SPARK SQL ASSIGNMENT 19.1

PROBLEM STATEMENT:

==========

Using spark-sql, Find:

19.1.1) What are the total number of gold medal winners every year

19.1.2) How many silver medals have been won by USA in each sport

==========

//Schema Definition for input test file Sports_data.txt

Sports_data.txt ->

firstname:String,

lastname:String,

sports:String,

medal_type:String,

age:Integer,

year:Integer,

country:String

//putting the sports_data.txt to HDFS from local filesystem

[acadgild@localhost]$ cd Downloads

[acadgild@localhost]$ ls

[acadgild@localhost Downloads]$ hadoop fs -put Sports_data.txt /user/acadgild/spark/

//Browsing the data of file put at HDFS

[acadgild@localhost Downloads]$ hadoop fs -cat /user/acadgild/spark/Sports_data.txt

//initiating the spark session

```
[acadgild@localhost Downloads]$ spark-shell
```

```
//IMPORTING THE SPARK SQL PACKAGES

import org.apache.spark.sql._

import SQLContext.implicits._

//conversion of text file into RDD with the help of SPARK CONTEXT object

val sportsRDD =
sc.textFile("hdfs://localhost:9000//user/acadgild/spark/Sports_data.txt")

//putting the first line of file into RDD which is header.

val header = sportsRDD.first()

//removing the header from RDD for data manipulation

val sdRDD  = sportsRDD.filter(record => (record != header))

//printing RDD which returns an Array[String]

sdRDD.foreach(println)

//class definition for sportsdata class and defining the respective schema

case class

sportsdata(firstname:String,

lastname:String,

sports:String,

medal_type:String,

age:Integer,

year:Integer,

country:String)

//mapping the record into word delimited by comma , defining the columns
and conversion to dataframe

val rec = sdRDD.map(x=> x.split(","))

.map(x=>sportsdata(x(0),x(1),x(2),x(3),x(4).toInt,x(5).toInt,x(6))).toDF
```

//registering temporary table sports for querying data

rec.registerTempTable("sports")

//querying the entire table via sqlContext object and displaying the same

sqlContext.sql("SELECT * FROM sports LIMIT 1").show()

==========

19.1.1) What are the total number of gold medal winners every year

Input Command:

```
sqlContext.sql("SELECT year, COUNT(medal_type) AS total_gold_medals
FROM sports where medal_type = 'gold' GROUP BY year").show()
```

Output Screenshots:

```
scala> sqlContext.sql("SELECT year, COUNT(medal_type) AS total_gold_medals FROM sports where medal_t
ype = 'gold' GROUP BY year").show()
+----+-----------------+
|year|total_gold_medals|
+----+-----------------+
|2014|                3|
|2015|                3|
|2016|                2|
|2017|                1|
+----+-----------------+
```

==========

19.1.2) How many silver medals have been won by USA in each sport

Input Command:

```
val resultdf = sqlContext.sql("SELECT sports,count(*) as
total_silver_medals_won_by_USA FROM sports where medal_type = 'silver'
and country = 'USA' GROUP BY sports")

resultdf.show()
```

Output Screenshots:

```
scala> val resultdf = sqlContext.sql("SELECT sports,count(*) as total_silver_medals_won_by_USA FROM
sports where medal_type = 'silver' and country = 'USA' GROUP BY sports")
resultdf: org.apache.spark.sql.DataFrame = [sports: string, total_silver_medals_won_by_USA: bigint]

scala> resultdf.show()
+--------+----------------------------+
|  sports|total_silver_medals_won_by_USA|
+--------+----------------------------+
|swimming|                           3|
+--------+----------------------------+
```

_

==========