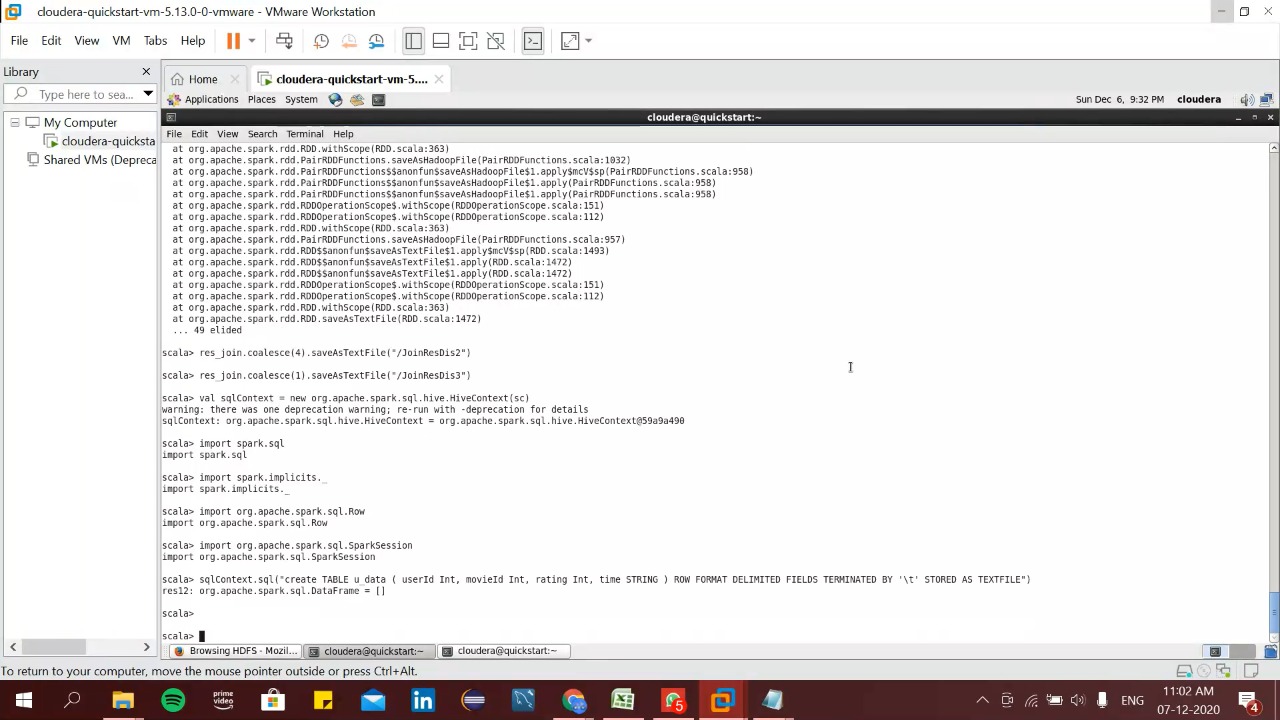
**KO: Query data using Spark SQL**

Step 1: Import the required packages in Scala and create the table for Spark SQL:



Here we uploaded the required packages and made a table for the Spark SQL.

Required packages:

val sqlContext = new org.apache.spark.sql.hive.HiveContext(sc)

