**SPARK ASSIGNMENT 18.3**

**===============================================================**

**S18\_Dataset\_User\_Details.txt -> userid,name,age**

**S18\_Dataset\_Transport.txt -> travel\_mode, cost\_per\_unit**

**S18\_Dataset\_Holidays.txt -> user\_id,source, destination , travel\_mode, distance , year\_of\_travel**

**Problem Statement**

**18.3.1) Considering age groups of < 20 , 20-35, 35 > ,Which age group spends the most amount of money travelling.**

**18.3.2) What is the amount spent by each age-group, every year in travelling?**

**Input Commands:**

**//putting the input files to HDFS**

**[acadgild@localhost ]$cd Downloads**

**[acadgild@localhost Downloads]$ hadoop fs -put S18\_Dataset\_User\_details.txt /user/acadgild/spark/**

**[acadgild@localhost Downloads]$ hadoop fs -put S18\_Dataset\_Transport.txt /user/acadgild/spark/**

**[acadgild@localhost Downloads]$ hadoop fs -put S18\_Dataset\_Holidays.txt /user/acadgild/spark/**

**//Initiating the spark-shell to run spark**

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

**//importing the spark packages**

**import org.apache.spark.sql.Row;**

**import org.apache.spark.sql.types.{StructType,StructField,StringType,NumericType,IntegerType};**

**import org.apache.spark.sql.\_**

**import sqlContext.implicits.\_**

**//Loading the input files to respective RDDs**

**val userdetailsRDD = sc.textFile("hdfs://localhost:9000//user/acadgild/spark/S18\_Dataset\_User\_details.txt")**

**val transportRDD = sc.textFile("hdfs://localhost:9000//user/acadgild/spark/S18\_Dataset\_Transport.txt")**

**val holidaysRDD = sc.textFile("hdfs://localhost:9000//user/acadgild/spark/S18\_Dataset\_Holidays.txt")**

**//Defining the schemas for each of the 3 files mentioned above respectively**

**val schemaStringu = "userid:integer,name:string,age:integer"**

**val schemaStringt = "travel\_mode:string,cost\_per\_unit:integer"**

**val schemaStringh = "userid:integer,source:string,destination:string,travel\_mode:string,distance:integer,year\_of\_travel:integer"**

**//Defining the Structtype and StructField for each DB2 Schema**

**val schemau = StructType(schemaStringu.split(",").map(fieldInfo => StructField(fieldInfo.split(":")(0),**

**if (fieldInfo.split(":")(1).equals("integer")) IntegerType else StringType,true)))**

**val schemat = StructType(schemaStringt.split(",").map(fieldInfo => StructField(fieldInfo.split(":")(0), if (fieldInfo.split(":")(1).equals("string")) StringType else IntegerType, true)))**

**val schemah = StructType(schemaStringh.split(",").map(fieldInfo => StructField(fieldInfo.split(":")(0), if (fieldInfo.split(":")(1).equals("string")) StringType else IntegerType, true)))**

