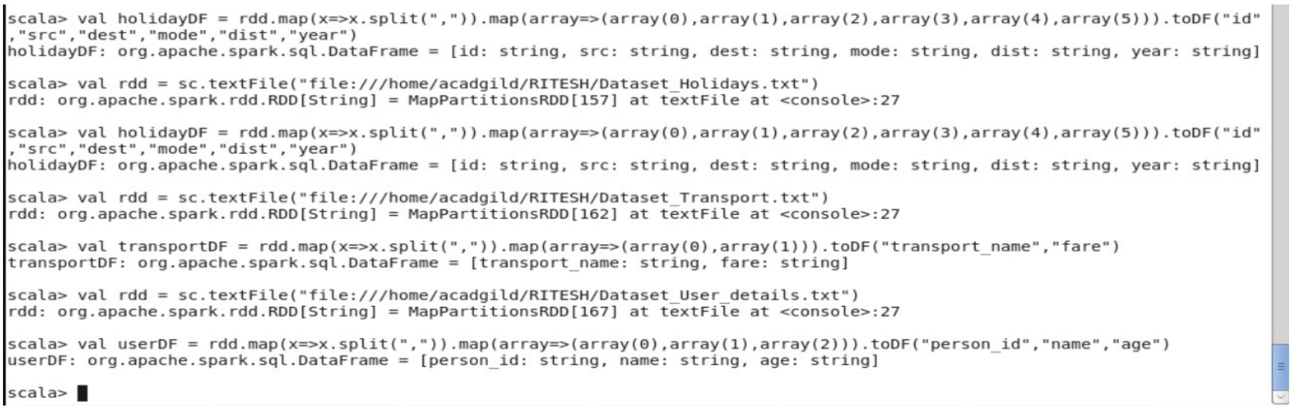
Bigdata Assignment 6.7

1) Considering age groups of < 20 , 20-35, 35 > ,Which age group spends the most
amount of money travelling.
2) What is the amount spent by each age-group, every year in travelling?

Solution -

- Loaded the datasets into dataframes using RDD.

```
val rdd =
sc.textFile(“file:///home/acadgild/RITESH/Dataset_Holidays.txt”)

val holidayDF =
rdd.map(x=>x.split(“,”)).map(array=>(array(0),array(1),array(2),array(3),array(4),array(5))).toDF(“id”,”src”,”dest”,”mode”,”dist”,”year”)

val rdd =
sc.textFile(“file:///home/acadgild/RITESH/Dataset_Transport.txt”)

val transportDF =
rdd.map(x=>x.split(“,”)).map(array=>(array(0),array(1))).toDF(“transport_name”,”transport_id”)

val rdd =
sc.textFile(“file:///home/acadgild/RITESH/Dataset_User_details.txt”)

val userDF =
rdd.map(x=>x.split(“,”)).map(array=>(array(0),array(1),array(2)).toDF(”person_id”,”name”,”age”)
```



```
scala> val holidayDF = rdd.map(x=>x.split(",")).map(array=>(array(0),array(1),array(2),array(3),array(4),array(5))).toDF("id"
,"src","dest","mode","dist","year")
holidayDF: org.apache.spark.sql.DataFrame = [id: string, src: string, dest: string, mode: string, dist: string, year: string]

scala> val rdd = sc.textFile("file:///home/acadgild/RITESH/Dataset_Holidays.txt")
rdd: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[157] at textFile at <console>:27

scala> val holidayDF = rdd.map(x=>x.split(",")).map(array=>(array(0),array(1),array(2),array(3),array(4),array(5))).toDF("id"
,"src","dest","mode","dist","year")
holidayDF: org.apache.spark.sql.DataFrame = [id: string, src: string, dest: string, mode: string, dist: string, year: string]

scala> val rdd = sc.textFile("file:///home/acadgild/RITESH/Dataset_Transport.txt")
rdd: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[162] at textFile at <console>:27

scala> val transportDF = rdd.map(x=>x.split(",")).map(array=>(array(0),array(1))).toDF("transport_name","fare")
transportDF: org.apache.spark.sql.DataFrame = [transport_name: string, fare: string]

scala> val rdd = sc.textFile("file:///home/acadgild/RITESH/Dataset_User_details.txt")
rdd: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[167] at textFile at <console>:27

scala> val userDF = rdd.map(x=>x.split(",")).map(array=>(array(0),array(1),array(2))).toDF("person_id","name","age")
userDF: org.apache.spark.sql.DataFrame = [person_id: string, name: string, age: string]

scala> █
```

1.

//udf function is created for assigning age grp given on the condition.
val ageGrp = udf((age: String) => { if(age.toInt < 20) { "<20"; } else { if(age.toInt > 35) { ">35"; } else { "20-35"; }}});

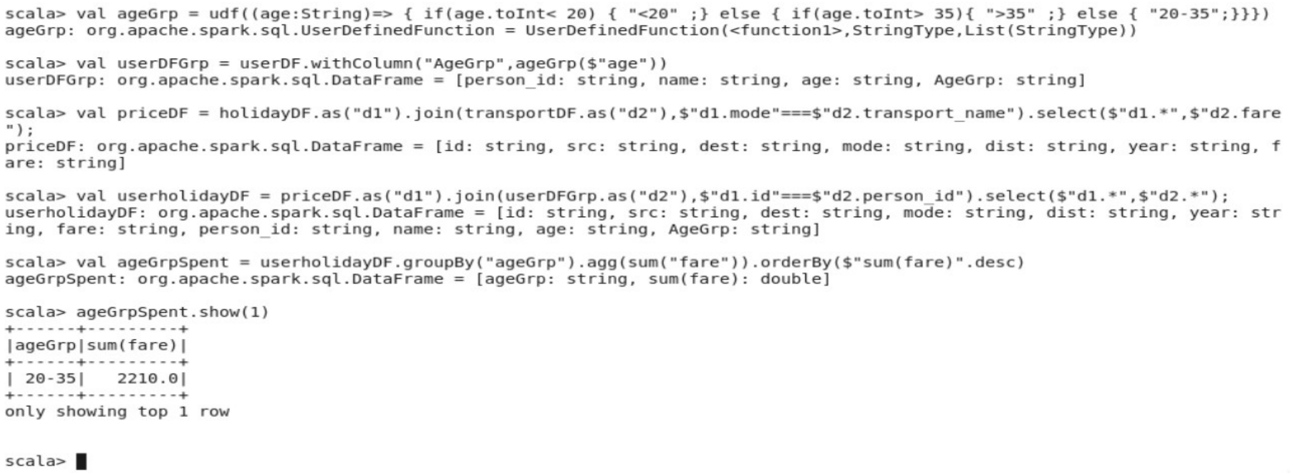
//New column is created depending on the value of age.
val userDFGrp = userDF.withColumn(“AgeGrp”,”ageGrp($”age”))

//A joined data frame is made of holiday and transport dataframes on mode.
val priceFare = holidayDF.as(“d1”).join(transportDF.as(“d2”), $”d1.mode”===$”d2.transport_name”).select($”d1.*”,$”d1.year”, $”d2.fare”);

//A joined data frame is made of priceDF and userDGGrp dataframes on id.
val userHolidayDF = priceDF.as(“d1”).join(userDFGrp.as(“d2”), $”d1.id”===$”d2.person_id”).select($”d1.*”,$”d2.*”);

//Grouped the dataframe by age group and its summation of fare is calculate and sorted in desceinding order.
val ageGrpSpent = userholidayDF.groupBy(“ageGrp”).agg(sum(“fare”)).orderBy($”sum(fare)”.desc)

