ASSIGNMENT 9.1

TASK1:

Table '**AthleteData**' is created and data '**olympix_data.csv**' is loaded into the table shown below.

acadgild@localhost:~

```
hive> CREATE TABLE AthleteData
    > (
    >  name STRING,
    >  age INT,
    >  country STRING,
    >  year1 STRING,
    >  close_date STRING,
    >  sport STRING,
    >  gold_medal INT,
    >  silver_medal INT,
    >  bronze_medal INT,
    >  total_medal INT)
    >  ROW FORMAT DELIMITED
    >  FIELDS TERMINATED BY '\t';
OK
Time taken: 0.133 seconds
hive>
```

acadgild@localhost:~

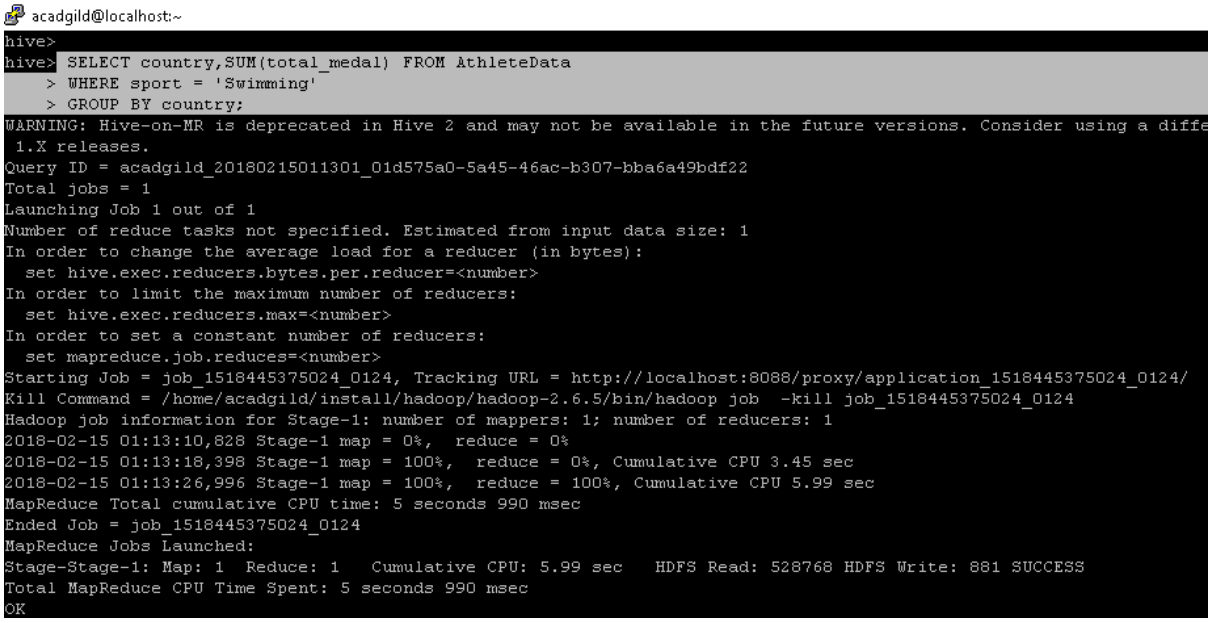
```
hive> LOAD DATA LOCAL INPATH '/home/acadgild/olympix_data.csv' INTO TABLE AthleteData;
Loading data to table olympic.athletedata
OK
Time taken: 0.668 seconds
hive> select * from AthleteData LIMIT  15;
OK
Michael Phelps  23      United States   2008    08-24-08        Swimming        8       0       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
Ian Thorpe      17      Australia       2000    10-01-00        Swimming        3       2       0       5
Dara Torres     33      United States   2000    10-01-00        Swimming        2       0       3       5
Cindy Klassen   26      Canada  2006    02-26-06        Speed Skating   1       2       2       5
Nastia Liukin   18      United States   2008    08-24-08        Gymnastics      1       3       1       5
Marit Bjørgen   29      Norway  2010    02-28-10        Cross Country Skiing    3       1       1       5
Time taken: 0.257 seconds, Fetched: 15 row(s)
hive>
```

