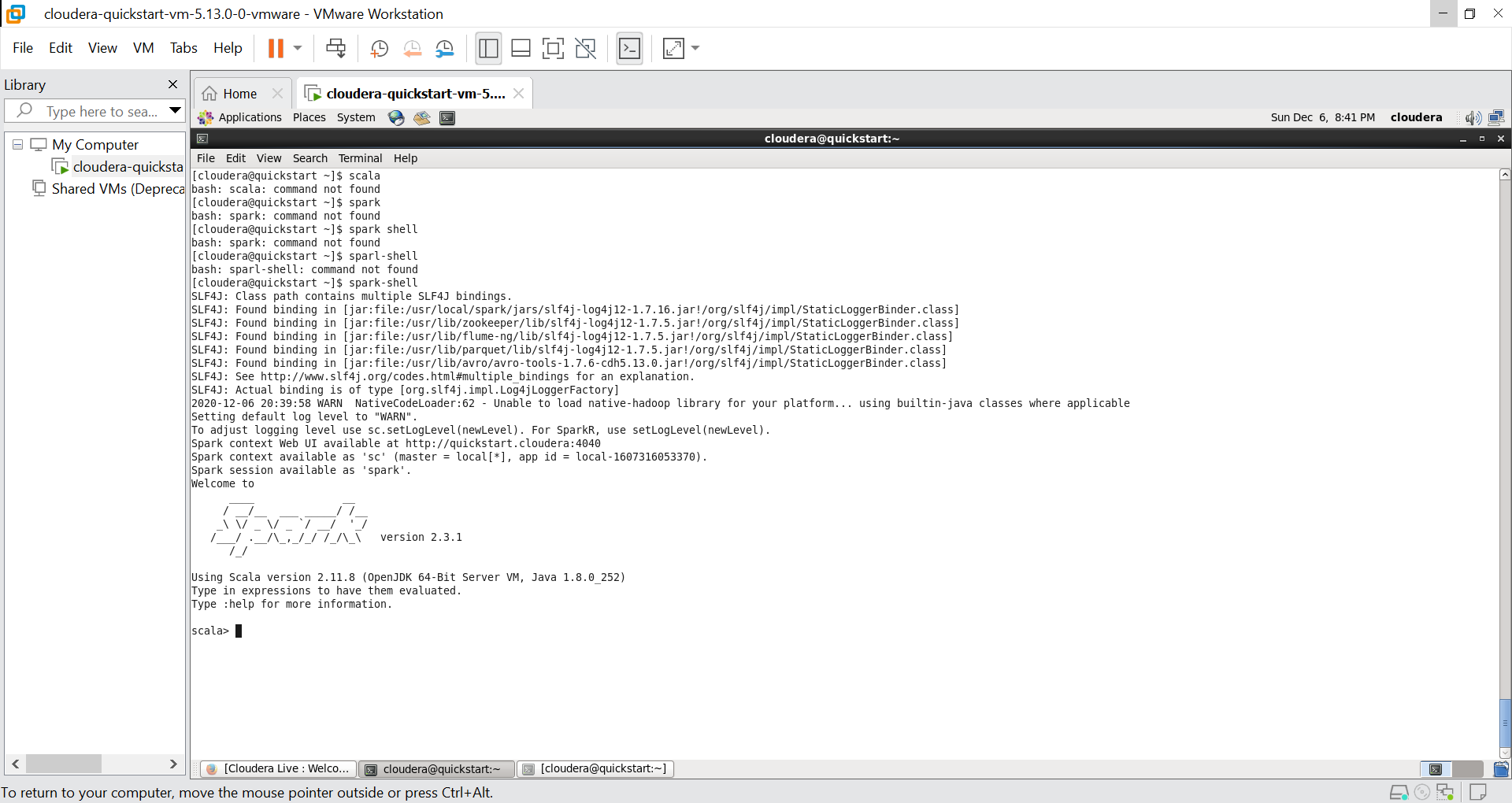
