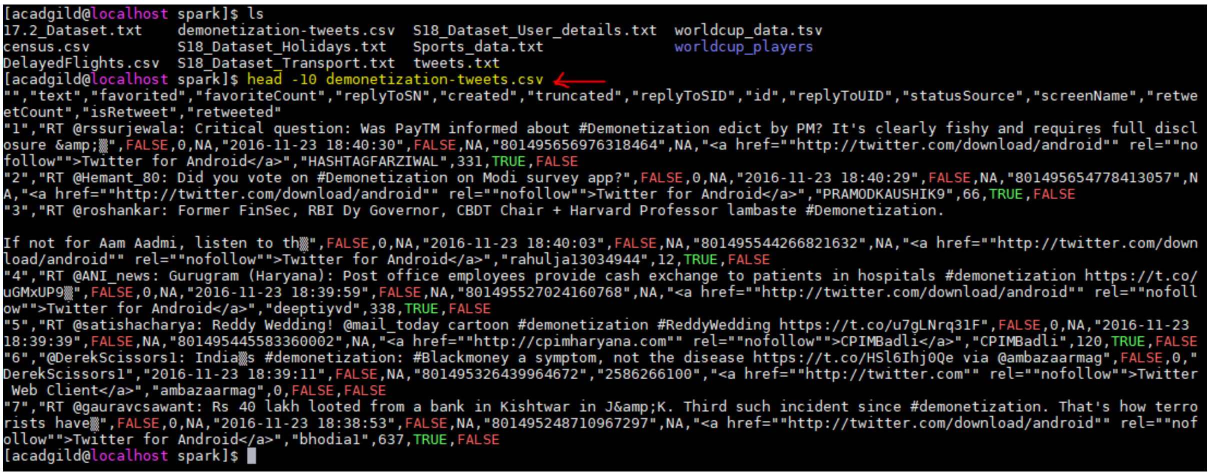
Assignment 22.2:

Problem Statement:

Downloaded the dataset demonetization-tweets.csv and AFFIN.txt and loaded in the local file