import spark.sql

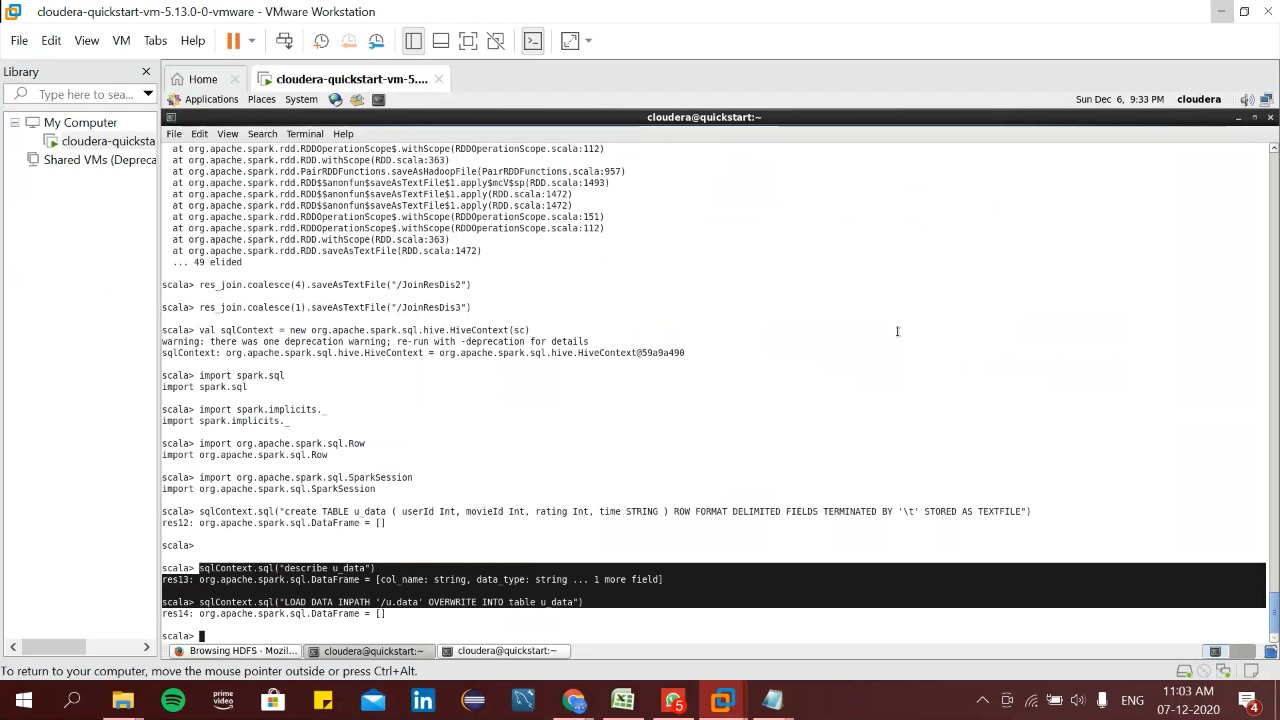
import spark.implicits.\_

import org.apache.spark.sql.Row

import org.apache.spark.sql.SparkSession

sqlContext.sql("create TABLE u\_data ( userId Int, movieId Int, rating Int, time STRING ) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE")

Step 2: Loading data to the table:



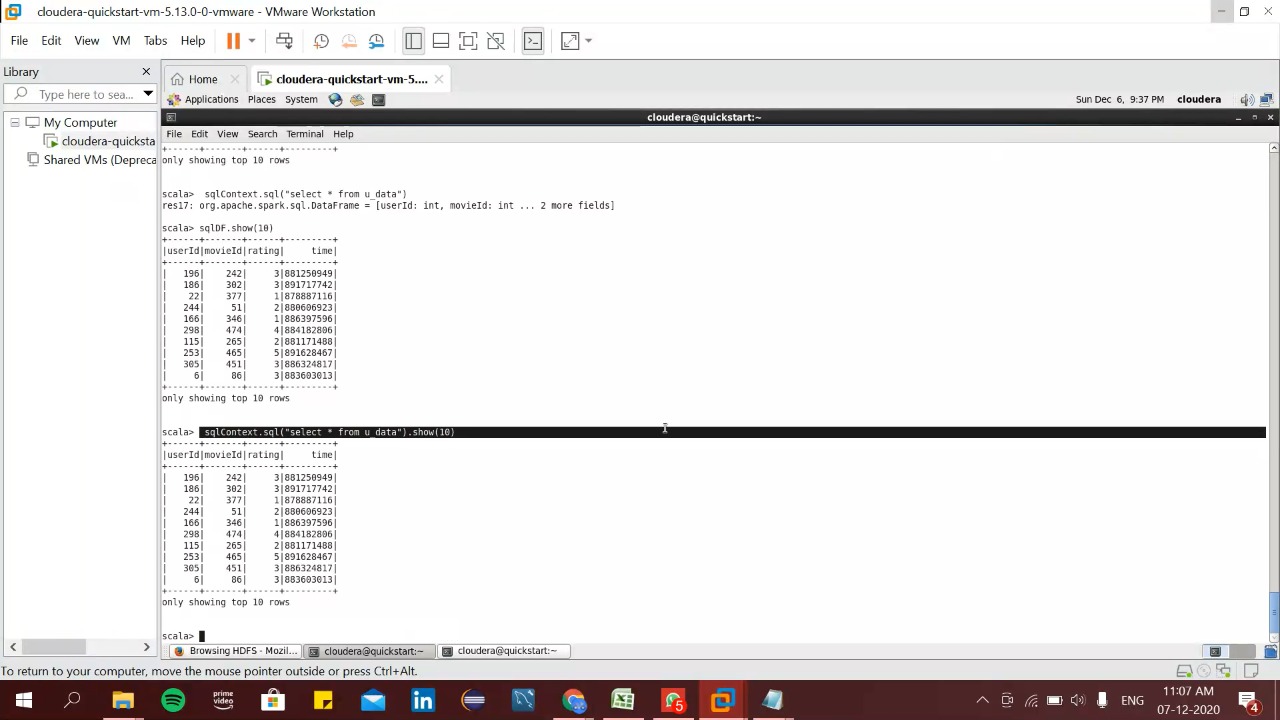
Step 4: We described the structure of the table using:

sqlContext.sql("describe u\_data")

Step5: Here we uploaded data to the table created using:

sqlContext.sql("LOAD DATA INPATH '/u.data' OVERWRITE INTO table u\_data")

Step 6: Displaying the first 10 rows:



Step 7: Printing the first 10 rows of the table:

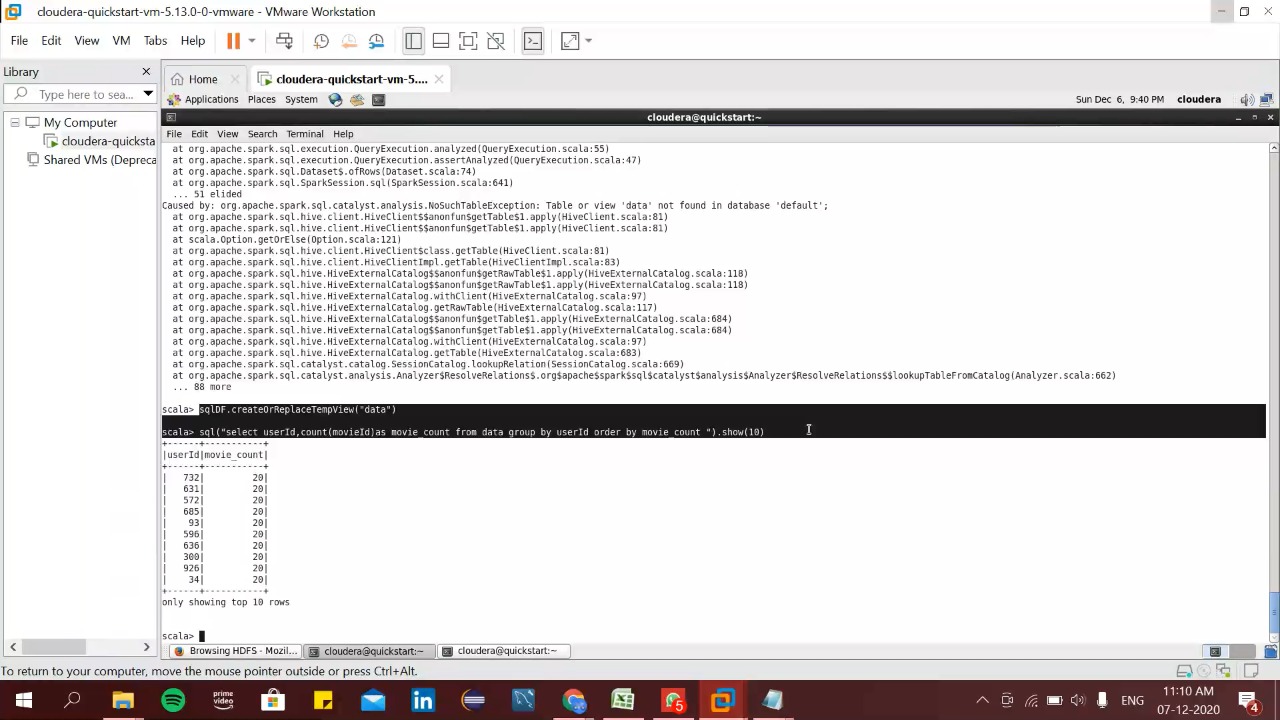
val sqlDF = sql("select \* from u\_data").show(10)

//This command will print the 10 rows of the table.

