Assignment 9 : Advanced Hive Assignment Problems

Associated Data Files:

This Data set is about Olympics. You can download the data set from the below link:

https://drive.google.com/open?id=0ByJLBTmJojjzV1czX3Nha0R3bTQ

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

Terminal Execution

```
[acadgild@localhost ~]$ jps
3054 Jps
[acadgild@localhost ~]$ sudo service sshd start
[sudo] password for acadgild:
[acadgild@localhost ~]$ sudo service mysqld start
Starting mysqld:                                           [  OK  ]
[acadgild@localhost ~]$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
18/07/17 06:02:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-namenode-localhost.localdomain.ou
t
localhost: starting datanode, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-datanode-localhost.localdomain.out
Starting secondary namenodes [0.0.0.0]
```

```
0.0.0.0: starting secondarynamenode, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-secondarynamenode-localhost.local
domain.out
18/07/17 06:03:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable
starting yarn daemons
starting resourcemanager, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/yarn-acadgild-resourcemanager-localhost.localdomai
n.out
localhost: starting nodemanager, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/yarn-acadgild-nodemanager-localhost.localdomain.ou
t
[acadgild@localhost ~]$ jps
3232 NameNode
3665 ResourceManager
3330 DataNode
3493 SecondaryNameNode
3766 NodeManager
3800 Jps
[acadgild@localhost ~]$

[acadgild@localhost ~]$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in
[jar:file:/home/acadgild/install/hive/apache-hive-2.3.3-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/
StaticLoggerBinder.class]
SLF4J: Found binding in
[jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/
org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in
jar:file:/home/acadgild/install/hive/apache-hive-2.3.3-bin/lib/hive-common-2.3.3.jar!/hive-log4j2.prope
rties 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> show databases;
OK
custom
default
Time taken: 11.729 seconds, Fetched: 2 row(s)
hive> use custom;
OK
Time taken: 0.097 seconds
hive> 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;
OK
Time taken: 2.127 seconds
hive> show create table olympic;
OK
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 SERDE
  'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
  'field.delim'='\t',
  'serialization.format'='\t')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  'hdfs://localhost:8020/user/hive/warehouse/custom.db/olympic'
TBLPROPERTIES (
  'transient_lastDdlTime'='1531790738')
Time taken: 0.521 seconds, Fetched: 24 row(s)

hive> load data local inpath '/home/acadgild/Desktop/olympix_data.csv' into table olympic;
Loading data to table custom.olympic
OK
Time taken: 1.694 seconds
```

Problem Statement

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;

 # Terminal Execution 1

```
hive> select country,SUM(total) from olympic 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_20180717070245_5d1158cc-61fd-41e0-9377-2ceb8016e83e
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_1531787606394_0002, Tracking URL =
http://localhost:8088/proxy/application_1531787606394_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1531787606394_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-17 07:03:02,298 Stage-1 map = 0%,  reduce = 0%
2018-07-17 07:03:13,092 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.1 sec
2018-07-17 07:03:25,291 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.6 sec
MapReduce Total cumulative CPU time: 5 seconds 600 msec
Ended Job = job_1531787606394_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.6 sec   HDFS Read: 528582 HDFS Write: 881
SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 600 msec
OK
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: 41.785 seconds, Fetched: 34 row(s)
hive>

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;

 # Terminal Execution 2

```
hive> select year,SUM(total) from olympic 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_20180717070459_a5c1d476-5cbb-4c07-8f41-6b8716d51535
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_1531787606394_0003, Tracking URL =
http://localhost:8088/proxy/application_1531787606394_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1531787606394_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-17 07:05:13,082 Stage-1 map = 0%,  reduce = 0%
2018-07-17 07:05:23,961 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.1 sec
2018-07-17 07:05:34,654 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.52 sec
MapReduce Total cumulative CPU time: 5 seconds 520 msec
Ended Job = job_1531787606394_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.52 sec   HDFS Read: 528570 HDFS Write: 163
SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 520 msec
OK
2000    1
2004    1
2008    3
2012    6
Time taken: 37.54 seconds, Fetched: 4 row(s)
```

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;
```

 # Terminal Execution 3

```
hive> select country,SUM(total) from olympic 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_20180717070646_3a7020bc-307d-4c4c-b7f6-71dc99d09bbd
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_1531787606394_0004, Tracking URL =
http://localhost:8088/proxy/application_1531787606394_0004/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1531787606394_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-17 07:06:59,953 Stage-1 map = 0%,  reduce = 0%
2018-07-17 07:07:09,404 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.1 sec
2018-07-17 07:07:20,963 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.37 sec
MapReduce Total cumulative CPU time: 4 seconds 370 msec
Ended Job = job_1531787606394_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.37 sec   HDFS Read: 527750 HDFS Write:
2742 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 370 msec
OK
Afghanistan     2
Algeria         8
Argentina       141
Armenia         10
Australia       609
Austria91
Azerbaijan      25
Bahamas         24
Bahrain         1
Barbados        1
Belarus         97
Belgium         18
Botswana        1
Brazil  221
Bulgaria        41
Cameroon        20
```

