**Session 17 Assignment 2**

* **PROBLEM STATEMENT – 1**

scala> val file = sc.textFile("/user/17.2\_Dataset.txt")

file: org.apache.spark.rdd.RDD[String] = /user/17.2\_Dataset.txt MapPartitionsRDD[3] at textFile at <console>:27

scala> val tuple = file.map( x => {

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

| (row.apply(0), row.apply(1), row.apply(2), row.apply(3).toInt, row.apply(4).toInt)

| } )

tuple: org.apache.spark.rdd.RDD[(String, String, String, Int, Int)] = MapPartitionsRDD[4] at map at <console>:29

scala> tuple.count()

res0: Long = 22

scala> val subjects = tuple.map(x => x.\_2).distinct()

subjects: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[8] at distinct at <console>:31

scala> subjects.foreach(println)

maths

history

science

scala> subjects.count()

res2: Long = 3

scala> val mathew55 = tuple.map(x => (x.\_1, x.\_4)).filter(x => (x.\_1 == "Mathew" && x.\_4 == 55))

<console>:31: error: value \_4 is not a member of (String, Int)

val mathew55 = tuple.map(x => (x.\_1, x.\_4)).filter(x => (x.\_1 == "Mathew" && x.\_4 == 55))

^

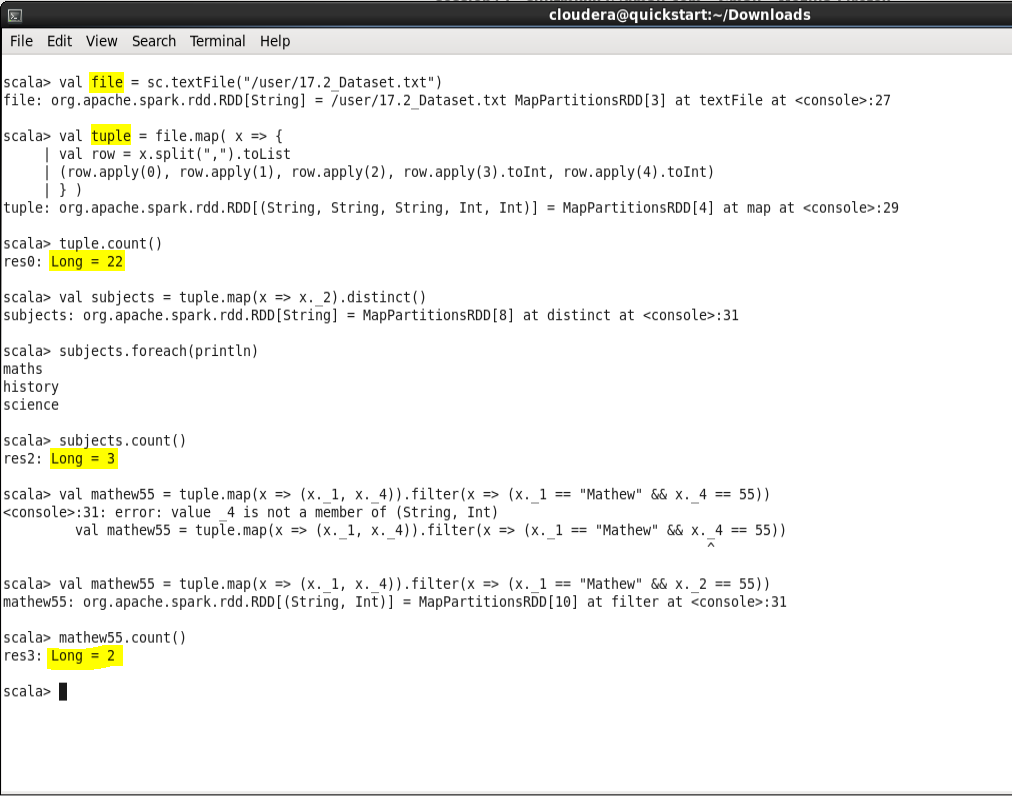
scala> val mathew55 = tuple.map(x => (x.\_1, x.\_4)).filter(x => (x.\_1 == "Mathew" && x.\_2 == 55))

mathew55: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[10] at filter at <console>:31

scala> mathew55.count()

res3: Long = 2

scala>



* **PROBLEM STATEMENT – 2**

scala> val studentsgrade = tuple.map(x => (x.\_3,1)).reduceByKey((x,y) => x+y)

studentsgrade: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[19] at reduceByKey at <console>:31

scala> studentsgrade.count()

res6: Long = 3

scala> studentsgrade.foreach(println)

