**Assignment 2 – Spark**

**Craft Brewery data**

Data Analysis using Spark Shell, language used is Scala on Craft Brewery Data

File required:

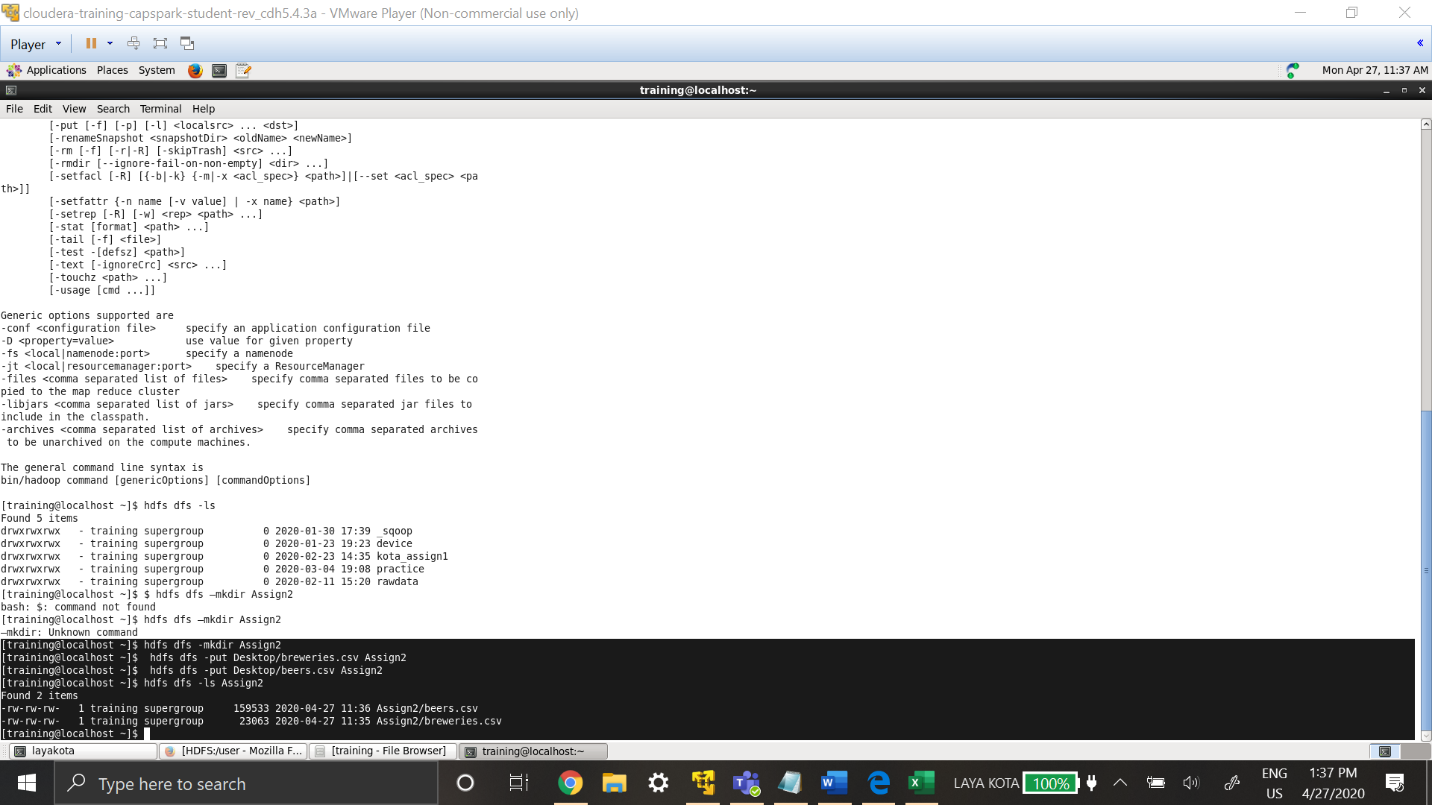
* Beers.csv
* Breweries.csv
* Dictionary.doc

**Step-1 :** Load data in HDFS

$ hdfs dfs –mkdir Assign2

$ hdfs dfs -put Desktop/breweries.csv Assign2

$ hdfs dfs -put Desktop/beers.csv Assign2



Dataset is stored in HDFS, these can be accessed directly from Spark-Shell.

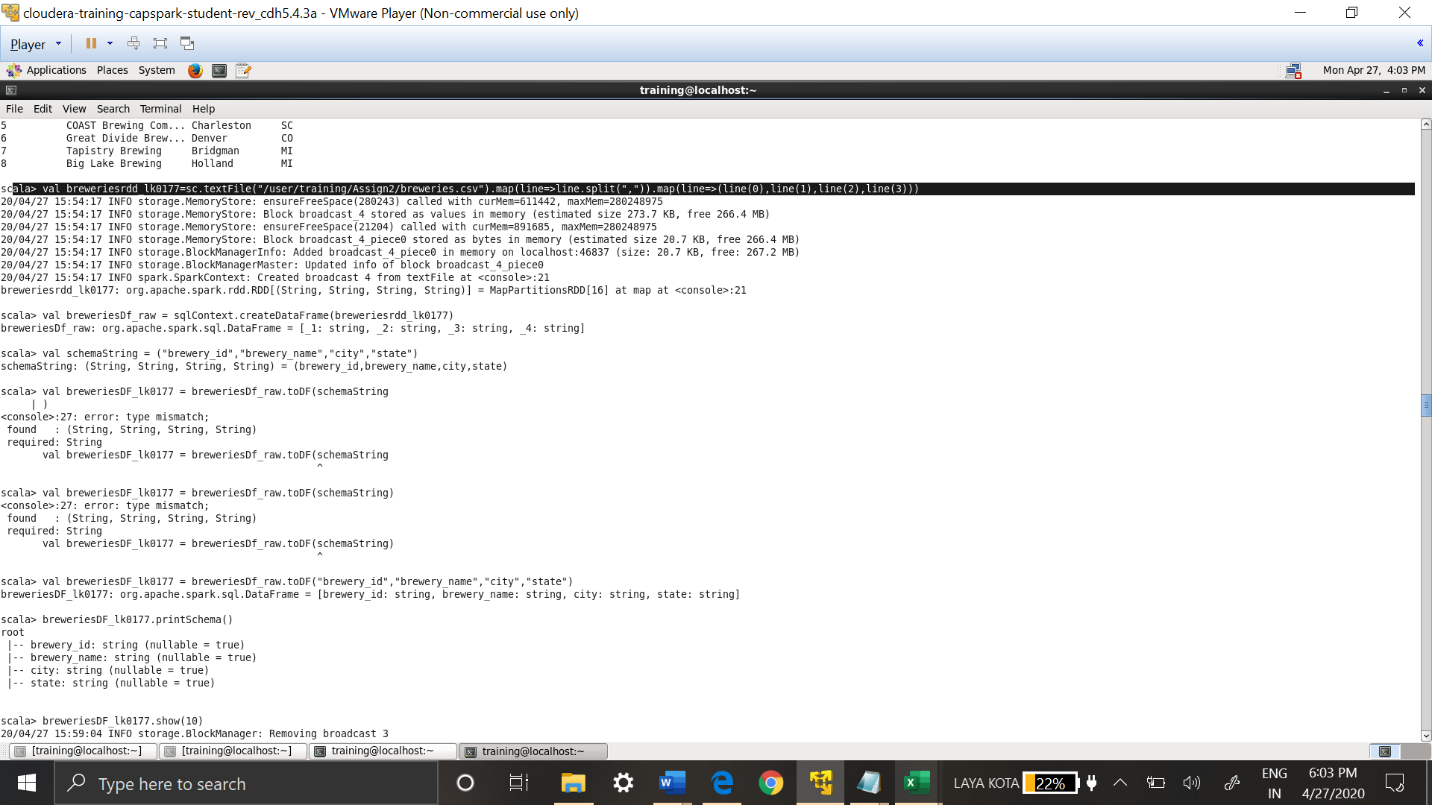
**Step-2 :** $ Spark-shell

**Step- 3:** Create RDD for Breweries.csv

Val breweriesrdd\_lk0177=sc.textFile("/user/training/Assign2/breweries.csv")

