**Spark**

Project 3

**Authored By**

Divya Jayaprakash (dxj160830)

BUAN 6346 – Big Data Analytics – Fall 2017

Prof. Kashif Saeed

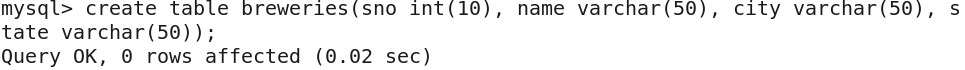
TA - Karthik

**1. Import the brewery.csv into Sqoop**

mysql -u root -p

use loudacre;

create table breweries(sno int(10), name varchar(50), city varchar(50), state varchar(50));



hdfs dfs -put Desktop/craft-cans/breweries.csv /breweries

sqoop export \

--connect jdbc:mysql://localhost/loudacre \

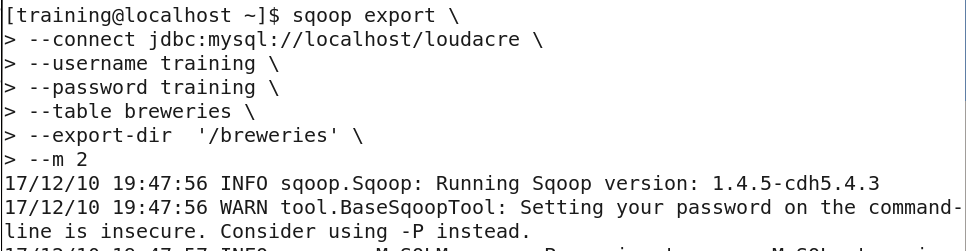
--username training \

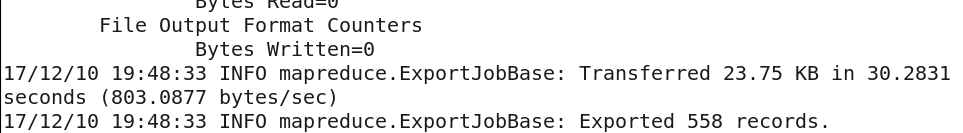
--password training \

--table breweries \

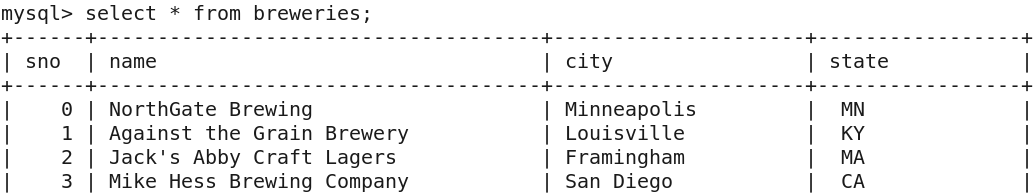
--export-dir '/breweries' \

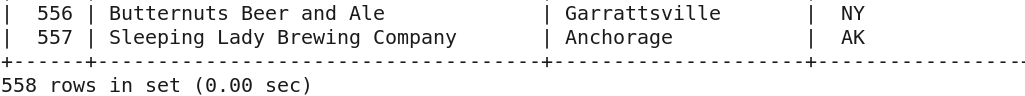
--m 2





Select \* from breweries;