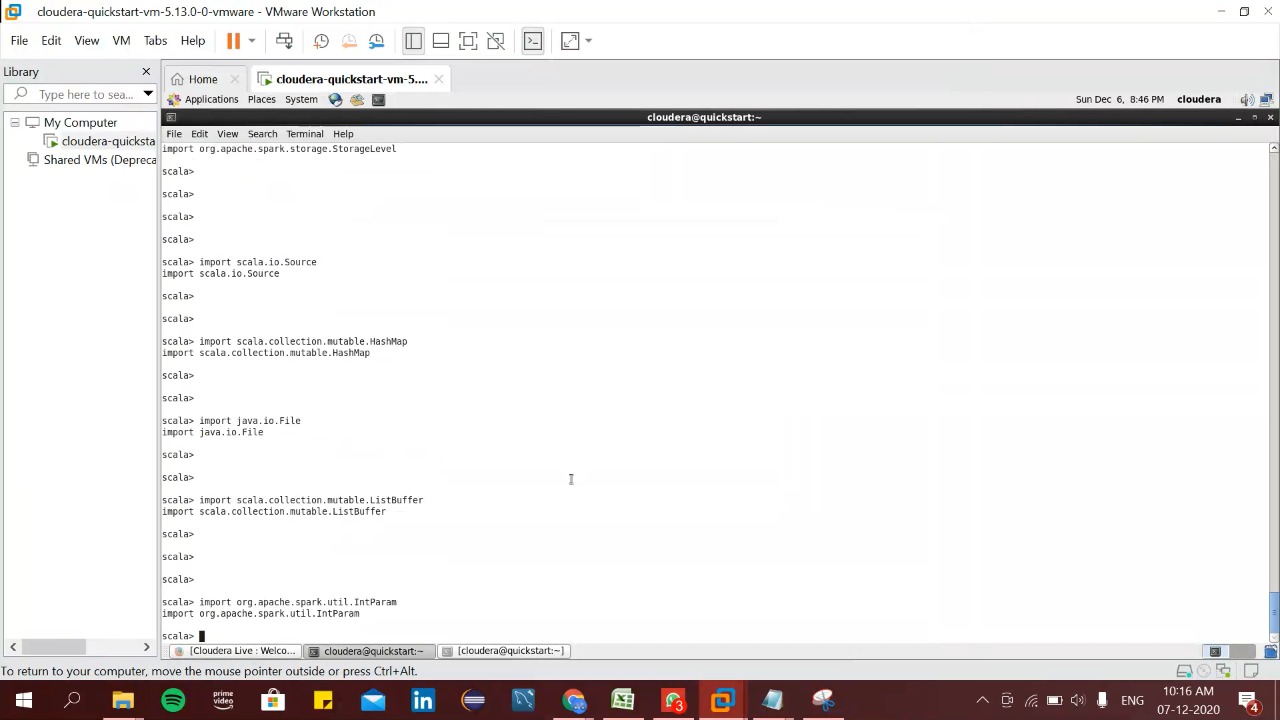
**KO: Analyze data using Data Frames**

