Assignment 9.1

Download file with Olympic data set which has following fields

1. The data set consists of the following fields.
2. Athlete: This field consists of the athlete name
3. Age: This field consists of athlete ages
4. Country: This fields consists of the country names which participated in Olympics
5. Year: This field consists of the year
6. Closing Date: This field consists of the closing date of ceremony
7. Sport: Consists of the sports name
8. Gold Medals: No. of Gold medals
9. Silver Medals: No. of Silver medals
10. Bronze Medals: No. of Bronze medals
11. Total Medals: Consists of total no. of medals

TASK-1

a) Write a Hive program to find the number of medals won by each country in swimming.
b) Write a Hive program to find the number of medals that India won year wise.
c) Write a Hive Program to find the total number of medals each country won.
d) Write a Hive program to find the number of gold medals each country won.

SOLUTION- Task-1

```
                    • MobaXterm 10.4 •
            (SSH client, X-server and networking tools)

→ SSH session to acadgild@192.168.56.2
  • SSH compression : v
  • SSH-browser     : v
  • X11-forwarding  : v  (remote display is forwarded through SSH)
  • DISPLAY         : v  (automatically set on remote server)

→ For more info, ctrl+click on help or visit our website
```

```
[acadgild.mmisra ~]$
```

#Launch Hive

```
[acadgild.mmisra ~]$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in
[jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-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-log
4j12-1.7.5.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-2.3.2-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 database called olympic

```
    >
    >
    >
    > create database olympic;
OK
Time taken: 0.048 seconds
hive>
```

#switch to database olympic

```
    > use olympic;
OK
Time taken: 0.028 seconds
```

#Create table called ‘data; with the required schema to hold the data as per the given dataset

```
hive> create table data( name 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';
OK
Time taken: 0.479 seconds
```

#Now load the data from the dataset file to the table

```
hive> LOAD DATA LOCAL INPATH 'olympix_data.csv' INTO TABLE data;
Loading data to table olympic.data
OK
Time taken: 1.339 seconds
```

to find number of medals won by each country in swimming we first group the data based on country name and total the number of medals for each row of the country using SUM and then we filter by the column sport=’swimming’

```
hive>
    > SELECT country, SUM(total) from data 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_20180724142100_17269bac-2eb8-4d66-b4da-d4053d4755a8
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_1532413643255_0015, Tracking URL =
http://localhost:8088/proxy/application_1532413643255_0015/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1532413643255_0015
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-24 14:21:16,288 Stage-1 map = 0%,  reduce = 0%
2018-07-24 14:21:24,186 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.04 sec
2018-07-24 14:21:31,911 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.03 sec
MapReduce Total cumulative CPU time: 7 seconds 30 msec
```

```
Ended Job = job_1532413643255_0015
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Reduce: 1   Cumulative CPU: 7.03 sec   HDFS Read: 528515 HDFS Write:
881 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 30 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: 33.641 seconds, Fetched: 34 row(s)
hive>
```

#To find the number of medals that India won year wise, we first group the table by the year, SUM number of medals for each year and then filter by column country='india'

```
    > SELECT year,SUM(total) from data 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_20180724142153_706e76e4-de7d-4620-950b-4d61e1ec4841
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_1532413643255_0016, Tracking URL =
http://localhost:8088/proxy/application_1532413643255_0016/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
```

```
job_1532413643255_0016
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-24 14:22:02,112 Stage-1 map = 0%,  reduce = 0%
2018-07-24 14:22:08,797 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.57 sec
2018-07-24 14:22:16,320 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6.67 sec
MapReduce Total cumulative CPU time: 6 seconds 670 msec
Ended Job = job_1532413643255_0016
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.67 sec   HDFS Read: 528510 HDFS Write:
163 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 670 msec
OK
2000    1
2004    1
2008    3
2012    6
Time taken: 23.756 seconds, Fetched: 4 row(s)
```

#To find the total number of medals each country won, we group the table by country and then SUM the number of medals for each row of the group

```
hive> SELECT country, SUM(total) from data  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_20180724142236_2ea61ae5-3436-4010-a7e1-4a1b03ae1e9b
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_1532413643255_0017, Tracking URL =
http://localhost:8088/proxy/application_1532413643255_0017/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1532413643255_0017
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-24 14:22:47,191 Stage-1 map = 0%,  reduce = 0%
2018-07-24 14:22:54,932 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.92 sec
2018-07-24 14:23:03,607 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.82 sec
MapReduce Total cumulative CPU time: 5 seconds 820 msec
Ended Job = job_1532413643255_0017
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.82 sec   HDFS Read: 527690 HDFS Write:
2742 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 820 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
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: 28.039 seconds, Fetched: 110 row(s)
```

#To find the number of gold medals each country won, we do same step as above but SUM only gold column of the group

