Assignment 19.2

Initial Steps:

* Load the Dataset Sport\_data.txt into RDD sports\_data\_with\_header
* Get the header from the sports\_data\_with\_header
* Remove the header from sports\_data\_with\_header by filtering and create RDD sports\_data
* Create class SportData with schema firstname, lastname,sports,medal\_type,age,year,country
* Splitting the fields of sports\_data records base on field separator comma (,) Populate SportsData class and convert to Data Frame sports\_data\_df

Code is as below:

val sports\_data\_with\_header = sc.textFile("/home/acadgild/assignment\_19.1/Sports\_data.txt")

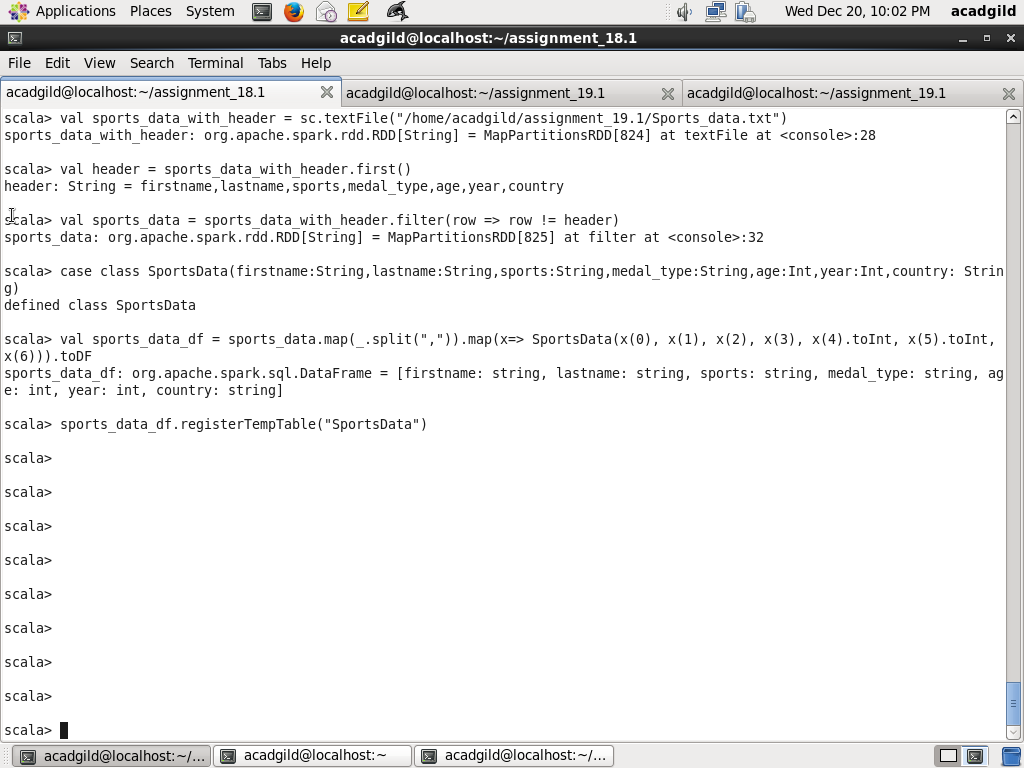
val header = sports\_data\_with\_header.first()

val sports\_data = sports\_data\_with\_header.filter(row => row != header)

case class SportsData(firstname:String,lastname:String,sports:String,medal\_type:String,age:Int,year:Int,country: String)

val sports\_data\_df = sports\_data.map(\_.split(",")).map(x=> SportsData(x(0), x(1), x(2), x(3), x(4).toInt, x(5).toInt, x(6))).toDF

Screenshot is as below:



Task1: Create a dataframe UDF to have firstname,lastname column into Mr.first\_two\_letters\_of\_first name<space>lastname

Step1: Create a UDF fullName which returns firstname,lastname column into Mr.first\_two\_letters\_of\_first name<space>lastname

Code is as below:

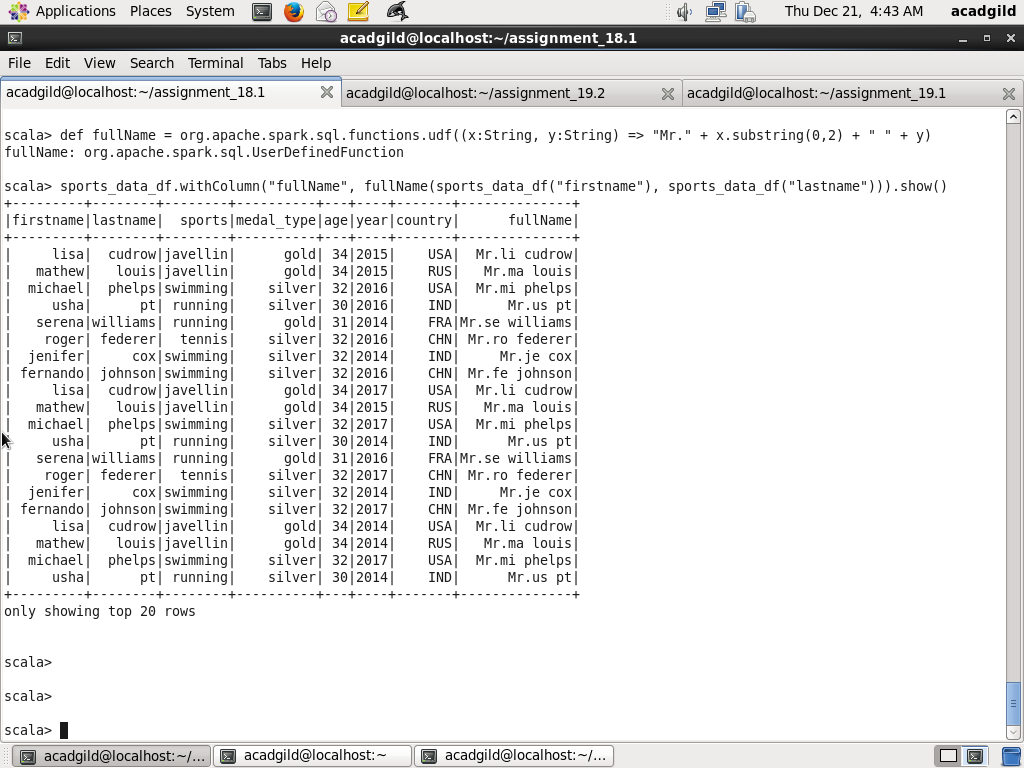
def fullName = org.apache.spark.sql.functions.udf((x:String, y:String) => "Mr." + x.substring(0,2) + " " + y)

Step2: Add a column fullName to sports\_data\_df and call the UDF fullName as created in step1 and taking arguments value of firstname and lastname and show the result

Code is as below:

sports\_data\_df.withColumn("fullName", fullName(sports\_data\_df("firstname"), sports\_data\_df("lastname"))).show()

Screenshot is as below:



Task2: Add a new column called ranking with UDF on database

Step1: Create a UDF findRanking which take two parameters x represenring medal\_type and y representing age. Based on value of medal\_type and age, following values are returned “pro”, “amateur”, “expert”, “”

Code is as below:

def findRanking = org.apache.spark.sql.functions.udf((x:String, y:Int) => {

if (x == "gold" && y >=32) "pro"

else if (x == "gold" && y <=31) "amateur"

else if (x == "silver" && y >= 32) "expert"

else if (x == "silver" && y <= 31) "rookie"

else ""

})

Step2: Add a column ranking to sports\_data\_df and call the UDF findRanking as created in step1 taking arguments, value of medal\_type, age and show the result

Code is as below:

sports\_data\_df.withColumn("ranking", findRanking(sports\_data\_df("medal\_type"), sports\_data\_df("age"))).show()

Screenshot is as below:

