**Session 18 Assignment 2**

scala> val transport = sc.textFile("/user/S18\_Dataset\_Transport.txt")

transport: org.apache.spark.rdd.RDD[String] = /user/S18\_Dataset\_Transport.txt MapPartitionsRDD[49] at textFile at <console>:27

scala> var transportTuples = transport.map(x=> {

| val row = x.split(",").toList

| (row.apply(0), row.apply(1).toInt)

| })

transportTuples: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[50] at map at <console>:29

scala> val user = sc.textFile("/user/S18\_Dataset\_User\_Details.txt")

user: org.apache.spark.rdd.RDD[String] = /user/S18\_Dataset\_User\_Details.txt MapPartitionsRDD[52] at textFile at <console>:27

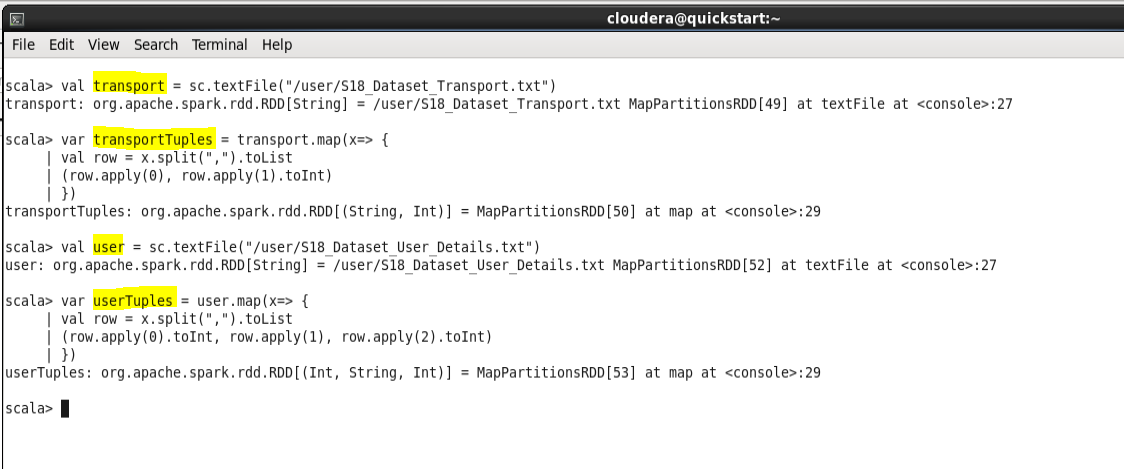
scala> var userTuples = user.map(x=> {

| val row = x.split(",").toList

| (row.apply(0).toInt, row.apply(1), row.apply(2).toInt)

| })

userTuples: org.apache.spark.rdd.RDD[(Int, String, Int)] = MapPartitionsRDD[53] at map at <console>:29



* **TASK - 1**

scala> val transportRdd = travelTuple.map(x=>(x.\_4,(x.\_1,x.\_6))).join(transportTuples).map(x => x.\_2).groupByKey()

transportRdd: org.apache.spark.rdd.RDD[((Int, Int), Iterable[Int])] = ShuffledRDD[74] at groupByKey at <console>:35

scala> val revenuePYrList = transportRdd.map(x=> (x.\_1,x.\_2.sum)).sortBy(x => -x.\_2)

revenuePYrList: org.apache.spark.rdd.RDD[((Int, Int), Int)] = MapPartitionsRDD[78] at sortBy at <console>:37

scala> revenuePYrList.first()

res15: ((Int, Int), Int) = ((1,1993),510)

scala> revenuePYrList.foreach(println)

((1,1993),510)

((7,1990),510)

((2,1991),340)

((4,1990),340)

((5,1992),340)

((9,1992),340)

((6,1991),340)

((2,1993),170)

((6,1993),170)

((10,1993),170)

((10,1992),170)

((10,1990),170)

((4,1991),170)

((5,1991),170)

((3,1993),170)

((1,1990),170)

((8,1990),170)

((5,1994),170)

((3,1991),170)

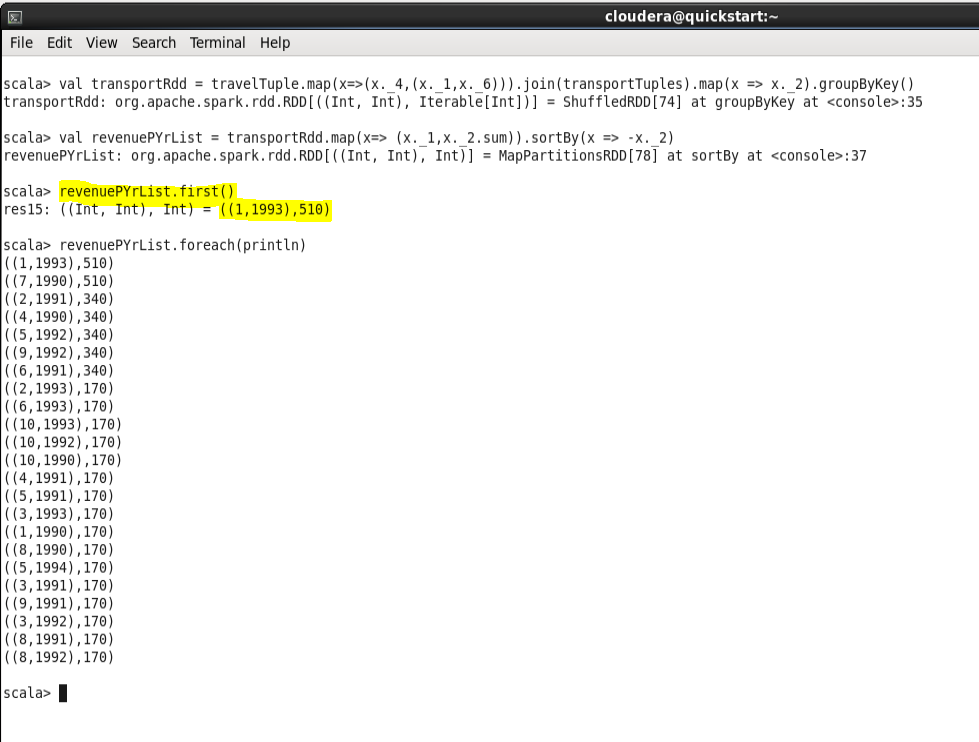
((9,1991),170)

((3,1992),170)

((8,1991),170)

((8,1992),170)

scala>



* **TASK - 2**

scala> val amountspent = travelTuple.map(x=>(x.\_4,(x.\_1,x.\_6))).join(transportTuples).map(x => x.\_2).groupByKey()

amountspent: org.apache.spark.rdd.RDD[((Int, Int), Iterable[Int])] = ShuffledRDD[90] at groupByKey at <console>:35

scala> val amountspentPYr = amountspent.map(x=> (x.\_1,x.\_2.sum)).sortBy(x => x.\_1.\_1)

amountspentPYr: org.apache.spark.rdd.RDD[((Int, Int), Int)] = MapPartitionsRDD[94] at sortBy at <console>:37

scala> amountspentPYr.foreach(println)

((1,1993),510)

((1,1990),170)

((2,1993),170)

((2,1991),340)

((3,1993),170)

((3,1991),170)

((3,1992),170)

((4,1990),340)

((4,1991),170)

((5,1992),340)

((5,1991),170)

((5,1994),170)

((6,1993),170)

((6,1991),340)

((7,1990),510)

((8,1990),170)

((8,1991),170)

((8,1992),170)

((9,1992),340)

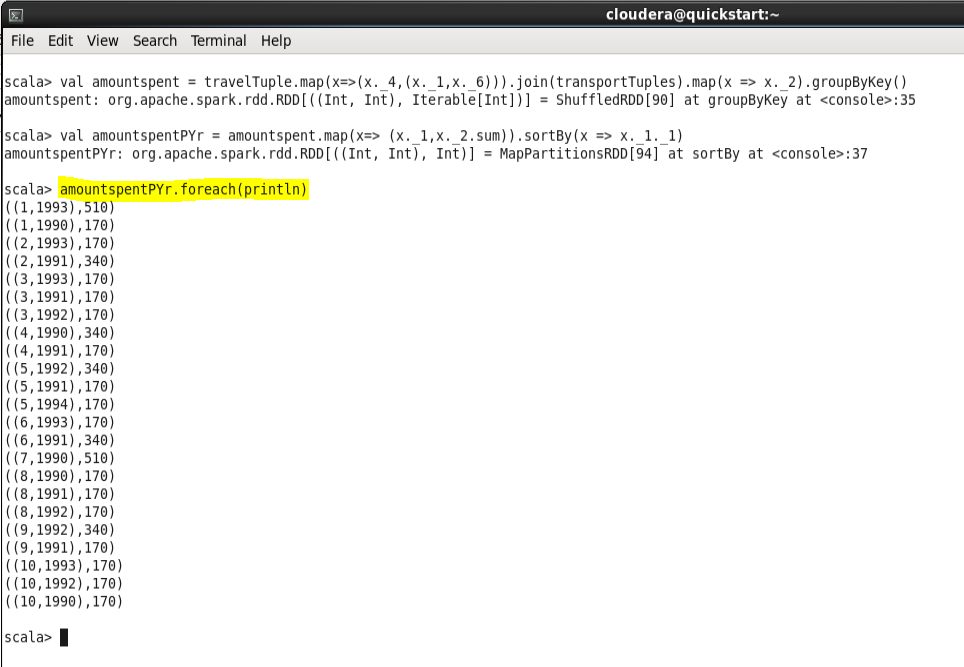
((9,1991),170)

((10,1993),170)

((10,1992),170)

((10,1990),170)

scala>



* **TASK - 3**

scala> val travelrdd = travelTuples.map(x => (x.\_1 ,(x.\_6, 1)))

travelrdd: org.apache.spark.rdd.RDD[(Int, (Int, Int))] = MapPartitionsRDD[25] at map at <console>:31

scala> val agegroup = userTuples.map(x=> x.\_1->{

| if (x.\_3<20) "<20"

| else if (x.\_3 > 35) ">35"

| else "20-35"

| }).join(travelrdd)

agegroup: org.apache.spark.rdd.RDD[(Int, (String, (Int, Int)))] = MapPartitionsRDD[29] at join at <console>:41

scala> val travellingmostagegroup = agegroup.map(x => ((x.\_2.\_1 , x.\_2.\_2.\_1) , x.\_2.\_2.\_2)).groupByKey().map(x=>(x.\_1 ,x.\_2.sum)).sortBy(x => -x.\_2)

travellingmostagegroup: org.apache.spark.rdd.RDD[((String, Int), Int)] = MapPartitionsRDD[35] at sortBy at <console>:39

scala> travellingmostagegroup.foreach(println)

((20-35,1990),5)

((<20,1993),5)

((20-35,1991),4)

((>35,1992),4)

((<20,1991),3)

((>35,1991),2)

((>35,1990),2)

((20-35,1992),2)

((20-35,1994),1)

((20-35,1993),1)

((<20,1992),1)

((<20,1990),1)

((>35,1993),1)

scala>

