Session 9 – Advance Hive (Assignment 9.1)

TASK 1:

Creation of Table in Hive and Loading of data

create table olympicdet (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;

```
hive> create table olympicdet (athelete STRING,age INT,country STRING,year STRING,closing STRING,sport STRING,gold INT,silver INT,bronze INT,total INT) row for
mat delimited fields terminated by '\t' stored as textfile;
OK
Time taken: 1.506 seconds
hive>
```

```
hive> select * from olympicdet;
OK
Time taken: 3.675 seconds
hive> describe olympicdet;
OK
athelete                string
age                     int
country                 string
year                    string
closing                 string
sport                   string
gold                    int
silver                  int
bronze                  int
total                   int
Time taken: 0.095 seconds, Fetched: 10 row(s)
hive>
```

load data local inpath '/home/acadgild/Desktop/Assignments/Assignment 9/olympic_data.csv' into table olympicdet;

Session 9 – Advance Hive (Assignment 9.1)

```
hive> LOAD DATA LOCAL INPATH '/home/acadgild/Desktop/Assignments/Assignment9/olympix_data.csv' into table olympicdet;
Loading data to table simplidb.olympicdet
OK
Time taken: 1.977 seconds
hive> select * from olympicdet;
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
Sun Yang        20      China   2012    08-12-12        Swimming        2       1       1       4
Kirsty Coventry 24      Zimbabwe        2008    08-24-08        Swimming        1       3       0       4
Libby Lenton-Trickett   23      Australia       2008    08-24-08        Swimming        2       1       1       4
Ryan Lochte     24      United States   2008    08-24-08        Swimming        2       0       2       4
Inge de Bruijn  30      Netherlands     2004    08-29-04        Swimming        1       1       2       4
Petria Thomas   28      Australia       2004    08-29-04        Swimming        3       1       0       4
Ian Thorpe      21      Australia       2004    08-29-04        Swimming        2       1       1       4
Inge de Bruijn  27      Netherlands     2000    10-01-00        Swimming        3       1       0       4
Gary Hall Jr.   25      United States   2000    10-01-00        Swimming        2       1       1       4
Michael Klim    23      Australia       2000    10-01-00        Swimming        2       2       0       4
Susie O'Neill   27      Australia       2000    10-01-00        Swimming        1       3       0       4
Jenny Thompson  27      United States   2000    10-01-00        Swimming        3       0       1       4
Pieter van den Hoogenband       22      Netherlands     2000    10-01-00        Swimming        2       0       2       4
An Hyeon-Su     20      South Korea     2006    02-26-06        Short-Track Speed Skating       3       0       1       4
Aliya Mustafina 17      Russia  2012    08-12-12        Gymnastics      1       1       2       4
Shawn Johnson   16      United States   2008    08-24-08        Gymnastics      1       3       0       4
Dmitry Sautin   26      Russia  2000    10-01-00        Diving  1       1       2       4
Leontien Zijlaard-van Moorsel   30      Netherlands     2000    10-01-00        Cycling 3       1       0       4
Petter Northug Jr.      24      Norway  2010    02-28-10        Cross Country Skiing    2       1       1       4
Ole Einar Bjørndalen    28      Norway  2002    02-24-02        Biathlon        4       0       0       4
```

set hive.cli.print.header=true;

```
hive> set hive.cli.print.header=true;
hive>
```

Above command is to see the Column names.

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

select country,SUM(total) as TotalMedals from olympicdet where sport = 'Swimming' GROUP BY country;

```
hive> select country,SUM(total) as TotalMedals from olympicdet 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 u
sing Hive 1.X releases.
Query ID = acadgild_20180812152749_ebfcf8af-b7e8-4d88-bc04-677c17775c56
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_1534063820121_0003, Tracking URL = http://192.168.0.10:8088/proxy/application_1534063820121_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534063820121_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-12 15:27:57,558 Stage-1 map = 0%,  reduce = 0%
2018-08-12 15:28:06,279 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.45 sec
2018-08-12 15:28:13,744 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.72 sec
MapReduce Total cumulative CPU time: 7 seconds 720 msec
Ended Job = job_1534063820121_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.72 sec   HDFS Read: 528630 HDFS Write: 881 SUCCESS
```

