SPARK ASSIGNMENT 20.2

==

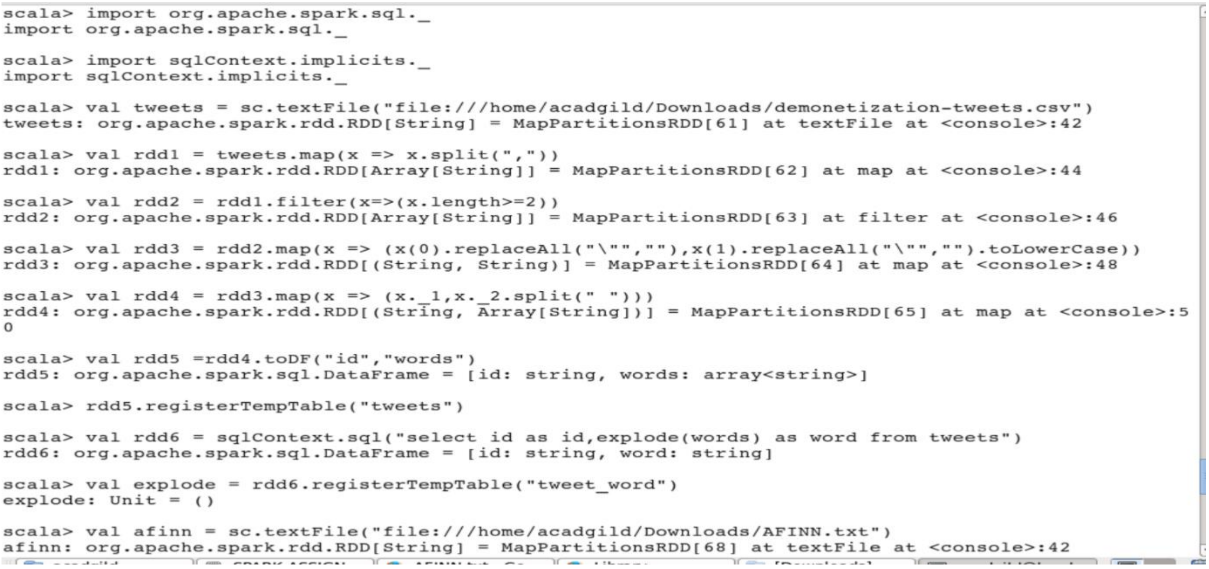
Let us find out the views of different people on the demonetization by analysing the tweets from twitter. Here is the dataset where twitter tweets are gathered in CSV format. You can download the dataset from the below link