Step 8: We can also use the command to print the 10 rows:

sqlDF.show(10)

Step 9: Creating a view and Grouping:



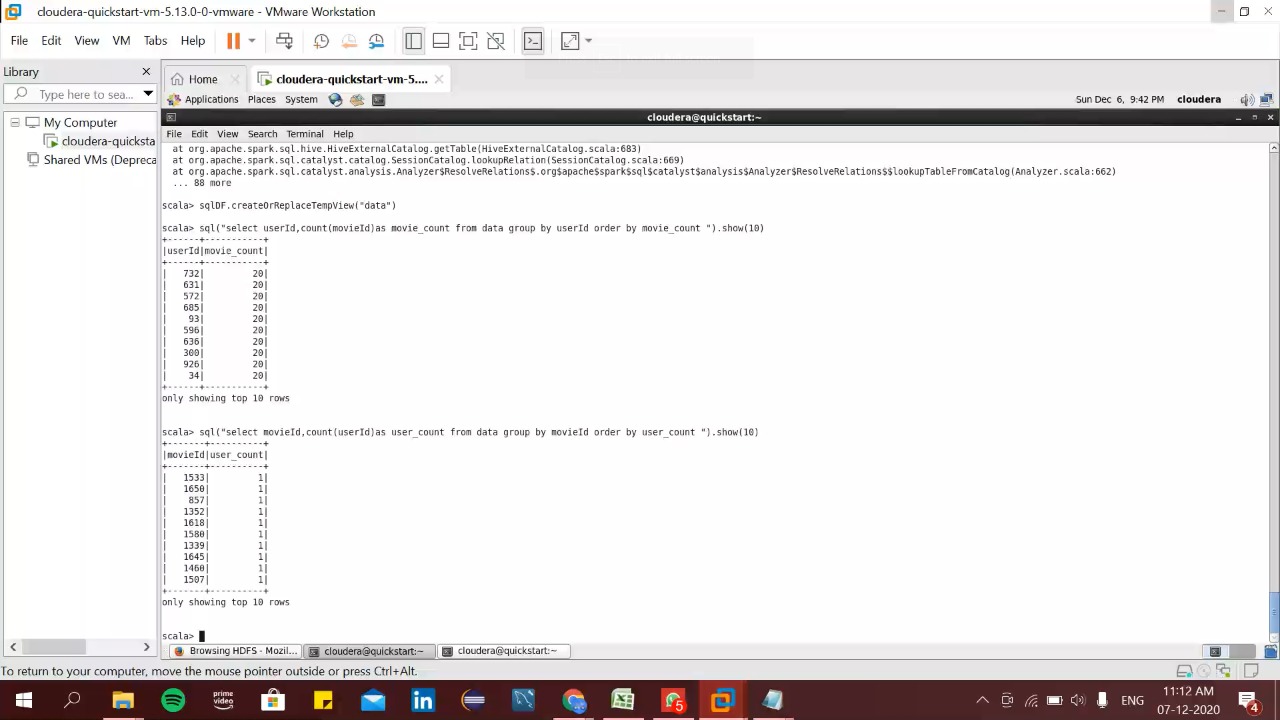
Here we have created the temporary view of the table to do SQL operations on the data.

sqlDF.createOrReplaceTempView(“data”)

then we have done SQL query for counting how many movie has been watched by each user

sql("select userId, count(movieId) as movie\_count from data group by userId order by movie\_count ").show(10)