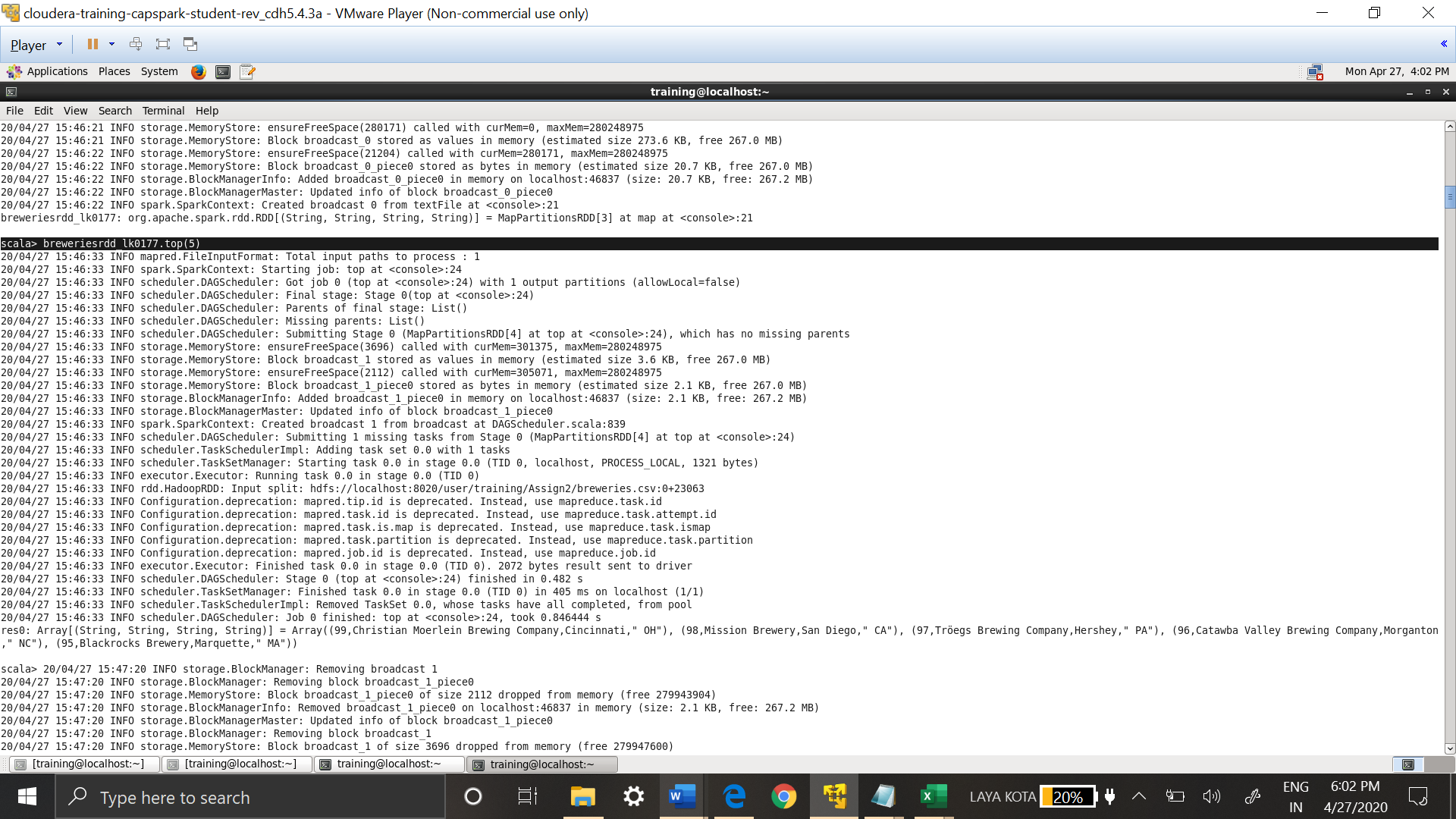
.map(line=>line.split(","))

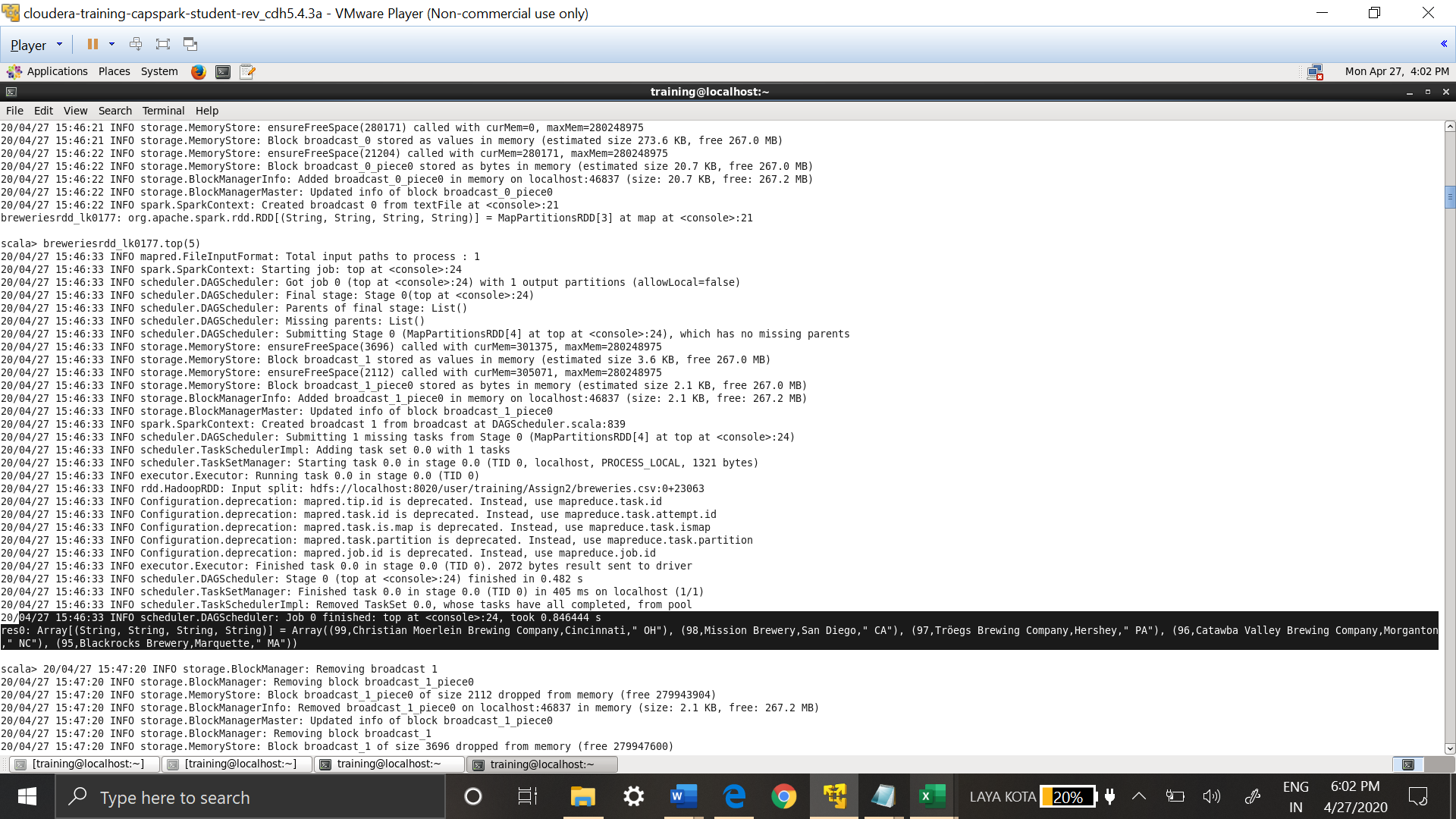
.map(line=>(line(0),line(1),line(2),line(3)))



RDD is created by accessing the file in HDFS, as it’s csv file the delimiter is ‘,’.Data in each row stored as it is with ‘,’.