Enter to the spark shell:



Import all the packages to run the spark



Following are the import statements:

import spark.sql

import spark.implicits.\_

import org.apache.spark.sql.Row

import org.apache.spark.sql.SparkSession

import org.apache.spark.\_

import org.apache.spark.rdd.RDD

import org.apache.spark.util.IntParam

import org.apache.spark.sql.SQLContext

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

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

import org.apache.spark.sql.\_

import org.apache.spark.mllib.stat.Statistics

import org.apache.spark.sql.SparkSession

import org.apache.spark.storage.StorageLevel

import scala.io.Source

import scala.collection.mutable.HashMap

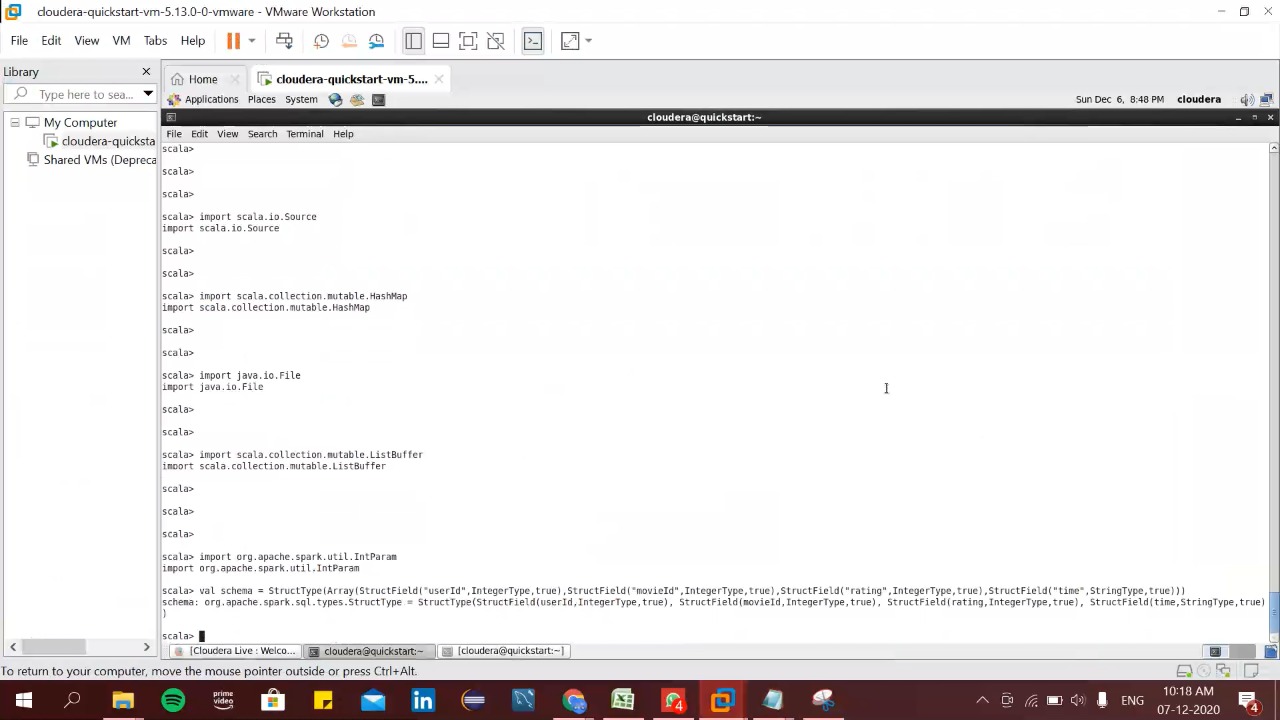
import java.io.File

import scala.collection.mutable.ListBuffer

import org.apache.spark.util.IntParam

Define schema to the dataset:

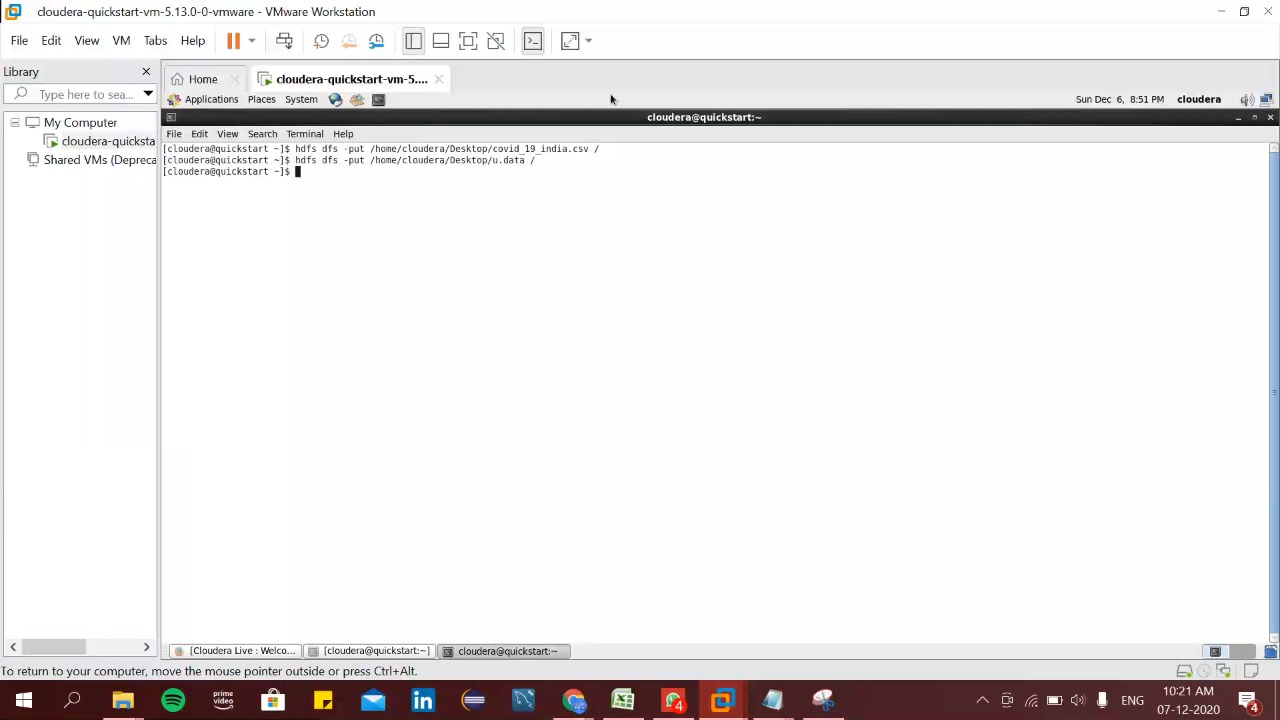
val schema = StructType(Array(StructField("userId", IntegerType, true), StructField("movieId", IntegerType, true), StructField("rating", IntegerType, true), StructField("time", StringType, true)))



