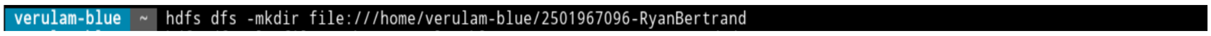
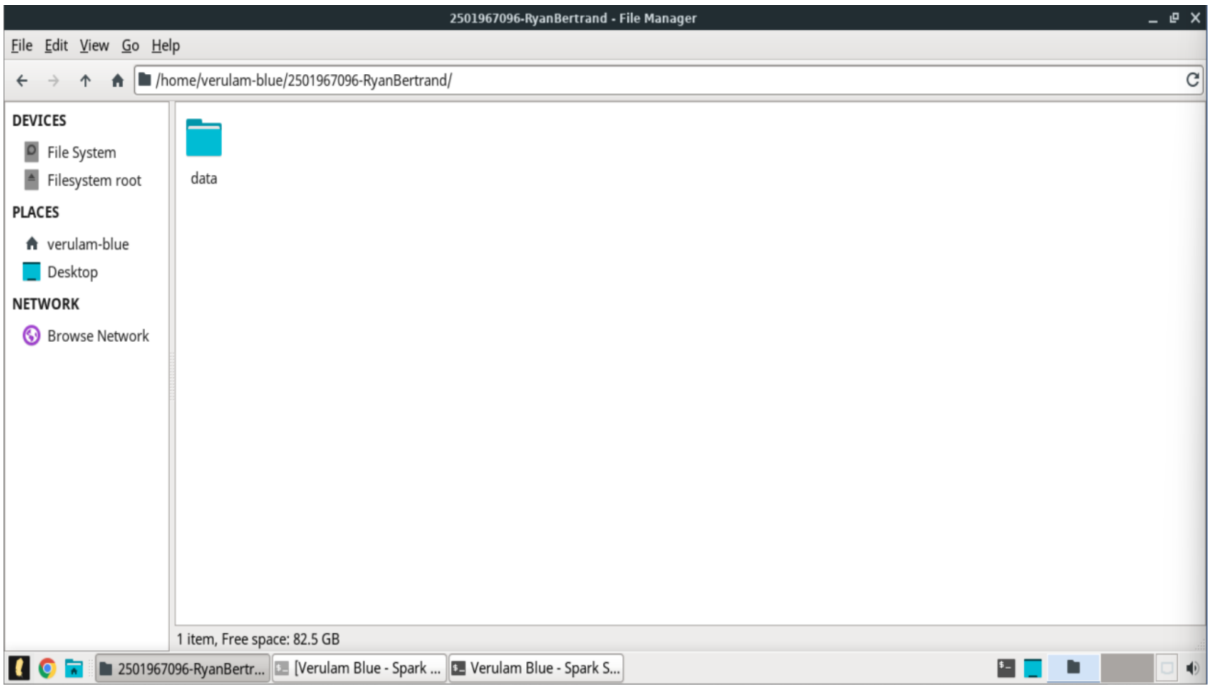
## Code

**Q1(2 points):** Create a directory in verulam-blue directory named YourStudentID-YourName. Display lists files in verulam-blue directory.

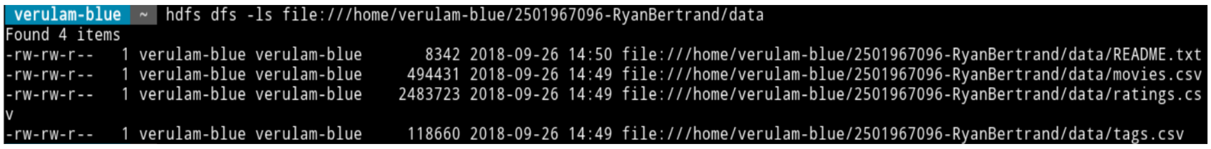


```
verulam-blue ~ hdfs dfs -mkdir file:///home/verulam-blue/2501967096-RyanBertrand
```

**Q2(2 points):** Move folder data into your working directory and display lists files in it

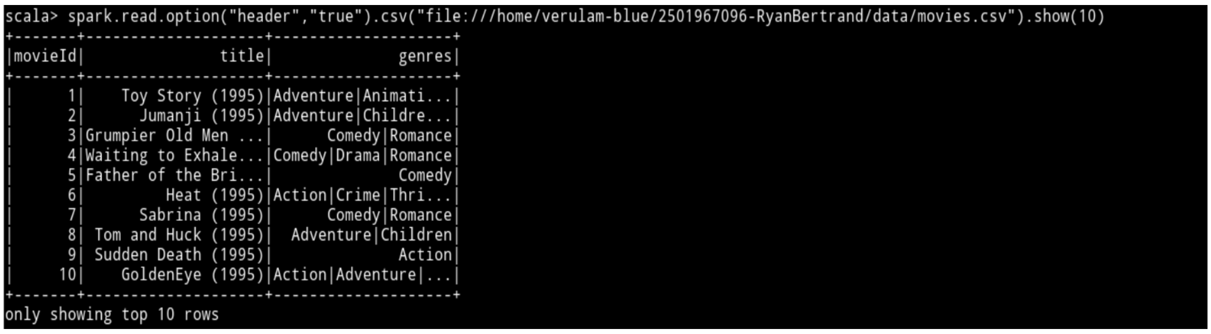


2501967096-RyanBertrand - File Manager
File Edit View Go Help
/home/verulam-blue/2501967096-RyanBertrand/
DEVICES
File System
Filesystem root
PLACES
verulam-blue
Desktop
NETWORK
Browse Network
data
1 item, Free space: 82.5 GB
2501967096-RyanBertr...
[Verulam Blue - Spark ...
Verulam Blue - Spark S...



