Assignment 9 Advanced Hive

I have created the table like below in custom database.

CREATE TABLE olympic_data(
Athlete STRING,

Age INT,

Country STRING,

Year BIGINT,

Closing_Date STRING,

Sport STRING,

Gold_Medals INT,

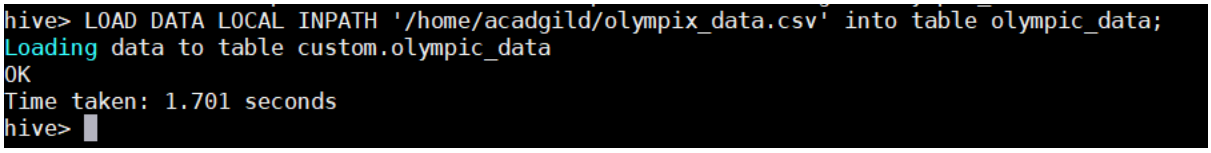
Silver_Medals INT,

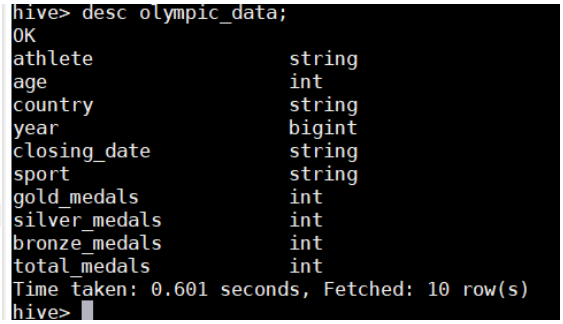
Bronze_Medals INT,

Total_Medals INT

)
row format delimited fields terminated by '\t';

LOAD DATA INPATH '/home/acadgild/olympix_data' into table olympic_data;





Task 1

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

Select sum (Total_Medals), Country from olympic_data where Sport = 'Swimming' group by Country;

```
hive> Select sum (Total_Medals), Country from olympic_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, te
sing Hive 1.X releases.
```

```
OK
1       Argentina
163     Australia
3       Austria
2       Belarus
8       Brazil
5       Canada
35      China
2       Costa Rica
1       Croatia
1       Denmark
39      France
32      Germany
11      Great Britain
9       Hungary
16      Italy
43      Japan
1       Lithuania
46      Netherlands
2       Norway
3       Poland
6       Romania
20      Russia
1       Serbia
2       Slovakia
1       Slovenia
11      South Africa
4       South Korea
3       Spain
9       Sweden
1       Trinidad and Tobago
3       Tunisia
7       Ukraine
267     United States
7       Zimbabwe
Time taken: 36.857 seconds, Fetched: 34 row(s)
```

2. **Write a Hive program to find the number of medals that India won year wise.**
   Select sum(Total_Medals),year from olympic_data where Country = 'India' group by year;

```
hive> Select sum(Total_Medals),Year from olympic_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 u
sing Hive 1.X releases.
Query ID = acadgild_20181208054756_9b53ba71-5d24-413c-8bba-6c97bcbe2332
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_1544146716043_0009, Tracking URL = http://localhost:8088/proxy/application_1544146716043_0009/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1544146716043_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-12-08 05:48:07,121 Stage-1 map = 0%,  reduce = 0%
2018-12-08 05:48:18,130 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.39 sec
2018-12-08 05:48:29,129 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.89 sec
MapReduce Total cumulative CPU time: 5 seconds 890 msec
Ended Job = job_1544146716043_0009
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.89 sec   HDFS Read: 529210 HDFS Write: 163 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 890 msec
OK
1       2000
1       2004
3       2008
6       2012
Time taken: 34.934 seconds, Fetched: 4 row(s)
hive>
```