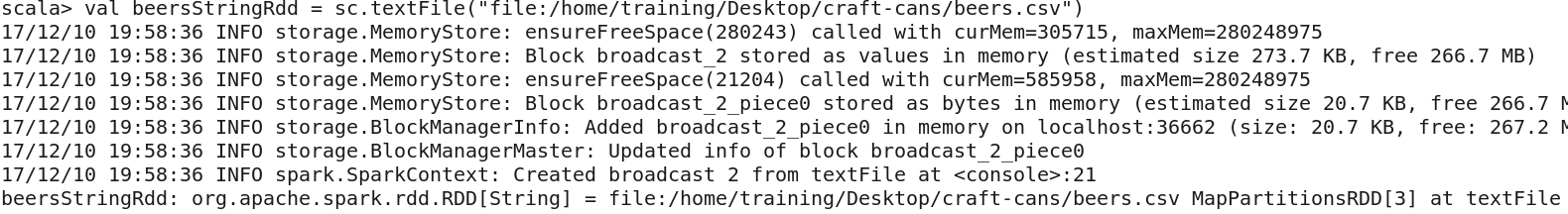




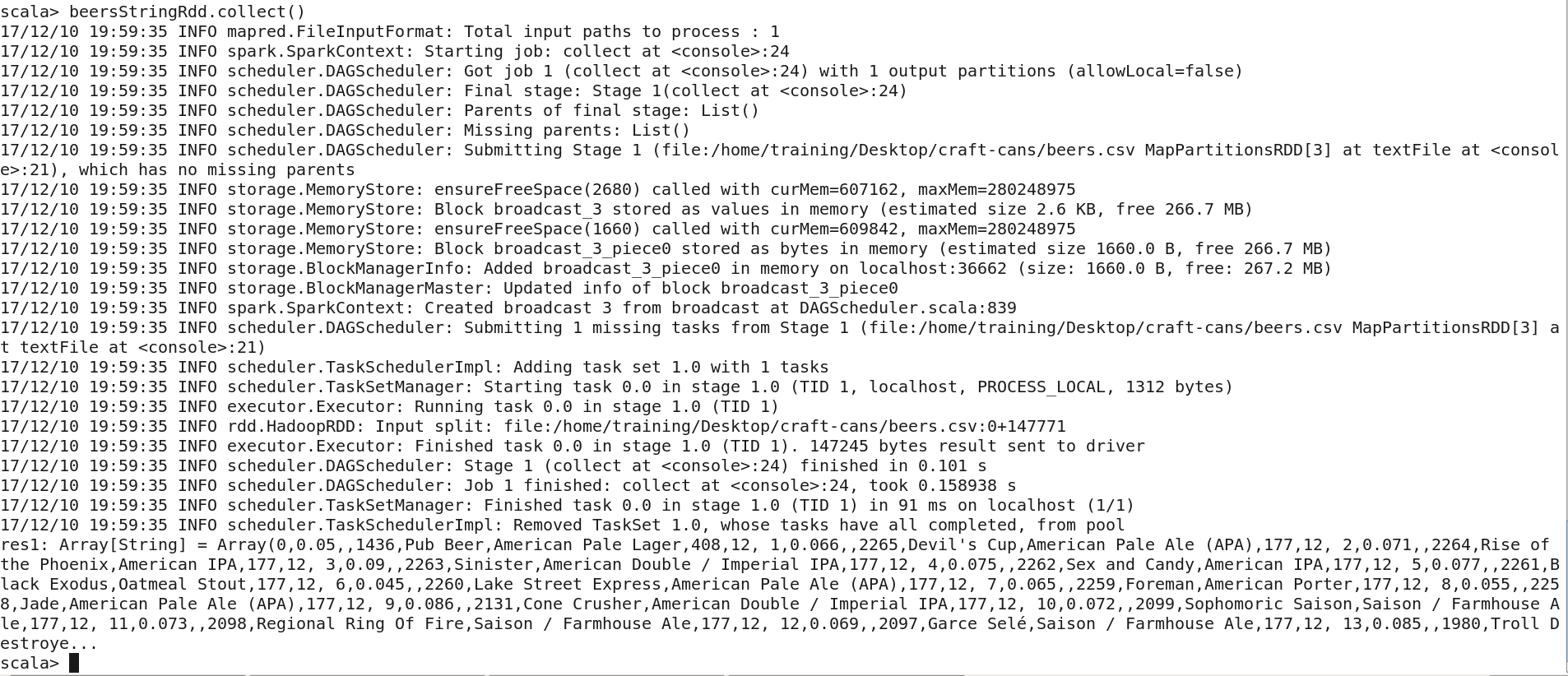
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**2. Create RDD for Beers.csv**

val beersStringRdd = sc.textFile("file:/home/training/Desktop/craft-cans/beers.csv")



beersStringRdd.collect()