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

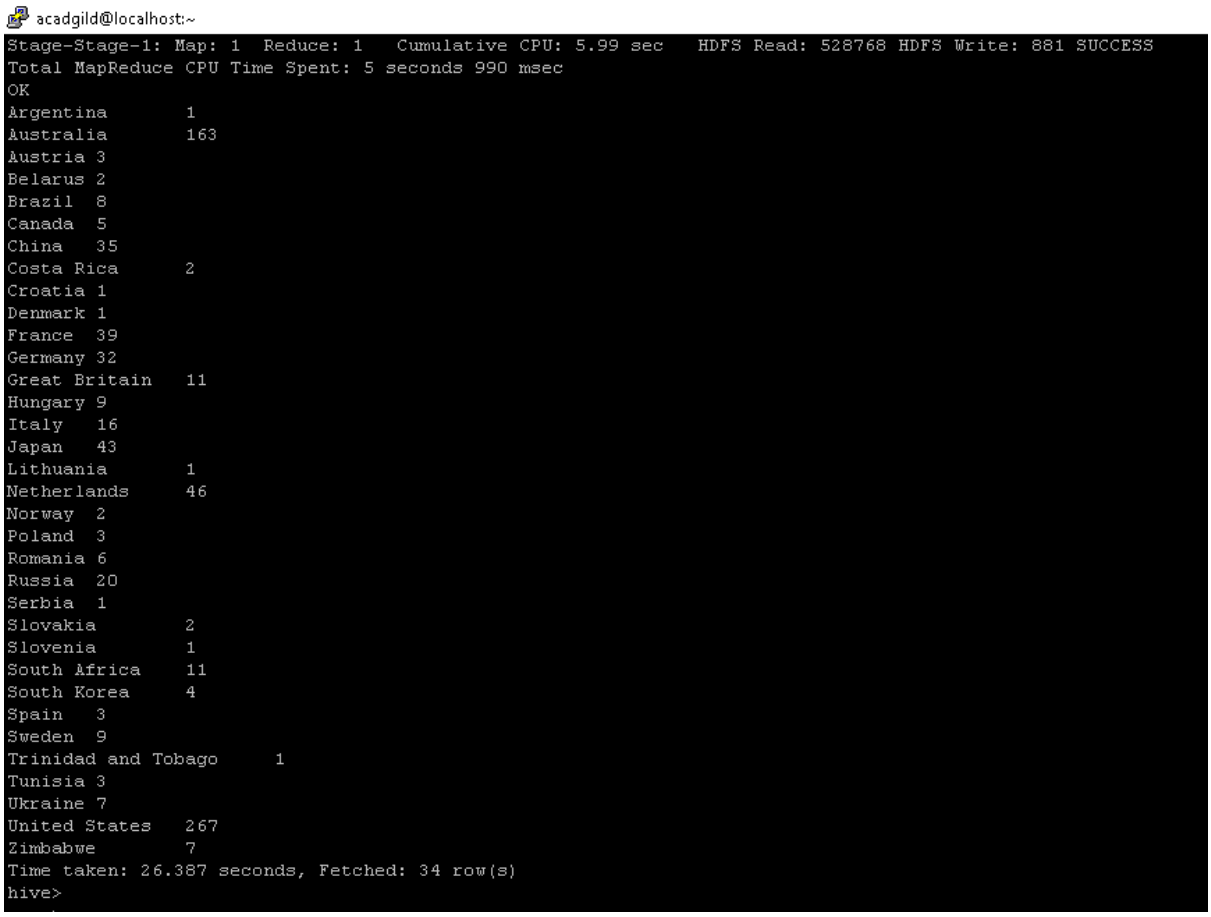
Syntax: SELECT country,SUM(total_medal) FROM AthleteData

WHERE sport = 'Swimming'

GROUP BY country;



Output:


acadgild@localhost:~
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.99 sec HDFS Read: 528768 HDFS Write: 881 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 990 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: 26.387 seconds, Fetched: 34 row(s)
hive>

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

Syntax: SELECT year1,total_medal FROM AthleteData

WHERE country = 'India'

GROUP BY year1,total_medal

ORDER BY year1;

```
acadgild@localhost:~
hive> SELECT year1,total_medal FROM AthleteData
    > WHERE country = 'India'
    > GROUP BY year1,total_medal
    > ORDER BY year1;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different e
 1.X releases.
Query ID = acadgild_20180215011903_b4d836df-15b1-4651-8f2a-4d598f8eb7e5
Total jobs = 2
Launching Job 1 out of 2
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_1518445375024_0125, Tracking URL = http://localhost:8088/proxy/application_1518445375024_0125/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1518445375024_0125
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-02-15 01:19:12,419 Stage-1 map = 0%,  reduce = 0%
2018-02-15 01:19:19,959 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.55 sec
2018-02-15 01:19:27,472 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6.09 sec
MapReduce Total cumulative CPU time: 6 seconds 90 msec
Ended Job = job_1518445375024_0125
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 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_1518445375024_0126, Tracking URL = http://localhost:8088/proxy/application_1518445375024_0126/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1518445375024_0126
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-02-15 01:19:40,674 Stage-2 map = 0%,  reduce = 0%
2018-02-15 01:19:47,148 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.59 sec
2018-02-15 01:19:54,652 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.28 sec
MapReduce Total cumulative CPU time: 4 seconds 280 msec
Ended Job = job_1518445375024_0126
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.09 sec   HDFS Read: 527603 HDFS Write: 188 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.28 sec   HDFS Read: 5683 HDFS Write: 163 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 370 msec
```

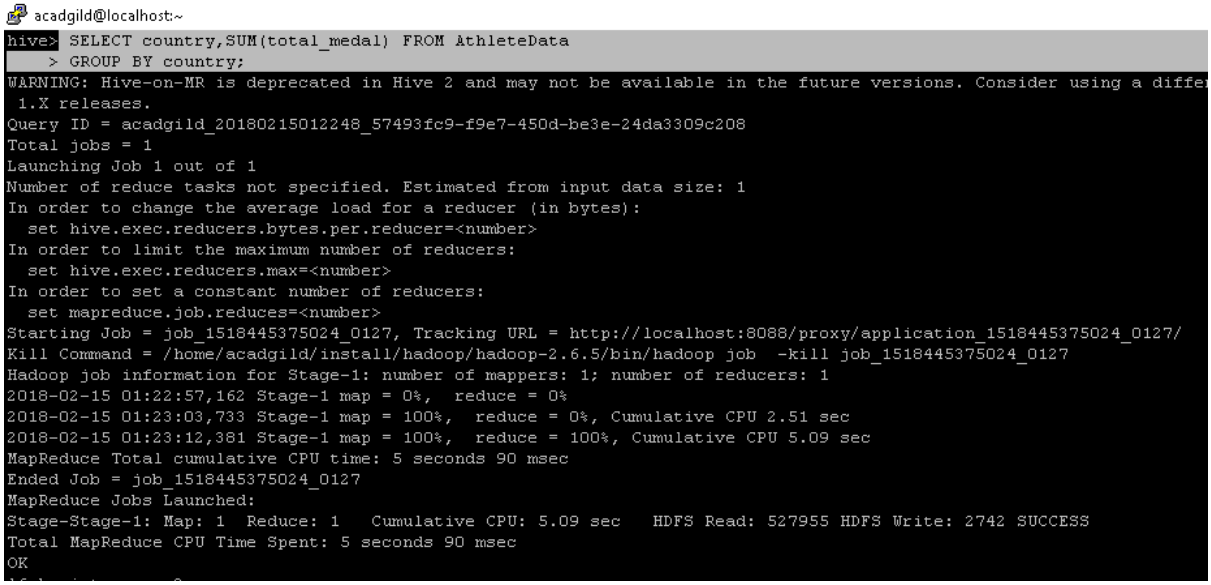
Output:

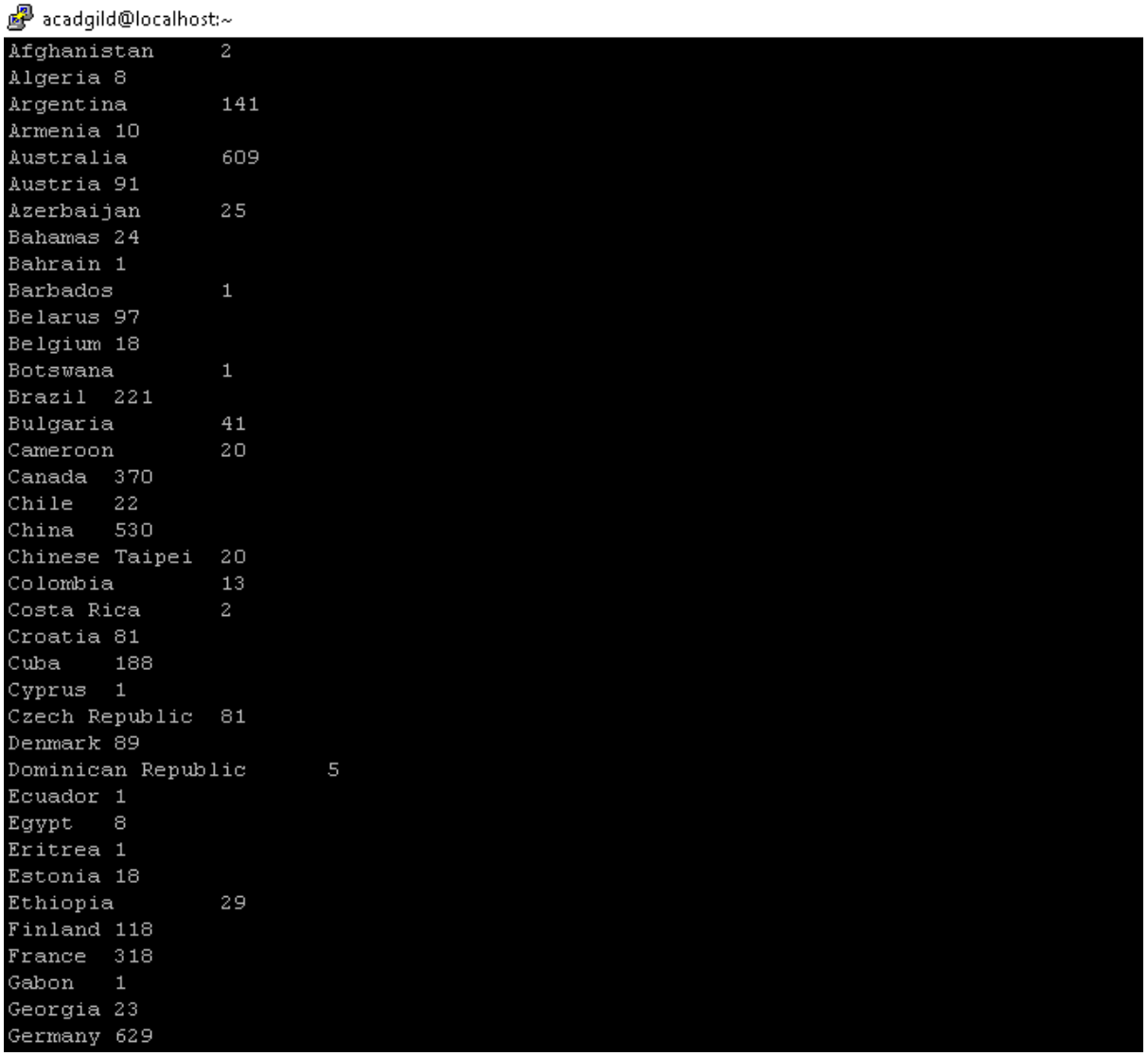
```
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.28 sec   HDFS Read: 5683 HDFS Write: 163 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 370 msec
OK
2000    1
2004    1
2008    1
2012    1
Time taken: 52.569 seconds, Fetched: 4 row(s)
hive>
```