https://drive.google.com/open?id=0ByJLBTmJojjzNkRsZWJiY1VGc28

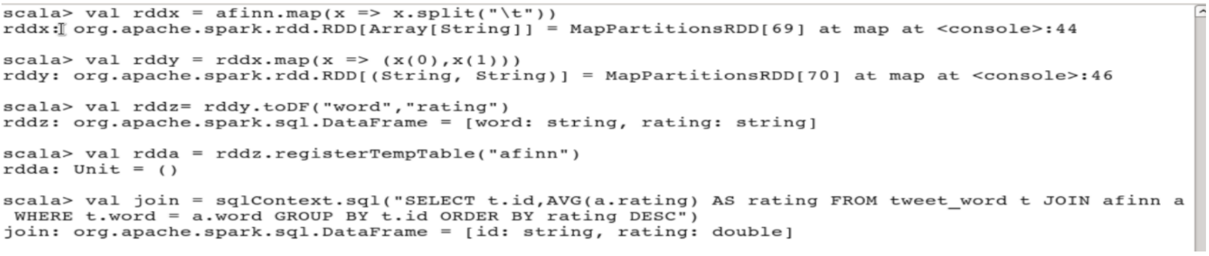
```
//Importing the spark packages and sqlContext Package to use dataframe

import org.apache.spark.sql._

import sqlContext.implicits._

//Reading the CSV file into an RDD from local filesystem with the help of SPARK CONTEXT OBJECT sc

val tweets = sc.textFile("file:///home/acadgild/Downloads/demonetization-tweets.csv")

//Mapping the fields delimited by comma

val rdd1 = tweets.map(x => x.split(","))

//filtering only those fields which are having length more than one

val rdd2 = rdd1.filter(x=>(x.length>=2))

//removing the double quotes from first field

//removing the double quotes from second field and converting it to lowercase

val rdd3 = rdd2.map(x => (x(0).replaceAll("\"",""),x(1).replaceAll("\"","").toLowerCase))

//Splitting the second part of key delimited by space

val rdd4 = rdd3.map(x => (x._1,x._2.split(" ")))

//Conversion of RDD into Spark Dataframe into columns id and words.

val rdd5df =rdd4.toDF("id","words")

//Create a temporary table tweets from dataframe

rdd5df.registerTempTable("tweets")

//Running select query on tweets table

val rdd6 = sqlContext.sql("select id as id,explode(words) as word from tweets")

//Saving the data to temporary new table

val explode = rdd6.registerTempTable("tweet_word")
```

```
//Reading the text file from local filesystem into SPARK RDD via SPARK CONTEXT object sc

val afinn = sc.textFile("file:///home/acadgild/Downloads/AFINN.txt")

//mapping the text file data delimited by tab

val rddx = afinn.map(x => x.split("\t"))

//Mapping the data into two columns delimited by comma

val rddy = rddx.map(x => (x(0),x(1)))

//Reading an RDD  into SPARK Dataframe and naming the columns of Dataframe

val rddzdf= rddy.toDF("word","rating")

//Creating a temporary table of DATAFRAME

val rdda = rddzdf.registerTempTable("afinn")

//Writing a join operation on two temporary tables created from Dataframes

val join = sqlContext.sql("SELECT t.id,AVG(a.rating) AS rating FROM tweet_word t JOIN afinn a WHERE t.word = a.word GROUP BY t.id ORDER BY rating DESC")

//Display the output of Join in SPARK SQL operation

join.show()

//Saving the output to csv file at local file system where columns are separated by underscore

join.map(_.mkString("_")).saveAsTextFile("file:/home/acadgild/Downloads/Assgn202.txt")

//Display the entire output of join operation

join.foreach(println)
```