3. **Write a Hive Program to find the total number of medals each country won.**
   Select sum (Total_Medals), Country from olympic_data group by Country;

```
hive> Select sum (Total_Medals), Country from olympic_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)
sing Hive 1.X releases.
Query ID = acadgild_20181208054946_9c24c20d-f8e6-436f-9243-89997ffeb84e
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_1544146716043_0010, Tracking URL = http://localhost:8088/proxy/application_1544146716043_0010/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1544146716043_0010
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-12-08 05:49:56,958 Stage-1 map = 0%,  reduce = 0%
2018-12-08 05:50:06,898 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.87 sec
2018-12-08 05:50:15,803 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.99 sec
MapReduce Total cumulative CPU time: 3 seconds 990 msec
Ended Job = job_1544146716043_0010
```

2	Afghanistan
8	Algeria
141	Argentina
10	Armenia
609	Australia
91	Austria
25	Azerbaijan
24	Bahamas
1	Bahrain
1	Barbados
97	Belarus
18	Belgium
1	Botswana
221	Brazil
41	Bulgaria
20	Cameroon
370	Canada
22	Chile
530	China
20	Chinese Taipei
13	Colombia
2	Costa Rica
81	Croatia
188	Cuba
1	Cyprus
81	Czech Republic
89	Denmark
5	Dominican Republic
1	Ecuador
8	Egypt
1	Eritrea
18	Estonia
29	Ethiopia
118	Finland
318	France
1	Gabon
23	Georgia
629	Germany
322	Great Britain
59	Greece
1	Grenada
1	Guatemala
3	Hong Kong
145	Hungary
15	Iceland
11	India
22	Indonesia
24	Iran
9	Ireland
4	Israel
331	Italy
80	Jamaica
282	Japan
42	Kazakhstan
39	Kenya
2	Kuwait
3	Kyrgyzstan
17	Latvia
30	Lithuania
1	Macedonia
3	Malaysia
1	Mauritius
38	Mexico
5	Moldova
10	Mongolia
14	Montenegro
11	Morocco
1	Mozambique
318	Netherlands
52	New Zealand
39	Nigeria
21	North Korea
192	Norway
1	Panama
17	Paraguay
80	Poland
9	Portugal
2	Puerto Rico
3	Qatar
123	Romania
768	Russia
6	Saudi Arabia
31	Serbia
38	Serbia and Montenegro
7	Singapore

```
35      Slovakia
25      Slovenia
25      South Africa
308     South Korea
205     Spain
1       Sri Lanka
1       Sudan
181     Sweden
93      Switzerland
1       Syria
3       Tajikistan
18      Thailand
1       Togo
19      Trinidad and Tobago
4       Tunisia
28      Turkey
1       Uganda
143     Ukraine
1       United Arab Emirates
1312    United States
1       Uruguay
19      Uzbekistan
4       Venezuela
2       Vietnam
7       Zimbabwe
```

4. **Write a Hive program to find the number of gold medals each country won.**
   Select sum (Gold_Medals), Country from olympic_data group by Country;

```
hive> Select sum (Gold_Medals), Country from olympic_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 eng
sing Hive 1.X releases.
Query ID = acadgild_20181208055345_79016fd5-881d-45cc-9ba4-10cdf74e8151
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:
```

0 Afghanistan
2 Algeria
49 Argentina
0 Armenia
163 Australia
36 Austria
6 Azerbaijan
11 Bahamas
0 Bahrain
0 Barbados
17 Belarus
2 Belgium
0 Botswana
46 Brazil
8 Bulgaria
20 Cameroon
168 Canada
3 Chile
234 China
2 Chinese Taipei
2 Colombia
0 Costa Rica
35 Croatia
57 Cuba
0 Cyprus
14 Czech Republic
46 Denmark

3	Dominican Republic
0	Ecuador
1	Egypt
0	Eritrea
6	Estonia
13	Ethiopia
11	Finland
108	France
0	Gabon
6	Georgia
223	Germany
124	Great Britain
12	Greece
1	Grenada
0	Guatemala
0	Hong Kong
77	Hungary
0	Iceland
1	India
5	Indonesia
10	Iran
1	Ireland
1	Israel
86	Italy
24	Jamaica
57	Japan
13	Kazakhstan
11	Kenya
0	Kuwait
0	Kyrgyzstan
3	Latvia
5	Lithuania
0	Macedonia
0	Malaysia
0	Mauritius
19	Mexico
0	Moldova
2	Mongolia
0	Montenegro
2	Morocco
1	Mozambique
101	Netherlands
18	New Zealand
6	Nigeria
6	North Korea
97	Norway