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

Syntax: SELECT country,SUM(total_medal) FROM AthleteData

GROUP BY country;




acadgild@localhost:~
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

acadgild@localhost:~

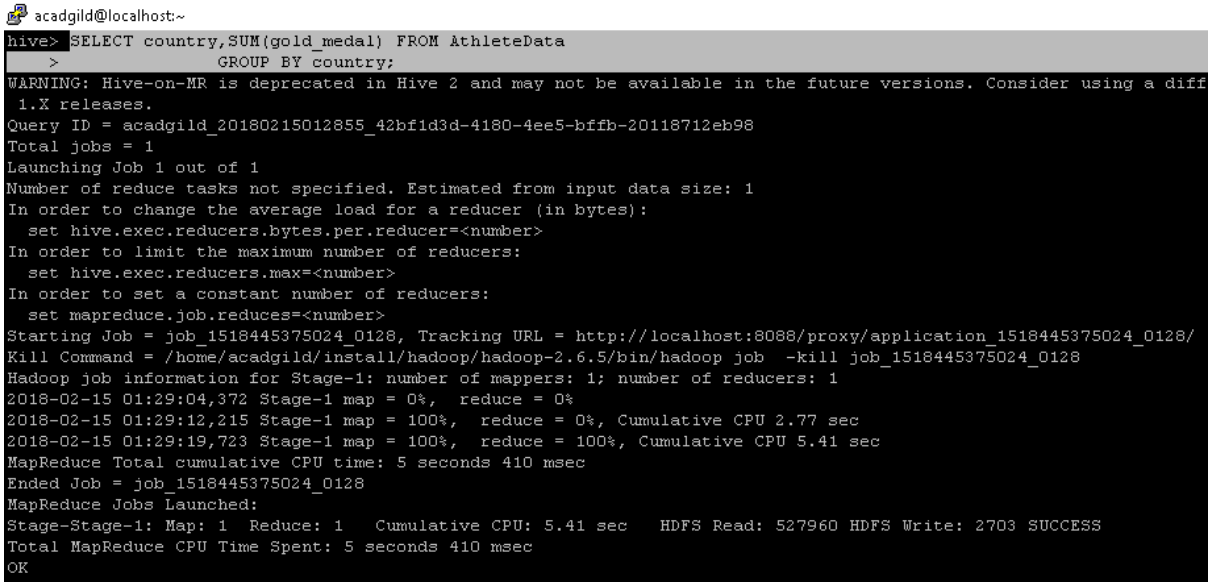
```
Georgia 23
Germany 629
Great Britain   322
Greece  59
Grenada 1
Guatemala       1
Hong Kong       3
Hungary 145
Iceland 15
India   11
Indonesia       22
Iran    24
Ireland 9
Israel  4
Italy   331
Jamaica 80
Japan   282
Kazakhstan      42
Kenya   39
Kuwait  2
Kyrgyzstan      3
Latvia  17
Lithuania       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
```

```
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: 25.57 seconds, Fetched: 110 row(s)
hive>
```

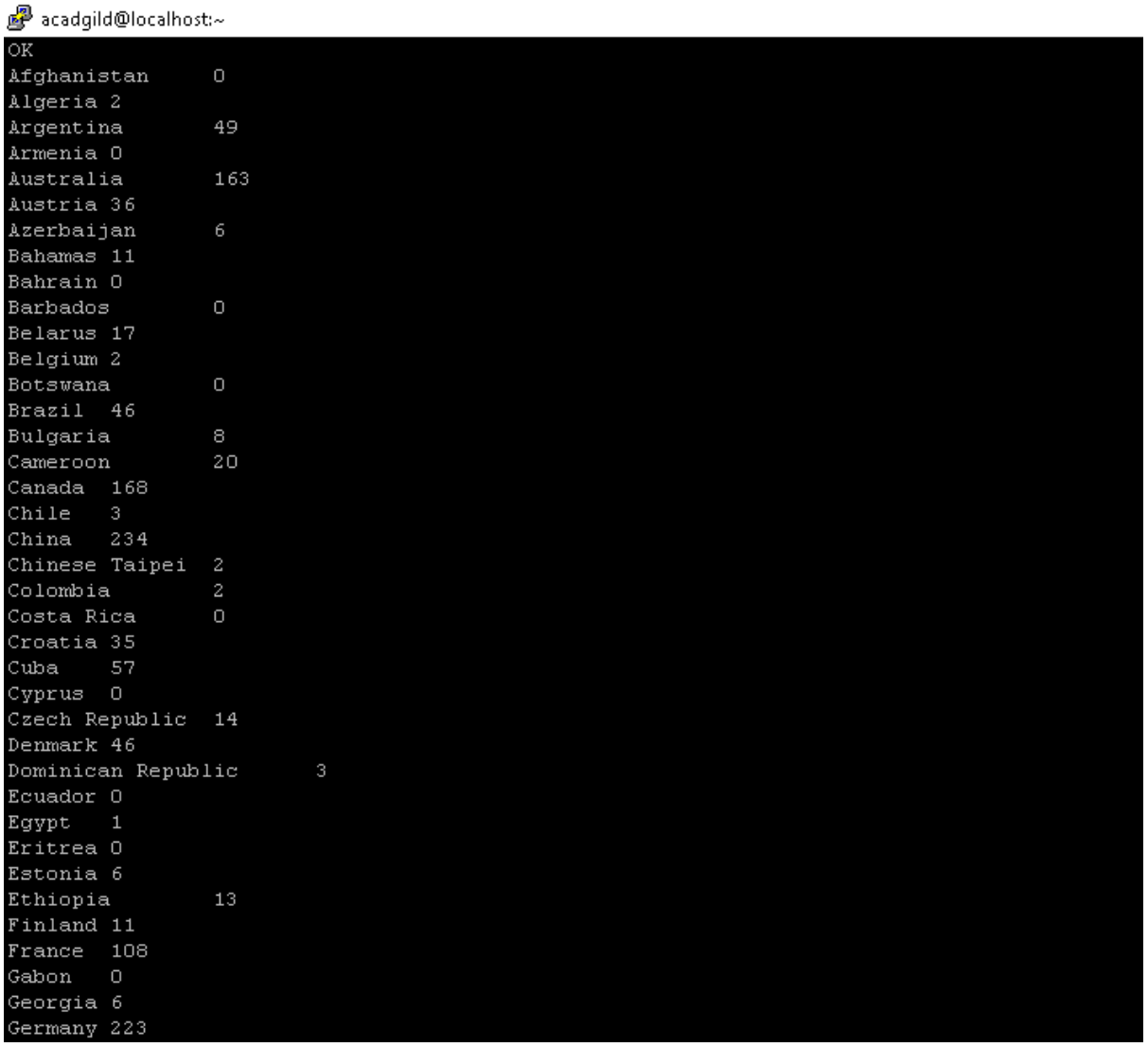
4. Write a Hive program to find the number of gold medals each country won.