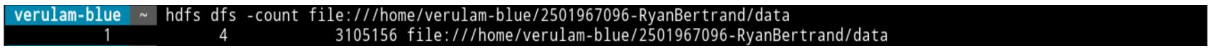
```
verulam-blue ~ hdfs dfs -ls file:///home/verulam-blue/2501967096-RyanBertrand/data
Found 4 items
-rw-rw-r--   1 verulam-blue verulam-blue       8342 2018-09-26 14:50 file:///home/verulam-blue/2501967096-RyanBertrand/data/README.txt
-rw-rw-r--   1 verulam-blue verulam-blue     494431 2018-09-26 14:49 file:///home/verulam-blue/2501967096-RyanBertrand/data/movies.csv
-rw-rw-r--   1 verulam-blue verulam-blue    2483723 2018-09-26 14:49 file:///home/verulam-blue/2501967096-RyanBertrand/data/ratings.cs
v
-rw-rw-r--   1 verulam-blue verulam-blue     118660 2018-09-26 14:49 file:///home/verulam-blue/2501967096-RyanBertrand/data/tags.csv
```

**Q3(2 points):** Show the first 10 contents of movies.csv file



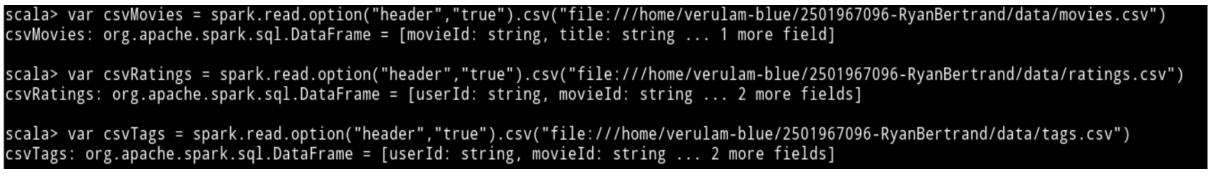
```
scala> spark.read.option("header","true").csv("file:///home/verulam-blue/2501967096-RyanBertrand/data/movies.csv").show(10)
+-------+--------------------+--------------------+
|movieId|               title|              genres|
+-------+--------------------+--------------------+
|      1|    Toy Story (1995)|Adventure|Animati...|
|      2|      Jumanji (1995)|Adventure|Childre...|
|      3|Grumpier Old Men ...|      Comedy|Romance|
|      4|Waiting to Exhale...|Comedy|Drama|Romance|
|      5|Father of the Bri...|              Comedy|
|      6|         Heat (1995)|Action|Crime|Thri...|
|      7|      Sabrina (1995)|      Comedy|Romance|
|      8| Tom and Huck (1995)|  Adventure|Children|
|      9| Sudden Death (1995)|              Action|
|     10|    GoldenEye (1995)|Action|Adventure|...|
+-------+--------------------+--------------------+
only showing top 10 rows
```

**Q4(2 points):** Display number of directory, number of files and file sizes of your working directory

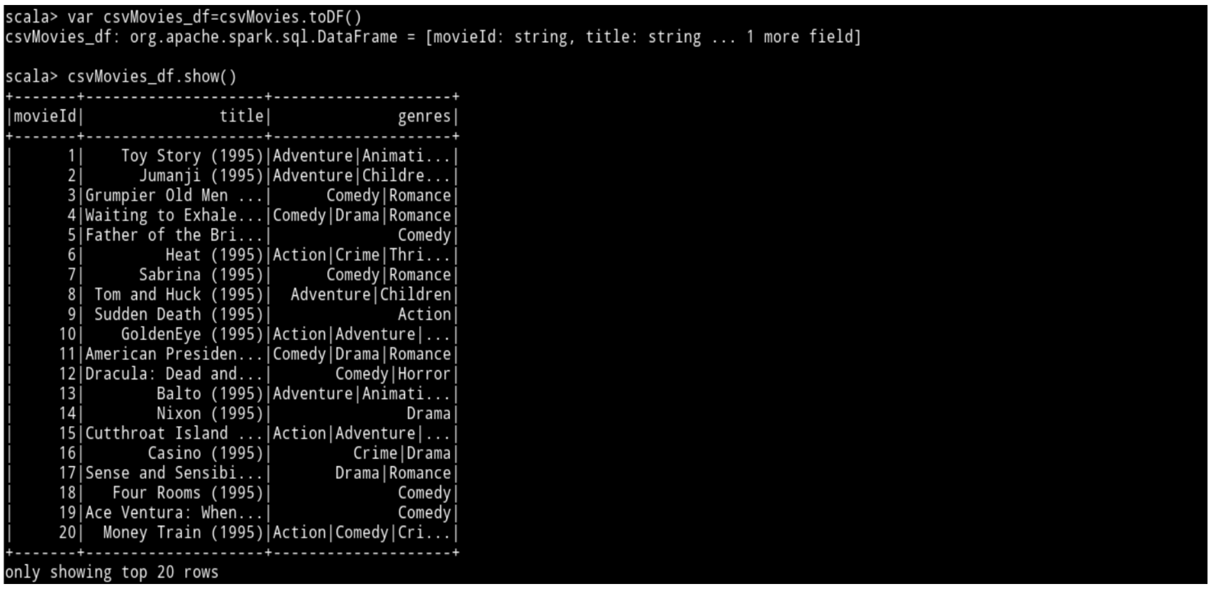


```
verulam-blue ~ hdfs dfs -count file:///home/verulam-blue/2501967096-RyanBertrand/data
           1            4            3105156 file:///home/verulam-blue/2501967096-RyanBertrand/data
```

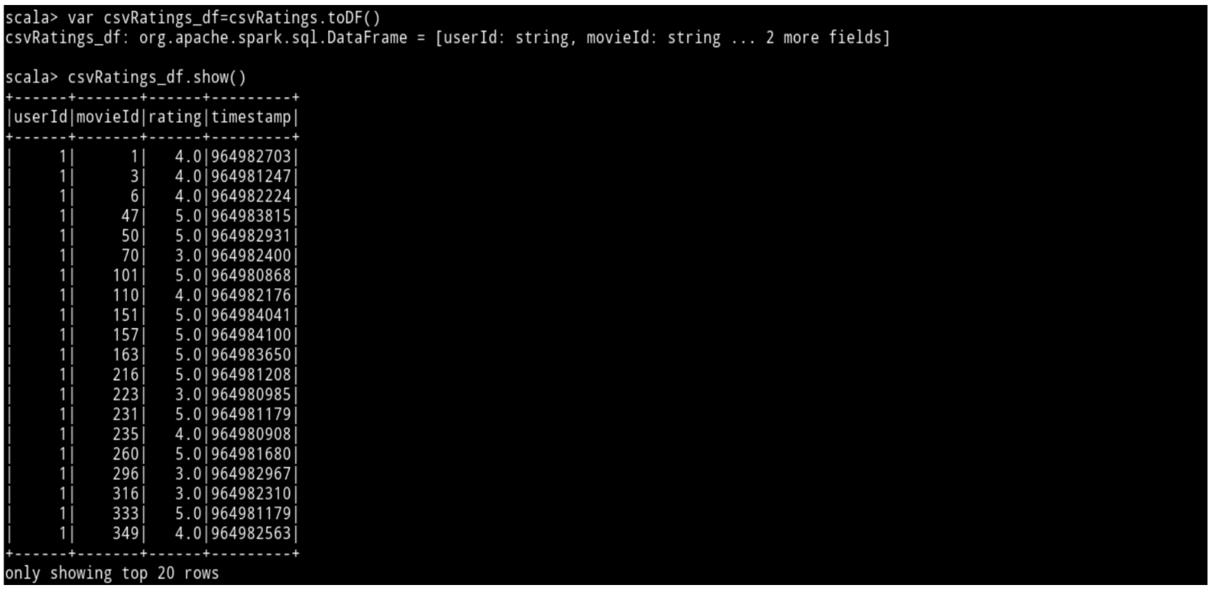
**Q5(3 points):** Create 3 dataframes from CSV filesand get a glimpse of the data.