1 Panama
0 Paraguay
20 Poland
1 Portugal
0 Puerto Rico
0 Qatar
57 Romania
234 Russia
0 Saudi Arabia
1 Serbia
11 Serbia and Montenegro
0 Singapore
10 Slovakia
5 Slovenia
10 South Africa
110 South Korea
19 Spain
0 Sri Lanka
0 Sudan
57 Sweden
21 Switzerland
0 Syria
0 Tajikistan
6 Thailand
0 Togo
1 Trinidad and Tobago
2 Tunisia
9 Turkey
1 Uganda
31 Ukraine
1 United Arab Emirates
552 United States
0 Uruguay
5 Uzbekistan
1 Venezuela
0 Vietnam
2 Zimbabwe
Time taken: 33.701 seconds, Fetched: 110 row(s)

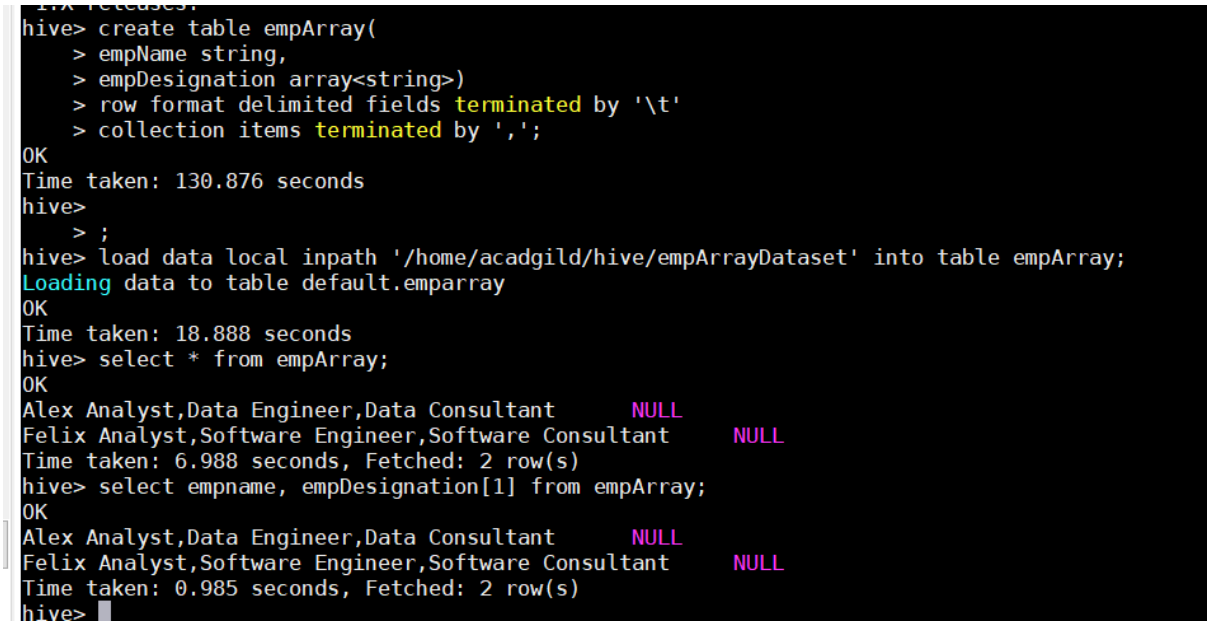
Task 2

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.