OUTPUT –

```
Total MapReduce CPU Time Spent: 7 seconds 720 msec
OK
country totalmedals
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: 26.799 seconds, Fetched: 34 row(s)
hive>
```

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

select year,country,SUM(total) as NumberOfMedals from olympicdet where country = 'India' GROUP BY year;

```
hive> select year,SUM(total) as NumberOfMedals from olympicdet 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 u
sing Hive 1.X releases.
Query ID = acadgild_20180812155547_ca68061f-9fef-4f68-8744-c2cea44fa674
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_1534069331974_0001, Tracking URL = http://localhost:8088/proxy/application_1534069331974_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534069331974_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-12 15:56:05,599 Stage-1 map = 0%,  reduce = 0%
2018-08-12 15:56:13,691 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.07 sec
2018-08-12 15:56:21,511 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6.87 sec
MapReduce Total cumulative CPU time: 6 seconds 870 msec
Ended Job = job_1534069331974_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.87 sec   HDFS Read: 528607 HDFS Write: 163 SUCCESS
```

OUTPUT:

```
Total MapReduce CPU Time Spent: 7 seconds 840 msec
OK
year    numberofmedals
2000    1
2004    1
2008    3
2012    6
Time taken: 26.459 seconds, Fetched: 4 row(s)
hive>
```

3. **Find the total number of medals each country won display the name along with total medals.**

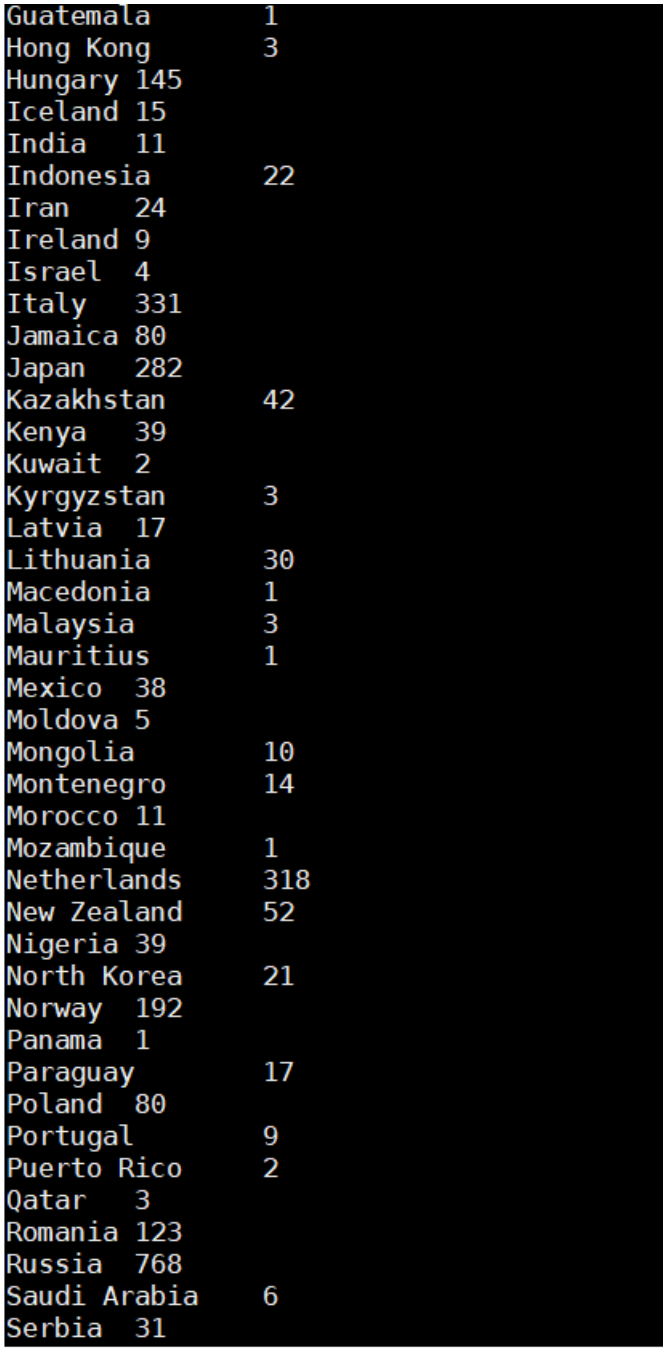
select country,SUM(total) as totalnumberofmedals from olympicdet GROUP BY country;

```
hive> select country,SUM(total) as totalnumberofmedals from olympicdet 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
sing Hive 1.X releases.
Query ID = acadgild_20180812160408_456d9bc1-58be-4cbf-8347-e70bf0d383d4
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_1534069331974_0003, Tracking URL = http://localhost:8088/proxy/application_1534069331974_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534069331974_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-12 16:04:17,513 Stage-1 map = 0%,  reduce = 0%
2018-08-12 16:04:25,050 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.62 sec
2018-08-12 16:04:32,569 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.53 sec
MapReduce Total cumulative CPU time: 5 seconds 530 msec
Ended Job = job_1534069331974_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.53 sec   HDFS Read: 527798 HDFS Write: 2742 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 530 msec
OK
```