(grade-3,4)

(grade-1,9)

(grade-2,9)

scala> val avg = tuple.map(x => ((x.\_1, x.\_3),x.\_4)).groupByKey().map(x => (x.\_1,(x.\_2.sum.toDouble/(x.\_2.size))))

avg: org.apache.spark.rdd.RDD[((String, String), Double)] = MapPartitionsRDD[22] at map at <console>:31

scala> avg.foreach(prntln)

<console>:34: error: not found: value prntln

avg.foreach(prntln)

^

scala> avg.foreach(println)

((Lisa,grade-1),24.0)

((Mark,grade-2),17.5)

((Lisa,grade-2),61.0)

((Mathew,grade-3),45.0)

((Andrew,grade-2),77.0)

((Andrew,grade-1),43.666666666666664)

((Lisa,grade-3),86.0)

((John,grade-1),38.666666666666664)

((John,grade-2),74.0)

((Mark,grade-1),84.0)

((Andrew,grade-3),35.0)

((Mathew,grade-2),65.66666666666667)

scala> val subjectavg = tuple.map(x => (x.\_2, x.\_4)).groupByKey().map(x => (x.\_1, (x.\_2.sum.toDouble/(x.\_2.size))))

subjectavg: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[31] at map at <console>:31

scala> subjectavg.foreach(println)

(maths,46.833333333333336)

(history,69.75)

(science,38.25)

scala> val subjectgradeavg = tuple.map(x => ((x.\_2, x.\_3), x.\_4)).groupByKey().map(x => (x.\_1, (x.\_2.sum.toDouble/(x.\_2.size))))

subjectgradeavg: org.apache.spark.rdd.RDD[((String, String), Double)] = MapPartitionsRDD[34] at map at <console>:31

scala> subjectgradeavg.foreach(println)

((history,grade-2),79.25)

((history,grade-3),86.0)

((maths,grade-1),46.0)

((science,grade-3),38.333333333333336)

((science,grade-1),50.0)

((science,grade-2),30.333333333333332)

((history,grade-1),51.666666666666664)

((maths,grade-2),48.5)

scala> val gradefilter = tuple.filter(x => x.\_3 == "grade-2")

gradefilter: org.apache.spark.rdd.RDD[(String, String, String, Int, Int)] = MapPartitionsRDD[35] at filter at <console>:31

scala> val grade2avg = gradefilter.map(x => (x.\_1, x.\_4.toDouble)).groupByKey().map(x => (x.\_1, (x.\_2.sum.toDouble/(x.\_2.size))))

grade2avg: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[38] at map at <console>:33

scala> val grade2avg50 = grade2avg.filter(x => x.\_2 > 50)

grade2avg50: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[39] at filter at <console>:35

scala> grade2avg50.foreach(println)

(Andrew,77.0)

(Mathew,65.66666666666667)

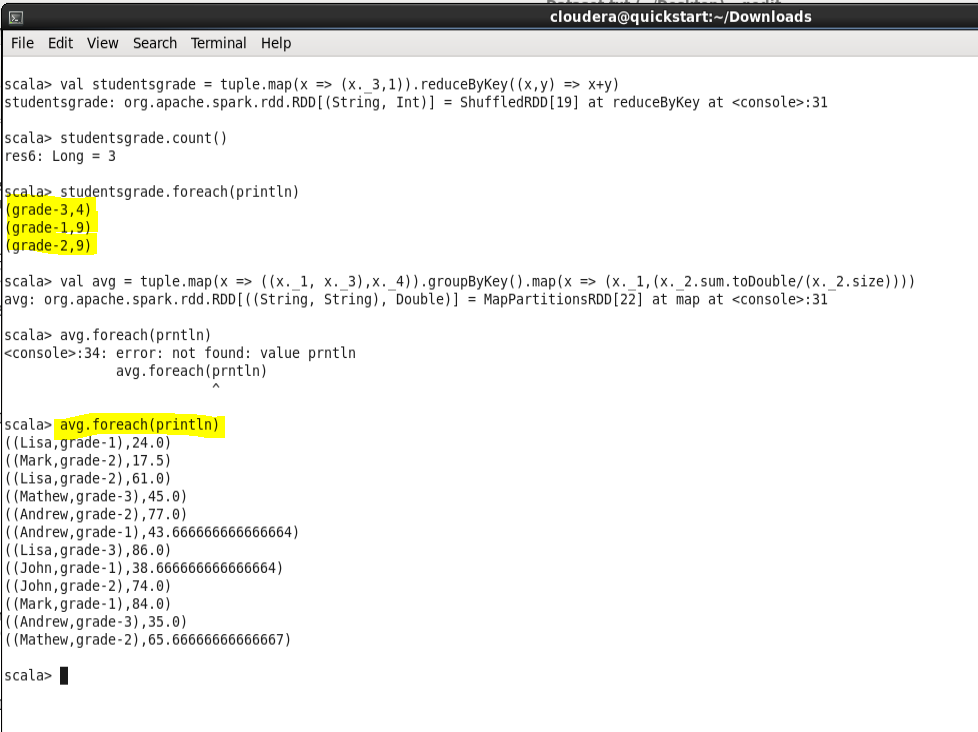
(John,74.0)

