Given a dataset of college students as a text file (name, subject, grade, marks) :

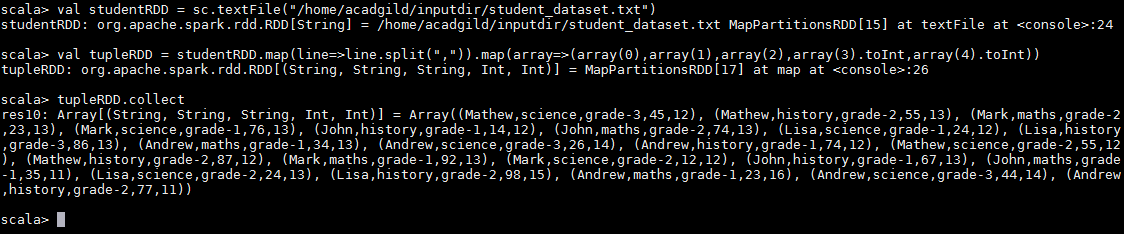
**Problem Statement 1:**

1. Read the text file, and create a tuple rdd.

**val studentRDD = sc.textFile("/home/acadgild/inputdir/student\_dataset.txt")**

**val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))**

**tupleRDD.collect**

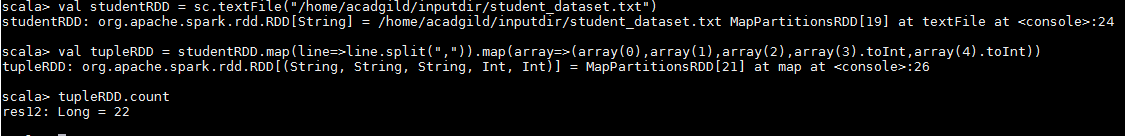


1. Find the count of total number of rows present.

**val studentRDD = sc.textFile("/home/acadgild/inputdir/student\_dataset.txt")**

**val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))**

**tupleRDD.count**



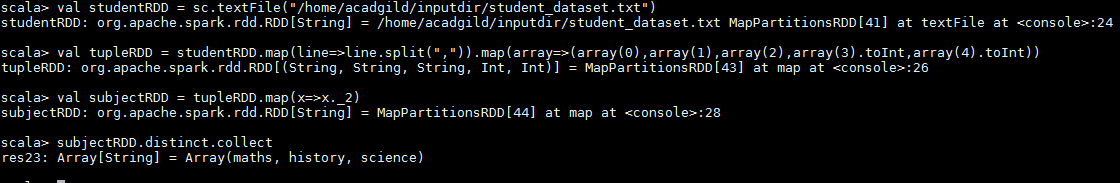
1. What is the distinct number of subjects present in the entire school

**val studentRDD = sc.textFile("/home/acadgild/inputdir/student\_dataset.txt")**

**val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))**

**val subjectRDD = tupleRDD.map(x=>x.\_2)**

**subjectRDD.distinct.collect**



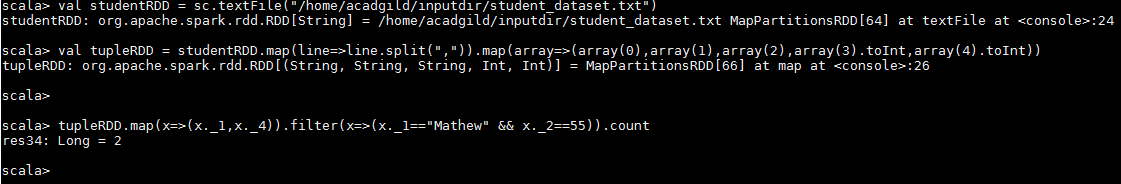
1. What is the count of the number of students in the school, whose name is Mathew and

marks is 55.