**//Mapping the data present in TEXT files at HDFS**

**val RDDu = userdetailsRDD.map(\_.split(",")).map(r => Row(r(0).toInt, r(1), r(2).toInt ))**

**val RDDt = transportRDD.map(\_.split(",")).map(r => Row(r(0), r(1).toInt ))**

**val RDDh = holidaysRDD.map(\_.split(",")).map(r => Row(r(0).toInt, r(1) , r(2), r(3) , r(4).toInt, r(5).toInt ))**

**//Defining the SQLCONTEXT object with the help of Spark Context object**

**val sqlContext = new org.apache.spark.sql.SQLContext(sc);**

**//Creating the Dataframe with the help of schema and data in text files**

**val uDF =sqlContext.createDataFrame(RDDu, schemau)**

**val tDF =sqlContext.createDataFrame(RDDt, schemat)**

**val hDF =sqlContext.createDataFrame(RDDh, schemah)**

**//Defining the temporary tables with the newly created dataframes**

**uDF.registerTempTable("userdetails")**

**tDF.registerTempTable("transport")**

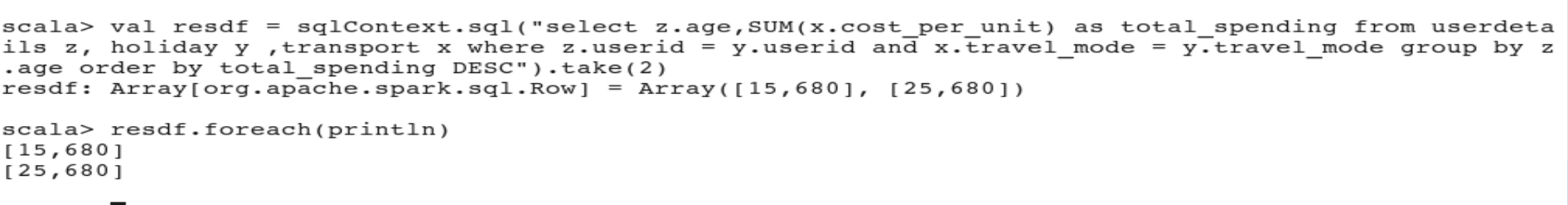
**hDF.registerTempTable("holiday")**

**1) Considering age groups of < 20 , 20-35, 35 > ,Which age group spends the most amount of money travelling.**

**val resdf = sqlContext.sql("select z.age,SUM(x.cost\_per\_unit) as total\_spending from userdetails z, holiday y ,transport x where z.userid = y.userid and x.travel\_mode = y.travel\_mode group by z.age order by total\_spending DESC").take(2)**

**resdf.show()**

**Output:**

****

**2) What is the amount spent by each age-group, every year in travelling?**

**val res1df = sqlContext.sql("select y.year\_of\_travel , SUM(x.cost\_per\_unit) as total\_spending\_of\_age\_below\_20 from userdetails z, holiday y ,transport x where z.userid = y.userid and x.travel\_mode = y.travel\_mode and z.age < 20 group by y.year\_of\_travel order by y.year\_of\_travel DESC")**

**val res2df = sqlContext.sql("select y.year\_of\_travel , SUM(x.cost\_per\_unit) as total\_spending\_of\_age\_bracket\_from\_20\_to\_35 from userdetails z, holiday y ,transport x where z.userid = y.userid and x.travel\_mode = y.travel\_mode and z.age >= 20 and z.age<=35 group by y.year\_of\_travel order by y.year\_of\_travel DESC")**

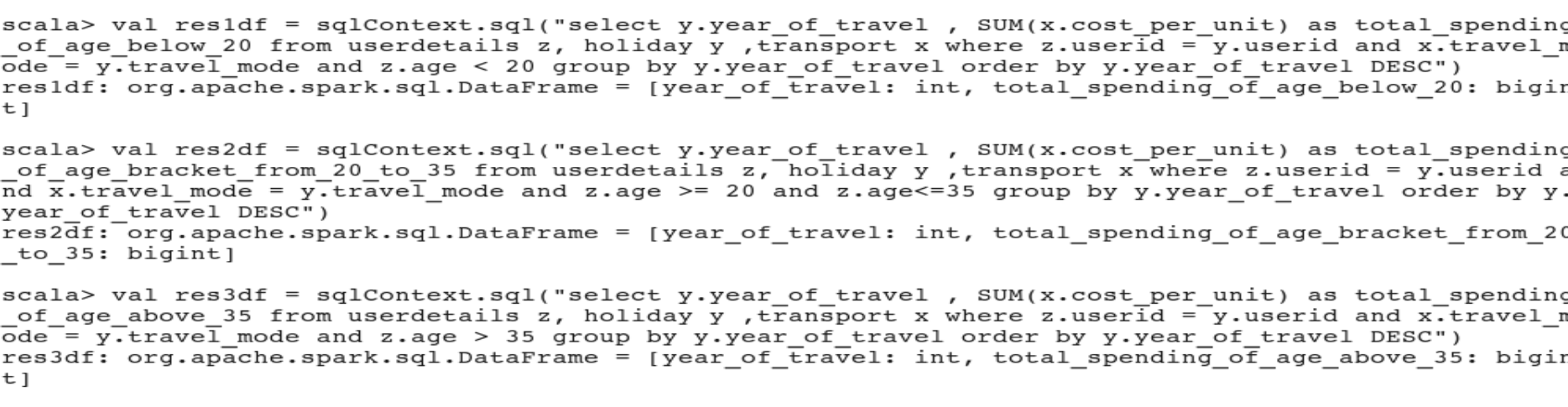
**val res3df = sqlContext.sql("select y.year\_of\_travel , SUM(x.cost\_per\_unit) as total\_spending\_of\_age\_above\_35 from userdetails z, holiday y ,transport x where z.userid = y.userid and x.travel\_mode = y.travel\_mode and z.age > 35 group by y.year\_of\_travel order by y.year\_of\_travel DESC")**