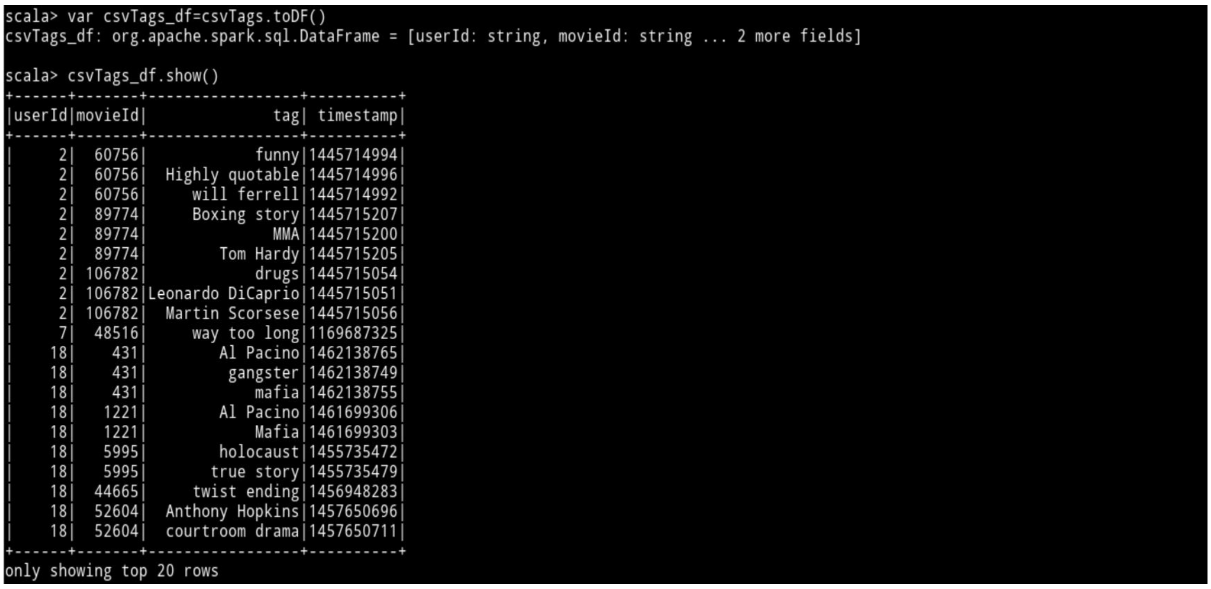
```
scala> var csvMovies = spark.read.option("header","true").csv("file:///home/verulam-blue/2501967096-RyanBertrand/data/movies.csv")
csvMovies: org.apache.spark.sql.DataFrame = [movieId: string, title: string ... 1 more field]

scala> var csvRatings = spark.read.option("header","true").csv("file:///home/verulam-blue/2501967096-RyanBertrand/data/ratings.csv")
csvRatings: org.apache.spark.sql.DataFrame = [userId: string, movieId: string ... 2 more fields]

scala> var csvTags = spark.read.option("header","true").csv("file:///home/verulam-blue/2501967096-RyanBertrand/data/tags.csv")
csvTags: org.apache.spark.sql.DataFrame = [userId: string, movieId: string ... 2 more fields]
```



```
scala> var csvMovies_df=csvMovies.toDF()
csvMovies_df: org.apache.spark.sql.DataFrame = [movieId: string, title: string ... 1 more field]

scala> csvMovies_df.show()
+-------+--------------------+--------------------+
|movieId|               title|              genres|
+-------+--------------------+--------------------+
|      1|    Toy Story (1995)|Adventure|Animati...|
|      2|      Jumanji (1995)|Adventure|Childre...|
|      3|Grumpier Old Men ...|      Comedy|Romance|
|      4|Waiting to Exhale...|Comedy|Drama|Romance|
|      5|Father of the Bri...|              Comedy|
|      6|         Heat (1995)|Action|Crime|Thri...|
|      7|      Sabrina (1995)|      Comedy|Romance|
|      8| Tom and Huck (1995)|  Adventure|Children|
|      9| Sudden Death (1995)|              Action|
|     10|    GoldenEye (1995)|Action|Adventure|...|
|     11|American Presiden...|Comedy|Drama|Romance|
|     12|Dracula: Dead and...|       Comedy|Horror|
|     13|        Balto (1995)|Adventure|Animati...|
|     14|        Nixon (1995)|               Drama|
|     15|Cutthroat Island ...|Action|Adventure|...|
|     16|       Casino (1995)|         Crime|Drama|
|     17|Sense and Sensibi...|       Drama|Romance|
|     18|   Four Rooms (1995)|              Comedy|
|     19|Ace Ventura: When...|              Comedy|
|     20|  Money Train (1995)|Action|Comedy|Cri...|
+-------+--------------------+--------------------+
only showing top 20 rows
```

