**Task1**

I have created the database called Custom.

Created the table as mentioned below

```
OK
CREATE TABLE `olympics`(
  `athlete` string,
  `age` int,
  `country` string,
  `year` int,
  `cdate` string,
  `sport` string,
  `gmedals` int,
  `smedals` int,
  `bmedals` int,
  `tomedals` int)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
  'field.delim'='\t',
  'line.delim'='\n',
  'serialization.format'='\t')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  'hdfs://localhost:8020/user/hive/warehouse/custom.db/olympics'
TBLPROPERTIES (
  'transient_lastDdlTime'='1528362496')
```

Load data into the Olympics table from local directory

```
hive> load data local inpath '/home/acadgild/Downloads/olympix_data.csv' into table olympics;
```

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

QUERY:

```
hive> select s.country,count(s.tomedals) from olympics s where s.sport='Swimming' group by s.country;
```

**OUTPUT:**

```
Argentina       1
Australia       92
Austria 2
Belarus 1
Brazil  7
Canada  5
China   29
Costa Rica      1
Croatia 1
Denmark 1
France  26
Germany 27
Great Britain   9
Hungary 7
Italy   13
Japan   30
Lithuania       1
Netherlands     32
Norway  2
Poland  1
Romania 4
Russia  19
Serbia  1
Slovakia        1
Slovenia        1
South Africa    8
South Korea     2
Spain   2
Sweden  7
Trinidad and Tobago     1
Tunisia 2
Ukraine 4
United States   145
Zimbabwe        2
```

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

**QUERY:**

```
hive> select s.country,s.year,count(s.tomedals) from olympics s where s.country='India' group by s.year,s.country;
```

**OUTPUT:**

```
OK
India   2000    1
India   2004    1
India   2008    3
India   2012    6
Time taken: 61.9 seconds, Fetched: 4 row(s)
```

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

**QUERY:**

```
hive> select s.country,sum(s.tomedals) from olympics s group by s.country;
```

**OUTPUT:**

```
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
```

```
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
```

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

**QUERY:**

```
hive> select s.country,count(s.gmedals) from olympics s group by s.country;
```

**OUTPUT:**

```
Afghanistan     2
Algeria 8
Argentina       141
Armenia 10
Australia       524
Austria 70
Azerbaijan      25
Bahamas 23
Bahrain 1
Barbados        1
Belarus 92
Belgium 18
Botswana        1
Brazil  220
Bulgaria        36
Cameroon        20
Canada  351
Chile   20
China   450
Chinese Taipei  20
Colombia        13
Costa Rica      1
Croatia 76
Cuba    188
Cyprus  1
Czech Republic  75
Denmark 89
Dominican Republic      5
Ecuador 1
Egypt   8
Eritrea 1
Estonia 16
Ethiopia        24
Finland 112
France  287
Gabon   1
Georgia 23
```

```
Germany 552
Great Britain   296
Greece  59
Grenada 1
Guatemala       1
Hong Kong       3
Hungary 132
Iceland 15
India   11
Indonesia       22
Iran    24
Ireland 9
Israel  4
Italy   307
Jamaica 61
Japan   259
Kazakhstan      42
Kenya   38
Kuwait  2
Kyrgyzstan      3
Latvia  17
Lithuania       30
Macedonia       1
Malaysia        3
Mauritius       1
Mexico  38
Moldova 5
Mongolia        10
Montenegro      14
Morocco 10
Mozambique      1
Netherlands     286
New Zealand     51
Nigeria 39
North Korea     21
Norway  158
Panama  1
Paraguay        17
```

**Task2**

**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.Code:**

**empData:**

```
1,Alex Analyst$Data Engineer$Data Consultant
2,Felix Analyst$Software Engineer$Software Consultant
3,venu  Analyst$Software Engineer$Software technologies
4,Ramesh Analyst$Software Engineer$Software tech
```

**Programcode**

```
package sample;

import java.util.ArrayList;
import org.apache.hadoop.hive.ql.exec.UDF;
public class udfarray extends UDF{
public String evaluate (String separator, ArrayList<String> array) {
StringBuffer sBuffer;
if (array == null) {
return null;
}
sBuffer = new StringBuffer();
sBuffer.append(array.get(0));
for (int i=1; i < array.size(); i++) {
sBuffer.append(separator);
sBuffer.append(array.get(i));
}
return sBuffer.toString();
}
}
```

Create table for array values.

```
> create table empArray(empName string, empDesignation array<string>) row format delimited fields terminated by ',' collection items terminated by '$';
```

Load data into the empArray and select the data from table

```
hive> load data local inpath 'empdata.txt' overwrite into table empArray;
Loading data to table custom.emparray
OK
Time taken: 0.895 seconds
hive> select * from empArray;
OK
1       ["Alex Analyst","Data Engineer","Data Consultant"]
2       ["Felix Analyst","Software Engineer","Software Consultant"]
3       ["venu  Analyst","Software Engineer","Software technologies"]
4       ["Ramesh Analyst","Software Engineer","Software tech"]
```

Create jar file and save in the directory.