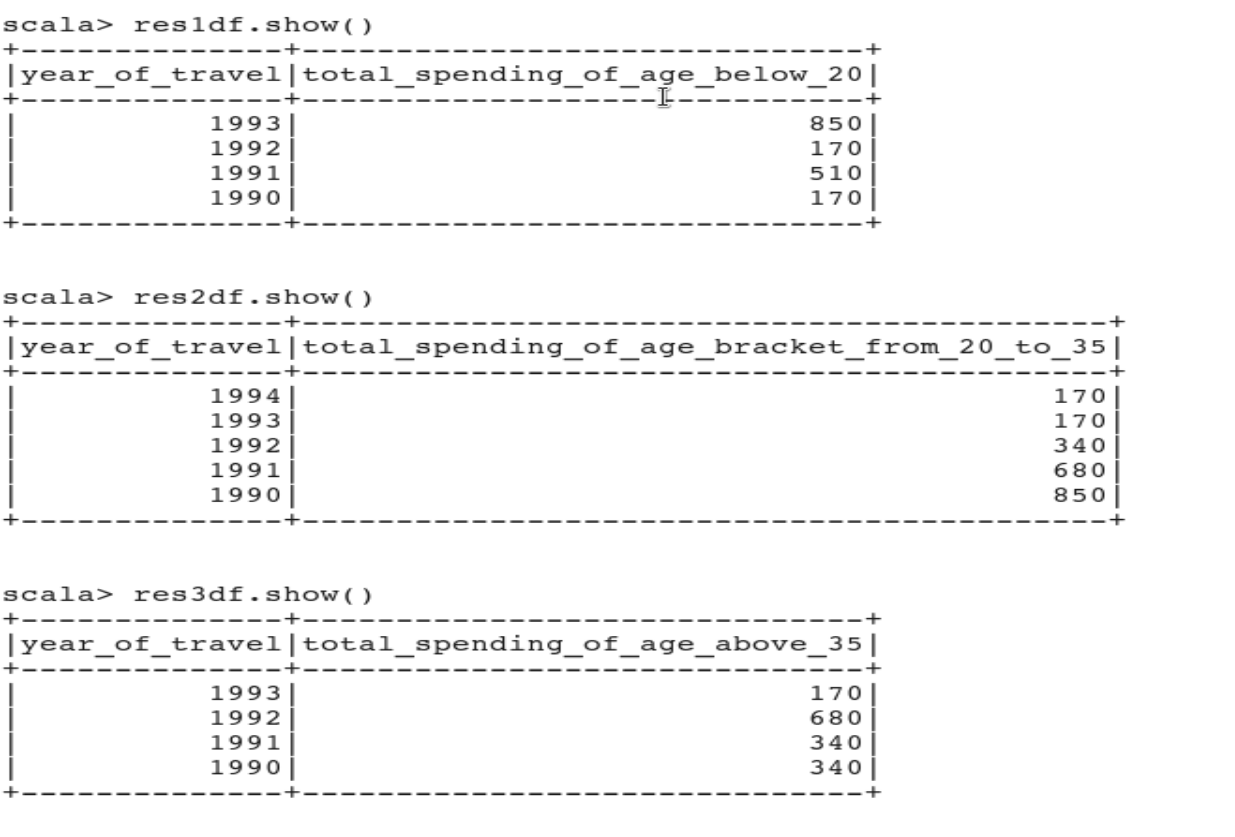
**res1df.show()**

**res2df.show()**

**res3df.show()**

**Output:**

****

****