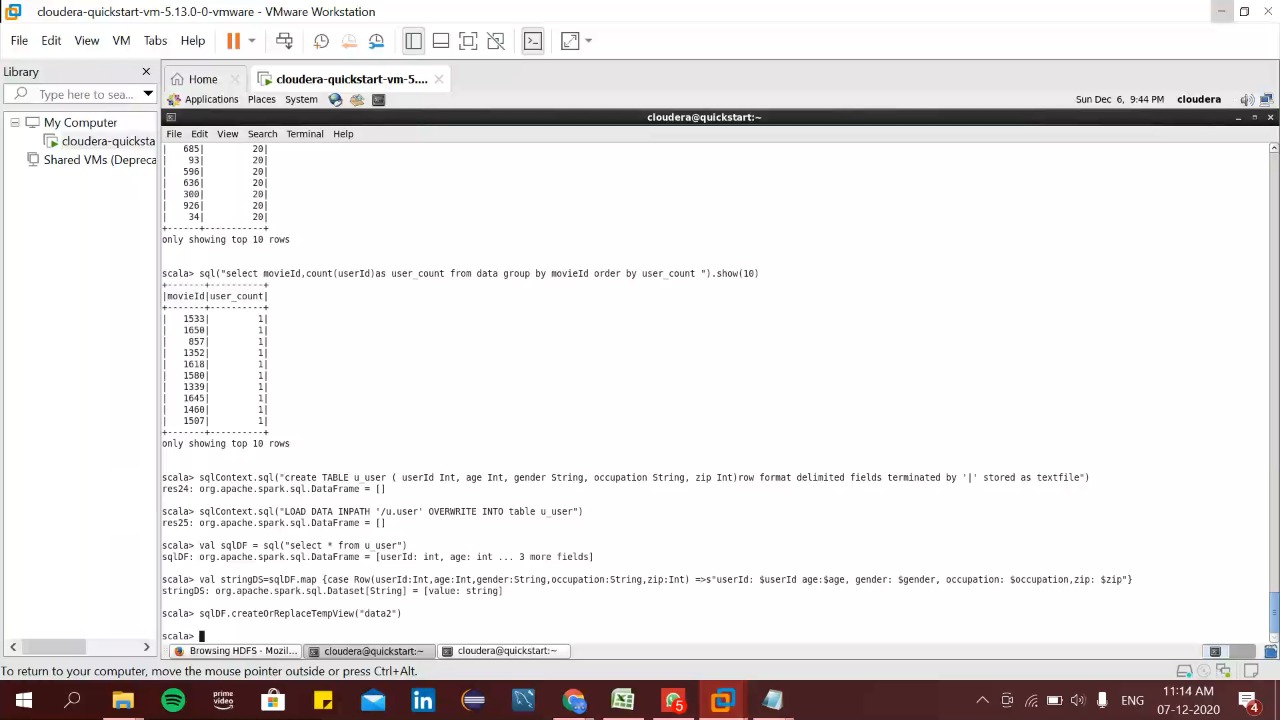
Grouping based on Movie\_id:



We have done SQL query to get the count of users who watched a particular movie:

sql("select movieId, count(userId) as user\_count from data group by movieId order by user\_count ").show(10)

Creating table u-user and get the view:



Now we have to create another table for u\_user csv .The steps are same as we have created at above.

sqlContext.sql("create TABLE u\_user ( userId Int, age Int, gender String, occupation String, zip Int)row format delimited fields terminated by '|' stored as textfile")

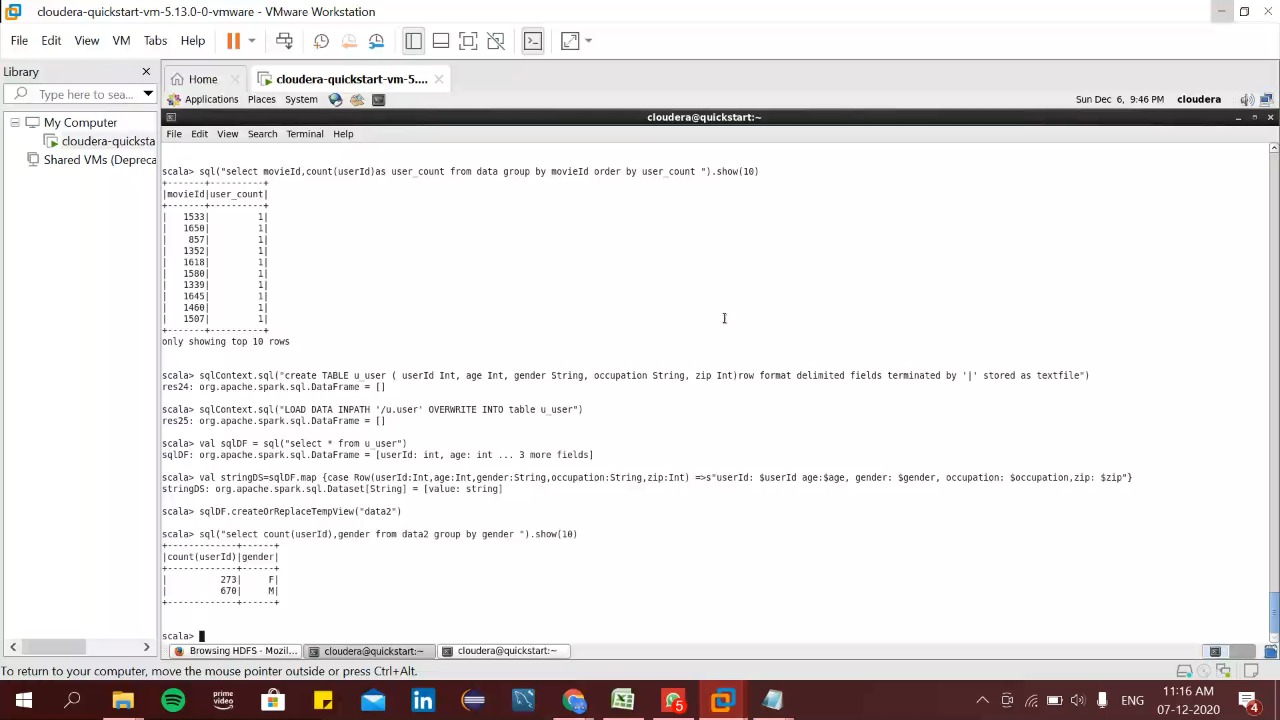
sqlContext.sql("LOAD DATA INPATH '/u.user' OVERWRITE INTO table u\_user")

