-148d3a4d48da
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_1524630371965_0016, Tracking URL = http://localhost:8088/proxy/application_1524630371965_0016/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1524630371965_0016
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-04-26 04:00:29,403 Stage-1 map = 0%, reduce = 0%
2018-04-26 04:00:41,022 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.61 sec
2018-04-26 04:00:52,761 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 6.35 sec
MapReduce Total cumulative CPU time: 6 seconds 350 msec
Ended Job = job_1524630371965_0016
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 6.35 sec HDFS Read: 528553 HDFS Write: 163 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 350 msec
OK
2000 1
2004 1
2008 3
2012 6
Time taken: 39.199 seconds, Fetched: 4 row(s)
hive>
```

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

select country, sum(total) from olympics group by country;

```
Applications Places System acadgild@localhost:~ Thu Apr 26, 4:05 AM
File Edit View Search Terminal Help
hive> select country,sum(total) 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 using Hive 1.X releases.
Query ID = acadgild_20180426040352_f3dc1ef2-4c73-460e-a42b-d423c6299caf
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_1524630371965_0017, Tracking URL = http://localhost:8088/proxy/application_1524630371965_0017/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1524630371965_0017
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-04-26 04:04:06,454 Stage-1 map = 0%, reduce = 0%
2018-04-26 04:04:16,965 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.43 sec
2018-04-26 04:04:28,380 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.95 sec
MapReduce Total cumulative CPU time: 4 seconds 950 msec
Ended Job = job_1524630371965_0017
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.95 sec HDFS Read: 527733 HDFS Write: 2742 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 950 msec
OK
Afghanistan 2
Algeria 8
Argentina 141
Armenia 10
Australia 609
Austria 91
Azerbaijan 25
Bahamas 24
Bahrain 1
Barbados 1
Belarus 97
Belgium 18
Botswana 1
Brazil 221
```

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

select country, sum(gold) from olympics group by country;

```
File Edit View Search Terminal Help
acadgild@localhost:~$
hive> select country, sum(gold) 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 using Hive 1.X releases.
Query ID = acadgild_20180426040823_08af3160-b225-4b3d-ac18-0e66dfcee6d3
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_1524630371965_0018, Tracking URL = http://localhost:8088/proxy/application_1524630371965_0018/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1524630371965_0018
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-04-26 04:08:36,853 Stage-1 map = 0%, reduce = 0%
2018-04-26 04:08:46,177 Stage-1 map = 100%, reduce = 0%
2018-04-26 04:08:57,654 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.86 sec
MapReduce Total cumulative CPU time: 4 seconds 860 msec
Ended Job = job_1524630371965_0018
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.86 sec HDFS Read: 527731 HDFS Write: 2703 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 860 msec
OK
country      gold
-----
Afghanistan    0
Algeria        2
Argentina      49
Armenia        0
Australia     163
Austria       36
Azerbaijan     6
Bahamas       11
Bahrain        0
Barbados       0
Belarus       17
Belgium        2
Botswana       0
Brazil       46
```

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.

The UDF has to be created by extending the `org.apache.hadoop.hive.ql.exec.UDF` class.

```
package com.acadgild.hiveudf;
```

```
import java.util.ArrayList;
```

```
import org.apache.commons.lang.StringUtils;
```

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

```
import org.apache.hadoop.io.Text;
```

```
public class StringConcatUDF extends UDF {
    private Text result = new Text();
    public Text evaluate(String sep, ArrayList<String> stringChars) {
        if (sep == null) {
            return null;
        }
        String tempstr = "";
        for (int i = 0; i <= stringChars.size() - 1; i++) {
            tempstr = tempstr + (stringChars.get(i) + sep);
        }
    }
}
```

```

        String finalstr = tempstr.substring(0, tempstr.length() - 1);
        result.set(finalstr);
        return result;
    }
    public Text evaluate(Text str) {
        if (str == null) {
            return null;
        }
        result.set(StringUtils.strip(str.toString()));
        return result;
    }
}

```

- Create a jar file for the java file.
- Add the jar in hive list of jars.

add jar '/location/of/the/jar/file'

Create a table with a column with array datatype.