**Step-4 :** Checking the data in RDD : breweriesrdd\_lk0177.top(5)

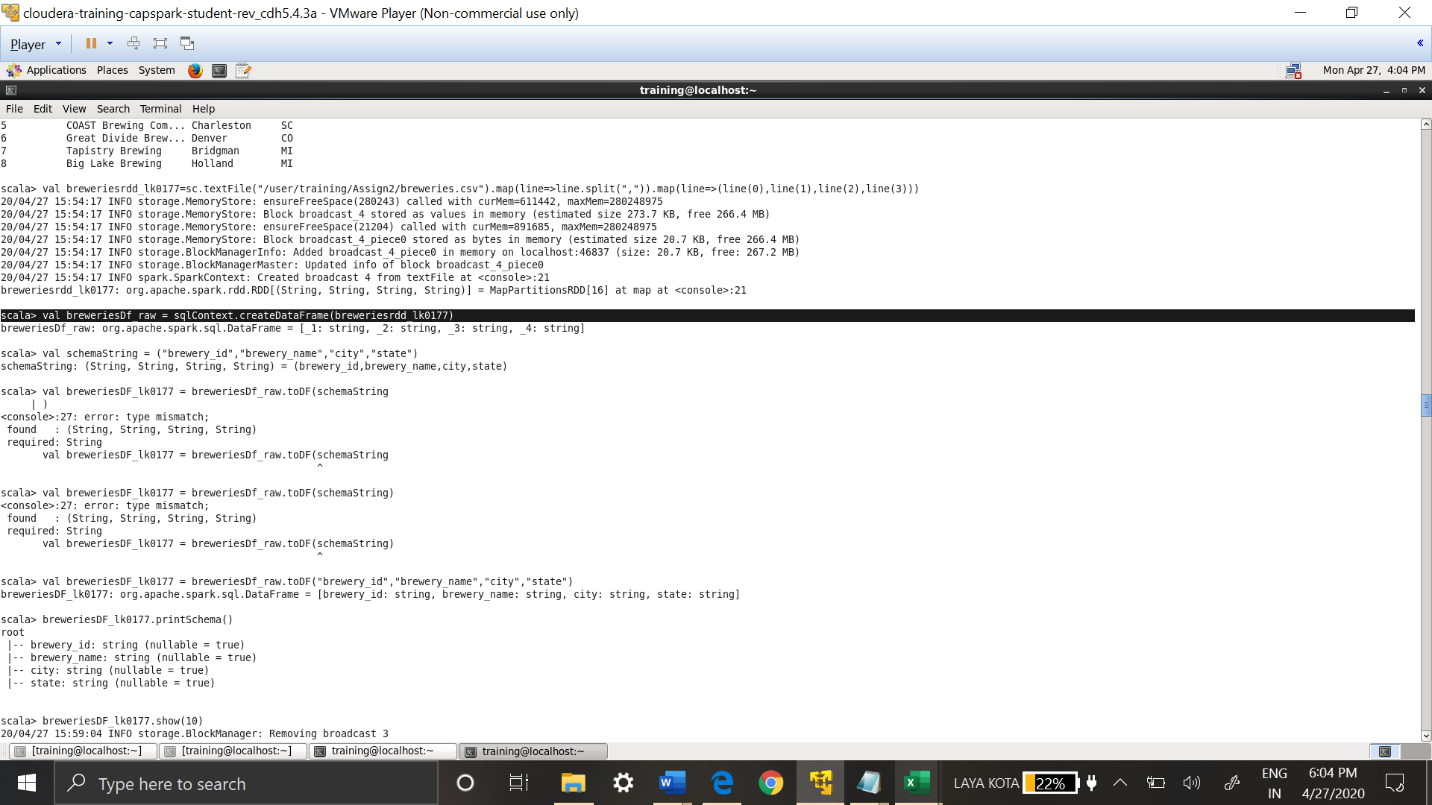




Data is stored as array, as mapped above row data is separated with ‘,’

**Step-5:** Creating DataFrame from RDD

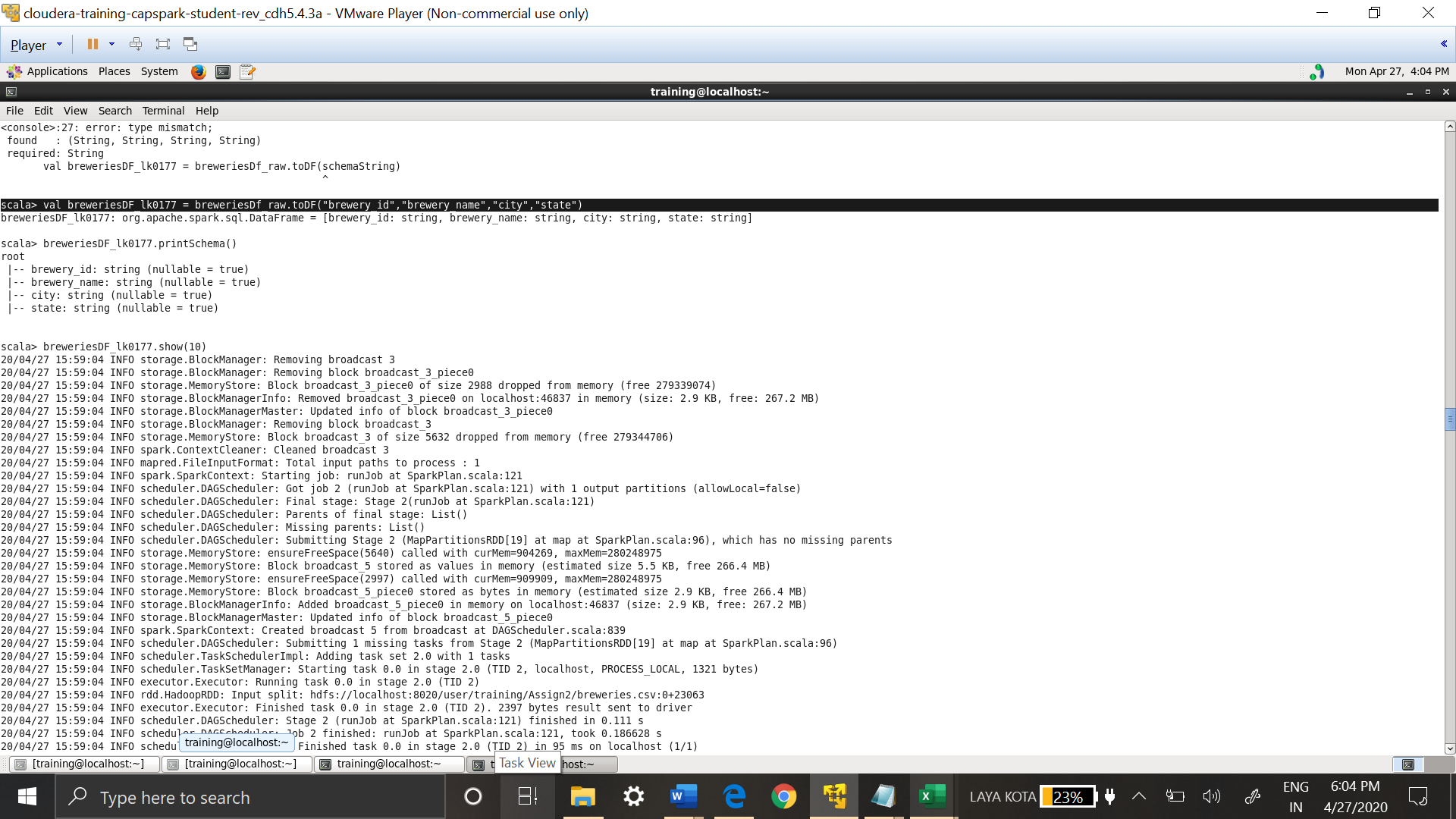
val breweriesDf\_raw = sqlContext.createDataFrame(breweriesrdd\_lk0177)



Sql Context class has method called **createDataFrame**, which takes RDD as input.

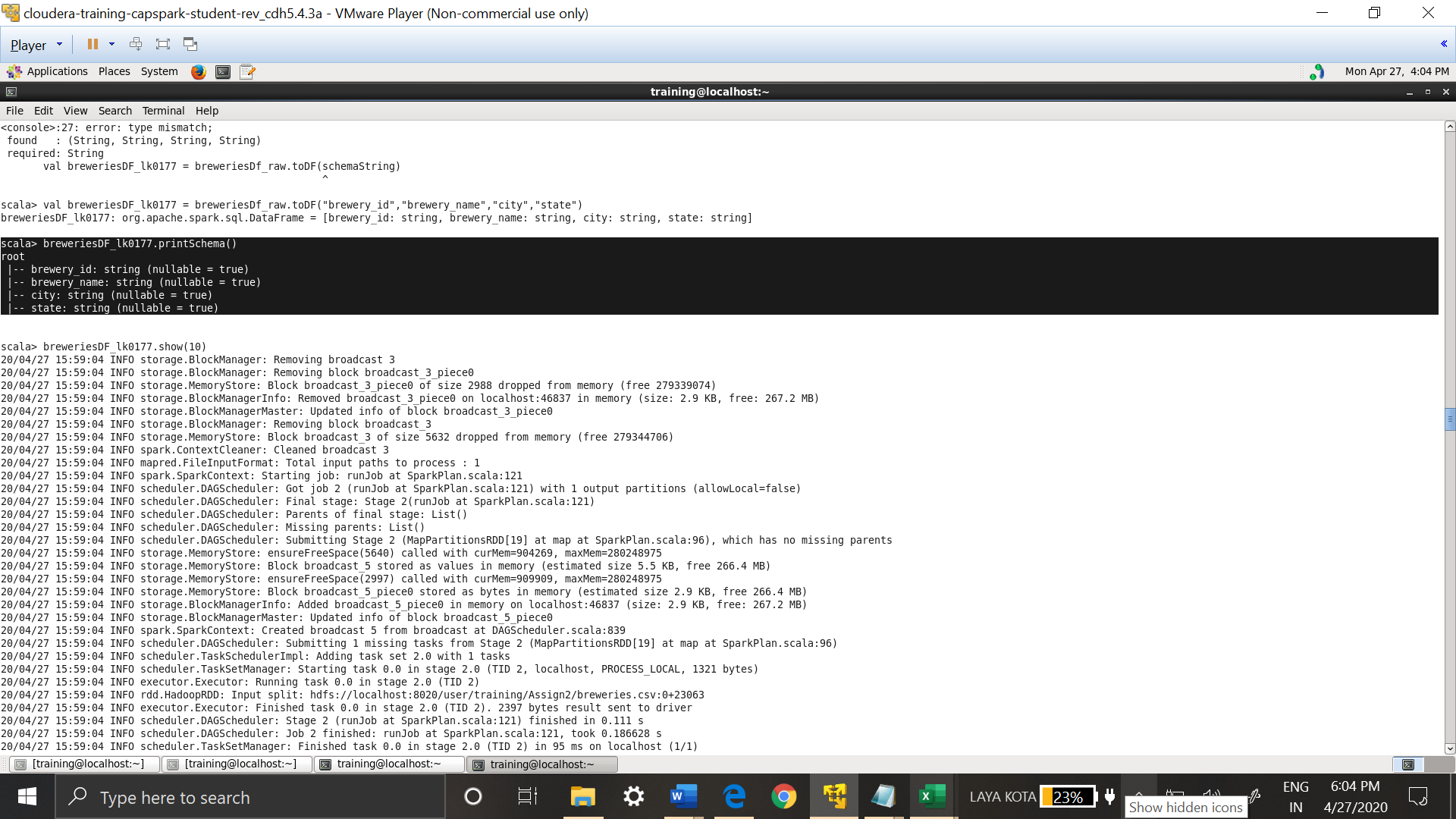
**Step-6 :** Schema is added to the Data Frame

val breweriesDF\_lk0177 = breweriesDf\_raw.toDF("brewery\_id","brewery\_name","city","state")