Syntax: SELECT country,SUM(gold_medal) FROM AthleteData

GROUP BY country;



Output:


acadgild@localhost:~
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

```
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: 25.64 seconds, Fetched: 110 row(s)
hive>
```

TASK2: Hive UDF that implements functionality of string concat_ws(string SEP, array<string>).

For this task table **employee** is created with compex data types and data is loaded from employee_data.txt as shown below:

acadgild@localhost:~

```
hive> CREATE TABLE employee
    > (
    >   name string,
    >   work_place ARRAY<string>,
    >   gender_age STRUCT<gender:string,age:int>,
    >   skills_score MAP<string,int>,
    >   depart_title MAP<STRING,ARRAY<STRING>>
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '|'
    > COLLECTION ITEMS TERMINATED BY ','
    > MAP KEYS TERMINATED BY ':';
OK
Time taken: 0.088 seconds
hive> select * from employee;
OK
Time taken: 0.18 seconds
hive> LOAD DATA LOCAL INPATH '/home/acadgild/employee_data.txt' OVERWRITE INTO TABLE employee;
Loading data to table olympic.employee
OK
Time taken: 0.491 seconds
hive> select * from employee;
OK
Michael ["Montreal","Toronto"]  {"gender":"Male","age":30}      {"DB":80}       {"Product":["Developer","Lead"]}
Will    ["Montreal","New York","Viginia"]       {"gender":"Male","age":35}      {"Perl":85}     {"Product":["Lead"],"Test":["Lead"]}
Shelley ["New York","Vancouver","Montreal"]     {"gender":"Female","age":27}    {"Python":80}   {"Test":["Lead"],"COE":["Architect"]}
Lucy    ["Vancouver","Montreal","Virginia","New York"]  {"gender":"Female","age":57}    {"Sales":89,"HR":94}    {"Sales":["Lead"]}
Time taken: 0.195 seconds, Fetched: 4 row(s)
hive>
```

Adding the jar file **concatWS.jar** of hive Generic UDF **CONCAT_WS**.

acadgild@localhost:~

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

Creating temporary function **concatws**.

acadgild@localhost:~

```
hive> create temporary function concatws  as 'Task4.GenericUDFConcatWS';
OK
Time taken: 0.002 seconds
hive>
```

Sample output using concatws hive udf from table employee:-

acadgild@localhost:~

```
hive> SELECT  name, concatws(name,work_place) from employee;
OK
Michael MontrealMichaelToronto
Will    MontrealWillNew YorkWillViginia
Shelley New YorkShelleyVancouverShelleyMontreal
Lucy    VancouverLucyMontrealLucyVirginiaLucyNew York
Time taken: 0.133 seconds, Fetched: 4 row(s)
hive> SELECT  name, concatws("++++",work_place) from employee;
OK
Michael Montreal++++Toronto
Will    Montreal++++New York++++Viginia
Shelley New York++++Vancouver++++Montreal
Lucy    Vancouver++++Montreal++++Virginia++++New York
Time taken: 0.156 seconds, Fetched: 4 row(s)
hive> SELECT  concatws("  ",work_place) from employee;
OK
Montreal  Toronto
Montreal  New York  Viginia
New York  Vancouver  Montreal
Vancouver  Montreal  Virginia  New York
Time taken: 0.149 seconds, Fetched: 4 row(s)
hive>
```

TASK3: Transaction in hive

Properties to be enable to work with transactions in hive:

acadgild@localhost:~

```
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;
hive>
```

Creating a sample table ‘bank’ with table property ‘transactional’

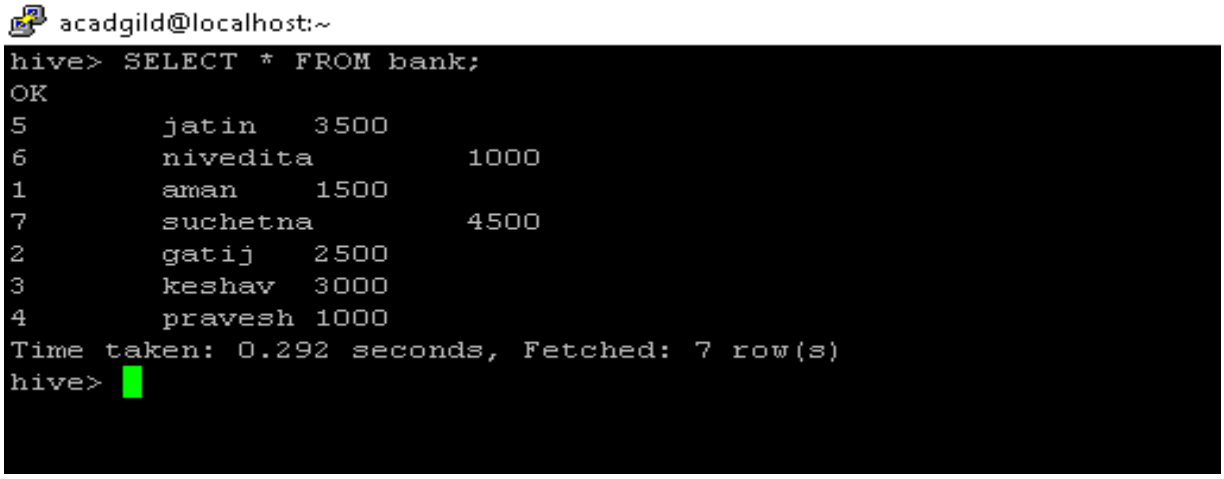
acadgild@localhost:~

```
hive> CREATE TABLE bank
    > (emp_id int,emp_name STRING,emp_salary INT)
    > clustered by (emp_id)
    > into 5 buckets stored as orc
    > TBLPROPERTIES('transactional'='true');
OK
Time taken: 7.54 seconds
hive>
```