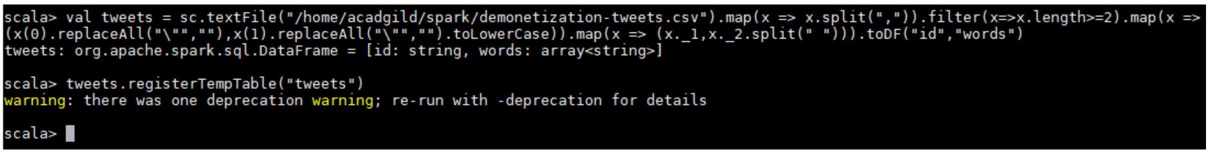


```
[acadgild@localhost spark]$ ls
17.2_Dataset.txt    demonetization-tweets.csv  S18_Dataset_User_details.txt  worldcup_data.tsv
census.csv          S18_Dataset_Holidays.txt   Sports_data.txt               worldcup_players
DelayedFlights.csv  S18_Dataset_Transport.txt  tweets.txt
[acadgild@localhost spark]$ head -10 demonetization-tweets.csv
"","text","favorited","favoriteCount","replyToSN","created","truncated","replyToSID","id","replyToUID","statusSource","screenName","retwe
etCount","isRetweet","retweeted"
"1","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full discl
osure &amp;▒",FALSE,0,NA,"2016-11-23 18:40:30",FALSE,NA,"801495656976318464",NA,"<a href=""http://twitter.com/download/android"" rel=""no
follow"">Twitter for Android</a>","HASHTAGFARZIWAL",331,TRUE,FALSE
"2","RT @Hemant_80: Did you vote on #Demonetization on Modi survey app?",FALSE,0,NA,"2016-11-23 18:40:29",FALSE,NA,"801495654778413057",N
A,"<a href=""http://twitter.com/download/android"" rel=""nofollow"">Twitter for Android</a>","PRAMODKAUSHIK9",66,TRUE,FALSE
"3","RT @roshankar: Former FinSec, RBI Dy Governor, CBDT Chair + Harvard Professor lambaste #Demonetization.

If not for Aam Aadmi, listen to th▒",FALSE,0,NA,"2016-11-23 18:40:03",FALSE,NA,"801495544266821632",NA,"<a href=""http://twitter.com/down
load/android"" rel=""nofollow"">Twitter for Android</a>","rahulja13034944",12,TRUE,FALSE
"4","RT @ANI_news: Gurugram (Haryana): Post office employees provide cash exchange to patients in hospitals #demonetization https://t.co/
uGMxUP9▒",FALSE,0,NA,"2016-11-23 18:39:59",FALSE,NA,"801495527024160768",NA,"<a href=""http://twitter.com/download/android"" rel=""nofoll
ow"">Twitter for Android</a>","deeptiyvd",338,TRUE,FALSE
"5","RT @satishacharya: Reddy Wedding! @mail_today cartoon #demonetization #ReddyWedding https://t.co/u7gLNrq31F",FALSE,0,NA,"2016-11-23
18:39:39",FALSE,NA,"801495445583360002",NA,"<a href=""http://cpimharyana.com"" rel=""nofollow"">CPIMBadli</a>","CPIMBadli",120,TRUE,FALSE
"6","@DerekScissors1: India▒s #demonetization: #Blackmoney a symptom, not the disease https://t.co/HSl6Ihj0Qe via @ambazaarmag",FALSE,0,"
DerekScissors1","2016-11-23 18:39:11",FALSE,NA,"801495326439964672","2586266100","<a href=""http://twitter.com"" rel=""nofollow"">Twitter
 Web Client</a>","ambazaarmag",0,FALSE,FALSE
"7","RT @gauravcsawant: Rs 40 lakh looted from a bank in Kishtwar in J&amp;K. Third such incident since #demonetization. That's how terro
rists have▒",FALSE,0,NA,"2016-11-23 18:38:53",FALSE,NA,"801495248710967297",NA,"<a href=""http://twitter.com/download/android"" rel=""nof
ollow"">Twitter for Android</a>","bhodial",637,TRUE,FALSE
[acadgild@localhost spark]$ 
```

1. First we will read the csv file and then split the columns to get and create a DataFrame.

Also, we will create a temporary table named tweets

val tweets = sc.textFile("/home/acadgild/sumona/demonetization-tweets.csv").map(x => x.split(",")).filter(x=>x.length>=2).map(x => (x(0).replaceAll("\"",""),x(1).replaceAll("\"","").toLowerCase)).map(x => (x._1,x._2.split(" "))).toDF("id","words").registerTempTable("tweets")

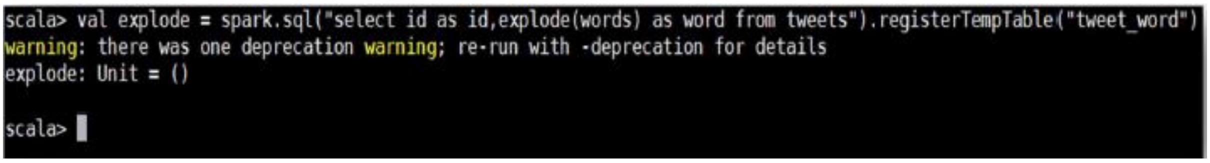


```
scala> val tweets = sc.textFile("/home/acadgild/spark/demonetization-tweets.csv").map(x => x.split(",")).filter(x=>x.length>=2).map(x =>
(x(0).replaceAll("\"",""),x(1).replaceAll("\"","").toLowerCase)).map(x => (x._1,x._2.split(" "))).toDF("id","words")
tweets: org.apache.spark.sql.DataFrame = [id: string, words: array<string>]

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

scala> 
```

2. Now from the above temporary table we will select the ID, words and form another

temporary table tweet_word

val explode = spark.sql("select id as id,explode(words) as word from tweets").registerTempTable("tweet_word")