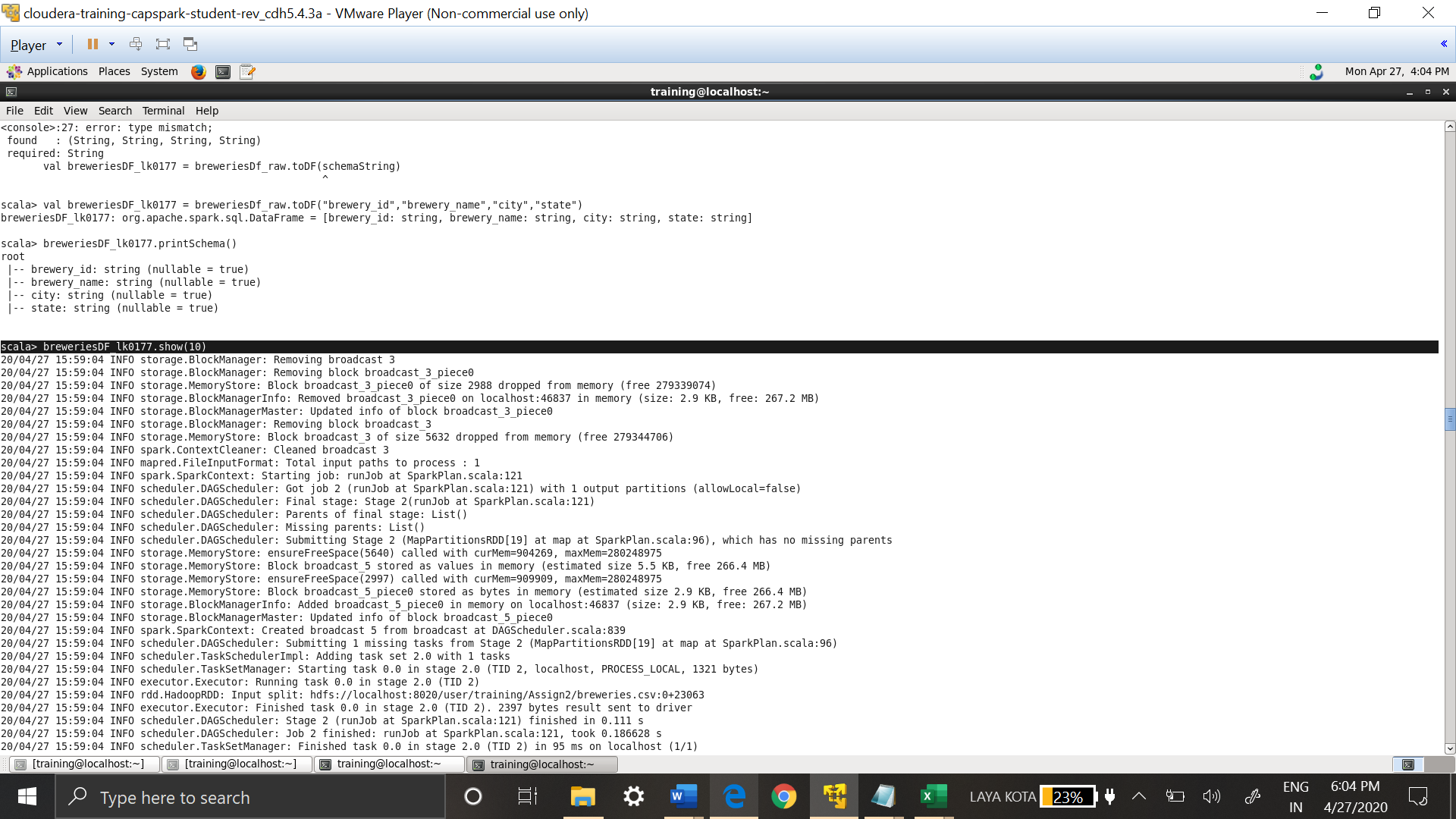
**Step-7:** Printing the Schema of the Data Frame

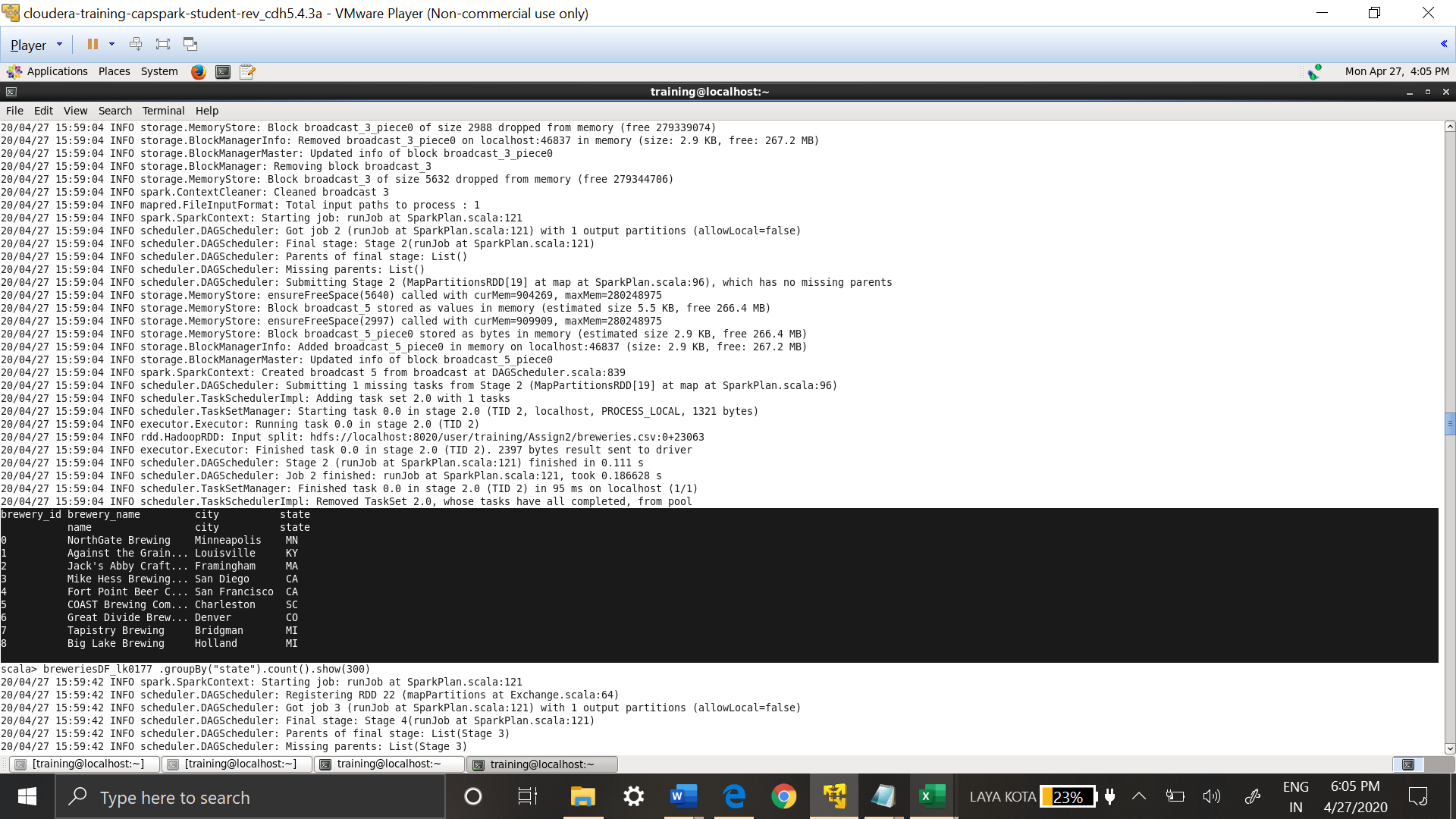
breweriesDF\_lk0177.printSchema()



**Step-8 :** Checking the Data Frame created

breweriesDF\_lk0177.show(10)

