Assignment Submission- Session 9

DATE SET DESCRIPTION
The data set consists of the following fields.
Athlete: This field consists of the athlete name
Age: This field consists of athlete ages
Country: This fields consists of the country names which participated in Olympics
Year: This field consists of the year
Closing Date: This field consists of the closing date of ceremony
Sport: Consists of the sports name
Gold Medals: No. of Gold medals
Silver Medals: No. of Silver medals
Bronze Medals: No. of Bronze medals
Total Medals: Consists of total no. of medals

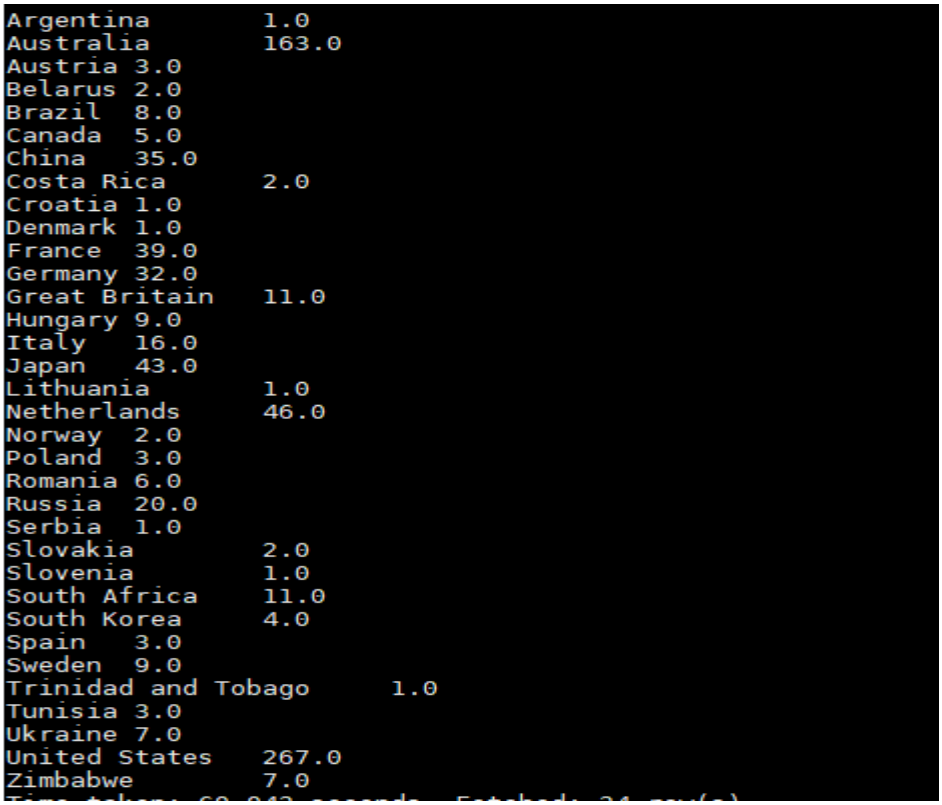
Task 1
Solution: First of all, creating a hive table and inserting the data into it.

```
CREATE TABLE OLYMPICS (name STRING, age INT, country STRING, yearParticipated
STRING, closingDate STRING, sports STRING, goldMedals INT, silverMedals INT,
bronzeMedals INT, totalMedals INT)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
   "separatorChar" = "\t"
)
STORED AS TEXTFILE;
```

LOADING DATA INTO TABLE

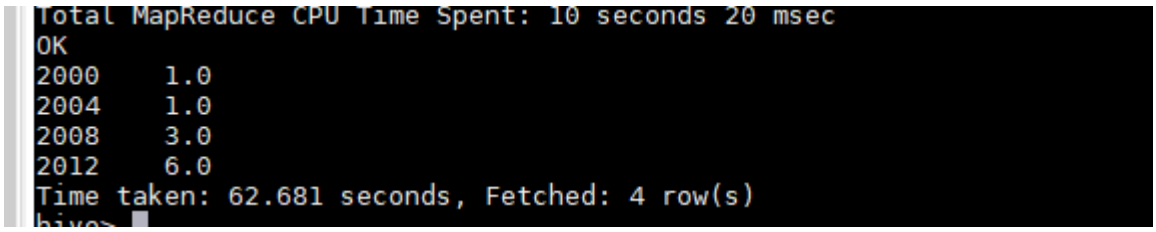
```
LOAD DATA LOCAL INPATH
'/home/acadgild/akshat/HIVE_SESSIONS/olympix_data.csv' INTO TABLE olympics;
```