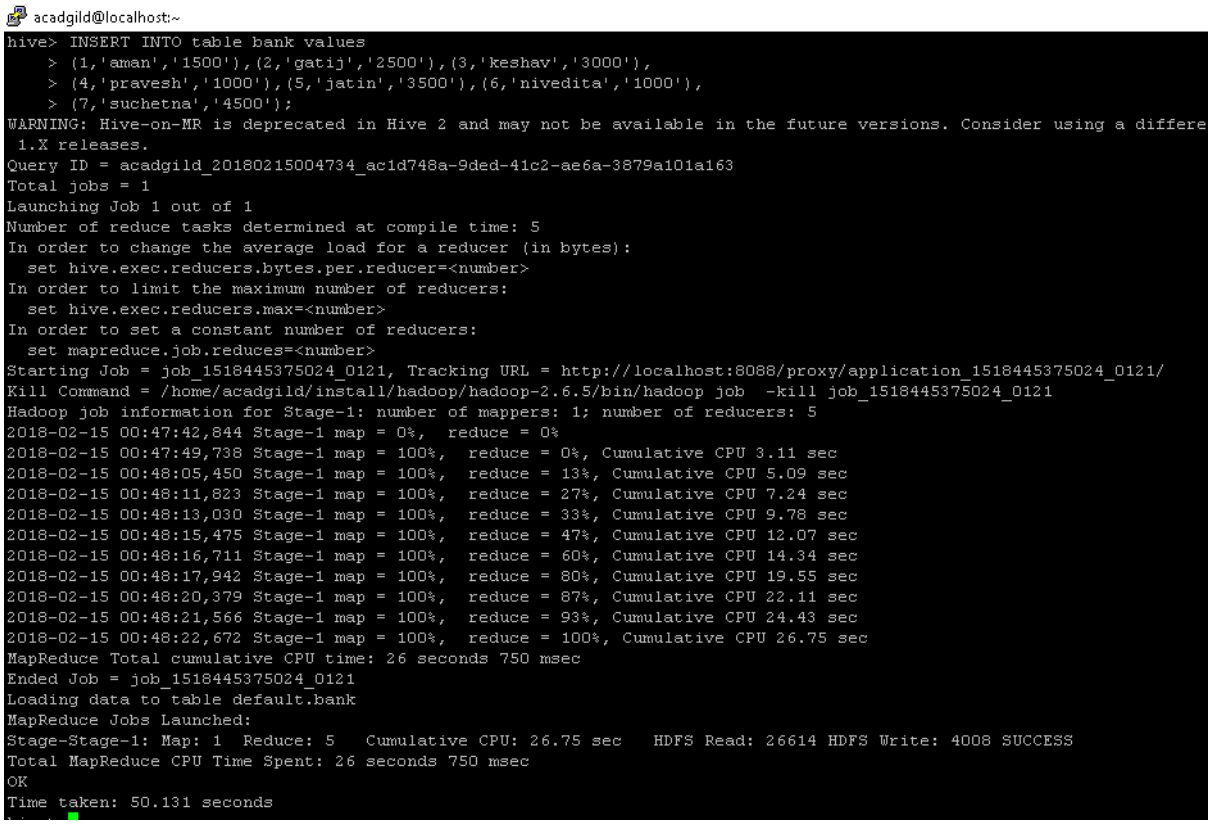
Inserting sample data row wise into hive table ‘bank’ .