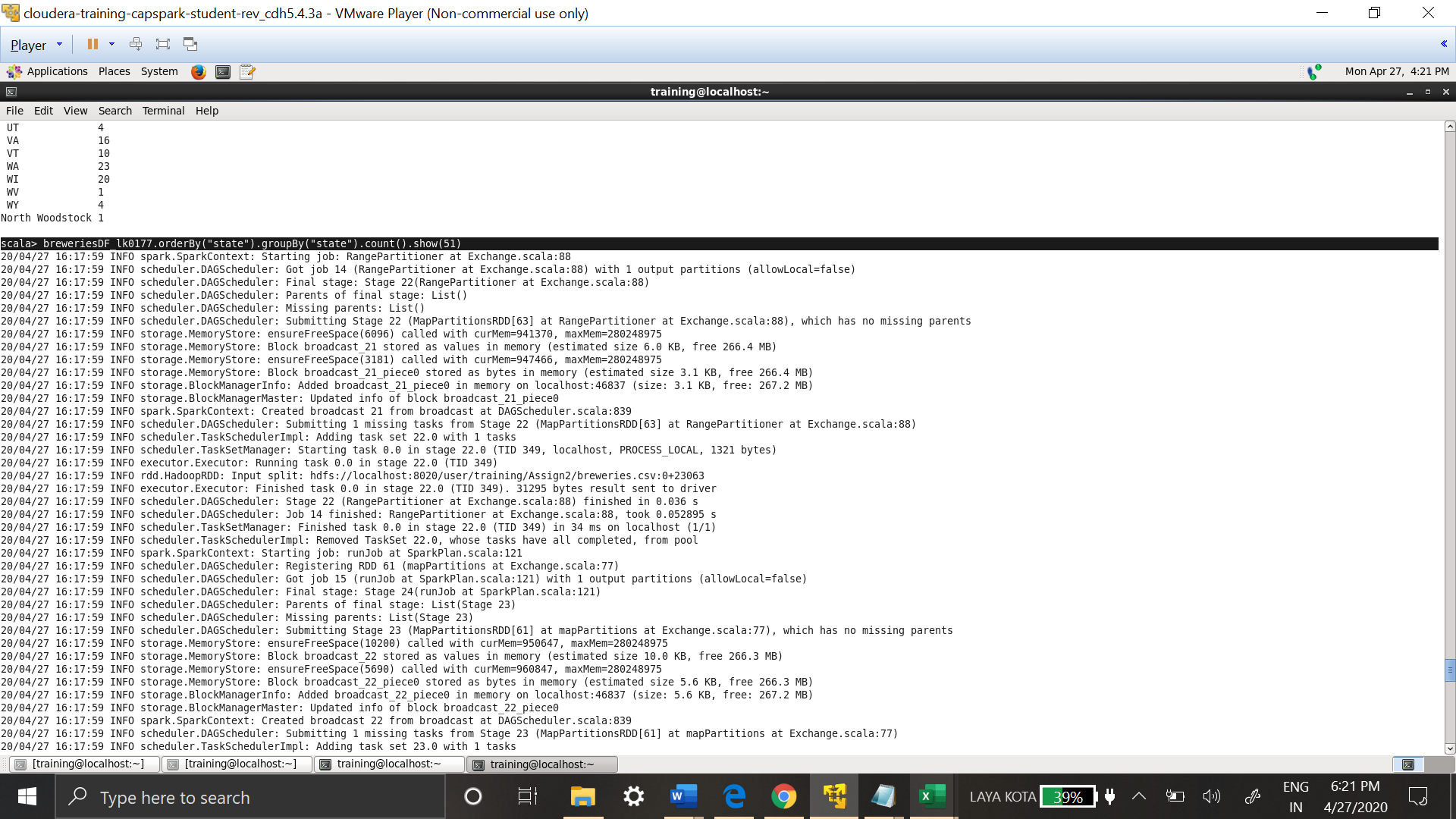




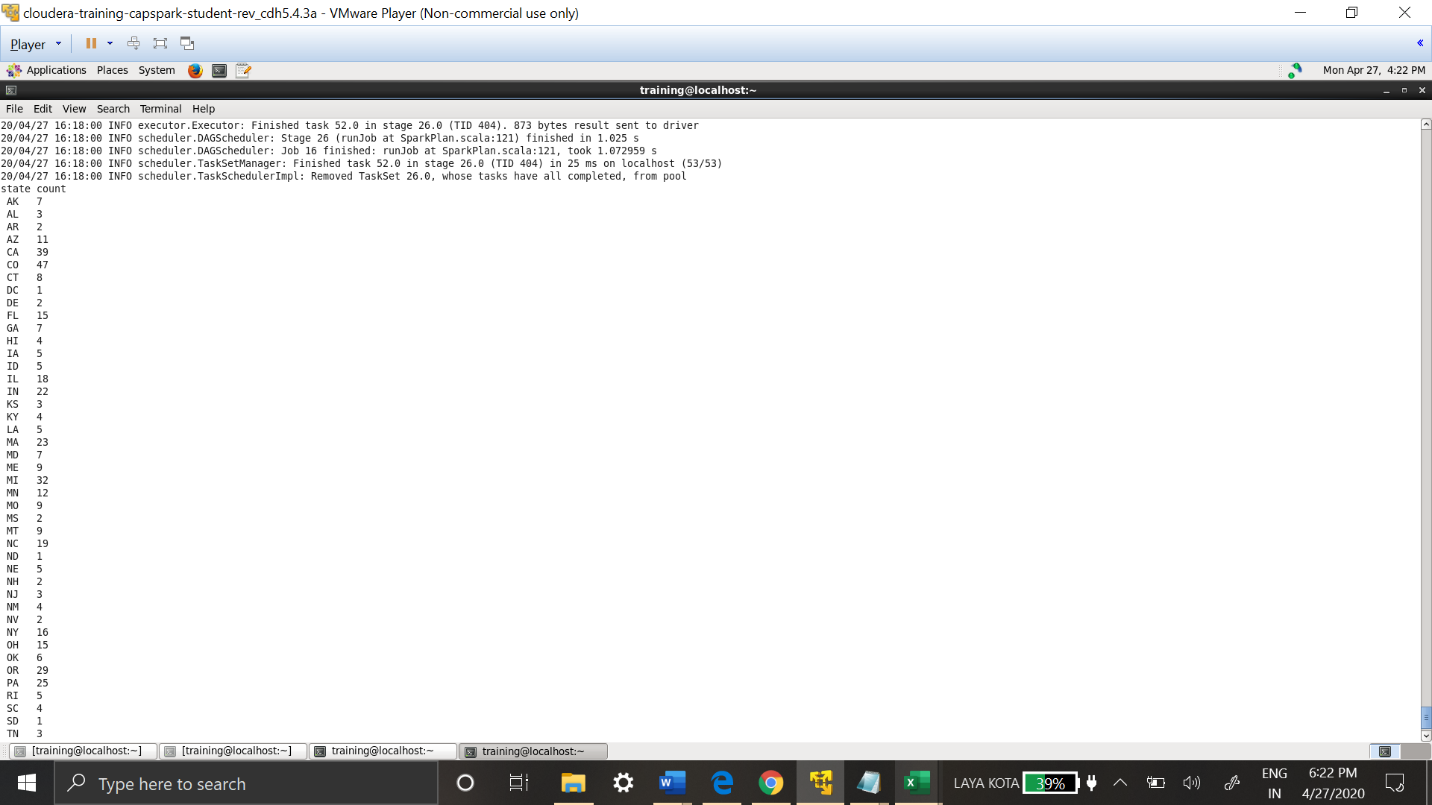
Top ten rows of the Data Frame are retrieved to check the data

1. **For the following questions use Spark SQL**
   1. **Determine the number of breweries in each state**

breweriesDF\_lk0177.orderBy("state").groupBy("state").count().show(51)



After looking at the data in excel, it looks like it’s USA related data. The results show number of breweries in each state