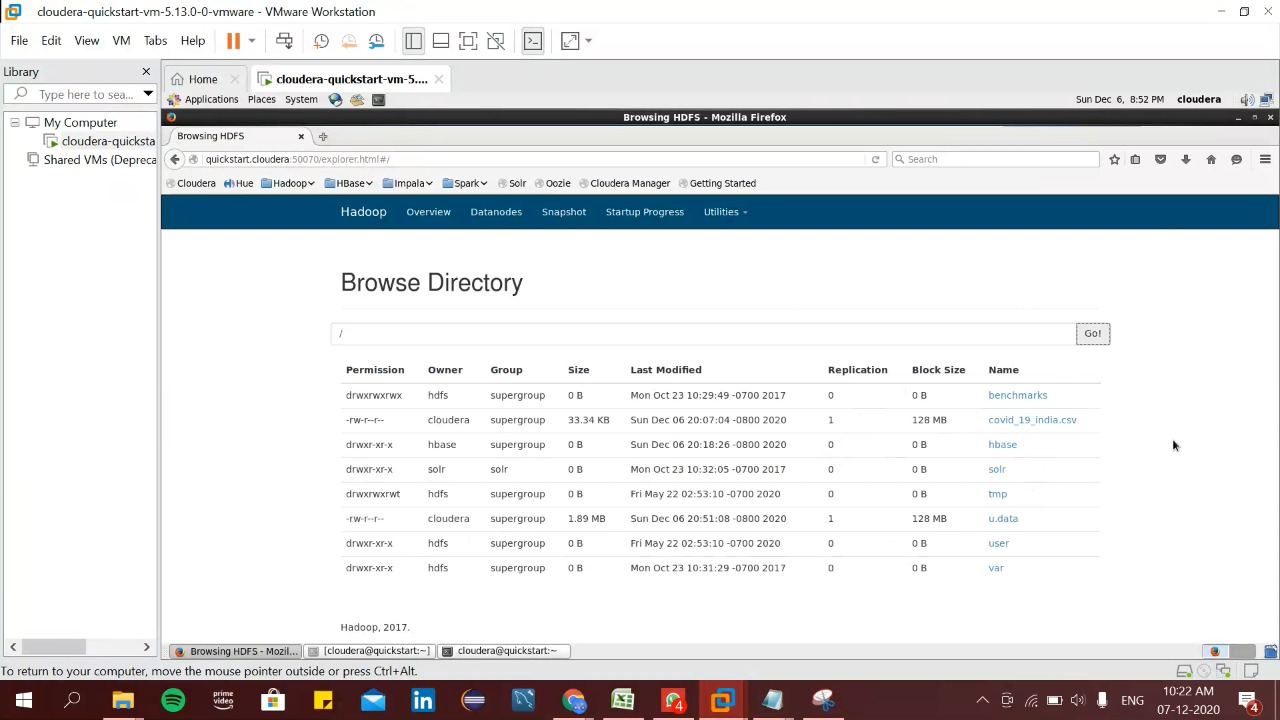
Upload the dataset to the HDFS:

hdfs dfs –put /home/cloudera/Desktop/covid\_19\_india.csv /

hdfs dfs –put /home/cloudera/Desktop/u.data /

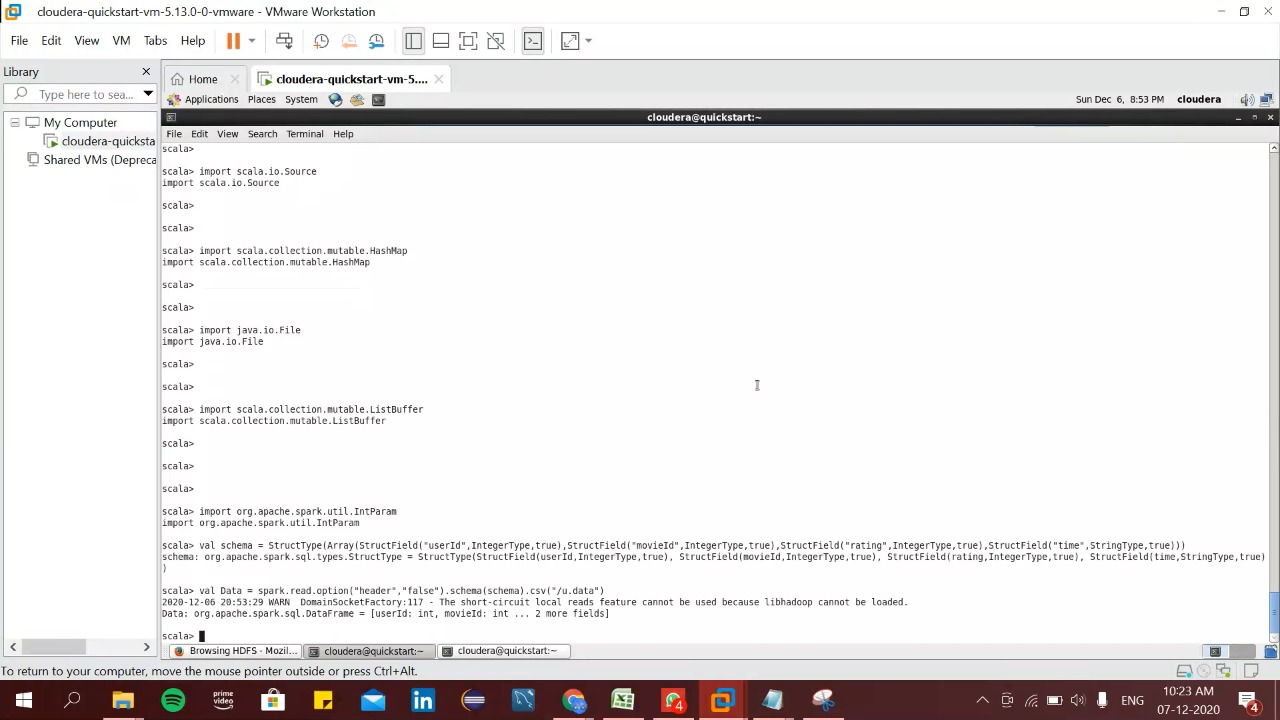


Recheck the file Covid\_19\_India whether it is present in hdfs or not:



We can see that the file covid\_19\_india is present in the hdfs directory.

Define the schema of the dataset:

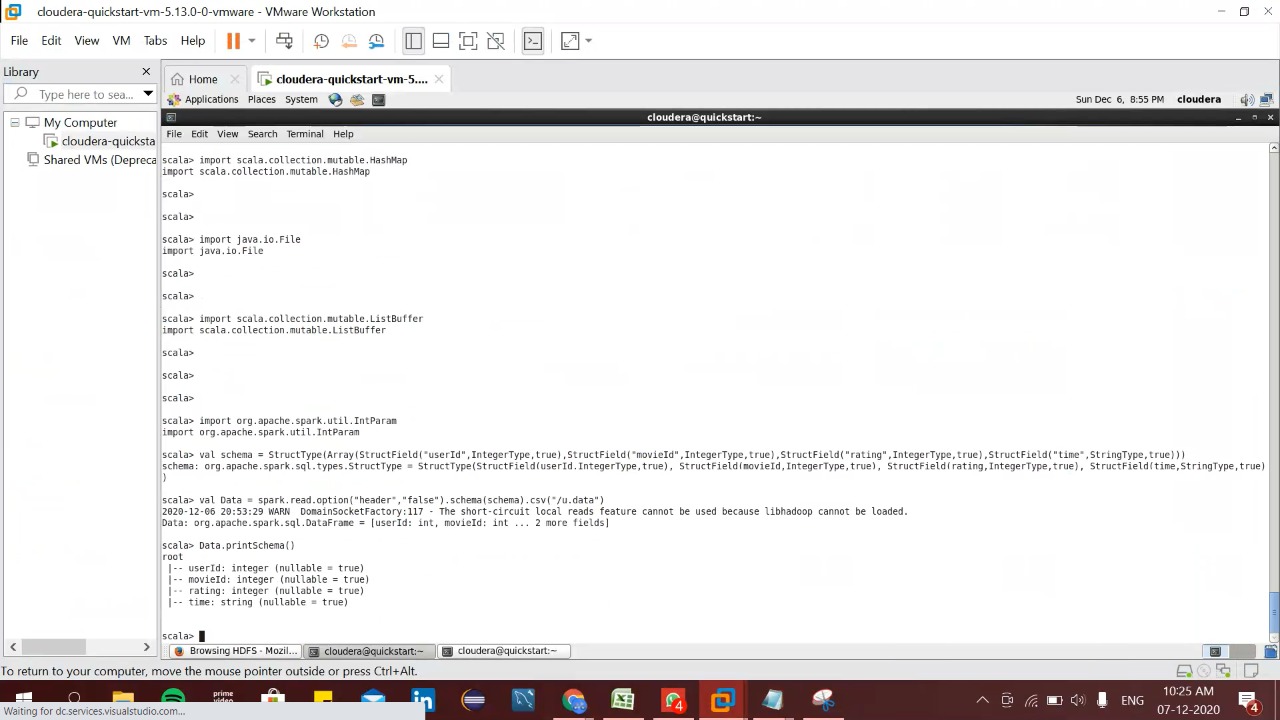


val Data = spark.read.option("header","false").schema(schema).csv("/udata.csv")

We can see that the schema is defined.

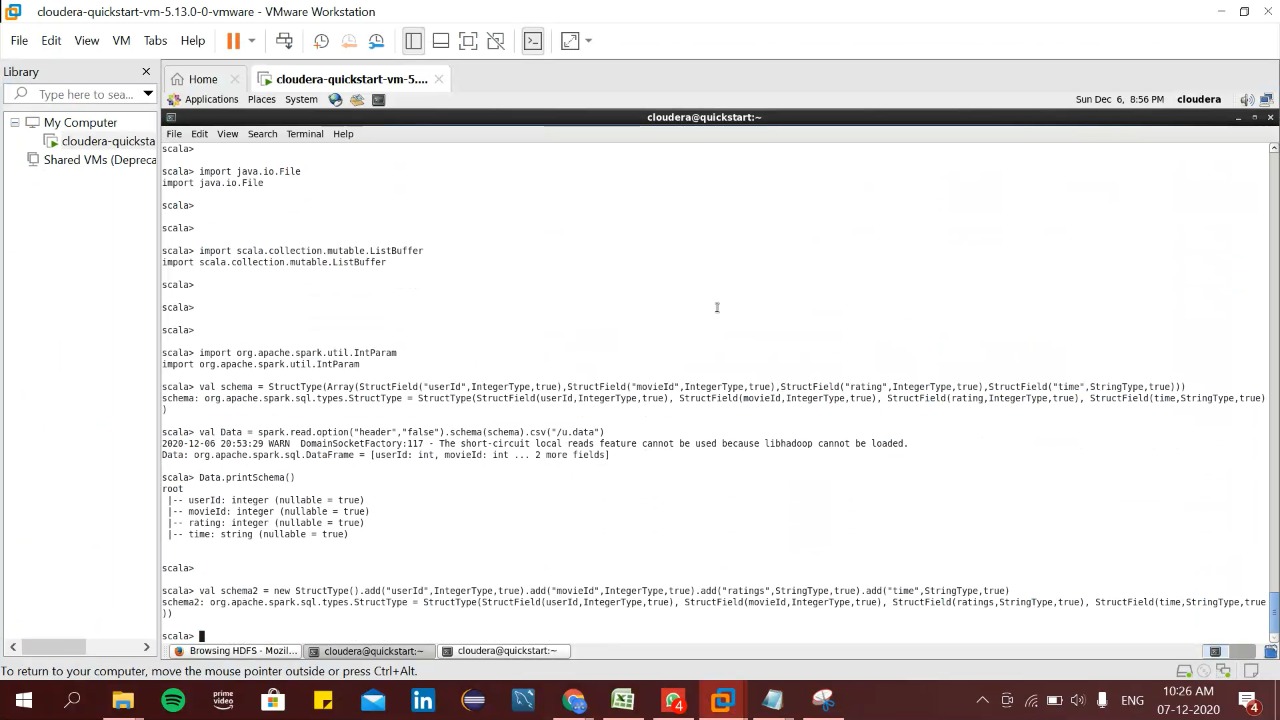
Print the schema of the dataset using the command:

Data.printSchema()



Here we can see schema of the dataset in the output.