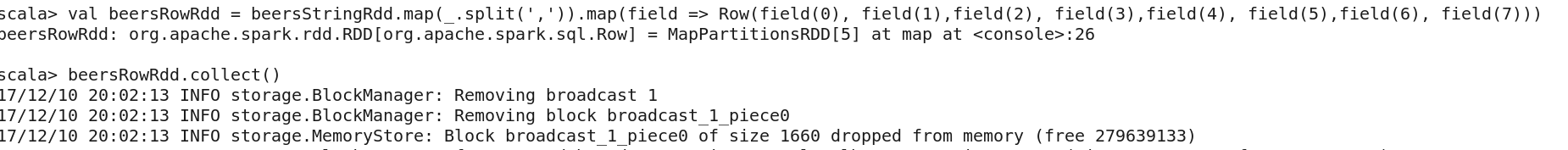


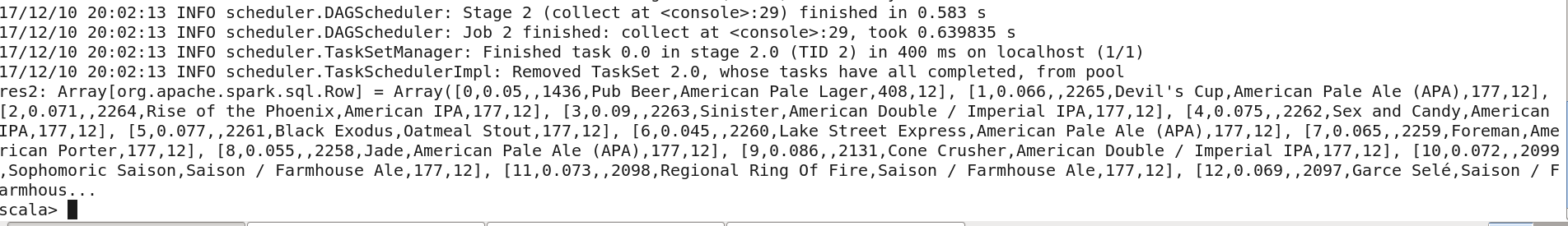
import org.apache.spark.sql.\_



val beersRowRdd = beersStringRdd.map(\_.split(',')).map(field => Row(field(0), field(1),field(2), field(3),field(4), field(5),field(6), field(7)))

beersRowRdd.collect()





Now trying to convert the RDD to Data frame in order to query on it

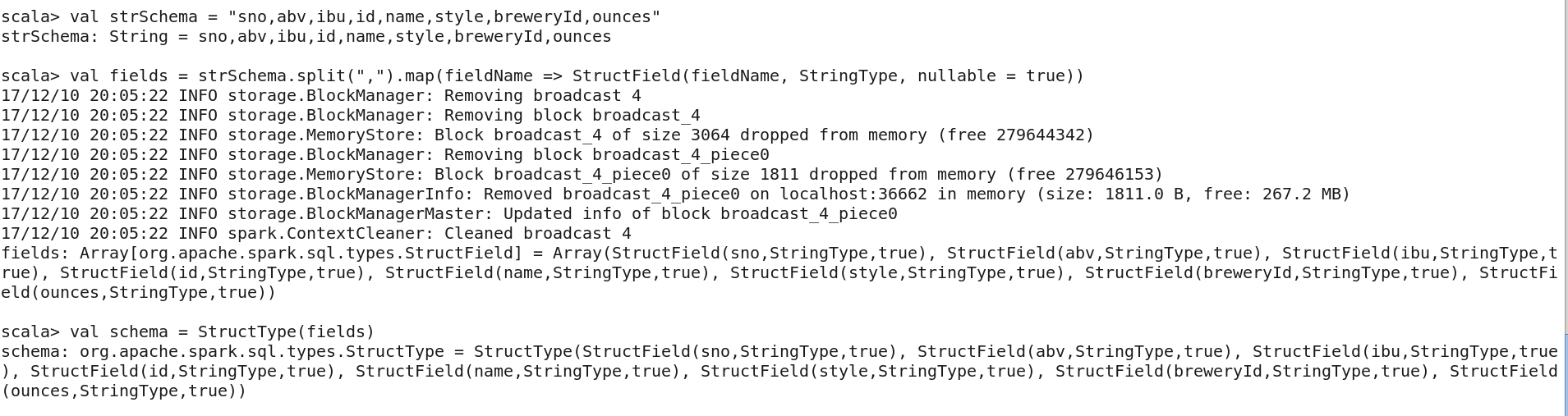
import org.apache.spark.sql.types.\_



val strSchema = "sno,abv,ibu,id,name,style,breweryId,ounces"

val fields = strSchema.split(",").map(fieldName => StructField(fieldName, StringType, nullable = true))

