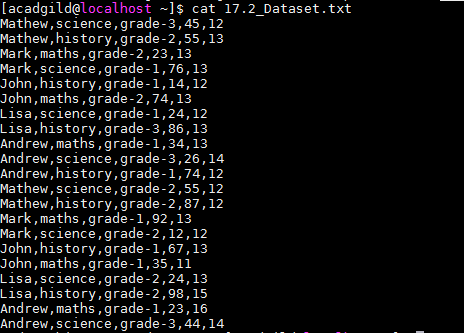
Given Dataset of college students as a text file (name,subject,grade,marks)

**Dataset screen shot**



**Problem Statement 1:**

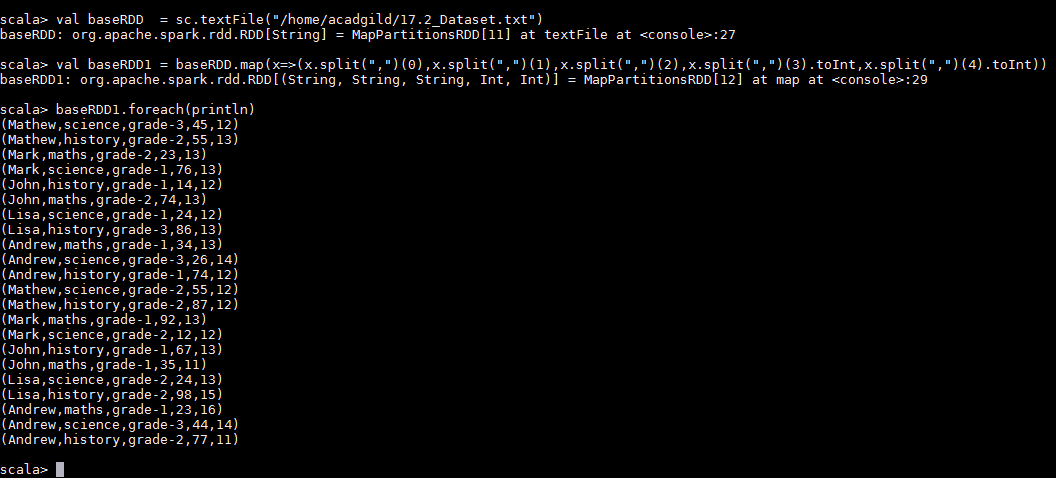
1. **Read the text file, and create a tupled rdd.**

**Solution :**

Val baseRDD = sc.textFile(“/home/acadgild/17.2\_Dataset.txt”).

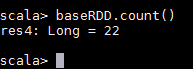
val baseRDD1 = baseRDD.map(x=>(x.split(",")(0),x.split(",")(1),x.split(",")(2),x.split(",")(3).toInt,x.split(",")(4).toInt))

baseRDD1.foreach(println)



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

**baseRDD.count()**



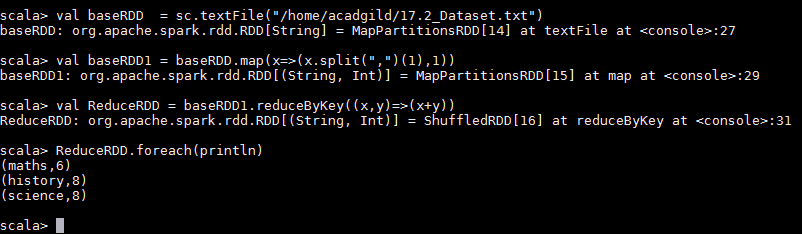
1. **What is the distinct number of subjects present in the entire school**

val baseRDD = sc.textFile(“/home/acadgild/17.2\_Dataset.txt”)

val baseRDD1 = baseRDD.map(x=>(x.split(",")(1),1))

val ReduceRDD = baseRDD1.reduceByKey((x,y)=>(x+y))

ReduceRDD.foreach(println)



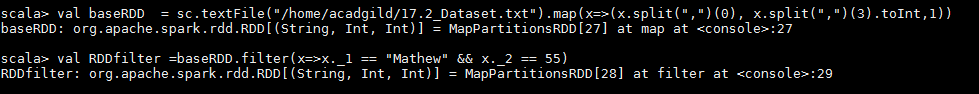
4. **What is the count of the number of students in the school, whose name is Mathew and**