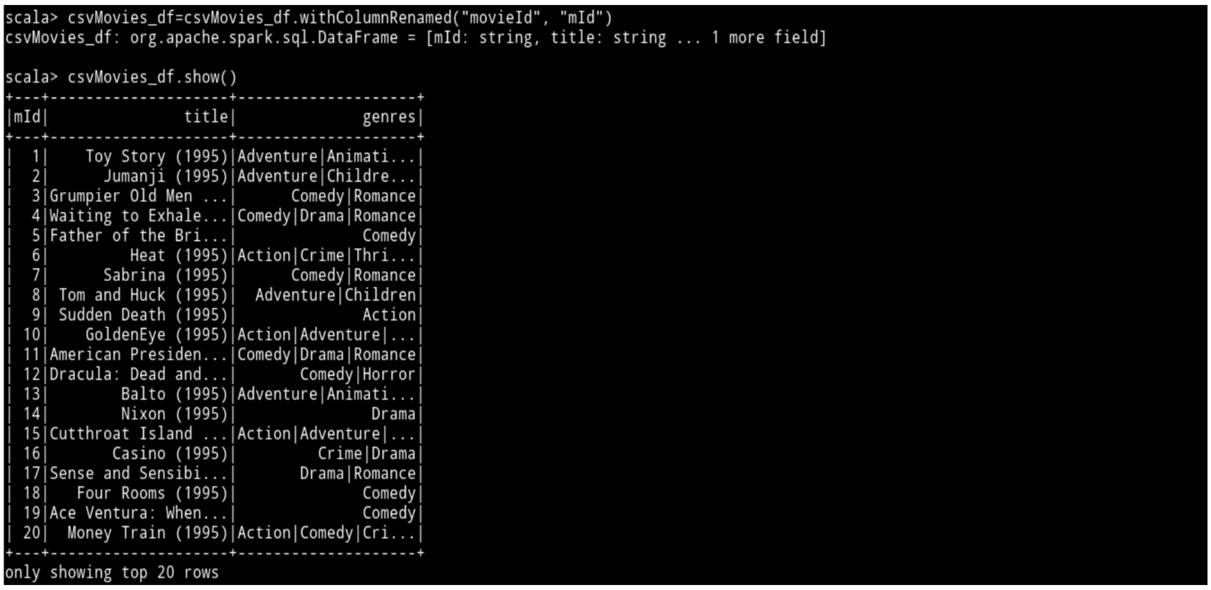


```
scala> var csvRatings_df=csvRatings.toDF()
csvRatings_df: org.apache.spark.sql.DataFrame = [userId: string, movieId: string ... 2 more fields]

scala> csvRatings_df.show()
+------+-------+------+---------+
|userId|movieId|rating|timestamp|
+------+-------+------+---------+
|     1|      1|   4.0|964982703|
|     1|      3|   4.0|964981247|
|     1|      6|   4.0|964982224|
|     1|     47|   5.0|964983815|
|     1|     50|   5.0|964982931|
|     1|     70|   3.0|964982400|
|     1|    101|   5.0|964980868|
|     1|    110|   4.0|964982176|
|     1|    151|   5.0|964984041|
|     1|    157|   5.0|964984100|
|     1|    163|   5.0|964983650|
|     1|    216|   5.0|964981208|
|     1|    223|   3.0|964980985|
|     1|    231|   5.0|964981179|
|     1|    235|   4.0|964980908|
|     1|    260|   5.0|964981680|
|     1|    296|   3.0|964982967|
|     1|    316|   3.0|964982310|
|     1|    333|   5.0|964981179|
|     1|    349|   4.0|964982563|
+------+-------+------+---------+
only showing top 20 rows
```



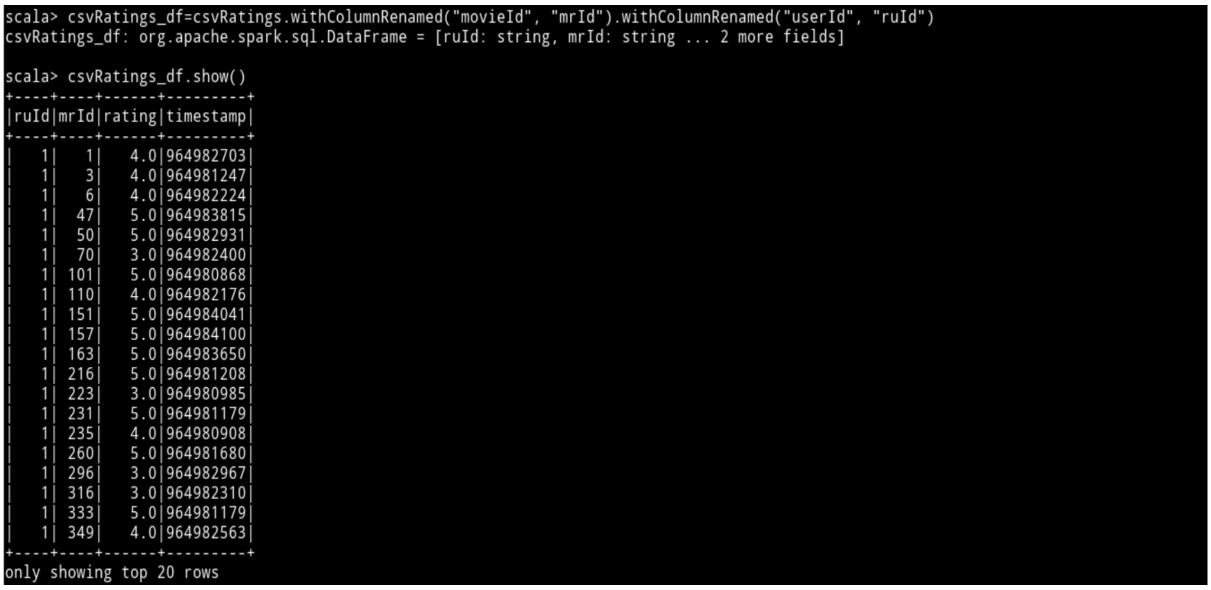
```
scala> var csvTags_df=csvTags.toDF()
csvTags_df: org.apache.spark.sql.DataFrame = [userId: string, movieId: string ... 2 more fields]

scala> csvTags_df.show()
+------+-------+-----------------+----------+
|userId|movieId|              tag| timestamp|
+------+-------+-----------------+----------+
|     2|  60756|            funny|1445714994|
|     2|  60756|  Highly quotable|1445714996|
|     2|  60756|     will ferrell|1445714992|
|     2|  89774|     Boxing story|1445715207|
|     2|  89774|              MMA|1445715200|
|     2|  89774|        Tom Hardy|1445715205|
|     2| 106782|            drugs|1445715054|
|     2| 106782|Leonardo DiCaprio|1445715051|
|     2| 106782|  Martin Scorsese|1445715056|
|     7|  48516|     way too long|1169687325|
|    18|    431|        Al Pacino|1462138765|
|    18|    431|         gangster|1462138749|
|    18|    431|            mafia|1462138755|
|    18|   1221|        Al Pacino|1461699306|
|    18|   1221|            Mafia|1461699303|
|    18|   5995|        holocaust|1455735472|
|    18|   5995|       true story|1455735479|
|    18|  44665|     twist ending|1456948283|
|    18|  52604|  Anthony Hopkins|1457650696|
|    18|  52604|  courtroom drama|1457650711|
+------+-------+-----------------+----------+
only showing top 20 rows
```