**val studentRDD = sc.textFile("/home/acadgild/inputdir/student\_dataset.txt")**

**val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))**

**tupleRDD.map(x=>(x.\_1,x.\_4)).filter(x=>(x.\_1=="Mathew" && x.\_2==55)).count**



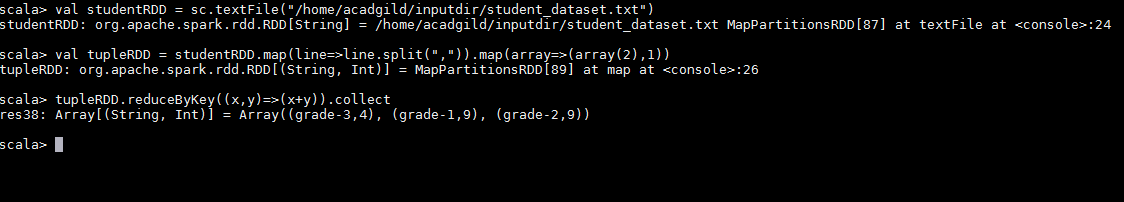
**Problem Statement 2:**

1. What is the count of students per grade in the school?

**val studentRDD = sc.textFile("/home/acadgild/inputdir/student\_dataset.txt")**

**val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(2),1))**

**tupleRDD.reduceByKey((x,y)=>(x+y)).collect**



1. Find the average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)

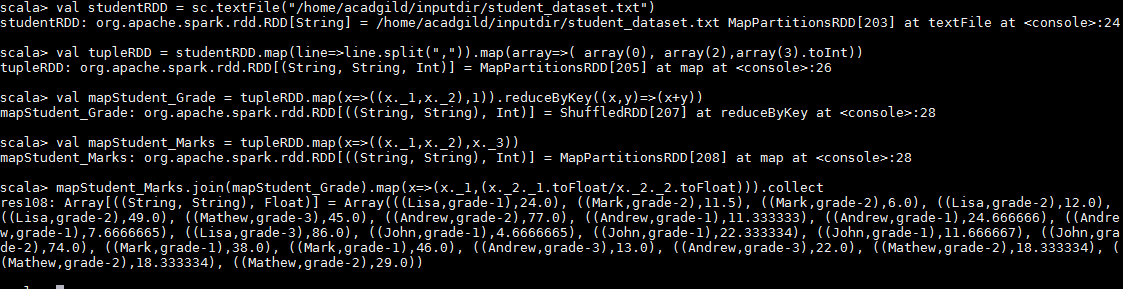
**val studentRDD = sc.textFile("/home/acadgild/inputdir/student\_dataset.txt")**

**val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>( array(0), array(2),array(3).toInt))**

**val mapStudent\_Grade = tupleRDD.map(x=>((x.\_1,x.\_2),1)).reduceByKey((x,y)=>(x+y))**

**val mapStudent\_Marks = tupleRDD.map(x=>((x.\_1,x.\_2),x.\_3))**

**mapStudent\_Marks.join(mapStudent\_Grade).map(x=>(x.\_1,(x.\_2.\_1.toFloat/x.\_2.\_2.toFloat))).collect**



1. What is the average score of students in each subject across all grades?

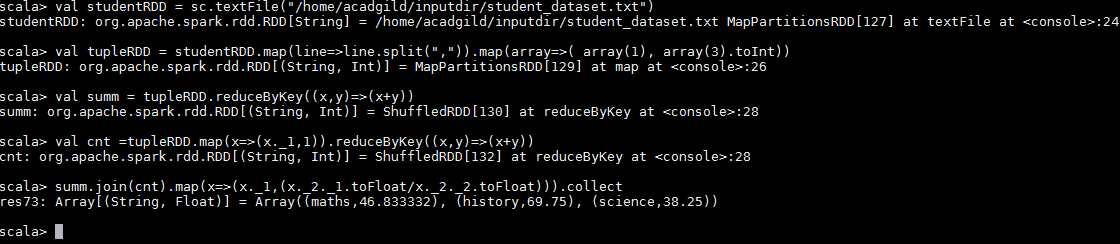
**val studentRDD = sc.textFile("/home/acadgild/inputdir/student\_dataset.txt")**

