**SESSION 22 ASSIGNMENT 1**

* **Importing APIs**

scala> import org.apache.spark.sql.Row

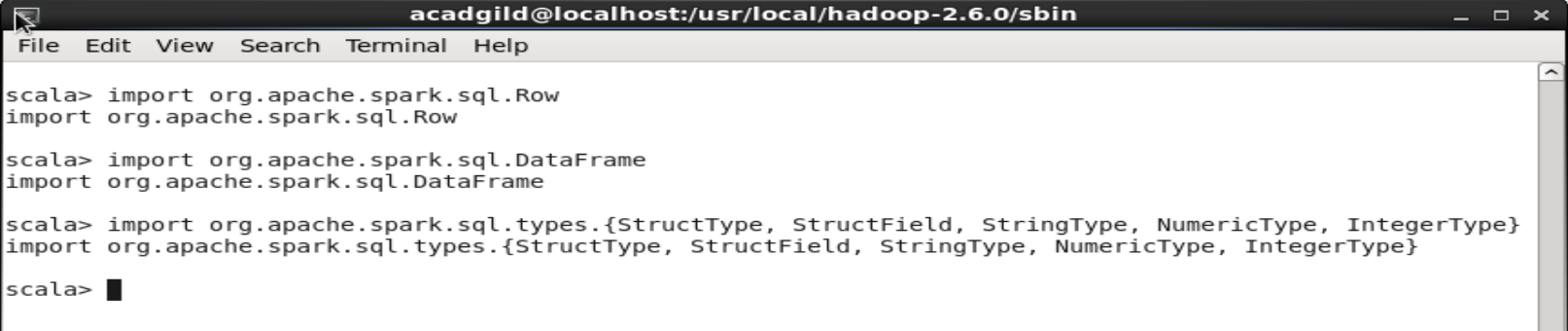
import org.apache.spark.sql.Row

scala> import org.apache.spark.sql.DataFrame

import org.apache.spark.sql.DataFrame

scala> import org.apache.spark.sql.types.{StructType, StructField, StringType, NumericType, IntegerType}

import org.apache.spark.sql.types.{StructType, StructField, StringType, NumericType, IntegerType}



* **Loading data and performing actions to get the desired result**

**// Loading tweets data separating with comma and creating data frame**

scala> val tweets = sc.textFile("file:///home/acadgild/Downloads/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>]

**// Creating the temp table for the above data**

scala> tweets.registerTempTable("tweets")

scala> val explode = sqlContext.sql("select id as id,explode(words) as word from tweets").registerTempTable("tweet\_word")

explode: Unit = ()

**// Loading the AFINN data and creating data frame**

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

afinn: Unit = ()

**// Writing a spark sql to get the id and avg rating for the tweets by joining tweets and afinn tables and ordering it by descending order of rating**

scala> val join = sqlContext.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