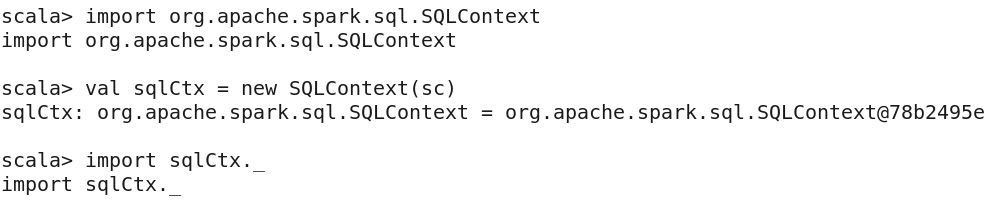
val schema = StructType(fields)



import org.apache.spark.sql.SQLContext

val sqlCtx = new SQLContext(sc)

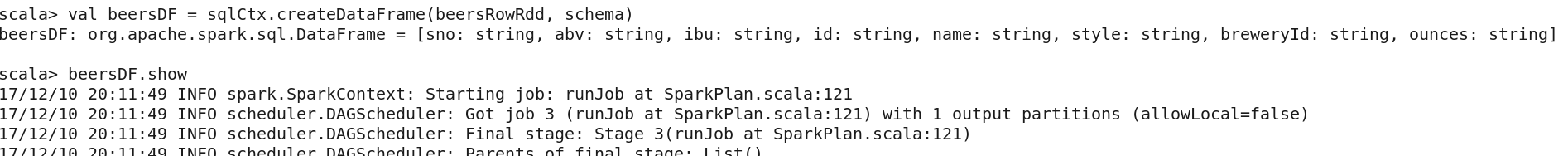
import sqlCtx.\_

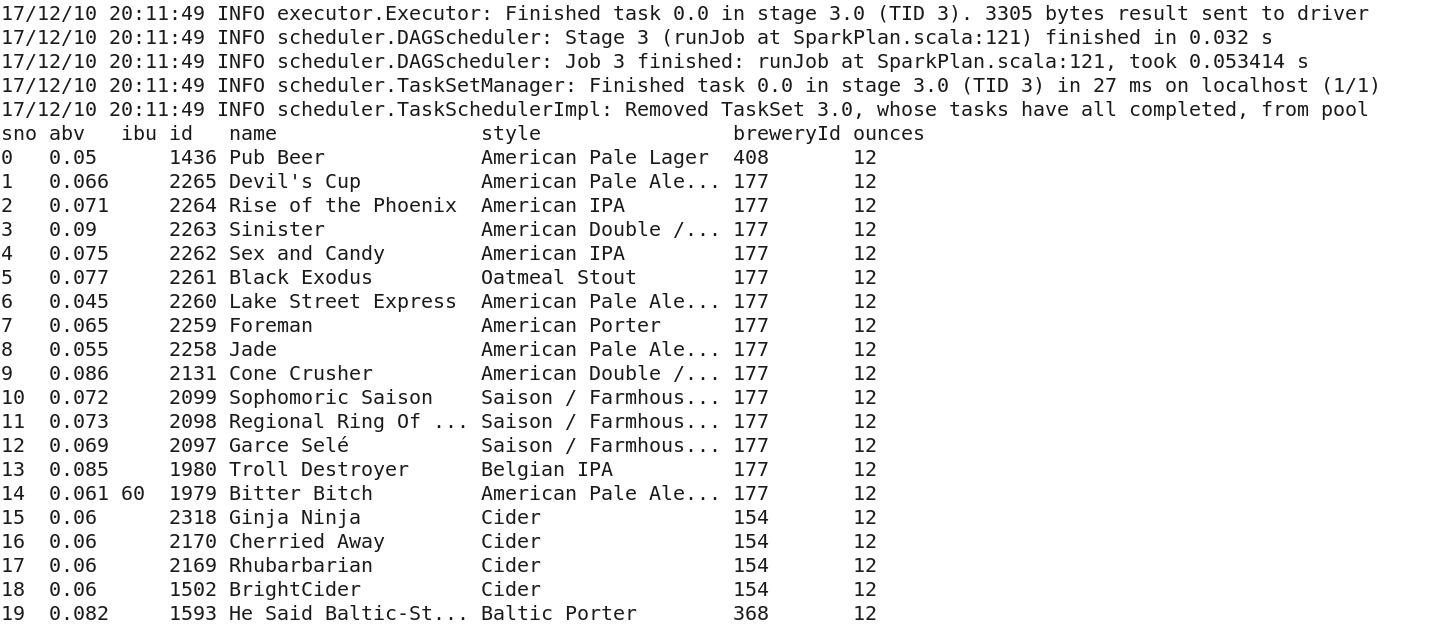


val beersDF = sqlCtx.createDataFrame(beersRowRdd, schema)

beersDF.show

beersDF.registerTempTable("beers")





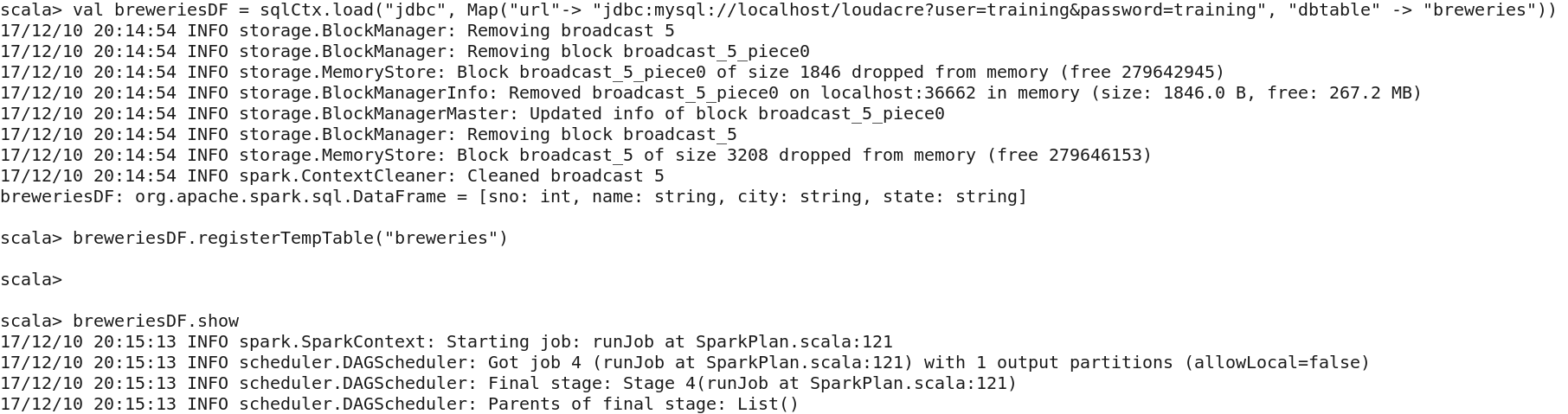