```
hive> SELECT country,SUM(gold) from data  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_20180724142319_ff24c376-2b46-4aaf-8f75-2869a53dd799
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_1532413643255_0018, Tracking URL =
http://localhost:8088/proxy/application_1532413643255_0018/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1532413643255_0018
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-24 14:23:26,306 Stage-1 map = 0%,  reduce = 0%
2018-07-24 14:23:31,768 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.56 sec
2018-07-24 14:23:39,308 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.93 sec
MapReduce Total cumulative CPU time: 3 seconds 930 msec
Ended Job = job_1532413643255_0018
```

```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.93 sec   HDFS Read: 527688 HDFS Write:
2703 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 930 msec
OK
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
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: 21.014 seconds, Fetched: 110 row(s)
hive>
```

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 Task#2

Below is the source code for UDF. We extend the base class UDF and define evaluate function which takes the required arguments. After processing we return a string with the concatenation of the separator and the array of strings passed to the evaluate function. We compile and create a jar called concat_ws.jar

```
package hive;


import org.apache.hadoop.io.Text;

import java.util.ArrayList;
import java.util.List;

import org.apache.hadoop.hive.ql.exec.UDF;
public class concat_ws extends UDF{

	public static Text evaluate(Text seperator, ArrayList<Text> mylist) {

		String tmp="";
		String sep="";
		if((mylist == null) || (seperator==null))
			return null;
		sep = seperator.toString();
		for(int i=0;i<mylist.size();i++)
		{
			String x = mylist.get(i).toString();
			tmp=tmp + x;
			if(i < mylist.size())
			{
			    tmp=tmp + sep;
			}

		}
		return new Text(tmp);

	}
}
```

```
Last login: Fri Jul 27 14:21:31 2018 from 192.168.56.1
[acadgild.mmisra ~]$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in
[jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-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-log
4j12-1.7.5.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-2.3.2-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> show databases;
OK
default
test
Time taken: 5.694 seconds, Fetched: 2 row(s)
hive> use test;
OK
Time taken: 0.029 seconds
```

we add concat_ws.jar as resource which is added to class path

```
hive> ADD JAR concat_ws.jar;
Added [concat_ws.jar] to class path
Added resources: [concat_ws.jar]
```

#we define a temporary function with name concat_ws which can be used in the query

```
hive> create TEMPORARY FUNCTION concat_ws as 'hive.concat_ws';
OK
Time taken: 0.023 seconds
```

#This is how we use the UDF concat_ws. We pass a separator and an array of strings and the result is concatenation of the two

```
hive> select concat_ws('^',Array('a','b','c','d'));
OK
a^b^c^d^
Time taken: 0.839 seconds, Fetched: 1 row(s)
hive> select concat_ws('*',Array('a','b','c','d'));
OK
a*b*c*d*
Time taken: 0.086 seconds, Fetched: 1 row(s)
hive> select concat_ws('*',Array('abc','def','ghi','pqr'));
OK
abc*def*ghi*pqr*
Time taken: 0.164 seconds, Fetched: 1 row(s)
hive>
```

Task 3

Explore row transactions in Hive.

Solution Task#3

Following is required for ACID property of the transaction in Hive

1. The table needs to be bucketed table
2. Certain properties of the hive to be set to allow transactions
3. Only ORC file format is supported at present
4. The table property should be set to transactional
5. A column which is bucketed can't be updated
6. There is no restriction for insert and delete

```
You have new mail in /var/spool/mail/acadgild
```

#Launch Hive

```
[acadgild.mmisra ~]$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in
[jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-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-log
4j12-1.7.5.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-2.3.2-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> show databases;
OK
Default
```

#Create a database called 'test'

```
Time taken: 5.024 seconds, Fetched: 1 row(s)
hive>
    >
    >
    >
    > create database test;
OK
Time taken: 0.182 seconds
hive> use test;
OK
```

#Setting required parameters in Hive to support transactions

```
Time taken: 0.026 seconds
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;
```

#Create a bucketed table called person_data with the name,age and country and where name column is used for clustering/hashing

```
hive> CREATE TABLE person_data(name STRING, age INT, country STRING) clustered by
    > (name) into 5 buckets stored as orc TBLPROPERTIES('transactional'='true');
OK
Time taken: 1.009 seconds
```

#Inserting 4 rows into the table

```
hive> INSERT INTO TABLE person_data
    > values('asha',29,'US'),('michael',39,'NZ'),('coco',15,'bhutan'),
('pico',63,'spain')
    > ;
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_20180727133625_ca0d9a3c-9864-4274-93b7-56fc3124246a
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_1532677324330_0001, Tracking URL =
http://localhost:8088/proxy/application_1532677324330_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1532677324330_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 5
2018-07-27 13:36:40,074 Stage-1 map = 0%,  reduce = 0%
2018-07-27 13:36:46,812 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.95 sec
2018-07-27 13:37:02,906 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 3.83 sec
2018-07-27 13:37:06,701 Stage-1 map = 100%,  reduce = 27%, Cumulative CPU 5.65 sec
2018-07-27 13:37:09,211 Stage-1 map = 100%,  reduce = 33%, Cumulative CPU 7.79 sec
2018-07-27 13:37:10,424 Stage-1 map = 100%,  reduce = 47%, Cumulative CPU 9.85 sec
2018-07-27 13:37:12,835 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 14.48 sec
2018-07-27 13:37:14,030 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 16.56 sec
2018-07-27 13:37:16,464 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 18.75 sec
2018-07-27 13:37:17,567 Stage-1 map = 100%,  reduce = 93%, Cumulative CPU 20.86 sec
2018-07-27 13:37:18,635 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 23.29 sec
MapReduce Total cumulative CPU time: 23 seconds 290 msec
Ended Job = job_1532677324330_0001
Loading data to table test.person_data
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 5   Cumulative CPU: 23.29 sec   HDFS Read: 26632 HDFS Write:
2965 SUCCESS
Total MapReduce CPU Time Spent: 23 seconds 290 msec
OK
Time taken: 55.445 seconds
```

#We see that 4 rows have been inserted