//Revenue spent by age group
ageGrpSpent.show(1);



```
scala> val ageGrp = udf((age:String)=> { if(age.toInt< 20) { "<20" ;} else { if(age.toInt> 35){ ">35" ;} else { "20-35";}}})
ageGrp: org.apache.spark.sql.UserDefinedFunction = UserDefinedFunction(<function1>,StringType,List(StringType))

scala> val userDFGrp = userDF.withColumn("AgeGrp",ageGrp($"age"))
userDFGrp: org.apache.spark.sql.DataFrame = [person_id: string, name: string, age: string, AgeGrp: string]

scala> val priceDF = holidayDF.as("d1").join(transportDF.as("d2"),$"d1.mode"===$"d2.transport_name").select($"d1.*",$"d2.fare
");
priceDF: org.apache.spark.sql.DataFrame = [id: string, src: string, dest: string, mode: string, dist: string, year: string, f
are: string]

scala> val userholidayDF = priceDF.as("d1").join(userDFGrp.as("d2"),$"d1.id"===$"d2.person_id").select($"d1.*",$"d2.*");
userholidayDF: org.apache.spark.sql.DataFrame = [id: string, src: string, dest: string, mode: string, dist: string, year: str
ing, fare: string, person_id: string, name: string, age: string, AgeGrp: string]

scala> val ageGrpSpent = userholidayDF.groupBy("ageGrp").agg(sum("fare")).orderBy($"sum(fare)".desc)
ageGrpSpent: org.apache.spark.sql.DataFrame = [ageGrp: string, sum(fare): double]

scala> ageGrpSpent.show(1)
+------+---------+
|ageGrp|sum(fare)|
+------+---------+
| 20-35|   2210.0|
+------+---------+
only showing top 1 row

scala> █
```

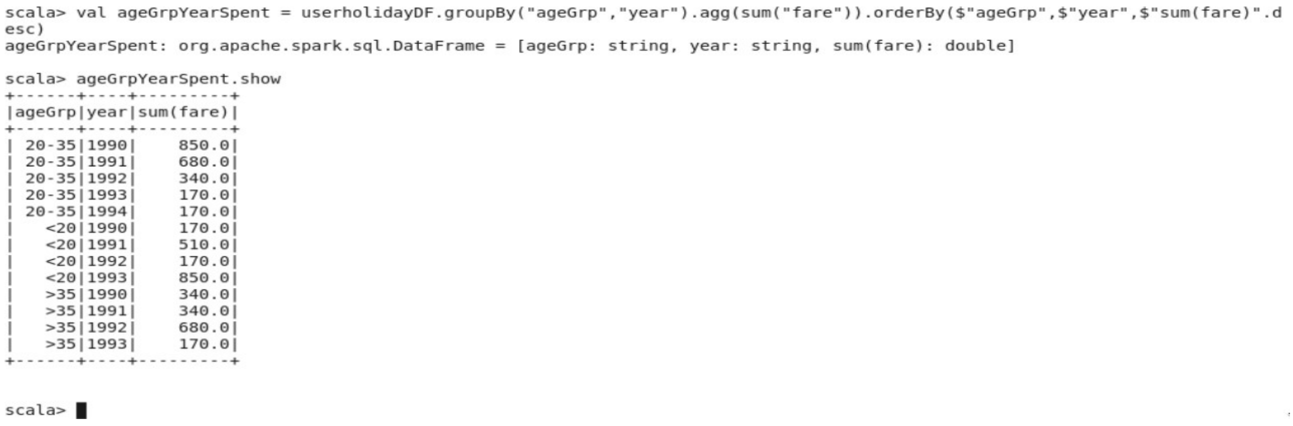
2.

//Grouped the dataframe by age group and its summation of fare is calculate and sorted in desceinding order.

val ageGrpSpent = userholidayDF.groupBy(“ageGrp”,”year”).agg(sum(“fare”)).orderBy($”sum(fare)”.desc)

//Displayed the amount spent by each age-group, every year in travelling

ageGrpYearSpent.show



```
scala> val ageGrpYearSpent = userholidayDF.groupBy("ageGrp","year").agg(sum("fare")).orderBy($"ageGrp",$"year",$"sum(fare)".d
esc)
ageGrpYearSpent: org.apache.spark.sql.DataFrame = [ageGrp: string, year: string, sum(fare): double]

scala> ageGrpYearSpent.show
+------+----+---------+
|ageGrp|year|sum(fare)|
+------+----+---------+
| 20-35|1990|    850.0|
| 20-35|1991|    680.0|
| 20-35|1992|    340.0|
| 20-35|1993|    170.0|
| 20-35|1994|    170.0|
|   <20|1990|    170.0|
|   <20|1991|    510.0|
|   <20|1992|    170.0|
|   <20|1993|    850.0|
|   >35|1990|    340.0|
|   >35|1991|    340.0|
|   >35|1992|    680.0|
|   >35|1993|    170.0|
+------+----+---------+

scala> █
```