1. Write a Hive program to find the number of medals won by each country in swimming.
   Hive query: SELECT country, sum(totalMedals) FROM olympics WHERE sports='Swimming' GROUP BY country;



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

Hive Query:
SELECT yearParticipated, sum(totalMedals) FROM olympics WHERE country='India' GROUP BY yearParticipated;

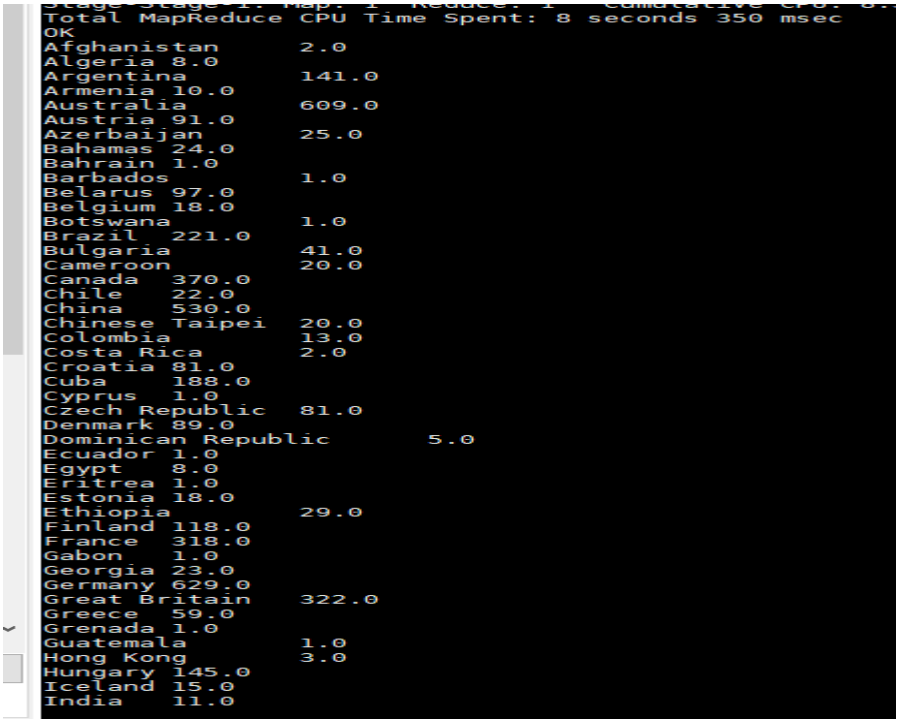


3. Write a Hive Program to find the total number of medals each country won.

Hive Query:

SELECT country, sum(totalMedals) FROM olympics GROUP BY country;

Output not completely captured in screenshot:

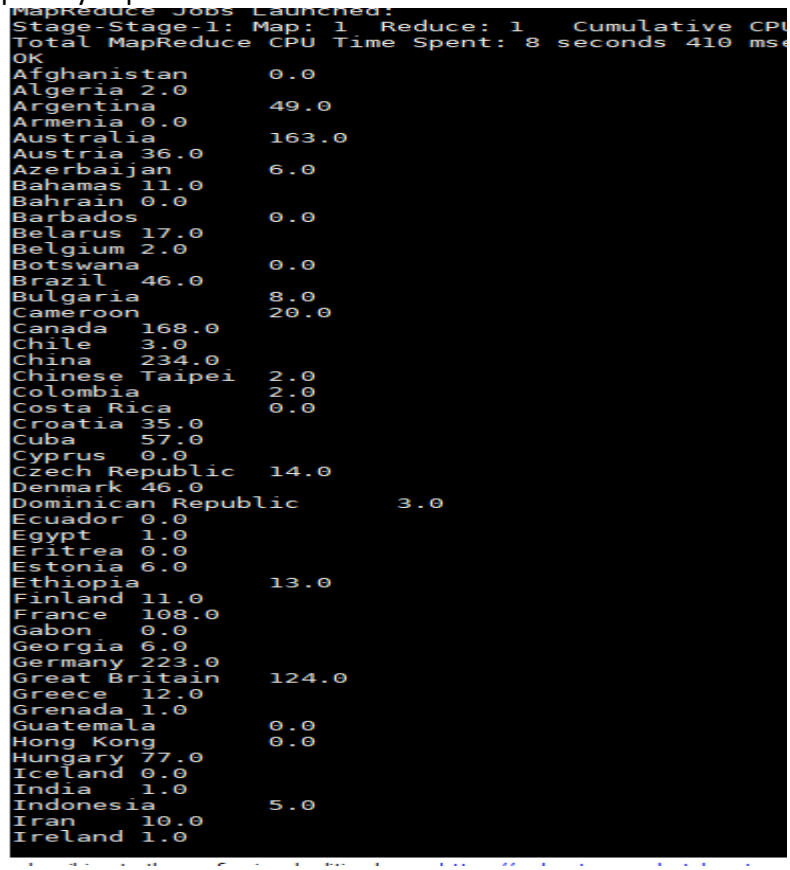


4. Write a Hive program to find the number of gold medals each country won.

Hive Query:

SELECT country, sum(goldMedals) FROM olympics GROUP BY country;

Output not completely captured in screenshot:



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.