**Q6(5 points):** Rename the following columns using operation for structured Transformation.
a. [movies dataframe] --> Rename movieId with mId



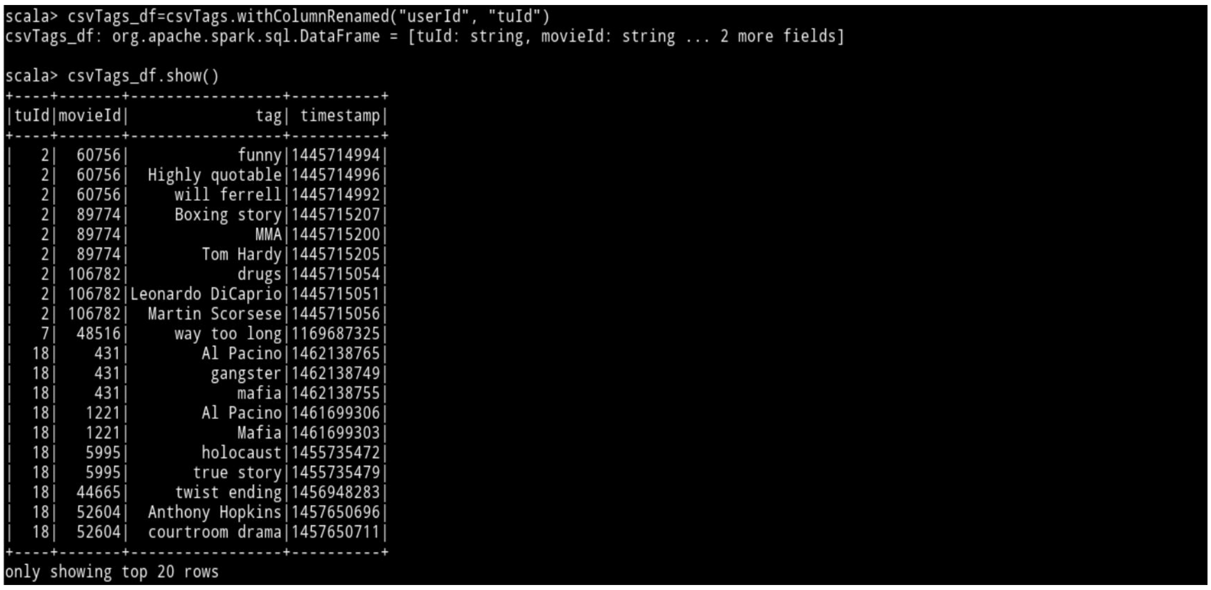
```
scala> csvMovies_df=csvMovies_df.withColumnRenamed("movieId", "mId")
csvMovies_df: org.apache.spark.sql.DataFrame = [mId: string, title: string ... 1 more field]

scala> csvMovies_df.show()
+---+--------------------+--------------------+
|mId|               title|              genres|
+---+--------------------+--------------------+
|  1|    Toy Story (1995)|Adventure|Animati...|
|  2|      Jumanji (1995)|Adventure|Childre...|
|  3|Grumpier Old Men ...|      Comedy|Romance|
|  4|Waiting to Exhale...|Comedy|Drama|Romance|
|  5|Father of the Bri...|              Comedy|
|  6|         Heat (1995)|Action|Crime|Thri...|
|  7|      Sabrina (1995)|      Comedy|Romance|
|  8| Tom and Huck (1995)|  Adventure|Children|
|  9| Sudden Death (1995)|              Action|
| 10|    GoldenEye (1995)|Action|Adventure|...|
| 11|American Presiden...|Comedy|Drama|Romance|
| 12|Dracula: Dead and...|       Comedy|Horror|
| 13|        Balto (1995)|Adventure|Animati...|
| 14|        Nixon (1995)|               Drama|
| 15|Cutthroat Island ...|Action|Adventure|...|
| 16|       Casino (1995)|         Crime|Drama|
| 17|Sense and Sensibi...|       Drama|Romance|
| 18|   Four Rooms (1995)|              Comedy|
| 19|Ace Ventura: When...|              Comedy|
| 20|  Money Train (1995)|Action|Comedy|Cri...|
+---+--------------------+--------------------+
only showing top 20 rows
```

b. [rating dataframe] --> Rename movieId with mrId & userId with ruId



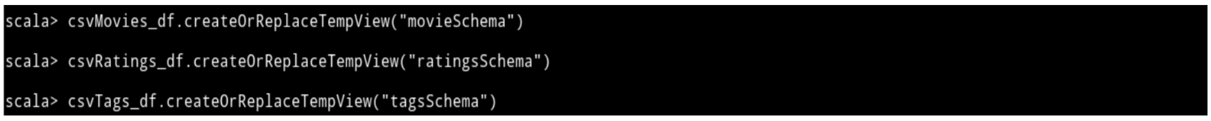
```
scala> csvRatings_df=csvRatings.withColumnRenamed("movieId", "mrId").withColumnRenamed("userId", "ruId")
csvRatings_df: org.apache.spark.sql.DataFrame = [ruId: string, mrId: string ... 2 more fields]

scala> csvRatings_df.show()
+----+----+------+---------+
|ruId|mrId|rating|timestamp|
+----+----+------+---------+
|   1|   1|   4.0|964982703|
|   1|   3|   4.0|964981247|
|   1|   6|   4.0|964982224|
|   1|  47|   5.0|964983815|
|   1|  50|   5.0|964982931|
|   1|  70|   3.0|964982400|
|   1| 101|   5.0|964980868|
|   1| 110|   4.0|964982176|
|   1| 151|   5.0|964984041|
|   1| 157|   5.0|964984100|
|   1| 163|   5.0|964983650|
|   1| 216|   5.0|964981208|
|   1| 223|   3.0|964980985|
|   1| 231|   5.0|964981179|
|   1| 235|   4.0|964980908|
|   1| 260|   5.0|964981680|
|   1| 296|   3.0|964982967|
|   1| 316|   3.0|964982310|
|   1| 333|   5.0|964981179|
|   1| 349|   4.0|964982563|
+----+----+------+---------+
only showing top 20 rows
```