Session 9 – Advance Hive (Assignment 9.1)

```
country totalnumberofmedals
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
```

Session 9 – Advance Hive (Assignment 9.1)



Session 9 – Advance Hive (Assignment 9.1)

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

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

select country,SUM(gold) as gold_number from olympicdet GROUP BY country;

```
hive> select country,SUM(gold) as gold_number from olympicdet 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 u
sing Hive 1.X releases.
Query ID = acadgild_20180812210034_705b717b-b553-4d18-8592-a957b56607bb
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_1534086915843_0001, Tracking URL = http://192.168.0.10:8088/proxy/application_1534086915843_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534086915843_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-12 21:00:55,109 Stage-1 map = 0%,  reduce = 0%
2018-08-12 21:01:02,960 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.62 sec
2018-08-12 21:01:11,924 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.69 sec
MapReduce Total cumulative CPU time: 5 seconds 690 msec
Ended Job = job_1534086915843_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.69 sec   HDFS Read: 527785 HDFS Write: 2703 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 690 msec
OK
country gold_number
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
```

Session 9 – Advance Hive (Assignment 9.1)

```
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
```

Session 9 – Advance Hive (Assignment 9.1)

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

Session 9 – Advance Hive (Assignment 9.1)

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.

We have to write the function with logic in eclipse after adding all Hive jars from VM. Below is the code snippet which is written as per the requirement provided above –

```
1  package com.acadgild.concatstrings;
2
3  import java.util.List;
4
5  import org.apache.hadoop.hive.ql.exec.UDF;
6  import org.apache.hadoop.io.Text;
7
8  public class StringConcat extends UDF {
9      public Text evaluate(Text SEPDEL, List<String> array) {
10         Text to_value = new Text("");
11         if (array != null) {
12             String word = "";
13                 for (int i=0; i<array.size(); i++) {
14                     if(i==0)
15                         word =word + array.get(i);
16                     else
17                         word = word+ SEPDEL + array.get(i);
18                 }
19                 to_value.set(word);
20         }
21         return to_value;
22
23     }
24 }
25
```

We need to add the jar created to the hive resources to execute this jar as an function in HIVE, which in turn we are running the UDF created as jar.

ADD JAR /home/acadgild/Desktop/Assignments/Assignment9/StringConcat.jar;

```
hive> ADD JAR /home/acadgild/Desktop/Assignments/Assignment9/StringConcat.jar;
Added [/home/acadgild/Desktop/Assignments/Assignment9/StringConcat.jar] to class path
Added resources: [/home/acadgild/Desktop/Assignments/Assignment9/StringConcat.jar]
```

Now we need to create a temporary function to run that as an function in Hive to execute queries against it.

CREATE TEMPORARY FUNCTION **concatstrings** as **'com.acadgild.concatstrings.StringConcat'**;

where concatstrings without quotes behaves as an function to be used and with quotes is the class we have created in our UDF, we have provided whole package path where class has been created.