**marks is 55**

val baseRDD = sc.textFile(“/home/acadgild/17.2\_Dataset.txt”).map(x=>(x.split(",")(0), x.split(",")(3).toInt,1))

val RDDfilter= baseRDD .filter(x=>x.\_1 ==’Mathew” &&x.\_2==55)

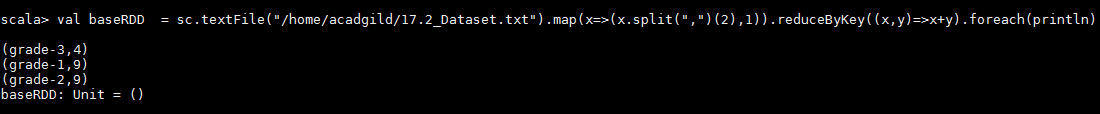
val ReduceRDD = RDDfilter.reduceByKey((x,y)=>(x+y)).foreach(println)



**Problem Statement 2:**

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

val baseRDD = sc.textFile(“/home/acadgild/17.2\_Dataset.txt”).map(x=>(x.split(",")(2), x.split(",")(3).toInt,1)).reduceByKey((x,y)=>x+y).foreach(println)



2**. Find the average of each student (Note - Mathew is grade-1, is different from Mathew in**

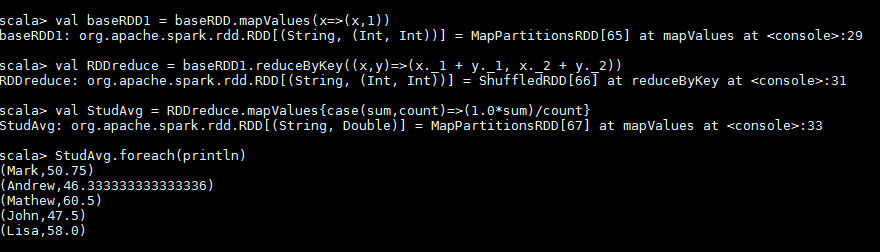
**some other grade!)**

val baseRDD = sc.textFile("/home/acadgild/17.2\_Dataset.txt").map(x=>(x.split(",")(0), x.split(",")(3).toInt))

val baseRDD1 = baseRDD.mapValues(x=>(x,1))

val RDDreduce = baseRDD1.reduceByKey((x,y)=>(x.\_1 + y.\_1, x.\_2 + y.\_2))

val StudAvg = RDDreduce.mapValues{case(sum,count)=>(1.0\*sum)/count}



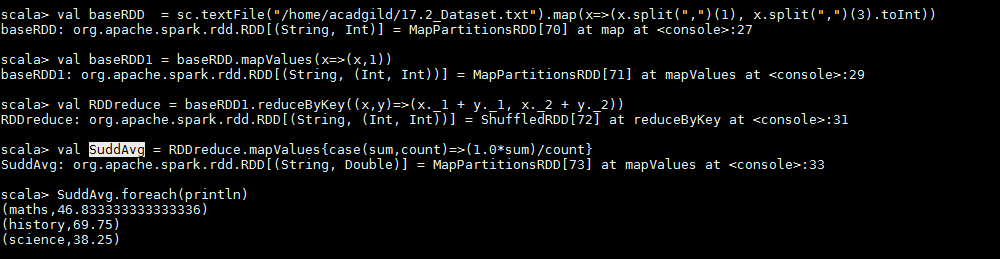
1. **What is the average score of students in each subject across all grades?**

val baseRDD = sc.textFile("/home/acadgild/17.2\_Dataset.txt").map(x=>(x.split(",")(1), x.split(",")(3).toInt))

val baseRDD1 = baseRDD.mapValues(x=>(x,1))

val RDDreduce = baseRDD1.reduceByKey((x,y)=>(x.\_1 + y.\_1, x.\_2 + y.\_2))