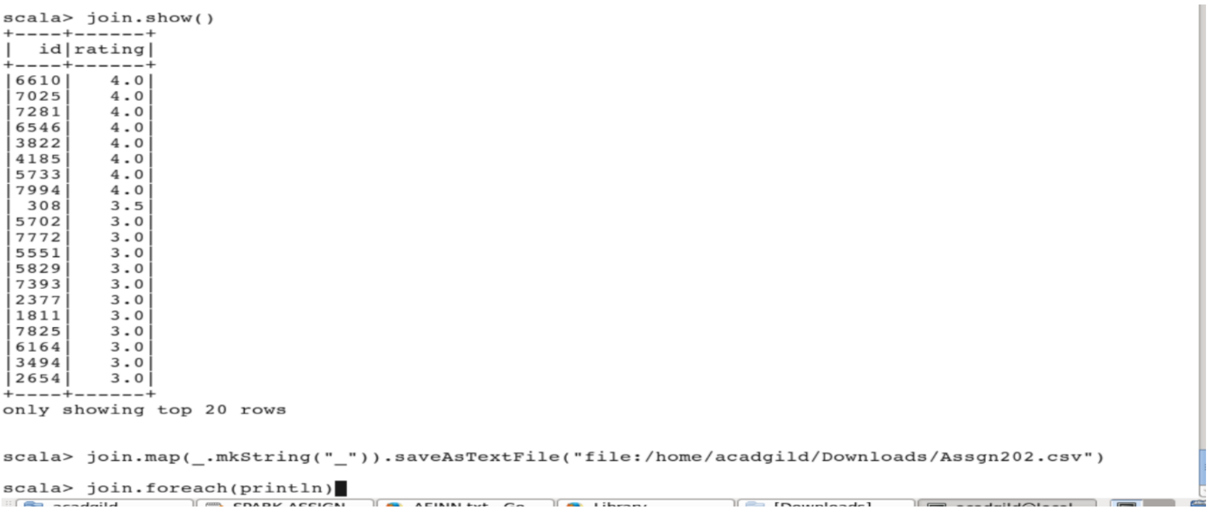
Screenshots:



```
scala> import org.apache.spark.sql._
import org.apache.spark.sql._

scala> import sqlContext.implicits._
import sqlContext.implicits._

scala> val tweets = sc.textFile("file:///home/acadgild/Downloads/demonetization-tweets.csv")
tweets: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[61] at textFile at <console>:42

scala> val rdd1 = tweets.map(x => x.split(","))
rdd1: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[62] at map at <console>:44

scala> val rdd2 = rdd1.filter(x=>(x.length>=2))
rdd2: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[63] at filter at <console>:46

scala> val rdd3 = rdd2.map(x => (x(0).replaceAll("\"",""),x(1).replaceAll("\"","").toLowerCase))
rdd3: org.apache.spark.rdd.RDD[(String, String)] = MapPartitionsRDD[64] at map at <console>:48

scala> val rdd4 = rdd3.map(x => (x._1,x._2.split(" ")))
rdd4: org.apache.spark.rdd.RDD[(String, Array[String])] = MapPartitionsRDD[65] at map at <console>:5
0

scala> val rdd5 =rdd4.toDF("id","words")
rdd5: org.apache.spark.sql.DataFrame = [id: string, words: array<string>]

scala> rdd5.registerTempTable("tweets")

scala> val rdd6 = sqlContext.sql("select id as id,explode(words) as word from tweets")
rdd6: org.apache.spark.sql.DataFrame = [id: string, word: string]

scala> val explode = rdd6.registerTempTable("tweet_word")
explode: Unit = ()

scala> val afinn = sc.textFile("file:///home/acadgild/Downloads/AFINN.txt")
afinn: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[68] at textFile at <console>:42
```



```
scala> val rddx = afinn.map(x => x.split("\t"))
rddx: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[69] at map at <console>:44

scala> val rddy = rddx.map(x => (x(0),x(1)))
rddy: org.apache.spark.rdd.RDD[(String, String)] = MapPartitionsRDD[70] at map at <console>:46

scala> val rddz= rddy.toDF("word","rating")
rddz: org.apache.spark.sql.DataFrame = [word: string, rating: string]

scala> val rdda = rddz.registerTempTable("afinn")
rdda: Unit = ()

scala> val join = sqlContext.sql("SELECT t.id,AVG(a.rating) AS rating FROM tweet_word t JOIN afinn a
 WHERE t.word = a.word GROUP BY t.id ORDER BY rating DESC")
join: org.apache.spark.sql.DataFrame = [id: string, rating: double]
```



```
scala> join.show()
+----+------+
|  id|rating|
+----+------+
|6610|   4.0|
|7025|   4.0|
|7281|   4.0|
|6546|   4.0|
|3822|   4.0|
|4185|   4.0|
|5733|   4.0|
|7994|   4.0|
| 308|   3.5|
|5702|   3.0|
|7772|   3.0|
|5551|   3.0|
|5829|   3.0|
|7393|   3.0|
|2377|   3.0|
|1811|   3.0|
|7825|   3.0|
|6164|   3.0|
|3494|   3.0|
|2654|   3.0|
+----+------+
only showing top 20 rows


scala> join.map(_.mkString("_")).saveAsTextFile("file:/home/acadgild/Downloads/Assgn202.csv")

scala> join.foreach(println)
```