# **SESSION 17 ASSIGNMENT 2**

**Problem Statement 1:**

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

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

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

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

marks is 55.

**Problem Statement 2:**

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

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

some other grade!)

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

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

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

**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.

**Solution:**

import org.apache.log4j.Logger

import org.apache.log4j.Level

import org.apache.spark.sql.SparkSession

object Assignment\_17\_2 extends App {

Logger.getLogger("org").setLevel(Level.OFF)

Logger.getLogger("akka").setLevel(Level.OFF)

val spark = SparkSession.builder()

.master("local")

.appName("example")

.getOrCreate()

val sc = spark.sparkContext

//Loading the text file

val csvDF1 = sc.textFile("C:/ACADGILD/Big Data/SESSION\_17/17.2\_Dataset.txt")

//We are creating a Resilient Distributed Dataset (RDD), the basic abstraction in Spark. Represents an immutable,

// partitioned collection of elements that can be operated on in parallel.

val arrayTuples = csvDF1.map(line => line.split(","))

.map(array => (array(0), array(1), array(2), array(3), array(4)))

.collect

println("Total number of rows = " + arrayTuples.length)

println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

//For geting distinct subjects we will use distinct function which returns distinct values of field.

val distinctRdd = arrayTuples.map(x => x.\_2).distinct.toList

println("Distinct subjects are :\n" + distinctRdd.mkString("\n"))

println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

println("Distinct number of subjects present in the entire school = "+(distinctRdd).length)

println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

//For selecting specific value we will use filter statement

//which returns the tuple with matching values

val distinctdata = arrayTuples.filter(x=>x.\_1=="Mathew" && x.\_4 == "55")

println("The count of the number of students in the school, whose name is Mathew and marks is 55 = "

+ distinctdata.length)

println("###########################################################################################")

val stud\_per\_grd = csvDF1.map(\_.split(",")).map(\_(2)).countByValue()

println("the count of students per grade in the school :\n" + stud\_per\_grd.mkString("\n"))

println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

val sum\_a = csvDF1.map(x=>x.split(","))

val sum\_b = sum\_a.map(line=>((line(0),line(2)),line(3).toFloat))

.groupByKey().mapValues(x => x.sum/x.size).collect()

println("Average of each student in each grade = " + sum\_b.mkString("\n") )

println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

//Grouping on the basis of key - In groupby, the RDD is reduced on the basis of key provided

val sum\_c = sum\_a.map(line =>(line(1),line(3).toFloat))

.groupByKey().mapValues(x => x.sum/x.size).collect()

println("The average score of students in each subject across all grades = " + sum\_c.mkString("\n") )

println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

val sum\_d = sum\_a.map(line =>((line(1),line(2)),line(3).toFloat))

.groupByKey().mapValues(x => x.sum/x.size).collect()

println("the average score of students in each subject per grade = " + sum\_d.mkString("\n") )

println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

val sum\_e = sum\_a.filter(x=>x(2)=="grade-2").map(line => (line(0),line(3).toFloat))

.groupByKey().mapValues(x => x.sum/x.size)

val sum\_ee = sum\_e.filter(\_.\_2 > 50.00).collect()

println("students in grade-2, having average score greater than 50 :" + sum\_ee.mkString("\n"))

println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

val gp\_ag = sum\_a.map(x => (x(0),x(3).toFloat)).groupByKey().mapValues(x => x.sum/x.size).collect()

val gp\_pg = sum\_a.map(x => ((x(0),x(2)),x(3).toFloat)).groupByKey().mapValues(x => x.sum/x.size).collect()

val gp = gp\_pg.map(x=>(x.\_1.\_1,x.\_2))

//Intersection is a function which provides the comoon values between two rdd

val gp\_pg\_1 = gp\_ag.intersect(gp)

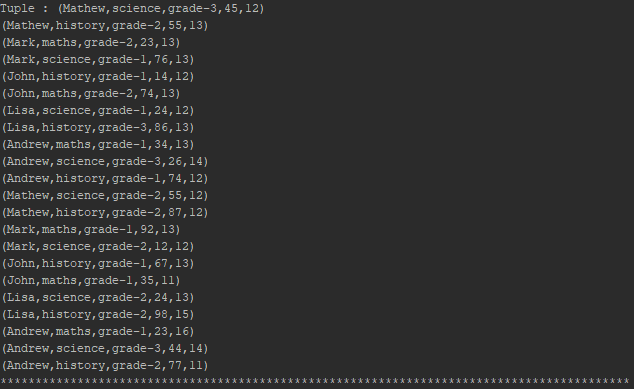
println("Average score per student\_name across all grades " + gp\_ag.mkString("\n"))

println("average score per student\_name per grade " + gp\_pg.mkString("\n"))

println("Number of students who satisfy the condition : " + gp\_pg\_1.length)

}

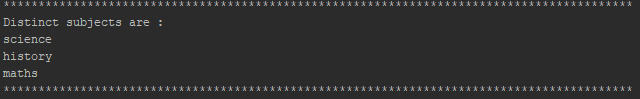
**Output:**

Read the text file, and create a tupled rdd

Find the count of total number of rows present



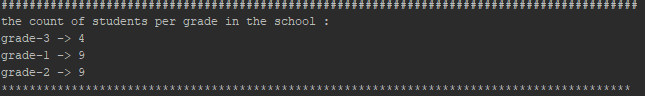
What is the distinct number of subjects present in the entire school



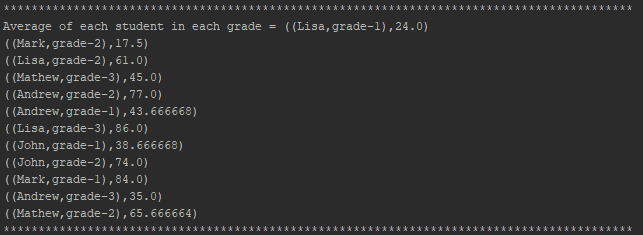
What is the count of the number of students in the school, whose name is Mathew and marks is 55



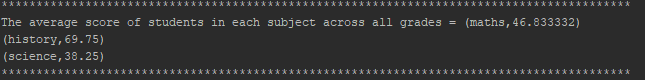
What is the count of students per grade in the school?



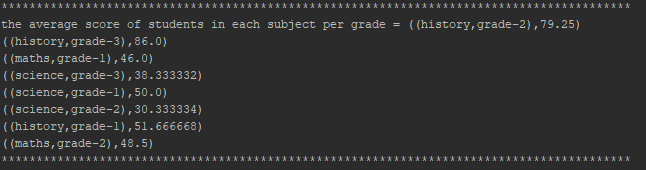
Find the average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)



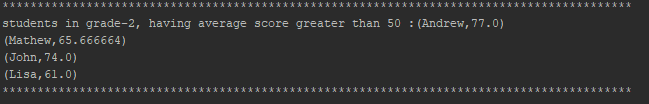
What is the average score of students in each subject across all grades?



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



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



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