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**3. Move the brewery.csv into Spark from**

val breweriesDF = sqlCtx.load("jdbc", Map("url"-> "jdbc:mysql://localhost/loudacre?user=training&password=training", "dbtable" -> "breweries"))

breweriesDF.registerTempTable("breweries")

breweriesDF.show

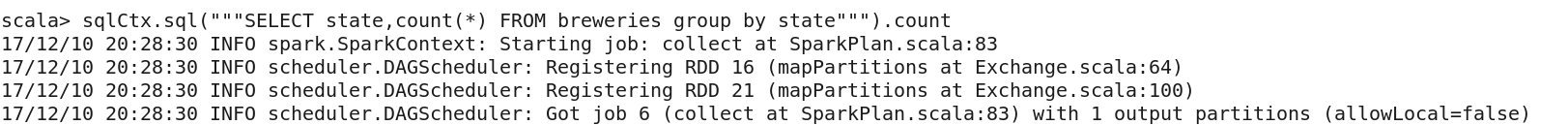


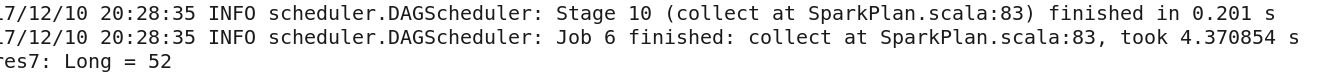
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***4.For the following questions use Spark SQL**