c. [tags dataframe] --> Rename userId with tuId



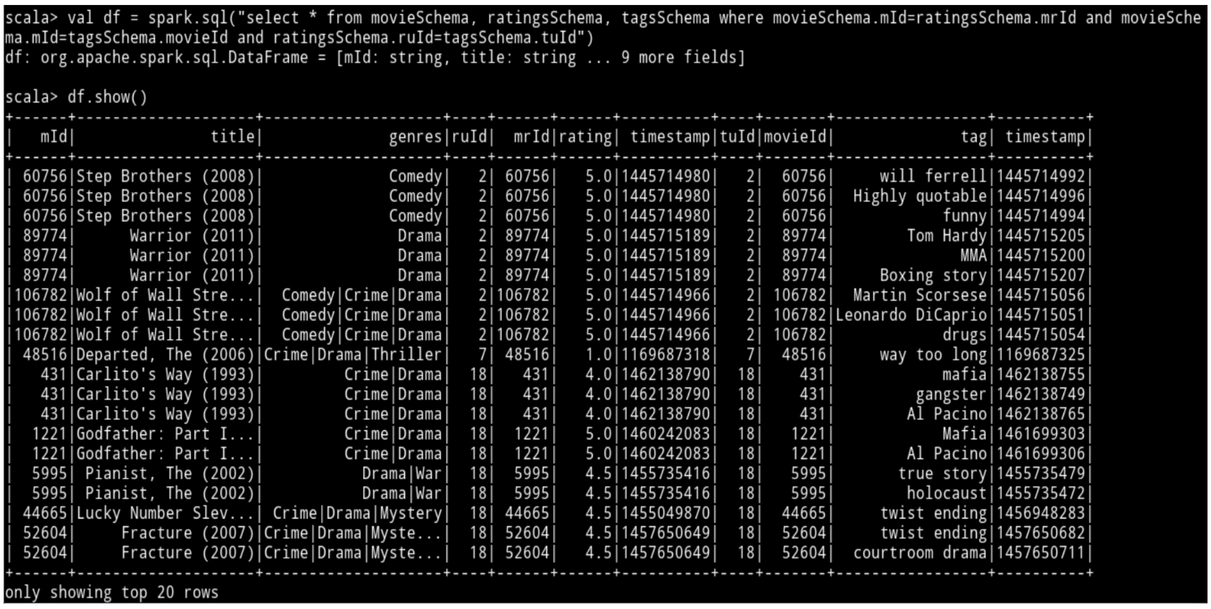
```
scala> csvTags_df=csvTags.withColumnRenamed("userId", "tuId")
csvTags_df: org.apache.spark.sql.DataFrame = [tuId: string, movieId: string ... 2 more fields]

scala> csvTags_df.show()
+----+-------+-----------------+----------+
|tuId|movieId|              tag| timestamp|
+----+-------+-----------------+----------+
|   2|  60756|            funny|1445714994|
|   2|  60756|  Highly quotable|1445714996|
|   2|  60756|     will ferrell|1445714992|
|   2|  89774|     Boxing story|1445715207|
|   2|  89774|              MMA|1445715200|
|   2|  89774|        Tom Hardy|1445715205|
|   2| 106782|            drugs|1445715054|
|   2| 106782|Leonardo DiCaprio|1445715051|
|   2| 106782|  Martin Scorsese|1445715056|
|   7|  48516|     way too long|1169687325|
|  18|    431|        Al Pacino|1462138765|
|  18|    431|         gangster|1462138749|
|  18|    431|            mafia|1462138755|
|  18|   1221|        Al Pacino|1461699306|
|  18|   1221|            Mafia|1461699303|
|  18|   5995|        holocaust|1455735472|
|  18|   5995|       true story|1455735479|
|  18|  44665|     twist ending|1456948283|
|  18|  52604|  Anthony Hopkins|1457650696|
|  18|  52604|  courtroom drama|1457650711|
+----+-------+-----------------+----------+
only showing top 20 rows
```

**Q7(3 points):** Register the data frames as 3 SQL tables.



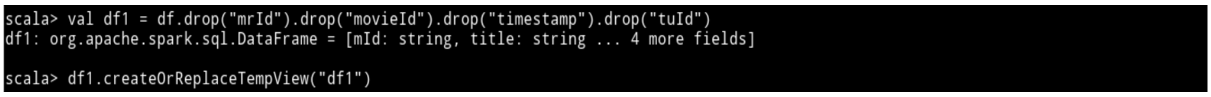
```
scala> csvMovies_df.createOrReplaceTempView("movieSchema")

scala> csvRatings_df.createOrReplaceTempView("ratingsSchema")

scala> csvTags_df.createOrReplaceTempView("tagsSchema")
```

**Q8(5 points):** Combine all records from three tables into one, named df. You just need records whenever movieId values from movies table are matching with movieId from rating and tag tables as well as userId values from rating table are matching with those in tag table, using an SQL Query