```

acadgild@localhost:~
File Edit View Search Terminal Help
Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (1. <del>spark</del> or using Hive 1.X releases)
hive> create table Employee(empname string, empdesignation array<string>)
> row format delimited
> fields terminated by '\t'
> collection items terminated by ',';
OK
Time taken: 14.786 seconds
hive> desc Employee;
OK
empname            string
empdesignation      array<string>
Time taken: 0.082 seconds, Fetched: 1 row(s)
hive> load data local inpath '/home/acadgild/user_acadgild/assignments/Hive/employee.txt' into table Employee;
Loading data to table default.Employee
OK
Time taken: 2.978 seconds
hive> select * from Employee;
OK
Alex      ["Analyst","Data Engineer","Big Data Consultant"]
Felix     ["Analyst","Software Engineer","Software Consultant"]
Time taken: 5.749 seconds, Fetched: 2 row(s)

```

With reference to the screenshot above,

- 1 : creating a table employee where the fields are delimited using a tab space and the values in an array are separated using comma.
- 2 : the datatype of the column is array.
- 3 : sample data from a text file is loaded.
- 4 : The table is loaded with the data and the array can be seen.

```

hive> ADD jar /home/acadgild/HiveUDF.jar;
Added [/home/acadgild/HiveUDF.jar] to class path
Added resources: [/home/acadgild/HiveUDF.jar]
hive> list jars;
/home/acadgild/HiveUDF.jar
hive> CREATE TEMPORARY FUNCTION concat_ws as 'com.acadgild.hiveudf.StringConcatUDF';
OK
Time taken: 0.156 seconds
hive> select concat_ws("HADOOP",empdesignation) from Employee;
AnalystHAD00PData EngineerHAD00PBig Data Consultant
AnalystHAD00PSoftware EngineerHAD00PSoftware Consultant
Time taken: 3.037 seconds, Fetched: 2 row(s)

```

With reference to the screenshot above,

1 : Adding jar to hive. Verifying the jar is added to hive, using 'list jars'.

2 : A temporary function is created with the classname to be used.

```
CREATE TEMPORARY FUNCTION concat_ws as 'com.acadgild.hiveudf.StringConcatUDF';
```

3 : Using the method.

```
select concat_ws("HADOOP",empdesignation) from Employee;
```

4 : The word HADOOP (1st argument) is concatenated between each field in the array.

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.

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.q1.Lockmgr.DbTxnManager;
hive> set hive.compactor.initiator.on = true;
hive> set hive.compactor.worker.threads = 1;
hive>

```

Creating a table to support 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');
```

```

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: 11.824 seconds
hive> show tables;
OK
college
employee
olympics
Time taken: 0.653 seconds, Fetched: 3 row(s)
hive>

```

The table name is displayed the list of 'show tables;'

Inserting data to Hive 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');

```

hive> desc college;
OK
clg_id          int
clg_name        string
clg_loc         string
Time taken: 0.384 seconds, Fetched: 3 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');
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 Hive 1.X releases.
Query ID = acadgild_20180430012309_50a10c62-56be-4e2f-8016-b0291ba5b3fb
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_1525006545565_0001, Tracking URL = http://localhost:8088/proxy/application_1525006545565_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1525006545565_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 5
2018-04-30 01:23:51,625 Stage-1 map = 0%, reduce = 0%
2018-04-30 01:24:11,439 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.83 sec
2018-04-30 01:25:00,789 Stage-1 map = 100%, reduce = 53%, Cumulative CPU 9.28 sec
2018-04-30 01:25:01,900 Stage-1 map = 100%, reduce = 67%, Cumulative CPU 10.66 sec
2018-04-30 01:25:22,323 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 27.24 sec
MapReduce Total cumulative CPU time: 28 seconds 630 msec
Ended Job = job_1525006545565_0001
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 5 Cumulative CPU: 28.63 sec HDFS Read: 26737 HDFS Write: 4001 SUCCESS
Total MapReduce CPU Time Spent: 28 seconds 630 msec
OK
Time taken: 140.7 seconds
hive>

```

Data is inserted into the table.

