Assignment - 9.1

Task1.1: Write a Hive program to find the number of medals won by each country in swimming.

Answer1.1: Here first we create the database named – 'olympix' .

```
hive> create database olympix;
OK
Time taken: 0.206 seconds
```

Create table 'Olympic' and load the dataset as an Internal table.

```
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: 0.117 seconds
hive> LOAD DATA LOCAL INPATH '/home/acadgild/olympix_data.txt' into table olympic;
```

Just perform a simple Select statement to ensure the data is loaded.

```
hive> select * from olympic;
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
```

Now to get thenumber od medals won by each country in swimming can be retrieved by below query.

```
hive> select country, sum(total)
    > from olympic
    > where sport = "Swimming"
    > group by country;
WARNING: Hive-on-MR is deprecated in Hive 2 an
ne (i.e. spark, tez) or using Hive 1.X release
```

Output:

```
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: 31.826 seconds, Fetched: 34 row(s)
hive>
```

Task1.2: Write a Hive program to find the number of medals that India won year wise.

Answer1.2:

```
hive> select year, sum(total)
    > from olympic
    > where country = "India"
    > group by (year);
WARNING: Hive-on-MR is deprecated in Hive 2 and
ne (i.e. spark, tez) or using Hive 1.X releases
```

Output:

```
Total MapReduce CPU Time Spent: 5 seconds 110
OK
2000    1
2004    1
2008    3
2012    6
Time taken: 32.022 seconds, Fetched: 4 row(s)
hive>
```

Task1.3: Write a Hive Program to find the total number of medals each country won.

Answer1.3:

```
hive> select country, sum(total)
    > from olympic
    > group by country;
WARNING: Hive-on-MR is deprecated in Hive 2 a
ne (i.e. spark, tez) or using Hive 1.X releas
```

Output:

```
Ended Job = job_1522603156069_0010
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumu
Total MapReduce CPU Time Spent: 3 secon
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
```

Task1.4: Write a Hive program to find the number of gold medals each country won.

Answer1.4:

```
hive> select country, sum(gold) from olympic group by country;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be availab
ne (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180401234131_3258ca5a-43ec-445a-9eee-d93799
Total jobs = 1
Launching Job 1 out of 1
```

Output:

```
MapReduce Total cumulative CPU time: 4 seconds
Ended Job = job_1522603156069_0011
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative C
Total MapReduce CPU Time Spent: 4 seconds 80 ms
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
```

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.

Task3: 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.

Answer3:

Creating a new database 'transactions' to perform all the hive transaction activity.

```
hive> create database transactions;
OK
Time taken: 0.256 seconds
hive> use transactions;
OK
Time taken: 0.034 seconds
hive>
```

Set all the configurations in Hive shell as below to perform Insert, Update and Delete operations.

```
hive> set hive.compactor.worker.threads = 3;
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 = 3;
hive>
```

Create table with enabling the transaction ability to be true in Hive.

```
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: 1.288 seconds
hive>
```

Inserting Data into a Hive Table

```
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. Cons
ne (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180403001657_08ca9fa7-7eda-4e21-9b0e-ecdda2fd9b5f
Total jobs = 1
```

The contents of the table can be viewed using the command **select * from college**

```
hive> select * from college;
OK
5       stanford        uk
6       JNTUA   atp
1       nec     nlr
7       cambridge       us
2       vit     vlr
3       srm     chen
4       lpu     del
Time taken: 0.345 seconds, Fetched: 7 row(s)
hive>
```

Now if we try to re-insert the same data again, it will be appended to the previous data as shown below:

```
hive> select * from college;
OK
5       stanford        uk
5       stanford        uk
6       JNTUA   atp
1       nec     nlr
6       JNTUA   atp
1       nec     nlr
7       cambridge       us
2       vit     vlr
7       cambridge       us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
Time taken: 0.22 seconds, Fetched: 14 row(s)
hive>
```

Updating the Data in Hive Table

Bucket Columns

```
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>
```

From the above image, we can see that we have received an error message. This means that the 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

Non-Bucket Columns

```
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
ne (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180403002735_dc9a9d3e-ed73-4a82-8af9-ff09137db
```

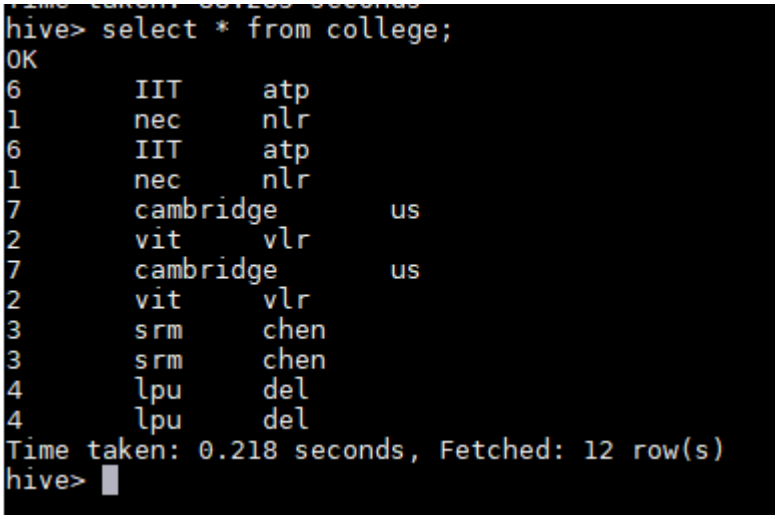
Output to check for update command.

```
hive> select * from college;
OK
5       stanford        uk
5       stanford        uk
6       IIT     atp
1       nec     nlr
6       IIT     atp
1       nec     nlr
7       cambridge       us
2       vit     vlr
7       cambridge       us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
Time taken: 0.231 seconds, Fetched: 14 row(s)
hive>
```

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 us
ne (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180403003236_d70b0d33-7ab8-4c3a-a45f-3df44df6539b
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_1522688971097_0005, Tracking URL = http://localhost:8088/proxy/application_152268897
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1522688971097_0005
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-04-03 00:32:47,355 Stage-1 map = 0%,  reduce = 0%
2018-04-03 00:33:23,921 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 5.65 sec
2018-04-03 00:33:27,670 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 11.95 sec
2018-04-03 00:33:29,002 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 15.28 sec
2018-04-03 00:33:31,563 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 15.98 sec
2018-04-03 00:33:32,791 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 16.54 sec
2018-04-03 00:33:54,677 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 17.57 sec
2018-04-03 00:33:55,927 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 19.75 sec
2018-04-03 00:33:58,511 Stage-1 map = 100%,  reduce = 53%, Cumulative CPU 20.85 sec
2018-04-03 00:33:59,816 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 22.0 sec
2018-04-03 00:34:01,129 Stage-1 map = 100%,  reduce = 73%, Cumulative CPU 23.01 sec
2018-04-03 00:34:02,336 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 25.49 sec
2018-04-03 00:34:03,498 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 27.5 sec
MapReduce Total cumulative CPU time: 27 seconds 500 msec
Ended Job = job_1522688971097_0005
Loading data to table transactions.college
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 27.5 sec   HDFS Read: 54954 HDFS Write: 780 SUCCESS
Total MapReduce CPU Time Spent: 27 seconds 500 msec
OK
Time taken: 88.285 seconds
hive>
```

Output:



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.