Create a new schema as schema 2:

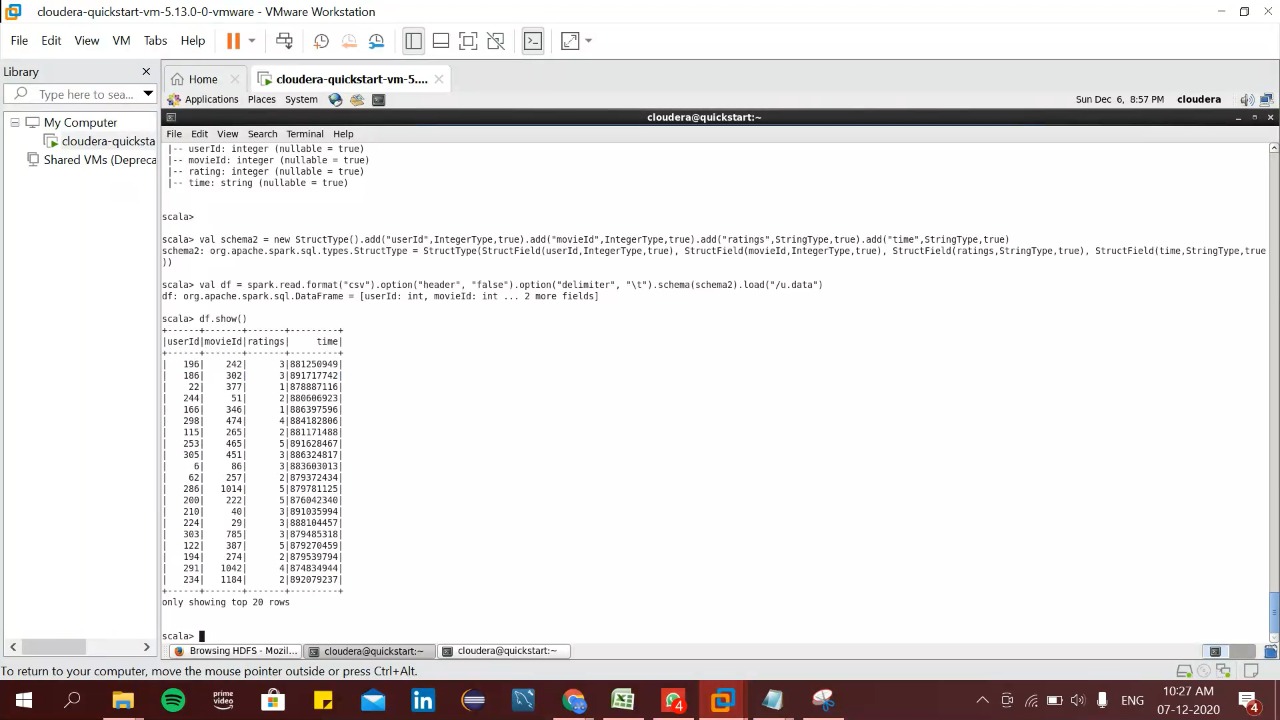


We have to use the command :

val schema2 = new StructType().add("userId",IntegerType,true).add("movieId",IntegerType,true).add("ratings",Strin gType,true).add("time",StringType,true)

for getting the schema.