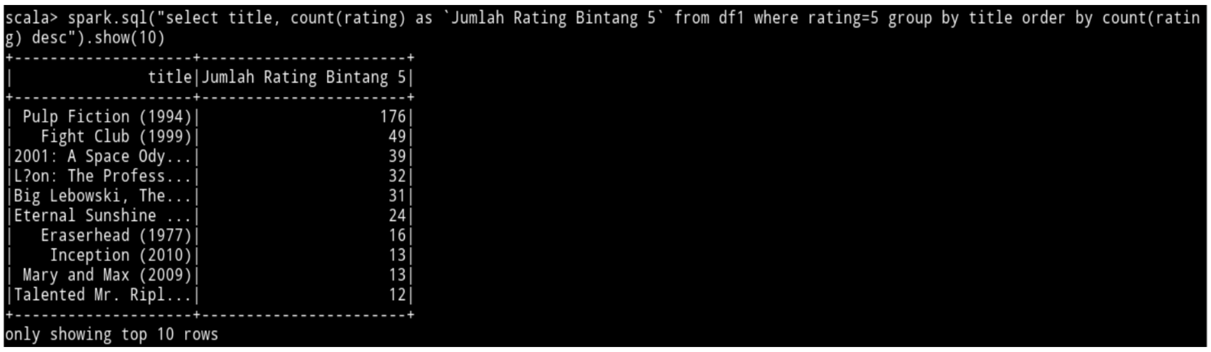
```
scala> val df = spark.sql("select * from movieSchema, ratingsSchema, tagsSchema where movieSchema.mId=ratingsSchema.mrId and movieSche
ma.mId=tagsSchema.movieId and ratingsSchema.ruId=tagsSchema.tuId")
df: org.apache.spark.sql.DataFrame = [mId: string, title: string ... 9 more fields]

scala> df.show()
+------+--------------------+--------------------+----+------+------+----------+----+-------+-----------------+----------+
|   mId|               title|              genres|ruId|  mrId|rating| timestamp|tuId|movieId|              tag| timestamp|
+------+--------------------+--------------------+----+------+------+----------+----+-------+-----------------+----------+
| 60756|Step Brothers (2008)|              Comedy|   2| 60756|   5.0|1445714980|   2|  60756|     will ferrell|1445714992|
| 60756|Step Brothers (2008)|              Comedy|   2| 60756|   5.0|1445714980|   2|  60756|  Highly quotable|1445714996|
| 60756|Step Brothers (2008)|              Comedy|   2| 60756|   5.0|1445714980|   2|  60756|            funny|1445714994|
| 89774|      Warrior (2011)|               Drama|   2| 89774|   5.0|1445715189|   2|  89774|        Tom Hardy|1445715205|
| 89774|      Warrior (2011)|               Drama|   2| 89774|   5.0|1445715189|   2|  89774|              MMA|1445715200|
| 89774|      Warrior (2011)|               Drama|   2| 89774|   5.0|1445715189|   2|  89774|     Boxing story|1445715207|
|106782|Wolf of Wall Stre...|  Comedy|Crime|Drama|   2|106782|   5.0|1445714966|   2| 106782|  Martin Scorsese|1445715056|
|106782|Wolf of Wall Stre...|  Comedy|Crime|Drama|   2|106782|   5.0|1445714966|   2| 106782|Leonardo DiCaprio|1445715051|
|106782|Wolf of Wall Stre...|  Comedy|Crime|Drama|   2|106782|   5.0|1445714966|   2| 106782|            drugs|1445715054|
| 48516|Departed, The (2006)|Crime|Drama|Thriller|   7| 48516|   1.0|1169687318|   7|  48516|     way too long|1169687325|
|   431|Carlito's Way (1993)|         Crime|Drama|  18|   431|   4.0|1462138790|  18|    431|            mafia|1462138755|
|   431|Carlito's Way (1993)|         Crime|Drama|  18|   431|   4.0|1462138790|  18|    431|         gangster|1462138749|
|   431|Carlito's Way (1993)|         Crime|Drama|  18|   431|   4.0|1462138790|  18|    431|        Al Pacino|1462138765|
|  1221|Godfather: Part I...|         Crime|Drama|  18|  1221|   5.0|1460242083|  18|   1221|            Mafia|1461699303|
|  1221|Godfather: Part I...|         Crime|Drama|  18|  1221|   5.0|1460242083|  18|   1221|        Al Pacino|1461699306|
|  5995| Pianist, The (2002)|           Drama|War|  18|  5995|   4.5|1455735416|  18|   5995|       true story|1455735479|
|  5995| Pianist, The (2002)|           Drama|War|  18|  5995|   4.5|1455735416|  18|   5995|        holocaust|1455735472|
| 44665|Lucky Number Slev...| Crime|Drama|Mystery|  18| 44665|   4.5|1455049870|  18|  44665|     twist ending|1456948283|
| 52604|     Fracture (2007)|Crime|Drama|Myste...|  18| 52604|   4.5|1457650649|  18|  52604|     twist ending|1457650682|
| 52604|     Fracture (2007)|Crime|Drama|Myste...|  18| 52604|   4.5|1457650649|  18|  52604|  courtroom drama|1457650711|
+------+--------------------+--------------------+----+------+------+----------+----+-------+-----------------+----------+
only showing top 20 rows
```

**Q9(3 points):** Remove the specified columns from dataframes mrId, movieId(from tags dataframe), timestamp, and tuId. Then, register the dataframe as an SQL table named df1



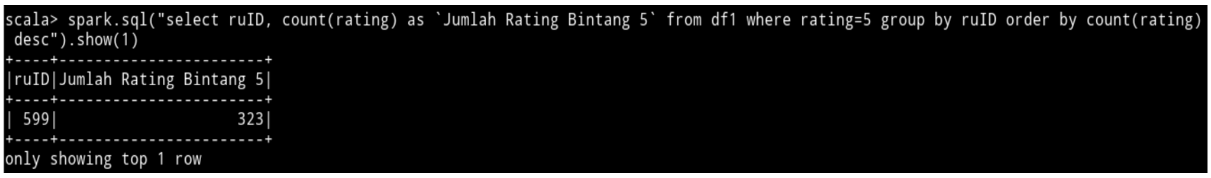
```
scala> val df1 = df.drop("mrId").drop("movieId").drop("timestamp").drop("tuId")
df1: org.apache.spark.sql.DataFrame = [mId: string, title: string ... 4 more fields]

scala> df1.createOrReplaceTempView("df1")
```

**Q10(5 points):** select the 10th most rated 5 star movies in a decresing count of times the movie got 5 star rating, using an SQL query