**val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>( array(1), array(3).toInt))**

**val summ = tupleRDD.reduceByKey((x,y)=>(x+y))**

**val cnt =tupleRDD.map(x=>(x.\_1,1)).reduceByKey((x,y)=>(x+y))**

**summ.join(cnt).map(x=>(x.\_1,(x.\_2.\_1.toFloat/x.\_2.\_2.toFloat))).collect**



1. What is the average score of students in each subject per grade?

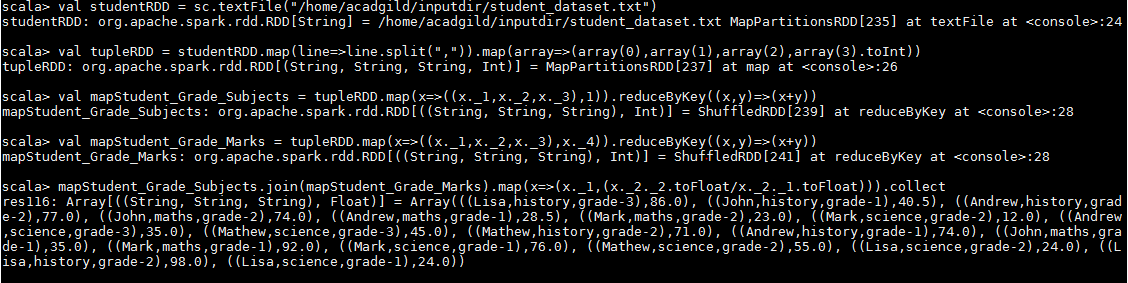
**val studentRDD = sc.textFile("/home/acadgild/inputdir/student\_dataset.txt")**

**val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt))**

**val mapStudent\_Grade\_Subjects = tupleRDD.map(x=>((x.\_1,x.\_2,x.\_3),1)).reduceByKey((x,y)=>(x+y))**

**val mapStudent\_Grade\_Marks = tupleRDD.map(x=>((x.\_1,x.\_2,x.\_3),x.\_4)).reduceByKey((x,y)=>(x+y))**

**mapStudent\_Grade\_Subjects.join(mapStudent\_Grade\_Marks).map(x=>(x.\_1,(x.\_2.\_2.toFloat/x.\_2.\_1.toFloat))).collect**



1. For all students in grade-2, how many have average score greater than 50?

**val studentRDD = sc.textFile("/home/acadgild/inputdir/student\_dataset.txt")**

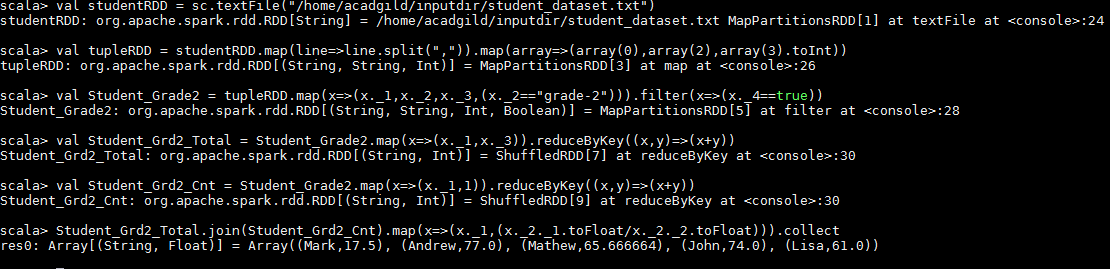
**val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(2),array(3).toInt))**