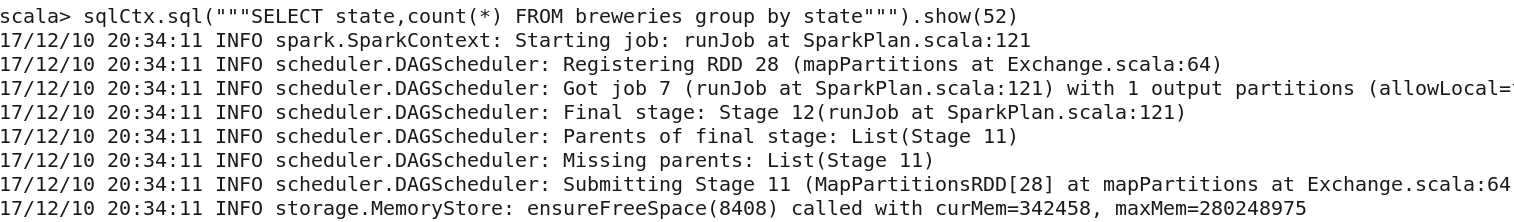
**a. Determine the number of breweries in each state**

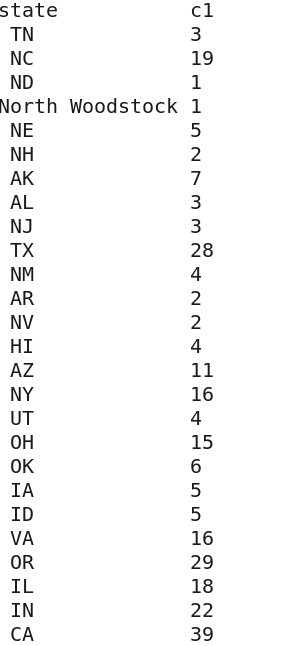
sqlCtx.sql("""SELECT state,count(\*) FROM breweries group by state""").count

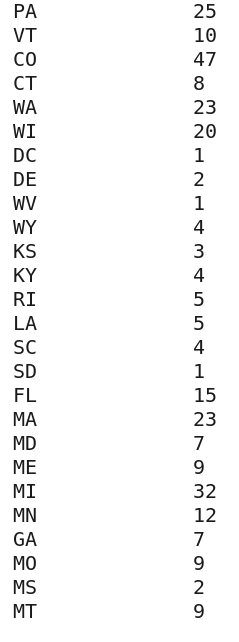
sqlCtx.sql("""SELECT state,count(\*) FROM breweries group by state""").show(52)









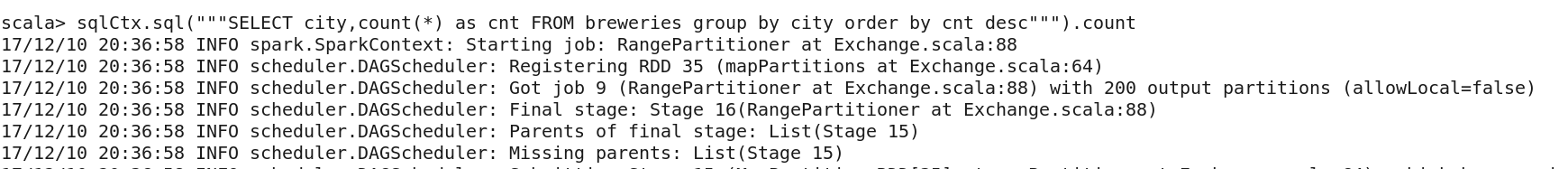


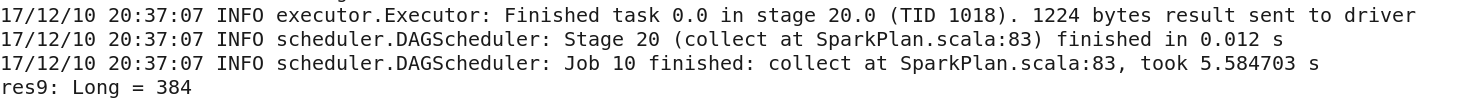
b. Determine the cities with most breweries

Portland has the highest of 17 breweries.