Canada	370
Chile	22
China	530
Chinese Taipei	20
Colombia	13
Costa Rica	2
Croatia	81
Cuba	188
Cyprus	1
Czech Republic	81
Denmark	89
Dominican Republic	5
Ecuador	1
Egypt	8
Eritrea	1
Estonia	18
Ethiopia	29
Finland	118
France	318
Gabon	1
Georgia	23
Germany	629
Great Britain	322
Greece	59
Grenada	1
Guatemala	1
Hong Kong	3
Hungary	145
Iceland	15
India	11
Indonesia	22
Iran	24
Ireland	9
Israel	4
Italy	331
Jamaica	80
Japan	282
Kazakhstan	42
Kenya	39
Kuwait	2
Kyrgyzstan	3
Latvia	17
Lithuania	30
Macedonia	1
Malaysia	3
Mauritius	1
Mexico	38
Moldova	5
Mongolia	10

Montenegro 14
Morocco 11
Mozambique 1
Netherlands 318
New Zealand 52
Nigeria 39
North Korea 21
Norway 192
Panama 1
Paraguay 17
Poland 80
Portugal 9
Puerto Rico 2
Qatar 3
Romania 123
Russia 768
Saudi Arabia 6
Serbia 31
Serbia and Montenegro 38
Singapore 7
Slovakia 35
Slovenia 25
South Africa 25
South Korea 308
Spain 205
Sri Lanka 1
Sudan 1
Sweden 181
Switzerland 93
Syria 1
Tajikistan 3
Thailand 18
Togo 1
Trinidad and Tobago 19
Tunisia 4
Turkey 28
Uganda 1
Ukraine 143
United Arab Emirates 1
United States 1312
Uruguay 1
Uzbekistan 19
Venezuela 4
Vietnam 2
Zimbabwe 7
Time taken: 35.802 seconds, Fetched: 110 row(s)

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;

 # Terminal Execution 4

```
hive> select country,SUM(gold) from olympic 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_20180717071000_c8e6337c-2964-4be5-ac03-afc6d29dbb88
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_1531787606394_0005, Tracking URL =
http://localhost:8088/proxy/application_1531787606394_0005/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1531787606394_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-17 07:10:13,575 Stage-1 map = 0%,  reduce = 0%
2018-07-17 07:10:23,009 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.04 sec
2018-07-17 07:10:34,618 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.32 sec
MapReduce Total cumulative CPU time: 4 seconds 320 msec
Ended Job = job_1531787606394_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.32 sec   HDFS Read: 527748 HDFS Write:
2703 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 320 msec
OK
Afghanistan     0
Algeria         2
Argentina       49
Armenia         0
Australia       163
Austria36
Azerbaijan      6
Bahamas         11
Bahrain         0
Barbados        0
```

Belarus	17
Belgium	2
Botswana	0
Brazil	46
Bulgaria	8
Cameroon	20
Canada	168
Chile	3
China	234
Chinese Taipei	2
Colombia	2
Costa Rica	0
Croatia	35
Cuba	57
Cyprus	0
Czech Republic	14
Denmark	46
Dominican Republic	3
Ecuador	0
Egypt	1
Eritrea	0
Estonia	6
Ethiopia	13
Finland	11
France	108
Gabon	0
Georgia	6
Germany	223
Great Britain	124
Greece	12
Grenada	1
Guatemala	0
Hong Kong	0
Hungary	77
Iceland	0
India	1
Indonesia	5
Iran	10
Ireland	1
Israel	1
Italy	86
Jamaica	24
Japan	57
Kazakhstan	13
Kenya	11
Kuwait	0
Kyrgyzstan	0
Latvia	3
Lithuania	5

Macedonia	0
Malaysia	0
Mauritius	0
Mexico	19
Moldova	0
Mongolia	2
Montenegro	0
Morocco	2
Mozambique	1
Netherlands	101
New Zealand	18
Nigeria	6
North Korea	6
Norway	97
Panama	1
Paraguay	0
Poland	20
Portugal	1
Puerto Rico	0
Qatar	0
Romania	57
Russia	234
Saudi Arabia	0
Serbia	1
Serbia and Montenegro	11
Singapore	0
Slovakia	10
Slovenia	5
South Africa	10
South Korea	110
Spain	19
Sri Lanka	0
Sudan	0
Sweden	57
Switzerland	21
Syria	0
Tajikistan	0
Thailand	6
Togo	0
Trinidad and Tobago	1
Tunisia	2
Turkey	9
Uganda	1
Ukraine	31
United Arab Emirates	1
United States	552
Uruguay	0
Uzbekistan	5
Venezuela	1

Vietnam 0
Zimbabwe 2
Time taken: 35.61 seconds, Fetched: 110 row(s)

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.