```

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: 1.483 seconds, Fetched: 7 row(s)

```

Updating Data in Hive table :

UPDATE college set clg_id = 8 where clg_id = 7;

Bucketed column cannot be updated. Only non bucketed columns can be updated.

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

```
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.
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 Hive 1.X releases.
Query ID = acadgild_20180430013148_5fa495ab-3470-4eed-a48e-d32029c4ffad
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_1525006545565_0002, Tracking URL = http://localhost:8088/proxy/application_1525006545565_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1525006545565_0002
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-04-30 01:32:11,081 Stage-1 map = 0%, reduce = 0%
2018-04-30 01:33:11,453 Stage-1 map = 0%, reduce = 0%, Cumulative CPU 12.1 sec
2018-04-30 01:33:26,500 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 23.5 sec
2018-04-30 01:34:14,958 Stage-1 map = 100%, reduce = 40%, Cumulative CPU 26.43 sec
2018-04-30 01:34:18,004 Stage-1 map = 100%, reduce = 53%, Cumulative CPU 28.88 sec
2018-04-30 01:34:19,446 Stage-1 map = 100%, reduce = 67%, Cumulative CPU 30.98 sec
2018-04-30 01:34:28,426 Stage-1 map = 100%, reduce = 93%, Cumulative CPU 37.14 sec
2018-04-30 01:34:29,521 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 39.21 sec
MapReduce Total cumulative CPU time: 39 seconds 210 msec
Ended Job = job_1525006545565_0002
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5 Reduce: 5 Cumulative CPU: 39.21 sec HDFS Read: 51649 HDFS Write: 937 SUCCESS
Total MapReduce CPU Time Spent: 39 seconds 210 msec
OK
Time taken: 164.052 seconds
hive>
```

The updated values are reflected in the table.

```
hive> select * from college;
OK
5      stanford      uk
6      IIT      atp
1      nec      nlr
7      cambridge      us
2      vit      vlr
3      srm      chen
4      lpu      del
Time taken: 0.535 seconds, Fetched: 7 row(s)
hive>
```

Deleting a row from the table :

delete from college where clg_id = 2;


```

hive> delete from college where clg_id=2;
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 Hive 1.X releases.
Query ID = acadgild_20180430013753_97826590-b260-40ac-bfdf-57caaf6fa362
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_1525006545565_0003, Tracking URL = http://localhost:8088/proxy/application_1525006545565_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1525006545565_0003
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-04-30 01:38:10,744 Stage-1 map = 0%, reduce = 0%
2018-04-30 01:39:12,398 Stage-1 map = 0%, reduce = 0%, Cumulative CPU 12.14 sec
2018-04-30 01:39:14,599 Stage-1 map = 60%, reduce = 0%, Cumulative CPU 16.88 sec
2018-04-30 01:39:15,725 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 20.37 sec
2018-04-30 01:39:54,824 Stage-1 map = 100%, reduce = 27%, Cumulative CPU 22.42 sec
2018-04-30 01:40:00,743 Stage-1 map = 100%, reduce = 53%, Cumulative CPU 26.04 sec
2018-04-30 01:40:02,403 Stage-1 map = 100%, reduce = 67%, Cumulative CPU 27.66 sec
2018-04-30 01:40:06,976 Stage-1 map = 100%, reduce = 80%, Cumulative CPU 31.18 sec
2018-04-30 01:40:09,683 Stage-1 map = 100%, reduce = 87%, Cumulative CPU 32.62 sec
2018-04-30 01:40:11,149 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 36.06 sec
MapReduce Total cumulative CPU time: 36 seconds 60 msec
Ended Job = job_1525006545565_0003
Loading data to table default.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5 Reduce: 5 Cumulative CPU: 36.06 sec HDFS Read: 49858 HDFS Write: 741 SUCCESS
Total MapReduce CPU Time Spent: 36 seconds 60 msec
OK
Time taken: 141.331 seconds
hive>

```

The data is reflected in the table.

```

hive> select * from college;
OK
5      stanford      uk
6      IIT           atp
1      nec           nlr
7      cambridge     us
3      srm           chen
4      lpu           del
Time taken: 0.514 seconds, Fetched: 6 row(s)
hive>

```

Row with clg_id 2 is deleted from the table.