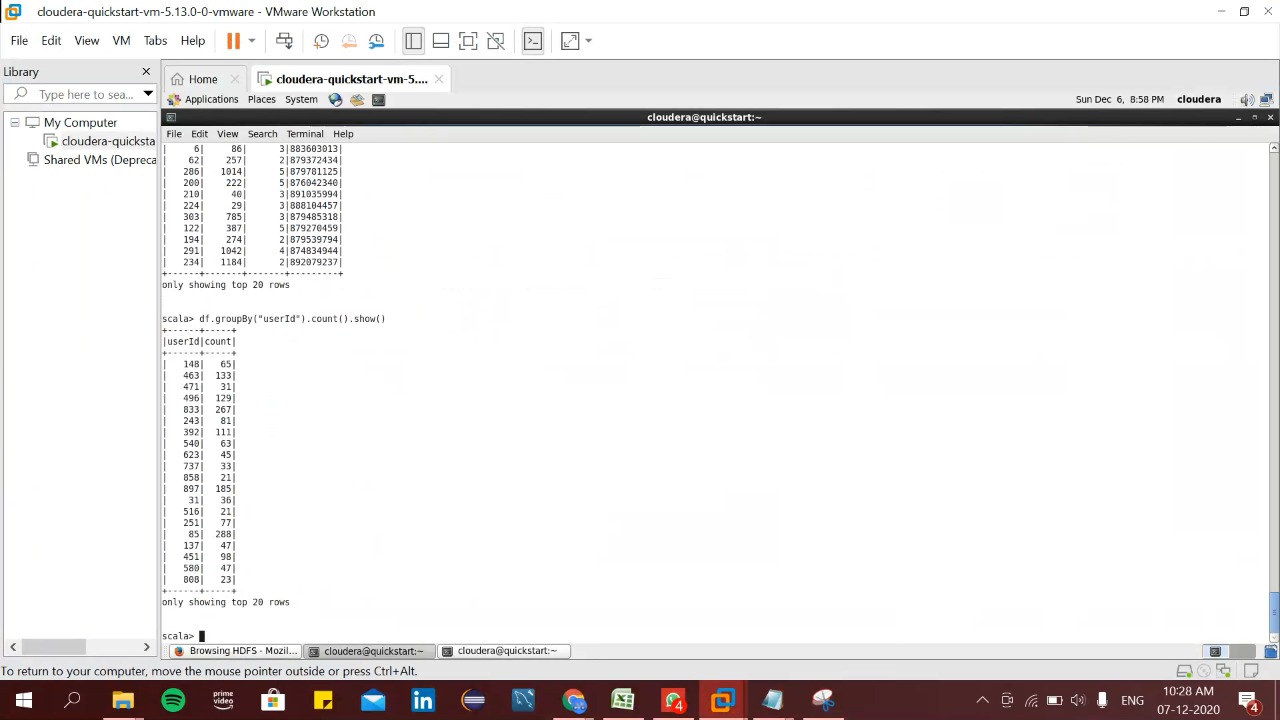
Print the top 20 rows of the new schema



We can see the top 20 rows of the dataset we have used using the command

df.show()

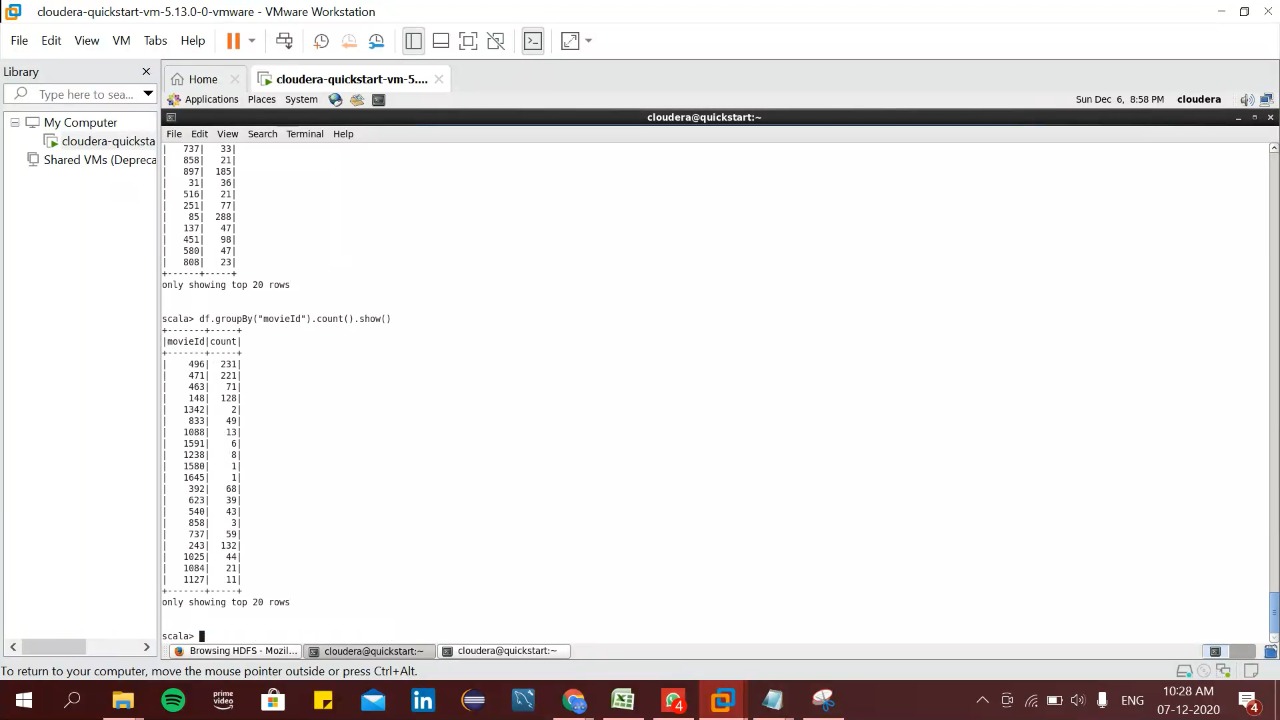
Show the grouping of the dataframe according to the user Id and display its count



Now we have to show the count of user Id grouped according to the user Id by using the command:

df.groupBy("userId").count().show()

Group the data frame according to the ovie Id and display its count



Here we displayed the total count of movie grouped by movie Id using the command:

df.groupBy("movieId").count().show()