acadgild@localhost:~

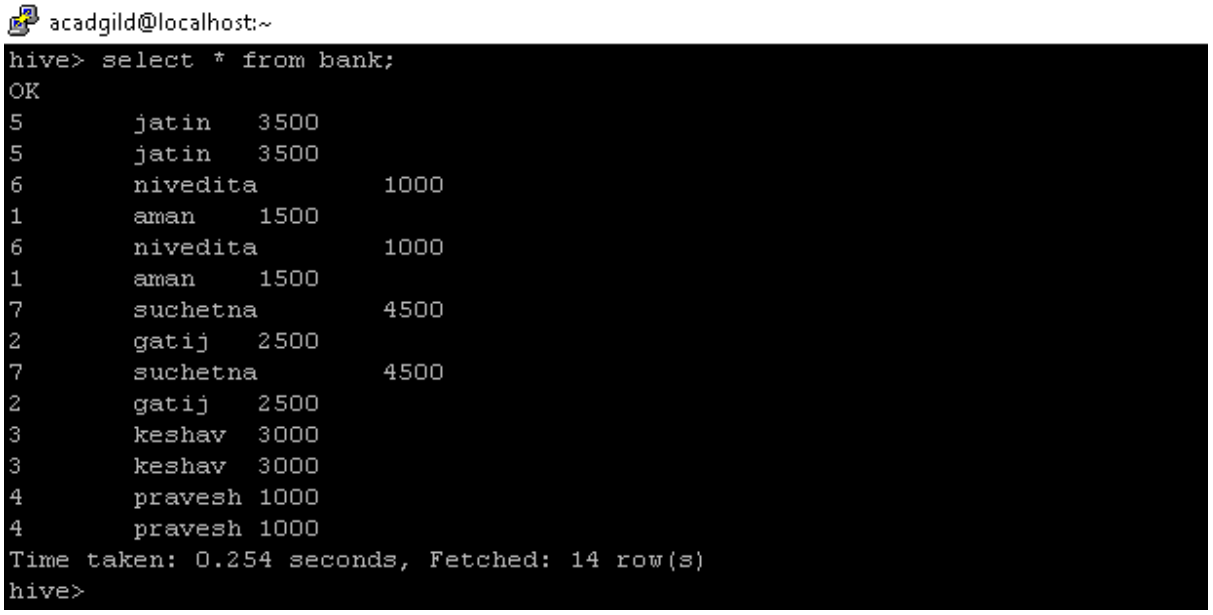
```
hive> INSERT INTO table bank values
    > (1,'aman','1500'),(2,'gatij','2500'),(3,'keshav','3000'),
    > (4,'pravesh','1000'),(5,'jatin','3500'),(6,'nivedita','1000'),
    > (7,'suchetna','4500');
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a diffe
 1.X releases.
Query ID = acadgild_20180215004454_6514a31a-f213-4291-a008-b74dc35a955d
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_1518445375024_0120, Tracking URL = http://localhost:8088/proxy/application_1518445375024_0120/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1518445375024_0120
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 5
2018-02-15 00:45:08,586 Stage-1 map = 0%,  reduce = 0%
2018-02-15 00:45:16,487 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.45 sec
2018-02-15 00:45:34,150 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 5.52 sec
2018-02-15 00:45:38,063 Stage-1 map = 100%,  reduce = 27%, Cumulative CPU 7.54 sec
2018-02-15 00:45:41,959 Stage-1 map = 100%,  reduce = 47%, Cumulative CPU 13.0 sec
2018-02-15 00:45:44,527 Stage-1 map = 100%,  reduce = 60%, Cumulative CPU 15.16 sec
2018-02-15 00:45:45,794 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 20.55 sec
2018-02-15 00:45:49,508 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 23.59 sec
2018-02-15 00:45:50,652 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 29.18 sec
MapReduce Total cumulative CPU time: 29 seconds 180 msec
Ended Job = job_1518445375024_0120
Loading data to table default.bank
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 5   Cumulative CPU: 29.18 sec   HDFS Read: 26759 HDFS Write: 4006 SUCCESS
Total MapReduce CPU Time Spent: 29 seconds 180 msec
OK
Time taken: 59.803 seconds
hive>
```



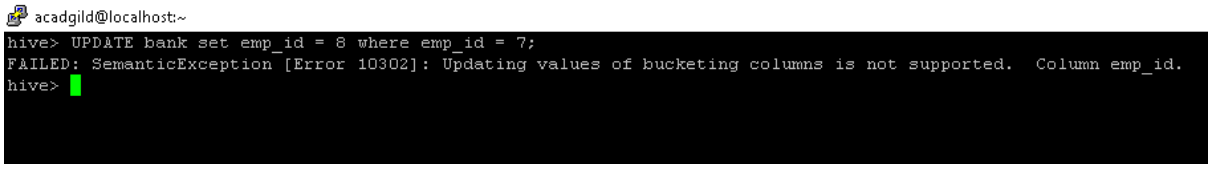
Re-Inserting the data in the table and it will append successfully.



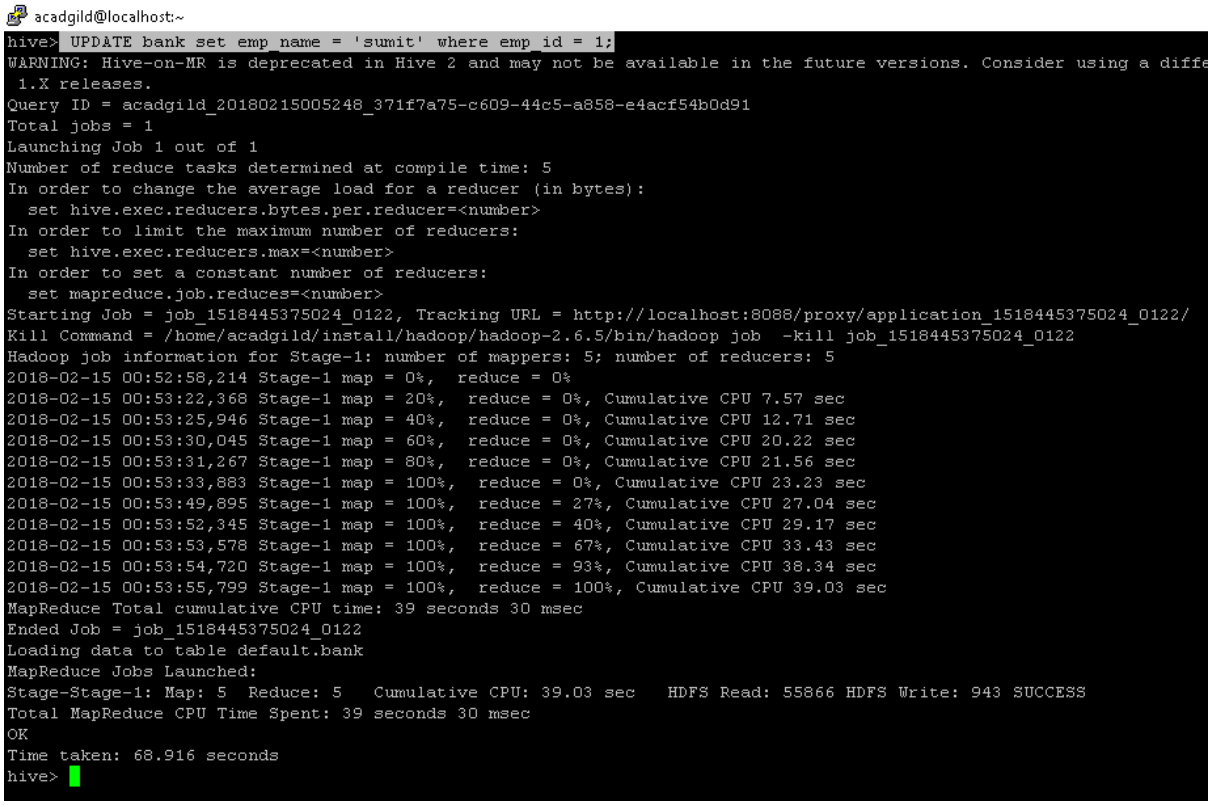
Output of appended data by INSERT clause.