Eclipse code: filename : concat_ws.java

```
package com.acadgild.hive.assignment;

import java.util.stream.Collectors;
import org.apache.avro.generic.GenericData.Array;
import org.apache.hadoop.hive.ql.exec.UDF;


public class concat_ws extends UDF{

            public String evaluate (String SEP, Array<String> stringList) {

                  return stringList.stream().map(u->
String.valueOf(u)).collect(Collectors.joining(SEP));
            }
      }
```

Terminal Execution:

hive> add jar /home/acadgild/Desktop/concat_delimiter.jar;
Added [/home/acadgild/Desktop/concat_delimiter.jar] to class path
Added resources: [/home/acadgild/Desktop/concat_delimiter.jar]

hive> create temporary function concat_delimit as 'com.acadgild.hive.assignment.concat_ws';
OK
Time taken: 0.319 seconds

hive> select concat_delimit('#',Array('Hadoop','is','awesome')) from college;
OK
Hadoop#is#awesome
Hadoop#is#awesome
Hadoop#is#awesome

Hadoop#is#awesome
Hadoop#is#awesome
Hadoop#is#awesome

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.

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

Terminal Execution :

```
hive> show tables in custom;
OK
olympic
temperature_data
temperature_data_vw
Time taken: 0.234 seconds, Fetched: 3 row(s)
```

```
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 = 1;
```

Creating a Table That Supports Hive Transactions

Terminal Execution :

```
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: 0.365 seconds

hive> show tables;
OK
college
olympic
temperature_data
temperature_data_vw
Time taken: 0.136 seconds, Fetched: 4 row(s)
```

Inserting Data into a Hive Table

Terminal Execution :

```
hive> INSERT INTO table college values(1,'nec','nlr');
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_20180717073515_7236feb8-a976-4da4-a5a5-987adea0a8a4
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_1531787606394_0006, Tracking URL =
http://localhost:8088/proxy/application_1531787606394_0006/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1531787606394_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 5
2018-07-17 07:35:34,493 Stage-1 map = 0%,  reduce = 0%
2018-07-17 07:35:46,922 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.83 sec
2018-07-17 07:36:25,206 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 3.79 sec
2018-07-17 07:36:34,580 Stage-1 map = 100%,  reduce = 27%, Cumulative CPU 5.48 sec
2018-07-17 07:36:38,392 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 6.68 sec
2018-07-17 07:36:40,117 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 9.16 sec
2018-07-17 07:36:52,564 Stage-1 map = 100%,  reduce = 73%, Cumulative CPU 12.12 sec
2018-07-17 07:36:58,860 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 18.33 sec
2018-07-17 07:37:00,442 Stage-1 map = 100%,  reduce = 93%, Cumulative CPU 21.61 sec
2018-07-17 07:37:02,479 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 24.92 sec
MapReduce Total cumulative CPU time: 24 seconds 920 msec
Ended Job = job_1531787606394_0006
Loading data to table custom.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 5   Cumulative CPU: 24.92 sec   HDFS Read: 26984 HDFS Write:
1771 SUCCESS
Total MapReduce CPU Time Spent: 24 seconds 920 msec
OK
Time taken: 111.258 seconds

hive> INSERT INTO table college values(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_20180717073837_a57ed0fe-1a70-4e58-a31c-f1d0dfc2ce2e
```

```
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_1531787606394_0007, Tracking URL =
http://localhost:8088/proxy/application_1531787606394_0007/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1531787606394_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 5
2018-07-17 07:38:54,149 Stage-1 map = 0%,  reduce = 0%
2018-07-17 07:39:06,244 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.74 sec
2018-07-17 07:39:43,788 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 3.68 sec
2018-07-17 07:39:52,563 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 6.22 sec
2018-07-17 07:39:55,978 Stage-1 map = 100%,  reduce = 53%, Cumulative CPU 7.51 sec
2018-07-17 07:39:57,655 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 8.69 sec
2018-07-17 07:40:09,458 Stage-1 map = 100%,  reduce = 73%, Cumulative CPU 11.69 sec
2018-07-17 07:40:14,144 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 14.69 sec
2018-07-17 07:40:17,003 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 17.64 sec
2018-07-17 07:40:18,411 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 23.29 sec
MapReduce Total cumulative CPU time: 23 seconds 290 msec
Ended Job = job_1531787606394_0007
Loading data to table custom.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 5   Cumulative CPU: 23.29 sec   HDFS Read: 26831 HDFS Write:
3971 SUCCESS
Total MapReduce CPU Time Spent: 23 seconds 290 msec
OK
Time taken: 103.471 seconds

hive> select * from college;
OK
5       stanford        uk
1       nec     nlr
6       JNTUA           atp
7       cambridge       us
2       vit     vlr
3       srm     chen
4       lpu     del
Time taken: 0.606 seconds, Fetched: 7 row(s)
```

Updating the Data in Hive Table

Update command is not supported on the columns that are bucketed.