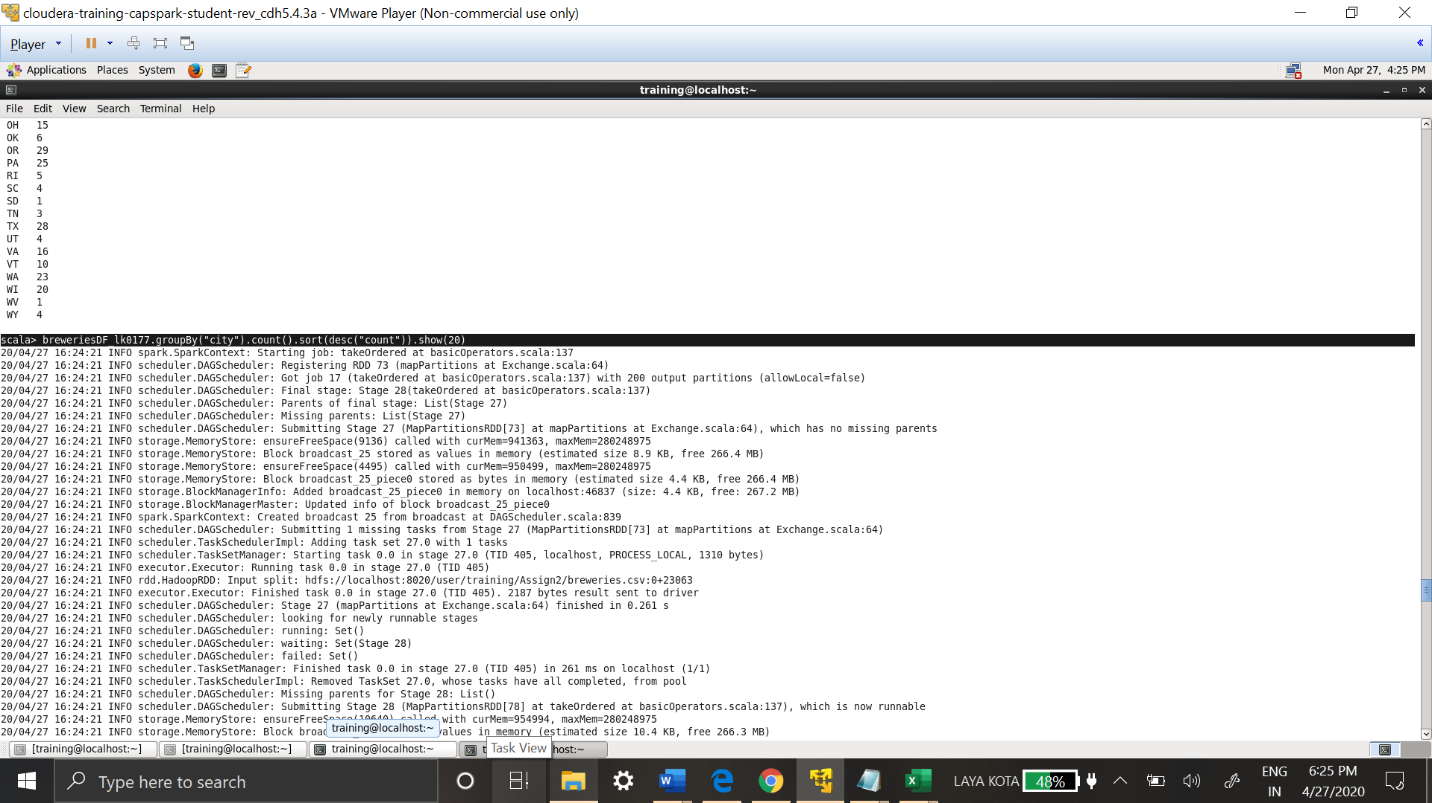


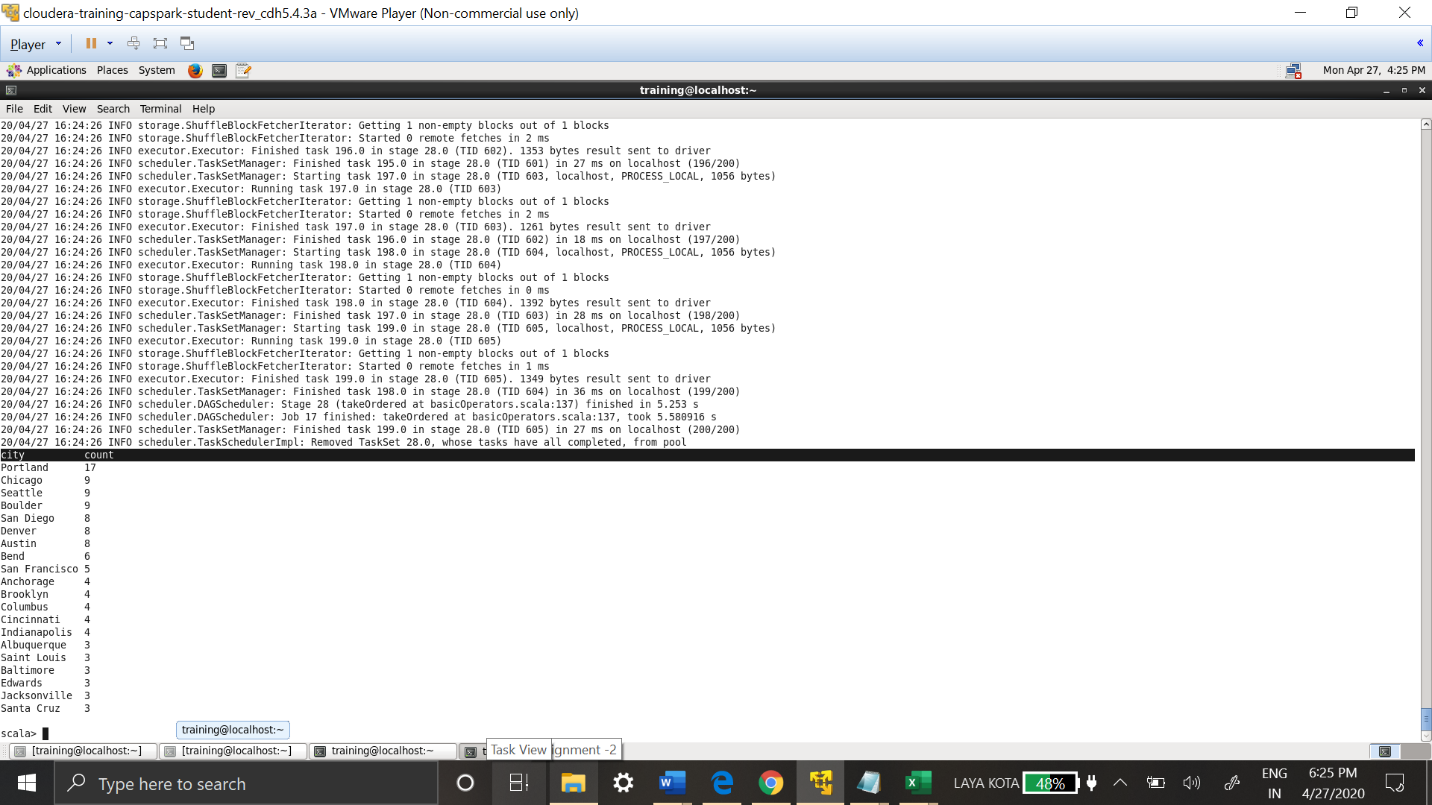


All the states are ordered alphabetically and grouped by state.

* 1. **Determine the cities with most breweries**

breweriesDF\_lk0177.groupBy("city").count().sort(desc("count")).show(20)





Top 20 Cities with most breweries are shown in the results, **Portland** at the top with 17 breweries.