Session 9 – Advance Hive (Assignment 9.1)

```
hive> CREATE TEMPORARY FUNCTION concatstrings as 'com.acadgild.concatstrings.StringConcat';
OK
Time taken: 0.004 seconds
hive>
```

Now, creating table assignhive with text and delimiters.

Create table assignhive (sepdel String, textarr array<string>) row format delimited fields terminated by ';' collection items terminated by ',';

```
hive> Create table assignhive (sepdel String, textarr array<string>) row format delimited  fields terminated by ';' collection items terminated by ',';
OK
Time taken: 0.957 seconds
hive> describe assignhive
    > ;
OK
sepdel                  string
textarr                 array<string>
Time taken: 0.295 seconds, Fetched: 2 row(s)
hive>
```

Input used to demonstrate UDF created –

```
[acadgild@localhost ~]$ cat /home/acadgild/Desktop/Assignments/Assignment9/udfinput.txt
-;This,is,assignment,session9, HIVE
+;UDF,demonstration,to,create,user,definedfunction[acadgild@localhost ~]$
```

Now, loading this data into the table created –

```
hive> load data local inpath '/home/acadgild/Desktop/Assignments/Assignment9/udfinput.txt' into table assignhive;
Loading data to table simplidb.assignhive
OK
Time taken: 1.508 seconds
hive> select * from assignhive;
OK
-       ["This","is","assignment","session9"," HIVE"]
+       ["UDF","demonstration","to","create","user","definedfunction"]
Time taken: 2.087 seconds, Fetched: 2 row(s)
hive>
```

No below is the output after using the function "concatstrings" which will concat the delimiter in input file with each words in the array of strings.

select concatstrings(**sepdel,textarr**) from **assignhive**;

```
hive> select concatstrings(sepdel,textarr) from assignhive;
OK
This-is-assignment-session9- HIVE
UDF+demonstration+to+create+user+definedfunction
Time taken: 0.488 seconds, Fetched: 2 row(s)
hive>
```

Session 9 – Advance Hive (Assignment 9.1)

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.

As per the blog we need to work on below operations in HIVE Demonstration –

1. **Insert**
2. **Update**
3. **Delete**

The below properties needs to be set appropriately in ***hive shell*** , order-wise to work with transactions in Hive:

```
set hive.support.concurrency = true;

set hive.enforce.bucketing = true;

set hive.exec.dynamic.partition.mode = nonstrict;

set hive.txn.manager = org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;

set hive.compactor.initiator.on = true;

set hive.compactor.worker.threads = a positive number on at least one
instance of the Thrift metastore service;
```

If the above properties are not set properly, the 'Insert' operation will work but 'Update' and 'Delete' will not work and you will receive the following error:

```
FAILED: SemanticException [Error 10294]: Attempt to do update or delete
usingtransaction manager that does not support these operations.
```

Creating a table to demonstrate INSERT, UPDATE and DELETE operations.

Create table StudentDetails(StuName string, StuID int, Stuadd string, Stuloc string) clustered by (StuID) into 5 buckets stored as orc TBLPROPERTIES('transactional'='true');

The above syntax will create a table with name 'StudentDetails' and the columns present in the table are 'StuName, StuID, Stuadd, Stuloc'. We are bucketing the table by 'StuID' and the table format is 'orc', also we are enabling the transactions in the table by specifying it inside
the TBLPROPERTIES as 'transactional'='true'

Session 9 – Advance Hive (Assignment 9.1)

```
hive> Create table StudentDetails(StuName string, StuID int, Stuadd string, Stuloc string) clustered by (StuID) into 5 buckets stored as orc TBLPROPERTIES('tra
nsactional'='true');
OK
Time taken: 0.595 seconds
hive> show tables;
OK
assignhive
buck_users
csv_table
emp_details
emp_details_partitioned
locations
mycustomer
olympicdet
studentdetails
txnrecords
users
Time taken: 0.059 seconds, Fetched: 11 row(s)
hive>
```

Table Creaed.