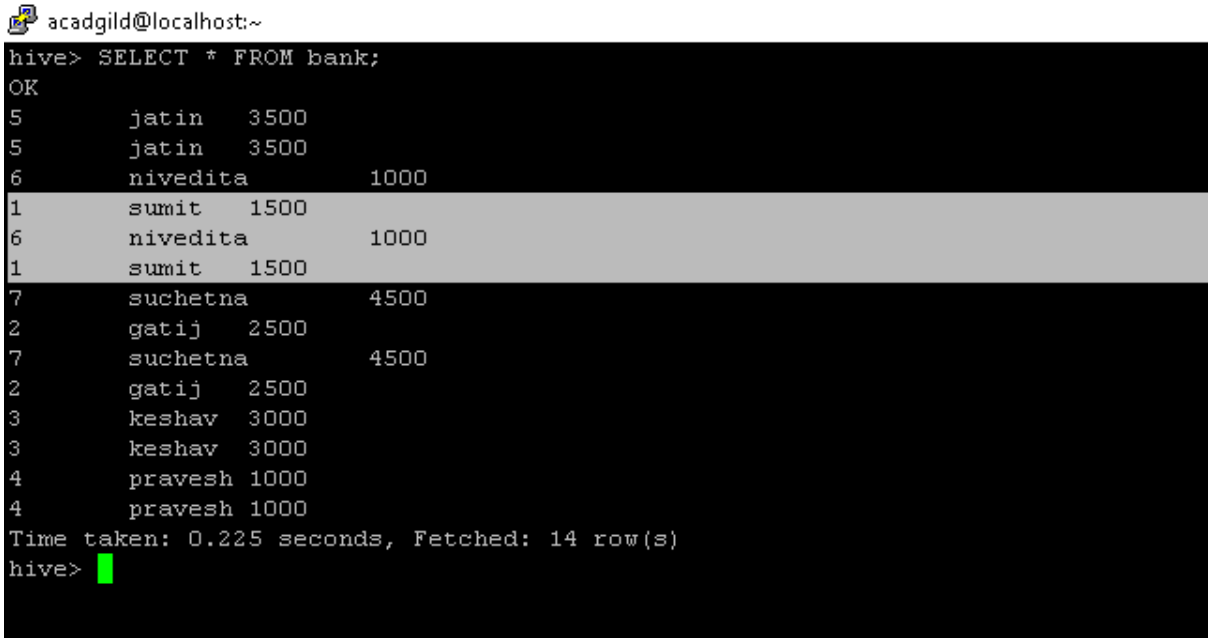
Below shows the error stating that UPDATE is not supported the bucked column



Updating the non bucket column.



Output of the UPDATE data



Deleting the row data from the table;

acadgild@localhost:~

```
hive> DELETE from bank where emp_id=5;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
 1.X releases.
Query ID = acadgild_20180215005844_e20e3d6c-aa93-4181-9e3b-4c9099f83d02
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_1518445375024_0123, Tracking URL = http://localhost:8088/proxy/application_1518445375024_0123/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1518445375024_0123
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-02-15 00:58:53,003 Stage-1 map = 0%,  reduce = 0%
2018-02-15 00:59:18,414 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 4.79 sec
2018-02-15 00:59:19,653 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 9.3 sec
2018-02-15 00:59:23,176 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 13.45 sec
2018-02-15 00:59:25,579 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 21.84 sec
2018-02-15 00:59:26,780 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 22.3 sec
2018-02-15 00:59:45,214 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 28.79 sec
2018-02-15 00:59:46,474 Stage-1 map = 100%,  reduce = 53%, Cumulative CPU 31.01 sec
2018-02-15 00:59:47,609 Stage-1 map = 100%,  reduce = 73%, Cumulative CPU 33.06 sec
2018-02-15 00:59:48,699 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 37.27 sec
MapReduce Total cumulative CPU time: 37 seconds 270 msec
Ended Job = job_1518445375024_0123
Loading data to table default.bank
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 37.27 sec   HDFS Read: 54063 HDFS Write: 740 SUCCESS
Total MapReduce CPU Time Spent: 37 seconds 270 msec
OK
Time taken: 66.538 seconds
hive>
```

acadgild@localhost:~

```
hive> SELECT * FROM bank;
OK
6       nivedita        1000
1       sumit   1500
6       nivedita        1000
1       sumit   1500
7       suchetna        4500
2       gatij   2500
7       suchetna        4500
2       gatij   2500
3       keshav  3000
3       keshav  3000
4       pravesh 1000
4       pravesh 1000
Time taken: 0.187 seconds, Fetched: 12 row(s)
hive>
```