val sqlDF = sql("select \* from u\_user")

val stringDS=sqlDF.map {case Row(userId:Int,age:Int,gender:String,occupation:String,zip:Int) =>s"userId: $userId age:$age, gender: $gender, occupation: $occupation,zip: $zip"}

sqlDF.createOrReplaceTempView("data2")

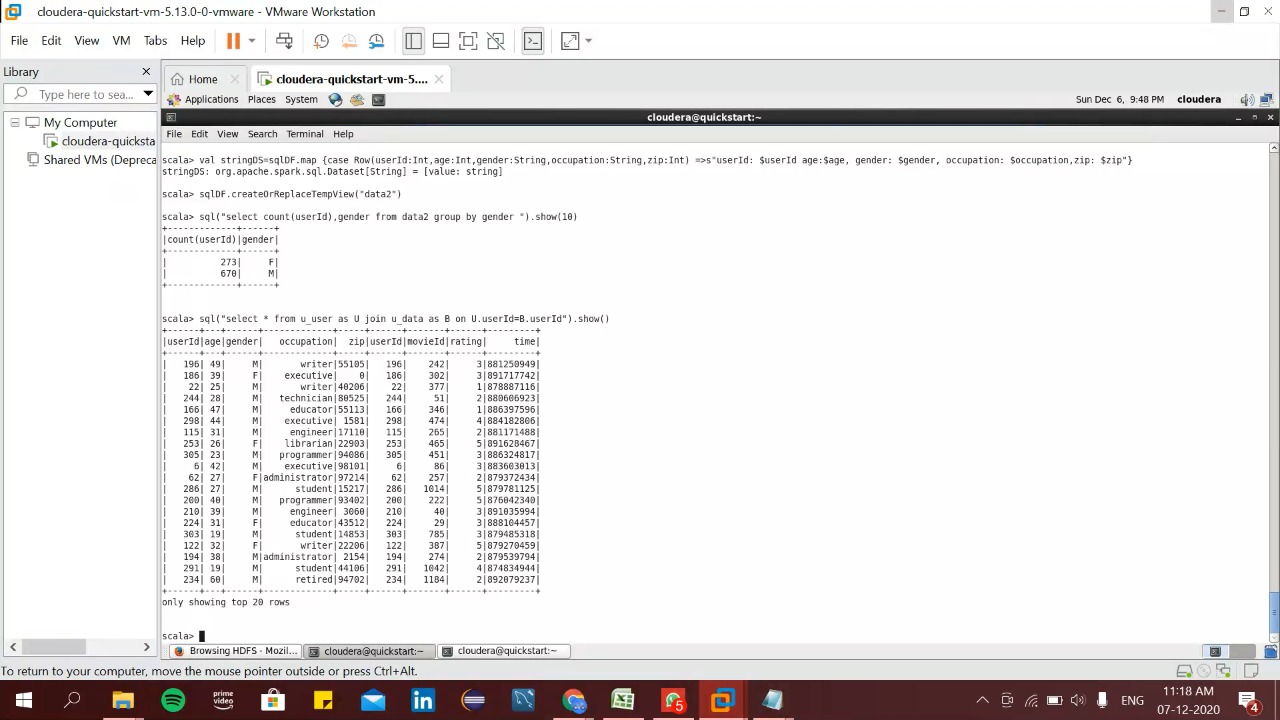
Grouping based on gender of u\_user:



Display the total number of males and females using:

sql("select count(userId),gender from data2 group by gender ").show(10)

Joining two tables based on id:



We have done a joining of the two tables we have created and displayed the result.

sql("select \* from u\_user as U join u\_data as B on U.userId=B.userId").show()