Add jar in the hive session like below

**Add jar con.jar**

**Create temporary function cancat as 'sample.udfarray';**

Sample is the package

udfarray is class name

Cancat is the function name.

Call the function from hive select statment

**OUTPUT:**

```
hive> select abc('|',empDesignation) from empArray;
OK
Alex Analyst|Data Engineer|Data Consultant
Felix Analyst|Software Engineer|Software Consultant
venu  Analyst|Software Engineer|Software technologies
Ramesh Analyst|Software Engineer|Software tech
NULL
```

**Task3**

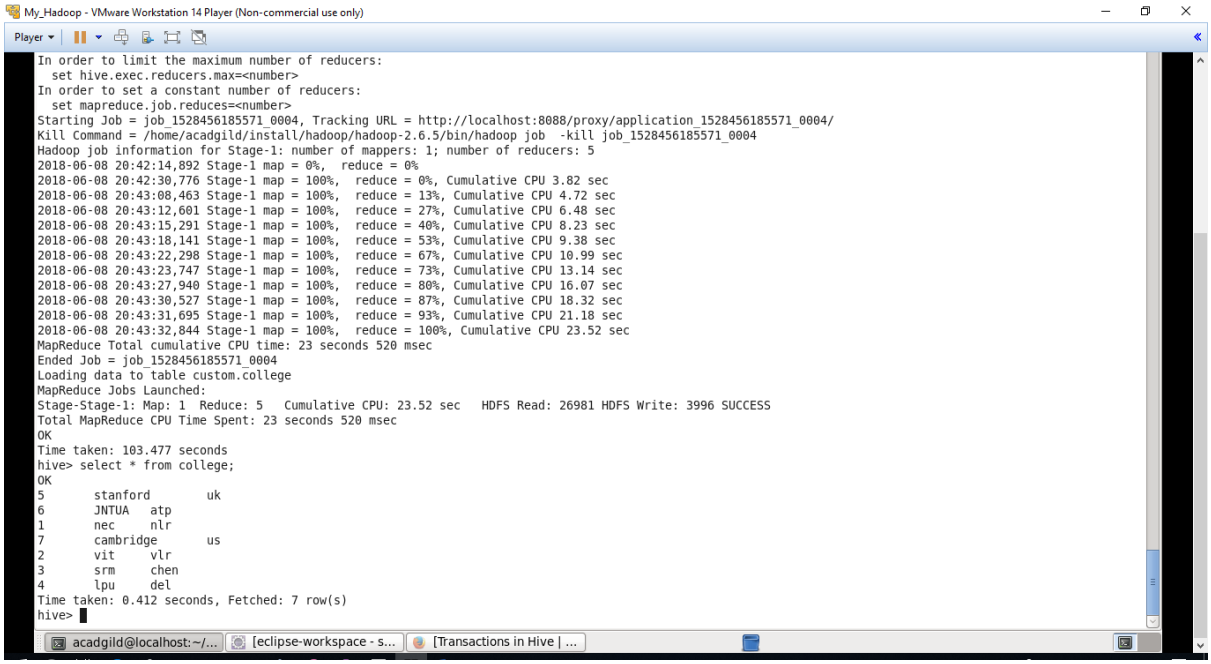
**Transctional  like** update insert and delete at row level

For transctions we should set the properties,

```
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 = a positive number on at least one ins
```

Below details are the create table college and inserting data into the row level.we are not using files to load data into the table.

```
hive> use custom;
OK
Time taken: 0.044 seconds
hive> CREATE TABLE college(clg_id int,clg_name string,clg_loc string) clustered by (clg_id) into 5 buckets stored as orc TBLPROPERTIES('transactional'='true'
);
OK
Time taken: 0.305 seconds
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'):
```




**Update**

Update Is not possible on buketted column

I will update the normal column using id

```
hive> UPDATE college set clg_name = 'bvrit' where clg_id = 7;
```

Table data:

```
hive> select * from college;
OK
5       stanford        uk
5       stanford        uk
6       JNTUA   atp
1       nec     nlr
6       JNTUA   atp
1       nec     nlr
7       bvrit   us
2       vit     vlr
7       bvrit   us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
```

After update statement execute and result.

```
hive> select * from college;
OK
5       stanford        uk
5       stanford        uk
5       JNTUA   atp
1       nec     nlr
5       JNTUA   atp
1       nec     nlr
7       bvrit   us
2       vit     vlr
7       bvrit   us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
```

**Delete**

We can delete data from row level

```
hive> select * from college;
OK
5       stanford        uk
5       stanford        uk
6       JNTUA   atp
1       nec     nlr
6       JNTUA   atp
1       nec     nlr
7       bvrit   us
2       vit     vlr
7       bvrit   us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
Time taken: 0.355 seconds, Fetched: 14 row(s)
hive> delete from college where clg_id=5;
```

After delete data from table id =5

```
hive> select * from college;
OK
6       JNTUA   atp
1       nec     nlr
6       JNTUA   atp
1       nec     nlr
7       bvrit   us
2       vit     vlr
7       bvrit   us
2       vit     vlr
3       srm     chen
3       srm     chen
4       lpu     del
4       lpu     del
Time taken: 0.392 seconds, Fetched: 12 row(s)
```