Inserting Data into a Hive Table

INSERT INTO table StudentDetails
values(‘LALALA’,001,'ADD1',’Kerela’),(‘ABCD’,002,'vit','vlr'),(‘SUBRAMANI’,3,'srm','chen'),(‘RAJU’,4,'lpu','del'),(‘RAMESH’,5,'stanford','uk'),(‘RAKESH’,6,'JNTUA','atp'),(‘REVANTH’,7,'cambridge','GERMANY');

```
hive> INSERT INTO table StudentDetails values('LALALA',001,'ADD1','Kerela'),('ABCD',002,'vit','vlr'),('SUBRAMANI',3,'srm','chen'),('RAJU',4,'lpu','del'),('RAME
SH',5,'stanford','uk'),('RAKESH',6,'JNTUA','atp'),('REVANTH',7,'cambridge','GERMANY');
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 u
sing Hive 1.X releases.
Query ID = acadgild_20180814031840_c56607a7-6e9f-4738-871c-893ffb68636f
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_1534187541703_0001, Tracking URL = http://localhost:8088/proxy/application_1534187541703_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534187541703_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 5
2018-08-14 03:18:54,078 Stage-1 map = 0%,  reduce = 0%
2018-08-14 03:19:01,911 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.17 sec
2018-08-14 03:19:19,958 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 4.81 sec
2018-08-14 03:19:24,918 Stage-1 map = 100%,  reduce = 27%, Cumulative CPU 6.71 sec
2018-08-14 03:19:27,438 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 8.82 sec
2018-08-14 03:19:28,640 Stage-1 map = 100%,  reduce = 47%, Cumulative CPU 12.27 sec
2018-08-14 03:19:29,874 Stage-1 map = 100%,  reduce = 60%, Cumulative CPU 14.32 sec
2018-08-14 03:19:32,983 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 19.89 sec
2018-08-14 03:19:35,440 Stage-1 map = 100%,  reduce = 87%, Cumulative CPU 23.01 sec
2018-08-14 03:19:36,664 Stage-1 map = 100%,  reduce = 93%, Cumulative CPU 26.33 sec
2018-08-14 03:19:38,851 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 28.87 sec
MapReduce Total cumulative CPU time: 28 seconds 870 msec
Ended Job = job_1534187541703_0001
Loading data to table simplidb.studentdetails
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 5   Cumulative CPU: 28.87 sec   HDFS Read: 29156 HDFS Write: 4580 SUCCESS
Total MapReduce CPU Time Spent: 28 seconds 870 msec
OK
Time taken: 60.167 seconds
hive>
```

```
hive> select * from studentdetails;
OK
RAMESH  5       stanford        uk
RAKESH  6       JNTUA   atp
LALALA  1       ADD1    Kerela
REVANTH 7       cambridge       GERMANY
ABCD    2       vit     vlr
SUBRAMANI       3       srm     chen
RAJU    4       lpu     del
Time taken: 0.284 seconds, Fetched: 7 row(s)
hive>
```

The above output shows the Insertion is successful.

Updating the Data in Hive Table

UPDATE Studentdetails set stuID = 8 where stuID = 7;

```
hive> UPDATE Studentdetails set stuID = 8 where stuID = 7;
FAILED: SemanticException [Error 10302]: Updating values of bucketing columns is not supported.  Column stuid.
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 ***'stuID*** column and performing the Update operation on the same column, so we have go the error

FAILED: SemanticException[Error 10302]: Updating values of bucketing columns is not supported. Column stuid

Performing operations on Non-Bucketed Columns.

UPDATE Studentdetails set stuname = 'BANKE' where STUID = 6;