→*Created Data set like below*
[acadgild@localhost hive]$ *cat empArrayDataset*
Alex Analyst,Data Engineer,Data Consultant
Felix Analyst,Software Engineer,Software Consultant

➔ *Create table*
Create table
create table empArray(
empName string,
empDesignation array<string>)
row format delimited fields terminated by '\t'
collection items terminated by ',';

load the data file
load data local inpath '/home/acadgild/hive/empArrayDataset' into table empArray;



Create jar file for the below code

```
[acadgild@localhost hive]$ cat JoinArray.java
import java.util.ArrayList;
import org.apache.hadoop.hive.ql.exec.UDF;
public class JoinArray 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();}}
[acadgild@localhost hive]$
```

Create UDF function like below.

add jar /home/acadgild/Desktop/empAr.jar;

create temporary function separate as 'JoinArray';

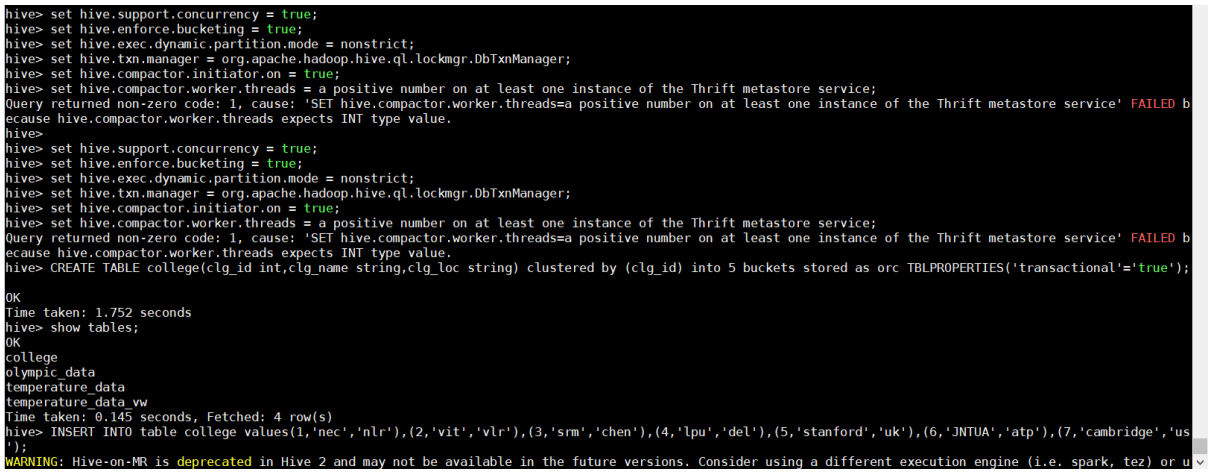
select separa(empName,empDesignation) from empArray;

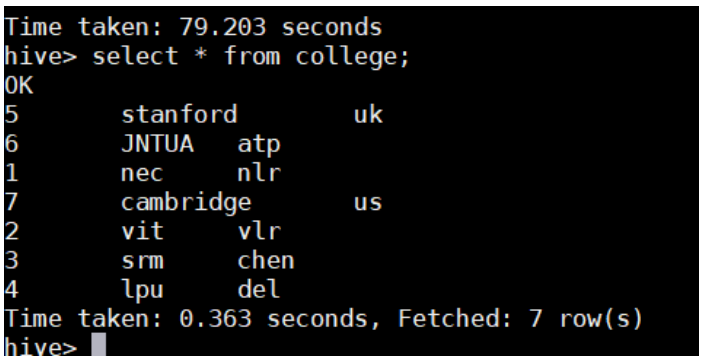
Task 3

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.

I have set the properties and created a table college and the columns present in the table are ‘clg_id, clg_name, clg_loc’. We are bucketing the table by ‘clg_id’ and the table format is ‘orc’, also we are enabling the transactions in the table by specifying it inside the TBLPROPERTIES as ‘transactional’=’true’

→*We have inserted few data.*





→*Inserted the data again*

```
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: 0.414 seconds, Fetched: 14 row(s)
hive>
```

→*We cannot update the bucketing row,*

```
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>
```


→*We can update on non bucketing rows,*
UPDATE college set clg_id = 8 where clg_id = 7;

```
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: 1.533 seconds, Fetched: 14 row(s)
hive>
```

→

We can perform delete on non bucketing rows

delete from college where clg_id=5;