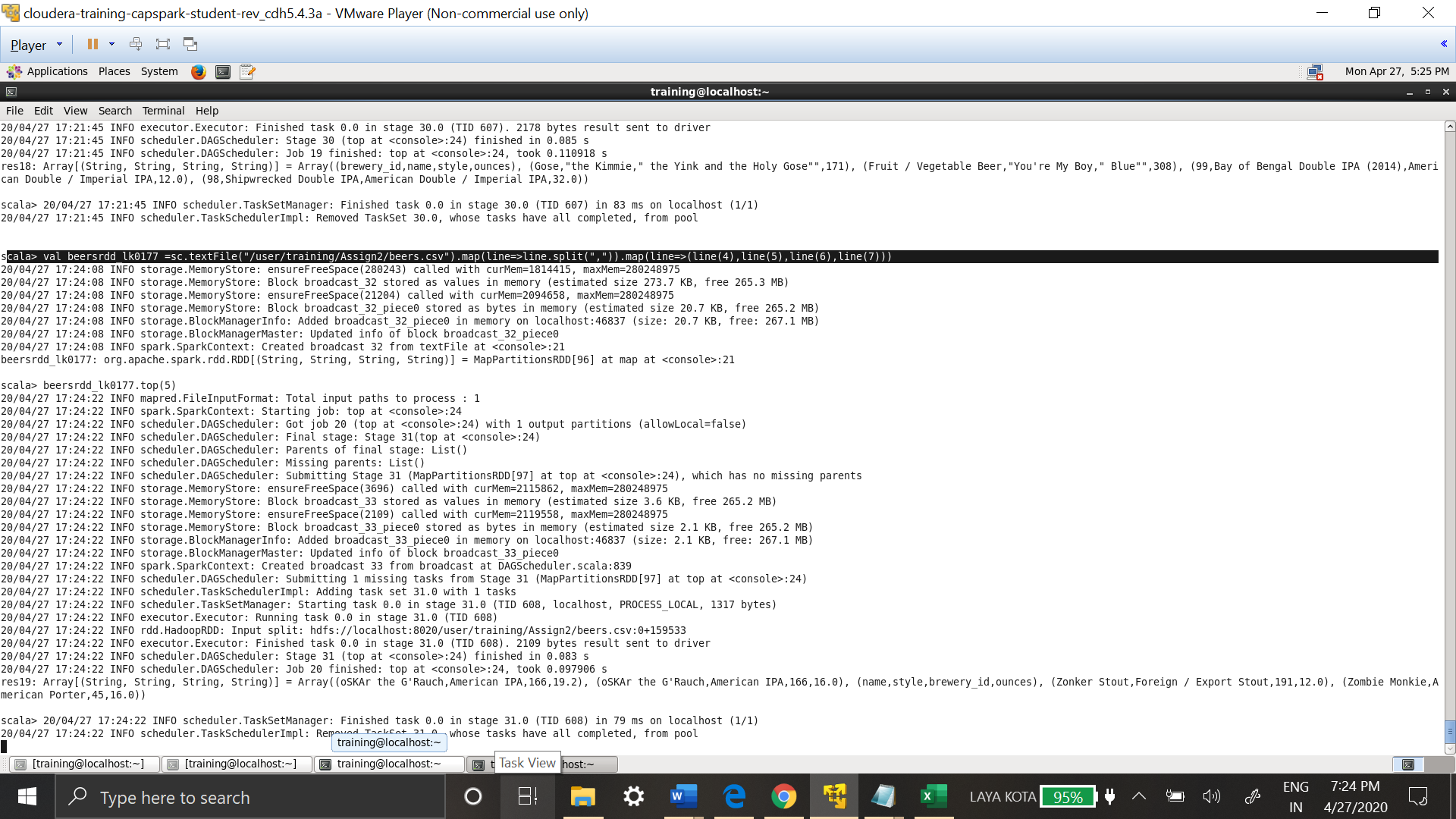
**Using Beers.csv Data**

**Step -1:** Creating RDD for Beers.csv file.

val beersrdd\_lk0177 =sc.textFile("/user/training/Assign2/beers.csv")

.map(line=>line.split(","))

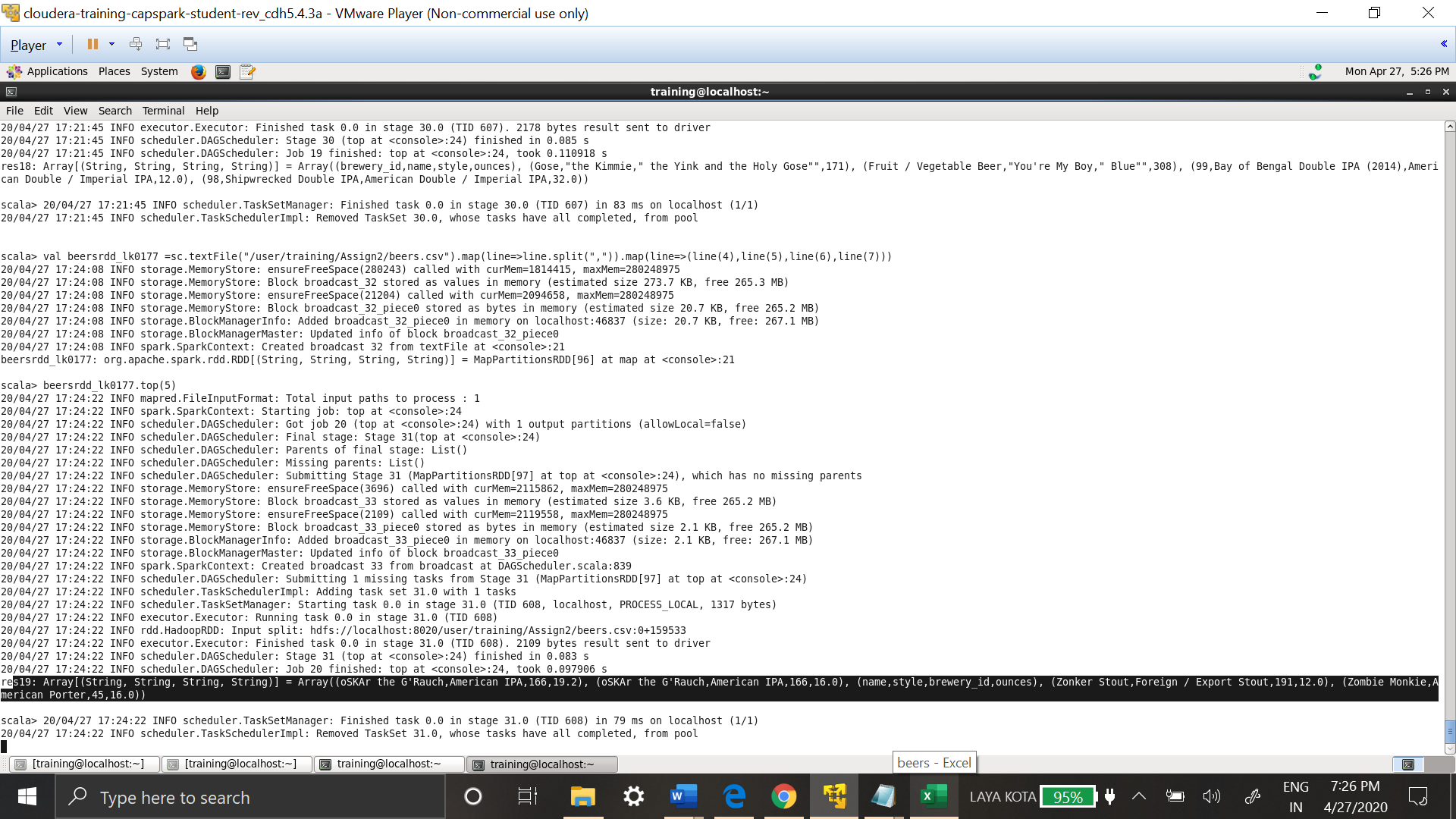
.map(line=>(line(4),line(5),line(6),line(7)))



Only last four columns are required as per Dictionary.doc file. So only those are read into RDD.

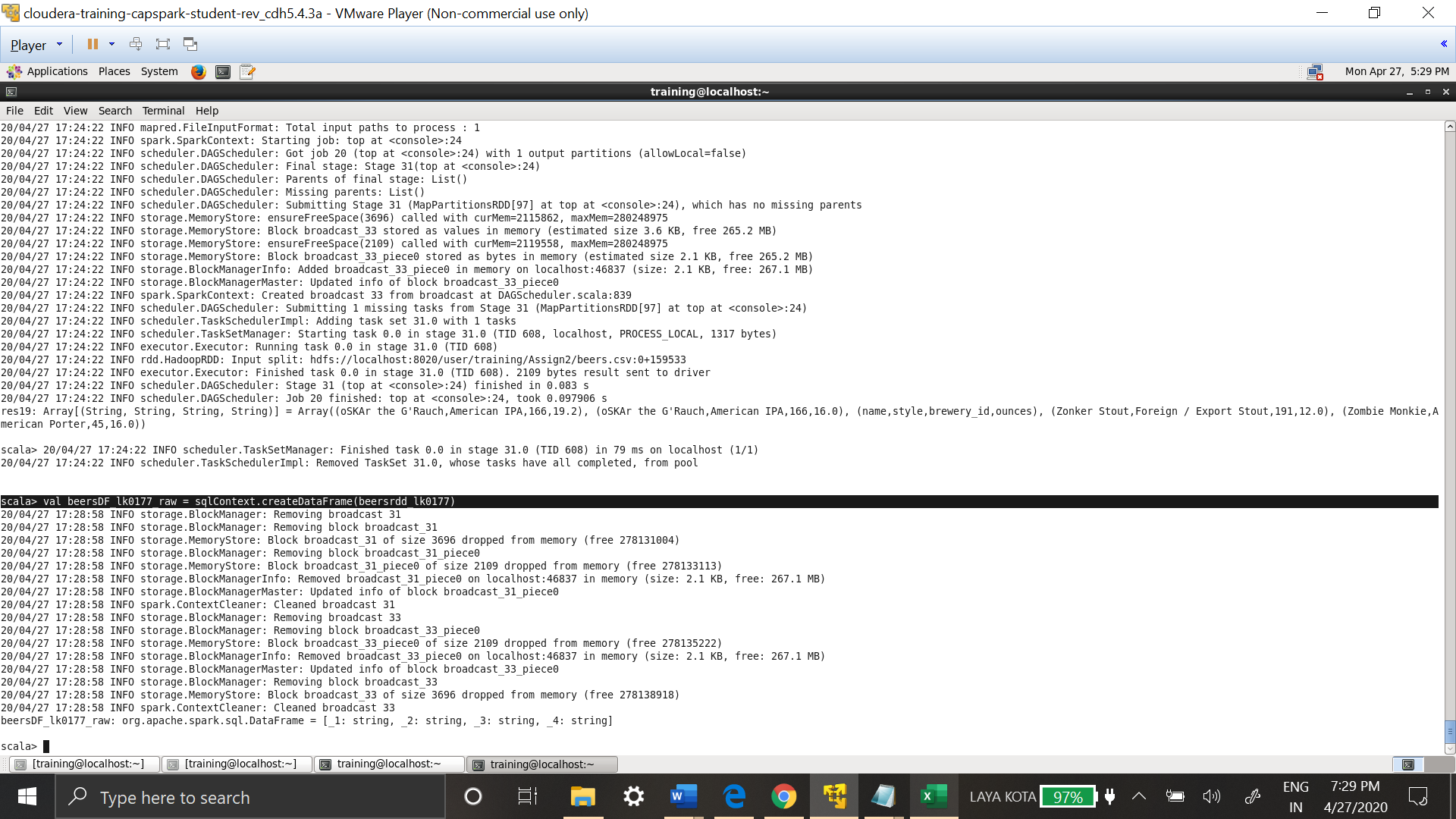
**Step-2:**Checking the Data in RDD

beersrdd\_lk0177.top(5)