val SuddAvg = RDDreduce.mapValues{case(sum,count)=>(1.0\*sum)/count}



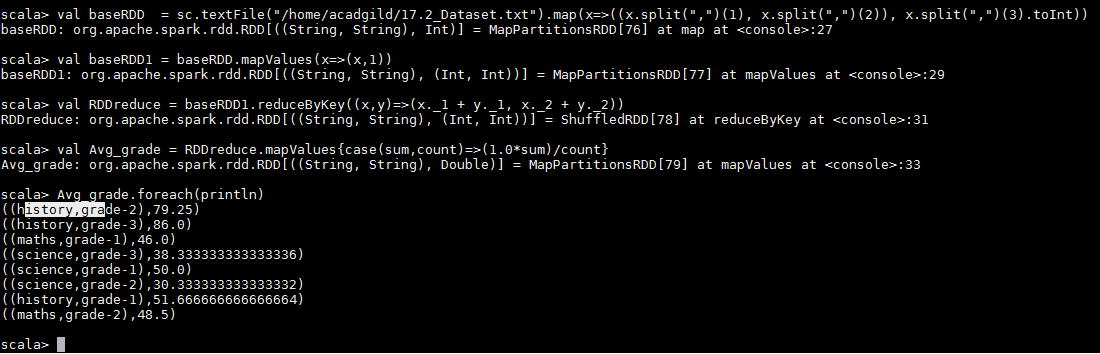
1. **What is the average score of students in each subject per grade?**

val baseRDD = sc.textFile("/home/acadgild/17.2\_Dataset.txt").map(x=>((x.split(",")(1), x.split(",")(2)), x.split(",")(3).toInt))

val baseRDD1 = baseRDD.mapValues(x=>(x,1))

val RDDreduce = baseRDD1.reduceByKey((x,y)=>(x.\_1 + y.\_1, x.\_2 + y.\_2))

val Avg\_grade = RDDreduce.mapValues{case(sum,count)=>(1.0\*sum)/count}



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

val baseRDD = sc.textFile("/home/acadgild/17.2\_Dataset.txt").map(x=>((x.split(",")(0), x.split(",")(2)), x.split(",")(3).toInt))

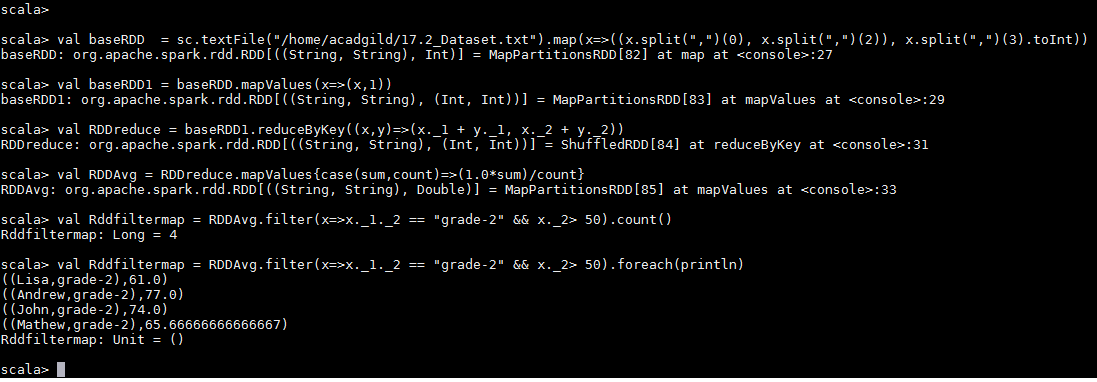
val baseRDD1 = baseRDD.mapValues(x=>(x,1))

val RDDreduce = baseRDD1.reduceByKey((x,y)=>(x.\_1 + y.\_1, x.\_2 + y.\_2))

val RDDAvg = RDDreduce.mapValues{case(sum,count)=>(1.0\*sum)/count}

val Rddfiltermap = RDDAvg.filter(x=>x.\_1.\_2 == “grade-2” && x.\_2> 50).count()

val Rddfiltermap = RDDAvg.filter(x=>x.\_1.\_2 == “grade-2” && x.\_2> 50).foreach(println))



**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

Hint - Use Intersection Property.

val baseRDD = sc.textFile("/home/acadgild/17.2\_Dataset.txt").map(x=>(x.split(",")(0), x.split(",")(3).toInt))

val stuAvg = baseRDD.mapValues(x=>(x,1)).foreach(println)

val RDDreduce = stuAvg.reduceByKey((x,y)=>(x.\_1 + y.\_1, x.\_2 + y.\_2))

val avg\_std = RDDreduce.mapValues{case(sum,count)=>(1.0\*sum)/count}