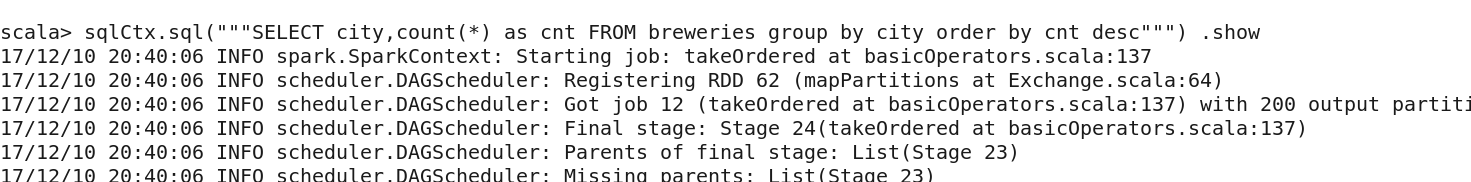
Next, Chicago, seattle, boulder have 9 breweries

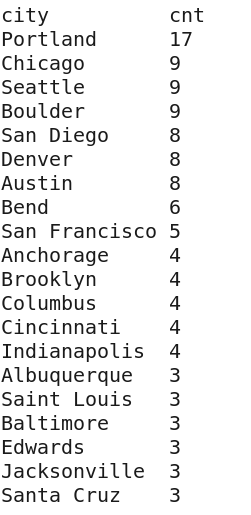
sqlCtx.sql("""SELECT city,count(\*) as cnt FROM breweries group by city order by cnt desc""") .count





sqlCtx.sql("""SELECT city,count(\*) as cnt FROM breweries group by city order by cnt desc""") .show





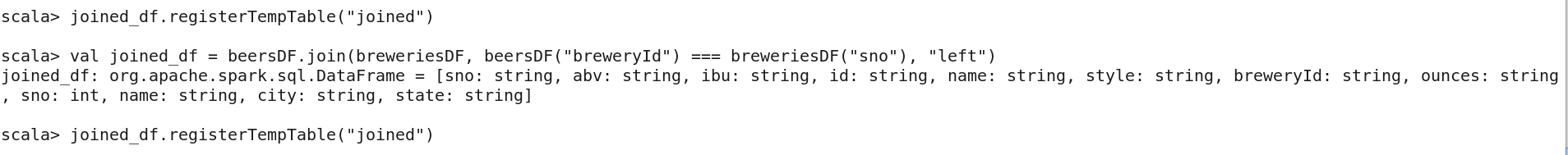
c. Calculate the average alcohol % by volume in each state

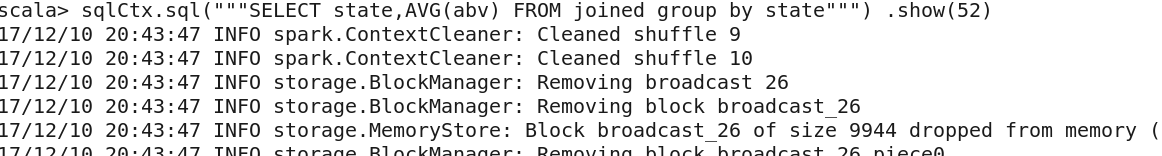
val joined\_df = beersDF.join(breweriesDF, beersDF("breweryId") === breweriesDF("sno"), "left")