```
hive> UPDATE Studentdetails set stuname = 'BANKE' where STUID = 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 u
sing Hive 1.X releases.
Query ID = acadgild_20180814032826_c3f8cd2b-60f0-46bc-bb3d-1bc3e168f83d
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_1534187541703_0002, Tracking URL = http://localhost:8088/proxy/application_1534187541703_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534187541703_0002
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-08-14 03:28:35,657 Stage-1 map = 0%,  reduce = 0%
2018-08-14 03:28:58,723 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 7.64 sec
2018-08-14 03:29:01,250 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 8.81 sec
2018-08-14 03:29:05,097 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 15.93 sec
2018-08-14 03:29:07,454 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 20.34 sec
2018-08-14 03:29:09,910 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 21.36 sec
2018-08-14 03:29:23,473 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 23.56 sec
2018-08-14 03:29:24,715 Stage-1 map = 100%,  reduce = 27%, Cumulative CPU 25.52 sec
2018-08-14 03:29:25,902 Stage-1 map = 100%,  reduce = 47%, Cumulative CPU 28.73 sec
2018-08-14 03:29:27,064 Stage-1 map = 100%,  reduce = 53%, Cumulative CPU 29.54 sec
2018-08-14 03:29:28,245 Stage-1 map = 100%,  reduce = 73%, Cumulative CPU 33.27 sec
2018-08-14 03:29:29,349 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 36.95 sec
MapReduce Total cumulative CPU time: 36 seconds 950 msec
Ended Job = job_1534187541703_0002
Loading data to table simplidb.studentdetails
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 36.95 sec   HDFS Read: 55452 HDFS Write: 1080 SUCCESS
Total MapReduce CPU Time Spent: 36 seconds 950 msec
OK
Time taken: 64.566 seconds
hive>
```

Session 9 – Advance Hive (Assignment 9.1)

```
hive> select * from studentdetails;
OK
RAMESH  5       stanford        uk
BANKE   6       JNTUA   atp
LALALA  1       ADD1    Kerela
REVANTH 7       cambridge       GERMANY
ABCD    2       vit     vlr
SUBRAMANI       3       srm     chen
RAJU    4       lpu     del
Time taken: 0.183 seconds, Fetched: 7 row(s)
hive>
```

The above Output shows the UPDATE query also successfully applied to the data.

Deleting a Row from Hive Table

delete from studentdetails where stuid=5;

```
OK
RAMESH  5       stanford        uk
BANKE   6       JNTUA   atp
LALALA  1       ADD1    Kerela
REVANTH 7       cambridge       GERMANY
ABCD    2       vit     vlr
SUBRAMANI       3       srm     chen
RAJU    4       lpu     del
Time taken: 0.184 seconds, Fetched: 7 row(s)
hive> delete from studentdetails where stuid=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 u
sing Hive 1.X releases.
Query ID = acadgild_20180814033403_b076702e-3e3a-42e6-84be-2b7f3b6d3a4d
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_1534187541703_0003, Tracking URL = http://localhost:8088/proxy/application_1534187541703_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534187541703_0003
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2018-08-14 03:34:12,373 Stage-1 map = 0%,  reduce = 0%
2018-08-14 03:34:36,754 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 4.29 sec
2018-08-14 03:34:40,385 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 15.56 sec
2018-08-14 03:34:42,893 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 17.53 sec
2018-08-14 03:34:47,845 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 21.59 sec
2018-08-14 03:35:02,531 Stage-1 map = 100%,  reduce = 13%, Cumulative CPU 23.5 sec
2018-08-14 03:35:03,786 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 28.01 sec
2018-08-14 03:35:06,112 Stage-1 map = 100%,  reduce = 73%, Cumulative CPU 33.33 sec
2018-08-14 03:35:07,179 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 37.13 sec
MapReduce Total cumulative CPU time: 37 seconds 130 msec
Ended Job = job_1534187541703_0003
Loading data to table simplidb.studentdetails
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 37.13 sec   HDFS Read: 52198 HDFS Write: 792 SUCCESS
Total MapReduce CPU Time Spent: 37 seconds 130 msec
OK
Time taken: 66.056 seconds
hive>
```

Delete operation also updated data successfully. Below is the output. We have had stuid = 5 as RAMESH(refer snapshot above).

But now that entry is deleted from the table **studentdetails**.

```
hive> select * from studentdetails;
OK
BANKE   6       JNTUA   atp
LALALA  1       ADD1    Kerela
REVANTH 7       cambridge       GERMANY
ABCD    2       vit     vlr
SUBRAMANI       3       srm     chen
RAJU    4       lpu     del
Time taken: 0.397 seconds, Fetched: 6 row(s)
hive>
```