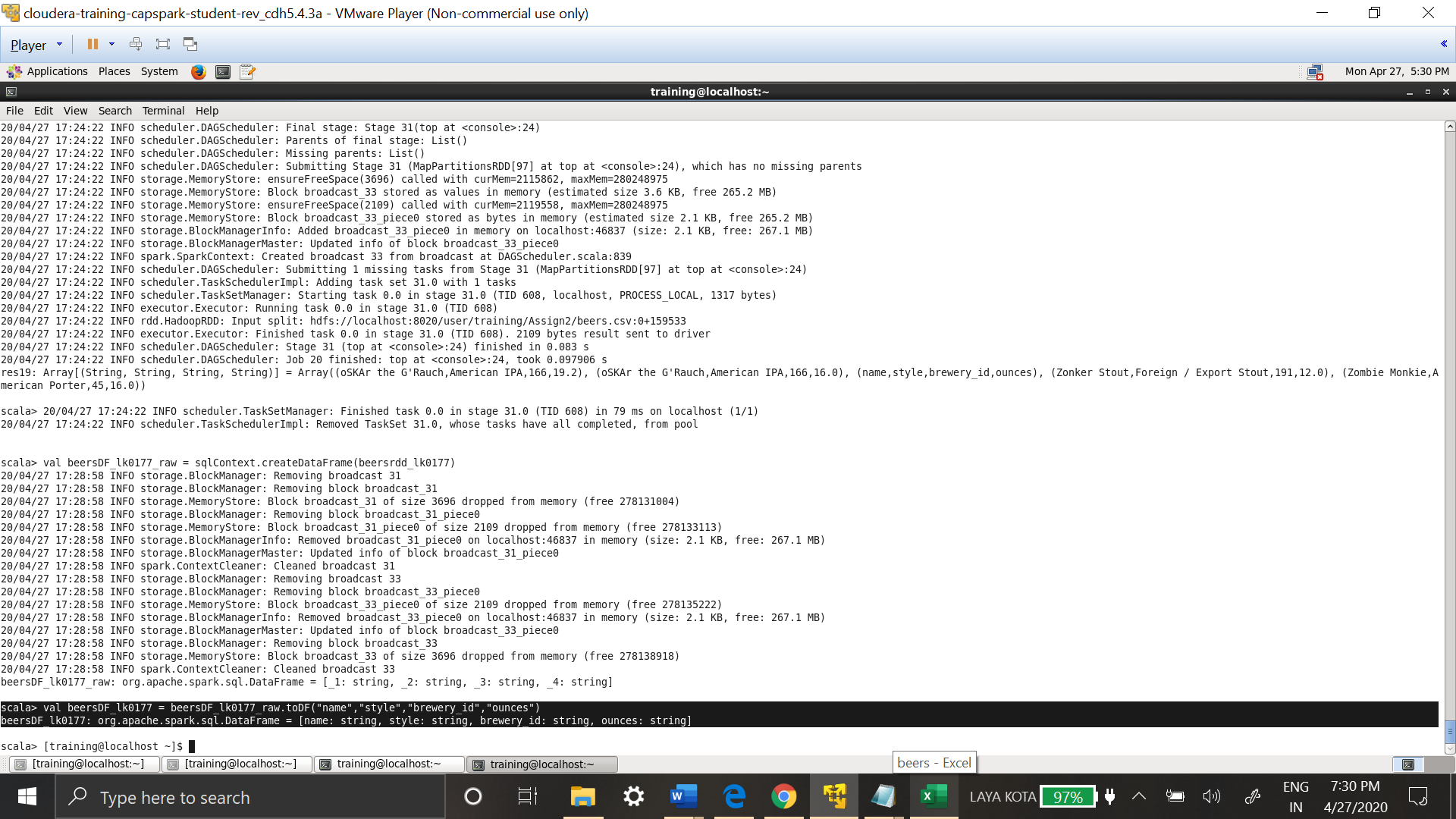
**Step-3:** Creating Data Frame from the RDD

val beersDF\_lk0177\_raw = sqlContext.createDataFrame(beersrdd\_lk0177)



**Step-4 :** Schema to the Data Frame created from RDD

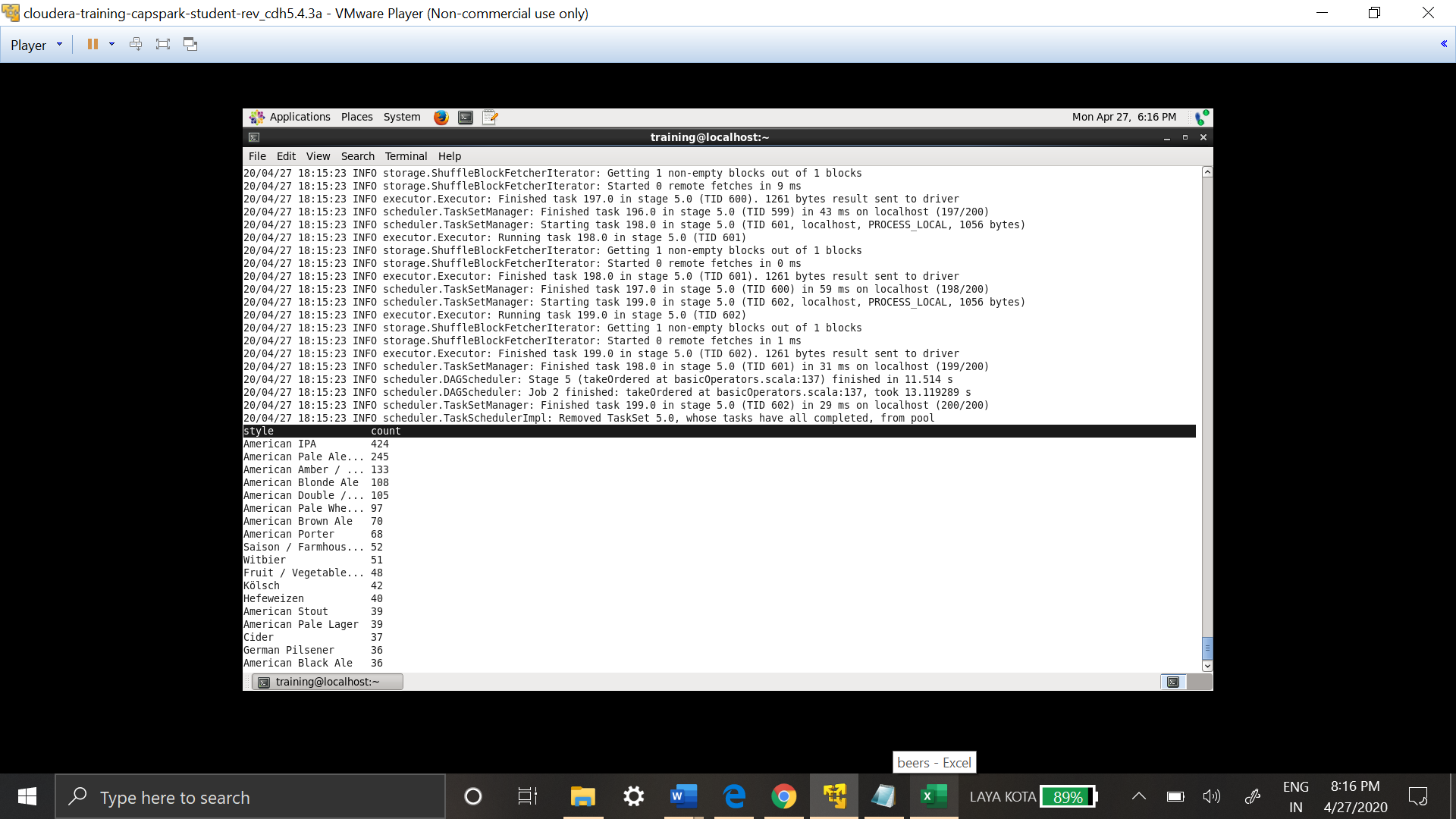
val beersDF\_lk0177 = beersDF\_lk0177\_raw.toDF("name","style","brewery\_id","ounces")



1. **Determine the most brewed beer style**

beersDF\_lk0177.groupBy("style").count().sort(desc("count")).show(20)





The most brewed style beer is American IPA, top 20 of the style are also displayed along with it. GroupBy function is used to group data. Count is used to get the total count and results are sorted in descending order.