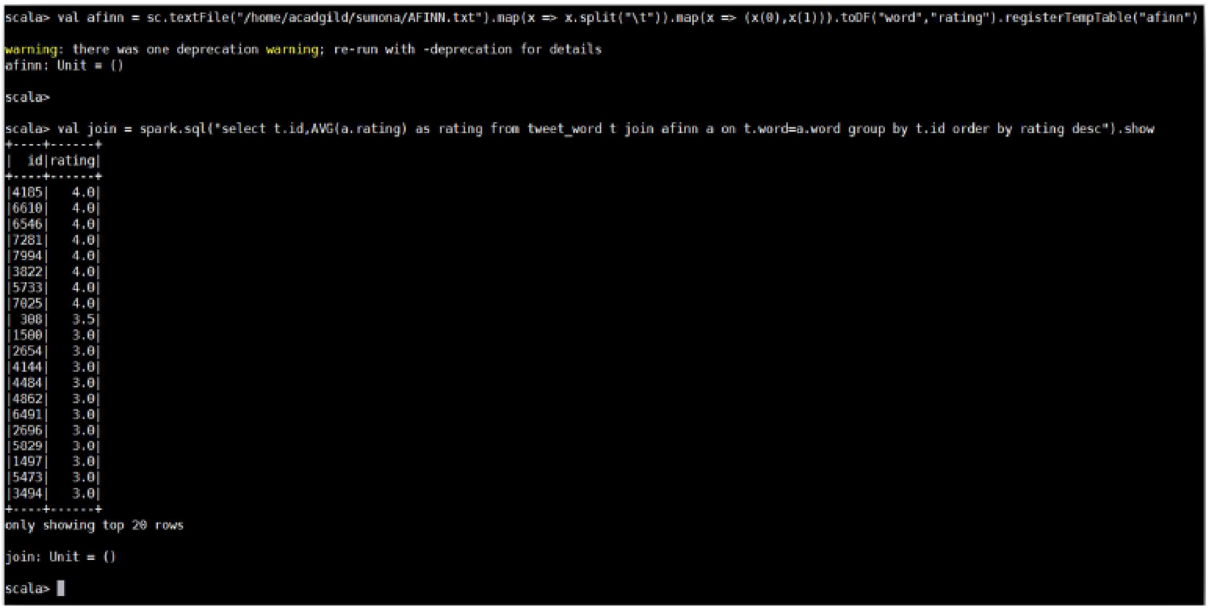
```
scala> val explode = spark.sql("select id as id,explode(words) as word from tweets").registerTempTable("tweet_word")
warning: there was one deprecation warning; re-run with -deprecation for details
explode: Unit = ()

scala> 
```

3. Here we will read the AFFIN file and create a temporary table affin

val afinn = sc.textFile("/home/acadgild/sumona/AFINN.txt").map(x => x.split("\t")).map(x => (x(0),x(1))).toDF("word","rating").registerTempTable("afinn")

Then we will join both the tables tweet_word and affin and get the views of different people on demonetization

val join = spark.sql("select t.id,AVG(a.rating) as rating from tweet_word t join afinn a on t.word=a.word group by t.id order by rating desc").show



```
scala> val afinn = sc.textFile("/home/acadgild/sumona/AFINN.txt").map(x => x.split("\t")).map(x => (x(0),x(1))).toDF("word","rating").registerTempTable("afinn")

warning: there was one deprecation warning; re-run with -deprecation for details
afinn: Unit = ()

scala>

scala> val join = spark.sql("select t.id,AVG(a.rating) as rating from tweet_word t join afinn a on t.word=a.word group by t.id order by rating desc").show
+----+------+
|  id|rating|
+----+------+
|4105|   4.0|
|6610|   4.0|
|6546|   4.0|
|7281|   4.0|
|7994|   4.0|
|3822|   4.0|
|5733|   4.0|
|7025|   4.0|
| 308|   3.5|
|1500|   3.0|
|2654|   3.0|
|4144|   3.0|
|4484|   3.0|
|4862|   3.0|
|6491|   3.0|
|2696|   3.0|
|5029|   3.0|
|1497|   3.0|
|5473|   3.0|
|3494|   3.0|
+----+------+
only showing top 20 rows

join: Unit = ()

scala>
```