```
hive> select * from person_data;
OK
coco    15      bhutan
michael 39      NZ
pico    63      spain
asha    29      US
Time taken: 0.291 seconds, Fetched: 4 row(s)
```

#We updated a column which is bucketed and we see that it is not allowed

```
hive> UPDATE person_data set name='John' WHERE age=15;
FAILED: SemanticException [Error 10302]: Updating values of bucketing columns is not
supported.  Column name.
```

#We update a column which is non-bucketed and we see that it gets updated

```
hive> UPDATE person_data set age=23 WHERE name='pico';
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_20180727133754_0ce70107-0a85-4a7d-8012-2c2f2038e119
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_1532677324330_0002, Tracking URL =
http://localhost:8088/proxy/application_1532677324330_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1532677324330_0002
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-07-27 13:38:02,086 Stage-1 map = 0%,  reduce = 0%
2018-07-27 13:38:20,028 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 3.95 sec
2018-07-27 13:38:23,685 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 7.64 sec
2018-07-27 13:38:27,528 Stage-1 map = 53%,  reduce = 0%, Cumulative CPU 10.73 sec
2018-07-27 13:38:28,781 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 10.95 sec
2018-07-27 13:38:30,212 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 14.57 sec
2018-07-27 13:38:32,534 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 18.25 sec
2018-07-27 13:38:43,501 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 20.43 sec
2018-07-27 13:38:44,688 Stage-1 map = 100%,  reduce = 20%, Cumulative CPU 20.87 sec
2018-07-27 13:38:45,850 Stage-1 map = 100%,  reduce = 47%, Cumulative CPU 24.95 sec
2018-07-27 13:38:46,975 Stage-1 map = 100%,  reduce = 60%, Cumulative CPU 26.32 sec
2018-07-27 13:38:48,070 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 28.64 sec
2018-07-27 13:38:49,108 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 31.11 sec
MapReduce Total cumulative CPU time: 31 seconds 110 msec
Ended Job = job_1532677324330_0002
Loading data to table test.person_data
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 31.11 sec   HDFS Read: 50685 HDFS Write:
975 SUCCESS
Total MapReduce CPU Time Spent: 31 seconds 110 msec
OK
Time taken: 56.158 seconds
hive> select * from person_data;
OK
coco    15      bhutan
michael 39      NZ
pico    23      spain
asha    29      US
Time taken: 0.165 seconds, Fetched: 4 row(s)
```

#We delete a particular row in the table and see that it is deleted

```
hive> DELETE from person_data WHERE name='asha';
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_20180727133926_19eba582-98b6-468f-9523-b2771a1245fb
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_1532677324330_0003, Tracking URL =
http://localhost:8088/proxy/application_1532677324330_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1532677324330_0003
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-07-27 13:39:33,010 Stage-1 map = 0%,  reduce = 0%
2018-07-27 13:39:51,844 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 3.63 sec
2018-07-27 13:39:53,024 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 7.03 sec
2018-07-27 13:39:59,203 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 16.57 sec
2018-07-27 13:40:00,419 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 16.75 sec
2018-07-27 13:40:01,732 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 17.48 sec
2018-07-27 13:40:15,025 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 19.55 sec
2018-07-27 13:40:16,270 Stage-1 map = 100%,  reduce = 20%, Cumulative CPU 20.02 sec
2018-07-27 13:40:17,569 Stage-1 map = 100%,  reduce = 53%, Cumulative CPU 24.58 sec
2018-07-27 13:40:18,691 Stage-1 map = 100%,  reduce = 60%, Cumulative CPU 25.17 sec
2018-07-27 13:40:19,756 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 27.73 sec
2018-07-27 13:40:20,797 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 30.92 sec
MapReduce Total cumulative CPU time: 30 seconds 920 msec
Ended Job = job_1532677324330_0003
Loading data to table test.person_data
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 30.92 sec   HDFS Read: 49022 HDFS Write:
757 SUCCESS
Total MapReduce CPU Time Spent: 30 seconds 920 msec
OK
Time taken: 56.457 seconds
hive> select * from person_data;
OK
coco    15      bhutan
michael 39      NZ
pico    23      spain
Time taken: 0.205 seconds, Fetched: 3 row(s)
hive>
```