

# Assignment\_46

## Problem Statement

## Task 1

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

## Dataset

### Problem Statement 1:

1. Read the text file, and create a tupled rdd.
2. Find the count of total number of rows present.

```
val student_rdd = sc.readFile("/home/acadgild/assignment/student_dataset")
val header = student_rdd.first()
val student_records = student_rdd.map(records => records != header)
println(student_rdd.count())
```

A screenshot of a Scala REPL terminal window titled "acadgild@localhost: ~/assignment". The window has a menu bar with File, Edit, View, Search, Terminal, Tabs, and Help. Below the menu bar are two tabs, both labeled "acadgild@localhost: ~/assignment". The terminal shows the following commands and output:

```
scala> val student_rdd = sc.textFile("/home/acadgild/assignment/student dataset")  
student_rdd: org.apache.spark.rdd.RDD[String] = /home/acadgild/assignment/student_dataset MapPartitionsRDD[34] at textFile at <console>:24  
  
scala> val header = student_rdd.first()  
header: String = name,subject,grade,marks,age  
  
scala> val student_records = student_rdd.map(x=> x != header)  
student_records: org.apache.spark.rdd.RDD[Boolean] = MapPartitionsRDD[35] at map at <console>:27  
  
scala> println(student_records.count())  
23  
  
scala>  
  
scala>  
  
scala>  
  
scala>  
  
scala>  
  
scala>  
  
scala>  
  
scala>  
  
scala>
```

The bottom status bar shows "acadgild@localhost: ~/..." and some system icons on the right.

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

Steps: Create a dataframe from the CSV file Register a temporary table Student

Create SQL Query which will return count of distinct subjects

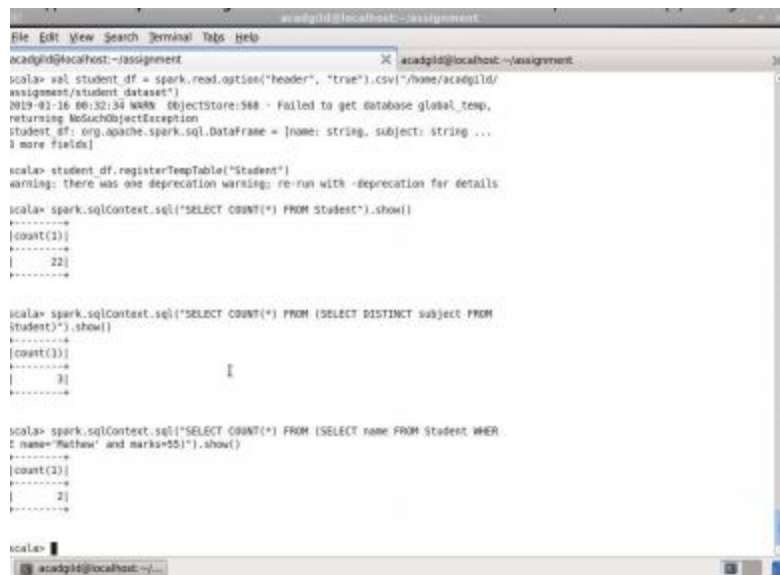
```
spark.sqlContext.sql("SELECT COUNT(*) FROM (SELECT DISTINCT subject FROM Student)").show()
```

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

- Create SQL with query criteria of name is Mathew and marks is 55 and get the count

```
spark.sqlContext.sql("SELECT COUNT(*) FROM (SELECT name FROM Student WHERE name='Mathew' and marks=55)").show()
```

Screenshots for the above code



```
scala> val student_df = spark.read.option("header", "true").csv("/home/acadgild/assignment/student_dataset")
2019-01-16 00:32:34 WARN ObjectStore:588 - Failed to get database global_temp, returning InMemoryObjectStoreException
student_df: org.apache.spark.sql.DataFrame = [name: string, subject: string ... 1 more fields]

scala> student_df.registerTempTable("Student")
warning: there was one deprecation warning; re-run with -deprecation for details

scala> spark.sqlContext.sql("SELECT COUNT(*) FROM Student").show()
=====
[count(1)]
=====
|      22|
=====

scala> spark.sqlContext.sql("SELECT COUNT(*) FROM (SELECT DISTINCT subject FROM Student)").show()
=====
[count(1)]
=====
|      3|
=====

scala> spark.sqlContext.sql("SELECT COUNT(*) FROM (SELECT name FROM Student WHERE name='Mathew' and marks=55)").show()
=====
[count(1)]
=====
|      2|
=====

scala>
```

## Task 2

- 1) What is the distribution of the total number of air-travelers per year
- 2) What is the total air distance covered by each user per year
- 3) Which user has travelled the largest distance till date
- 4) What is the most preferred destination for all users.

```
import org.apache.spark.sql.functions._
val df = spark.read.option("header","true").csv("/FileStore/tables/S18_Dataset.txt")
println(s"The distribution of total number of air-travelers per year:-")
df.groupBy("year_of_travel").agg(count("user_id") as "no.of traveller").show()
```

The distribution of total number of air-travelers per year:-

```
+-----+-----+
|year_of_travel|no.of traveller|
+-----+-----+
|      1992|          7|
|      1994|          1|
|      1993|          7|
|      1990|          8|
|      1991|          9|
+-----+-----+
```

2) What is the total air distance covered by each user per year

```
import org.apache.spark.sql.functions._
df: org.apache.spark.sql.DataFrame = [user_id: string, src: string ... 4 more fields]

println(s"The total air distance covered by each user per year:-")
df.groupBy("user_id","year_of_travel").agg(sum("distance") as "total distance").show()
```

The total air distance covered by each user per year:-

```
+-----+-----+-----+
|user_id|year_of_travel|total distance|
+-----+-----+-----+
|    1|      1990|      200.0|
|   10|      1992|      200.0|
```

	7	1990	600.0
	6	1993	200.0
	10	1990	200.0
	8	1990	200.0
	8	1991	200.0
	3	1991	200.0
	1	1993	600.0
	9	1992	400.0
	2	1991	400.0
	3	1993	200.0
	4	1990	400.0
	2	1993	200.0
	4	1991	200.0
	10	1993	200.0
	9	1991	200.0
	8	1992	200.0
	3	1992	200.0
	5	1992	400.0

+-----+-----+-----+

only showing top 20 rows

3) Which user has travelled the largest distance till date

```
println(s"The below user has travelled the largest distance till date :-")
```

```
df.groupBy("user_id").agg(sum("distance") as "total distance").show()
```

The below user has travelled the largest distance till date :-

+-----+-----+

user_id total distance
------------------------

+-----+-----+

	7	600.0
--	---	-------

	3	600.0
	8	600.0
	5	800.0
	6	600.0
	9	600.0
	1	800.0
	10	600.0
	4	600.0
	2	600.0
+-----+-----+		

4) What is the most preferred destination for all users.

```
println(s"The most preferred destination for all users :-")
df.groupBy("dest").agg(count("dest") as "preferred destination").show()
```

The most preferred destination for all users :-

+----+-----+	
dest	preferred destination
+----+-----+	
AUS	5
PAK	5
RUS	6
IND	9
CHN	7
+----+-----+	