In this table, we have bucketed the ***'clg_id'*** column and performing the Update operation on the same column, so we have go the error

Terminal Execution

```
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.
```

Now on non bucketed item :

update operation on Non bucketed column **college_name :**

Terminal Execution

```
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_20180717074717_1ce6ebe7-6d7a-409a-afe8-660189b0318f
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_1531787606394_0008, Tracking URL =
http://localhost:8088/proxy/application_1531787606394_0008/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1531787606394_0008
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-07-17 07:47:35,578 Stage-1 map = 0%,  reduce = 0%
2018-07-17 07:48:35,999 Stage-1 map = 0%,  reduce = 0%, Cumulative CPU 4.82 sec
2018-07-17 07:48:48,058 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 13.31 sec
2018-07-17 07:48:49,569 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 14.99 sec
```

```
2018-07-17 07:48:51,434 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 16.69 sec
2018-07-17 07:48:53,217 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 18.44 sec
2018-07-17 07:48:55,100 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 20.08 sec
2018-07-17 07:49:41,610 Stage-1 map = 100%,  reduce = 53%, Cumulative CPU 23.36 sec
2018-07-17 07:49:45,215 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 26.59 sec
2018-07-17 07:49:55,339 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 30.81 sec
2018-07-17 07:49:56,584 Stage-1 map = 100%,  reduce = 93%, Cumulative CPU 32.36 sec
2018-07-17 07:49:57,686 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 34.18 sec
MapReduce Total cumulative CPU time: 34 seconds 180 msec
Ended Job = job_1531787606394_0008
Loading data to table custom.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 34.18 sec   HDFS Read: 53794 HDFS Write: 944
SUCCESS
Total MapReduce CPU Time Spent: 34 seconds 180 msec
OK
Time taken: 163.267 seconds

hive> select * from college;
OK
5       stanford        uk
1       nec     nlr
6       IIT     atp
7       cambridge       us
2       vit     vlr
3       srm     chen
4       lpu     del
Time taken: 0.656 seconds, Fetched: 7 row(s)
```

Deleting a Row from Hive Table :

```
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 Hive 1.X releases.
Query ID = acadgild_20180717075205_2db744a7-dc13-4971-ad8f-dd1c538ecd2e
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_1531787606394_0009, Tracking URL =
```

```
http://localhost:8088/proxy/application_1531787606394_0009/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1531787606394_0009
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-07-17 07:52:24,097 Stage-1 map = 0%,  reduce = 0%
2018-07-17 07:53:25,369 Stage-1 map = 0%,  reduce = 0%, Cumulative CPU 12.22 sec
2018-07-17 07:53:27,159 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 13.53 sec
2018-07-17 07:53:28,899 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 14.8 sec
2018-07-17 07:53:30,764 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 16.03 sec
2018-07-17 07:53:34,302 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 18.77 sec
2018-07-17 07:54:13,002 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 19.68 sec
2018-07-17 07:54:14,642 Stage-1 map = 100%,  reduce = 27%, Cumulative CPU 20.96 sec
2018-07-17 07:54:16,380 Stage-1 map = 100%,  reduce = 53%, Cumulative CPU 23.72 sec
2018-07-17 07:54:19,704 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 25.65 sec
2018-07-17 07:54:26,408 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 28.46 sec
2018-07-17 07:54:27,789 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 30.37 sec
2018-07-17 07:54:29,138 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 33.12 sec
MapReduce Total cumulative CPU time: 33 seconds 120 msec
Ended Job = job_1531787606394_0009
Loading data to table custom.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 33.12 sec   HDFS Read: 52001 HDFS Write: 736
SUCCESS
Total MapReduce CPU Time Spent: 33 seconds 120 msec
OK
Time taken: 146.7 seconds


hive> select * from college;
OK
1       nec     nlr
6       IIT     atp
7       cambridge       us
2       vit     vlr
3       srm     chen
4       lpu     del
Time taken: 0.684 seconds, Fetched: 6 row(s)
```

We can see that there is no row with ***clg_id =1***. This means that we have successfully deleted the row from the Hive table.

This is how the transactions or row-wise operations are performed in Hive.