**val Student\_Grade2 = tupleRDD.map(x=>(x.\_1,x.\_2,x.\_3,(x.\_2=="grade-2"))).filter(x=>(x.\_4==true))**

**val Student\_Grd2\_Total = Student\_Grade2.map(x=>(x.\_1,x.\_3)).reduceByKey((x,y)=>(x+y))**

**val Student\_Grd2\_Cnt = Student\_Grade2.map(x=>(x.\_1,1)).reduceByKey((x,y)=>(x+y))**

**Student\_Grd2\_Total.join(Student\_Grd2\_Cnt).map(x=>(x.\_1,(x.\_2.\_1.toFloat/x.\_2.\_2.toFloat))).collect**



**Problem Statement 3:**

Are there any students in the college that satisfy the below criteria :

1. Average score per student\_name across all grades is same as average score per

student\_name per grade.

**val studentRDD = sc.textFile("/home/acadgild/inputdir/student\_dataset.txt")**

**val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(3).toInt))**

**val Student\_Total = tupleRDD.reduceByKey((x,y)=>(x+y))**

**val Student\_Count = tupleRDD.map(x=>(x.\_1,1)).reduceByKey((x,y)=>(x+y))**

**val Student\_avg = Student\_Total.join(Student\_Count).map(x=>(x.\_1,(x.\_2.\_1.toFloat/x.\_2.\_2.toFloat)))**

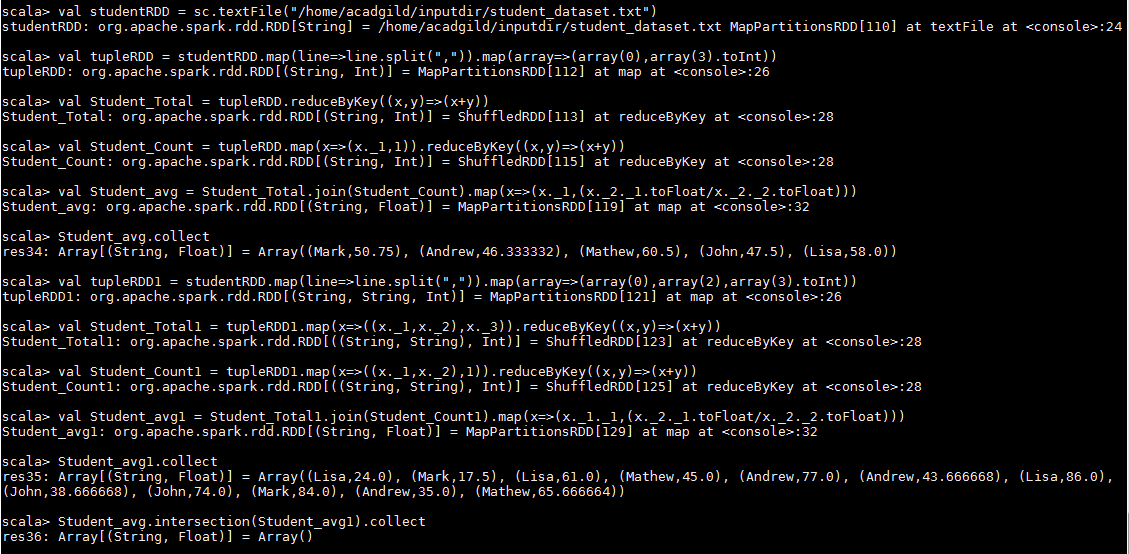
**val tupleRDD1 = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(2),array(3).toInt))**

**val Student\_Total1 = tupleRDD1.map(x=>((x.\_1,x.\_2),x.\_3)).reduceByKey((x,y)=>(x+y))**

**val Student\_Count1 = tupleRDD1.map(x=>((x.\_1,x.\_2),1)).reduceByKey((x,y)=>(x+y))**

**val Student\_avg1 = Student\_Total1.join(Student\_Count1).map(x=>(x.\_1.\_1,(x.\_2.\_1.toFloat/x.\_2.\_2.toFloat)))**

**Student\_avg.intersection(Student\_avg1).collect**



**There are no students who satisfy the criteria.**