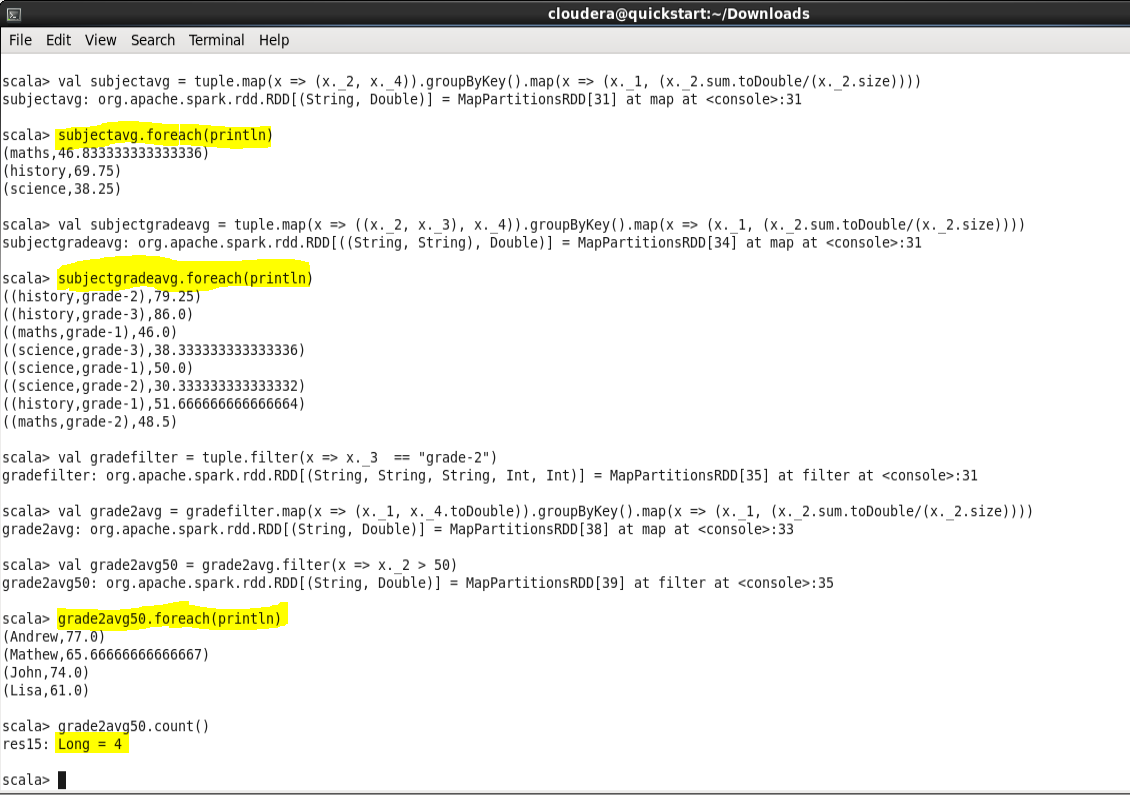
(Lisa,61.0)

scala> grade2avg50.count()

res15: Long = 4

scala>





* **PROBLEM STATEMENT - 3**

scala> val studentavg = tuple.map(x => (x.\_1,x.\_4)).groupByKey().map(x => (x.\_1,(x.\_2.sum.toDouble/(x.\_2.size))))

studentavg: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[62] at map at <console>:31

scala> val studentavggrade = tuple.map(x => ((x.\_1 + "," + x.\_3), x.\_4)).groupByKey().map(x => (x.\_1,(x.\_2.sum.toDouble/(x.\_2.size))))

studentavggrade: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[65] at map at <console>:31

scala> val intersection = studentavg.intersection(studentavggrade)

intersection: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[71] at intersection at <console>:35

scala> intersection.count()

res21: Long = 0

scala>