Solution:

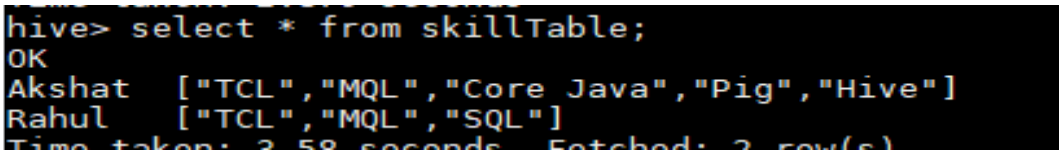
- Created a table in hive:

create table skillTable(name string, skill array<string>) row format delimited fields terminated by '\t' collection items terminated by ',';

- Loaded data into table:

load data local inpath '/home/acadgild/akshat/HIVE_SESSIONS/skillData.txt' into table skillTable;

- Select all data from table:



- Created a jar file for udf.

UDF program code:

```
import java.util.ArrayList;
import org.apache.hadoop.hive.ql.exec.UDF;
public class ConcatArrayElements extends UDF{
        public String evaluate(String del, ArrayList<String> arrayelements) {
                if(arrayelements==null) {
                        return null;

                }
                StringBuffer sbuf = new StringBuffer();
                sbuf.append(arrayelements.get(0));
                for (int i = 1; i < arrayelements.size(); i++) {
                        sbuf.append(del);
                        sbuf.append(arrayelements.get(i));
                }
                return sbuf.toString();
        }
}
```

- Added jar to hive.

```
hive> add jar /home/acadgild/akshat/HIVE_SESSIONS/ConcatArrElements.jar;
Added [/home/acadgild/akshat/HIVE_SESSIONS/ConcatArrElements.jar] to class path
Added resources: [/home/acadgild/akshat/HIVE_SESSIONS/ConcatArrElements.jar]
```

- Created temporary function.

Jar hive -exec couldn't be downloaded so couldn't proceed further for running the function.

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.

Solution:

Turning ON features to support transactions:

```
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;
```

Creating table using ORC format

CREATE TABLE company(cmp_id int,cmp_name string,cmp_loc string) clustered by (cmp_id) into 3 buckets stored as orc TBLPROPERTIES('transactional'='true');

```
hive> CREATE TABLE company(cmp_id int,cmp_name string,cmp_loc string) clustered by (cmp_id) into 3 buckets stored as orc TBLPROPERTIES('transactional'='true');
OK
Time taken: 2.526 seconds
```

Inserting data into table created

INSERT INTO table company values
(1,'tata','india'),(2,'facebook','US'),(3,'Sopra','France'),(4,'Barclays','England');

```
hive> INSERT INTO table company values(1,'tata','india'),(2,'facebook','US'),(3,'Sopra','France'),(4,'Barclays','England');
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 using Hive 1.X releases.
Query ID = acadgild_20180925214038_0b440751-33ec-44da-a506-d24bca4722fa
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 3
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
```

Data after inserting in table:

```
OK
3       Sopra   France
4       Barclays        England
1       tata    india
2       facebook        US
```

Updating record: (Update doesn't work on bucketed column)

UPDATE company set cmp_name = 'wipro' where cmp_id = 1;

```
hive> UPDATE company set cmp_name = 'wipro' where cmp_id = 1;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in
Query ID = acadgild_20180925214813_bd90eb89-8ad0-410a-8e62-d8e82cf046d0
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 3
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:
```

Deleting record:

delete from company where cmp_id=3;

```
hive> delete from company where cmp_id=3;
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. s
Query ID = acadgild_20180925215135_f1b7b390-5a13-43ec-af3b-4601112bb378
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 3
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>
```

Data post update and deletion:

```
hive> SELECT * FROM company;
OK
4       Barclays        England
1       wipro   india
2       facebook        US
Time taken: 0.524 seconds, Fetched: 3 row(s)
hive>
```