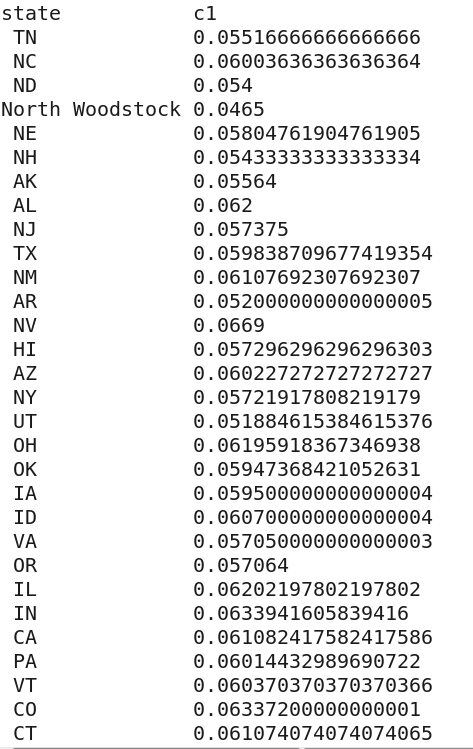
joined\_df.show

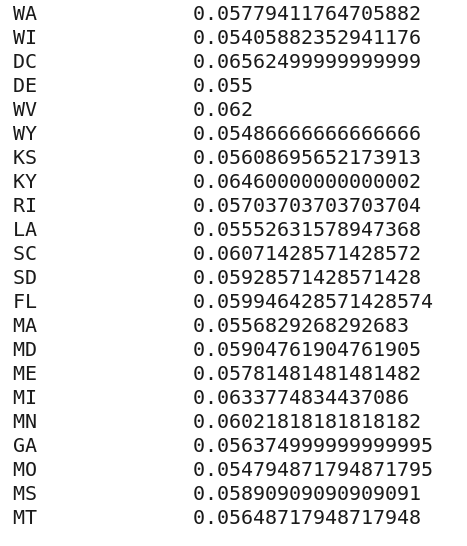
joined\_df.registerTempTable("joined")

sqlCtx.sql("""SELECT state,AVG(abv) FROM joined group by state""") .show(52)





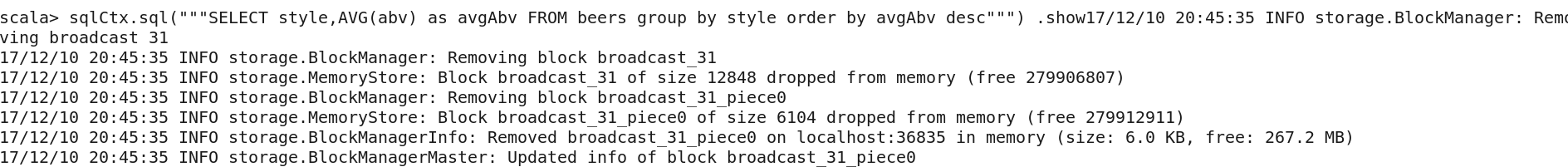


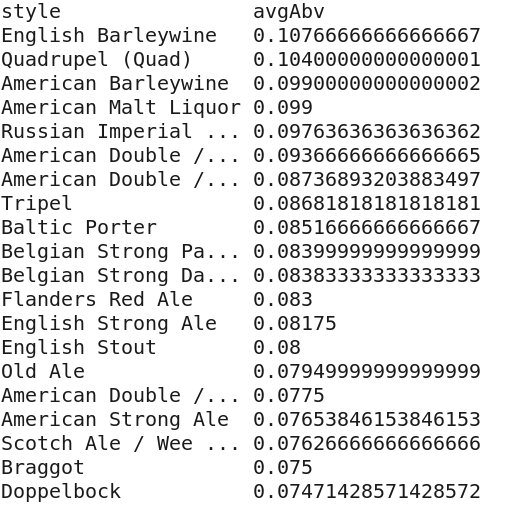


d. Determine the beer styles with highest average alcohol by volume

Ans: English Barleywine, quadrupel (quad), American Barleywine with 0.1077,0.104, 0.099 avg abv respectively

sqlCtx.sql("""SELECT style,AVG(abv) as avgAbv FROM beers group by style order by avgAbv desc""") .show





E.Determine the most brewed beer style

**The most brewed beer style is American IPA brewed for 424 beer brands**

sqlCtx.sql("""Select style,count(\*) as cnt from beers group by style order by cnt desc""").show(5)