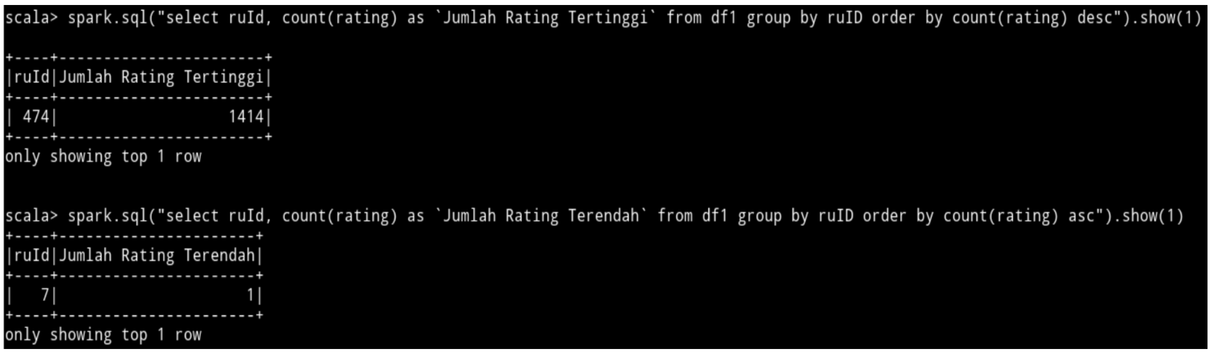
```
scala> spark.sql("select title, count(rating) as `Jumlah Rating Bintang 5` from df1 where rating=5 group by title order by count(ratin
g) desc").show(10)
+--------------------+-----------------------+
|               title|Jumlah Rating Bintang 5|
+--------------------+-----------------------+
| Pulp Fiction (1994)|                    176|
|   Fight Club (1999)|                     49|
|2001: A Space Ody...|                     39|
|L?on: The Profess...|                     32|
|Big Lebowski, The...|                     31|
|Eternal Sunshine ...|                     24|
|   Eraserhead (1977)|                     16|
|    Inception (2010)|                     13|
| Mary and Max (2009)|                     13|
|Talented Mr. Ripl...|                     12|
+--------------------+-----------------------+
only showing top 10 rows
```

**Q11(5 points):** Which user gave most '5' ratings? Return the userID and number of ratings, using an SQL query



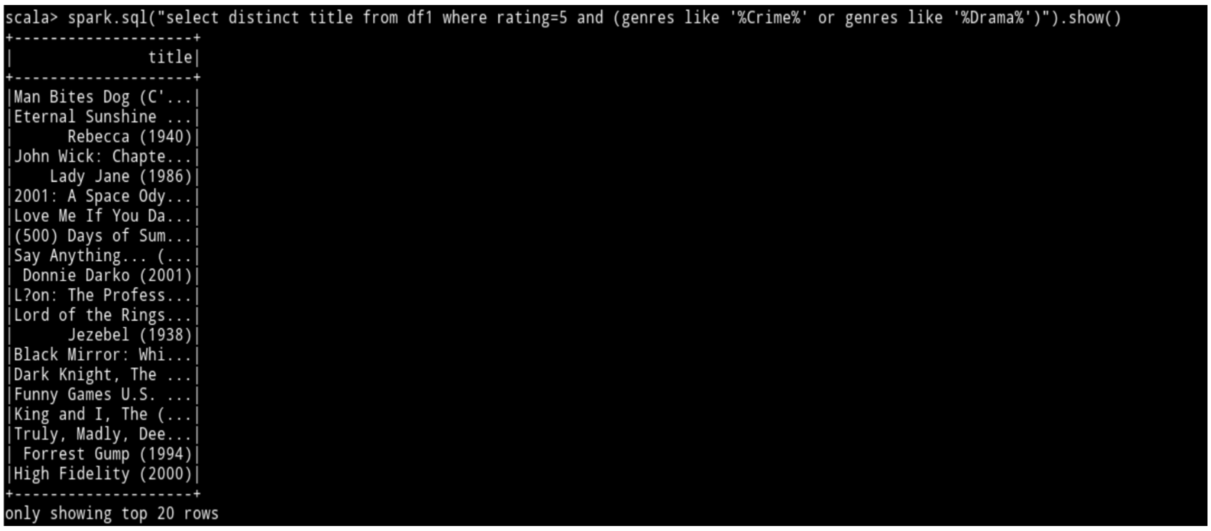
```
scala> spark.sql("select ruID, count(rating) as `Jumlah Rating Bintang 5` from df1 where rating=5 group by ruID order by count(rating)
 desc").show(1)
+----+-----------------------+
|ruID|Jumlah Rating Bintang 5|
+----+-----------------------+
| 599|                    323|
+----+-----------------------+
only showing top 1 row
```

**Q12(8 points):** check the users that have rated the most and the least number of movies, using an SQL Query



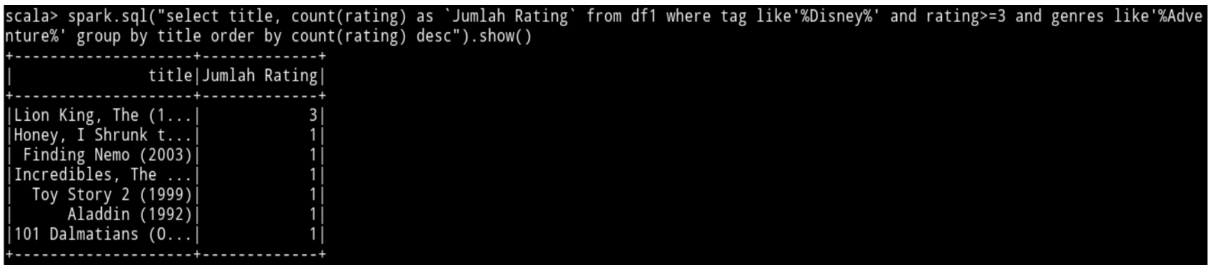
```
scala> spark.sql("select ruId, count(rating) as `Jumlah Rating Tertinggi` from df1 group by ruID order by count(rating) desc").show(1)

+----+-----------------------+
|ruId|Jumlah Rating Tertinggi|
+----+-----------------------+
| 474|                   1414|
+----+-----------------------+
only showing top 1 row


scala> spark.sql("select ruId, count(rating) as `Jumlah Rating Terendah` from df1 group by ruID order by count(rating) asc").show(1)
+----+----------------------+
|ruId|Jumlah Rating Terendah|
+----+----------------------+
|   7|                     1|
+----+----------------------+
only showing top 1 row
```

**Q13(10 points):** Get a distinct list of all movie titles that are five-star rated in the Crime/Drama genre, using an SQL Query



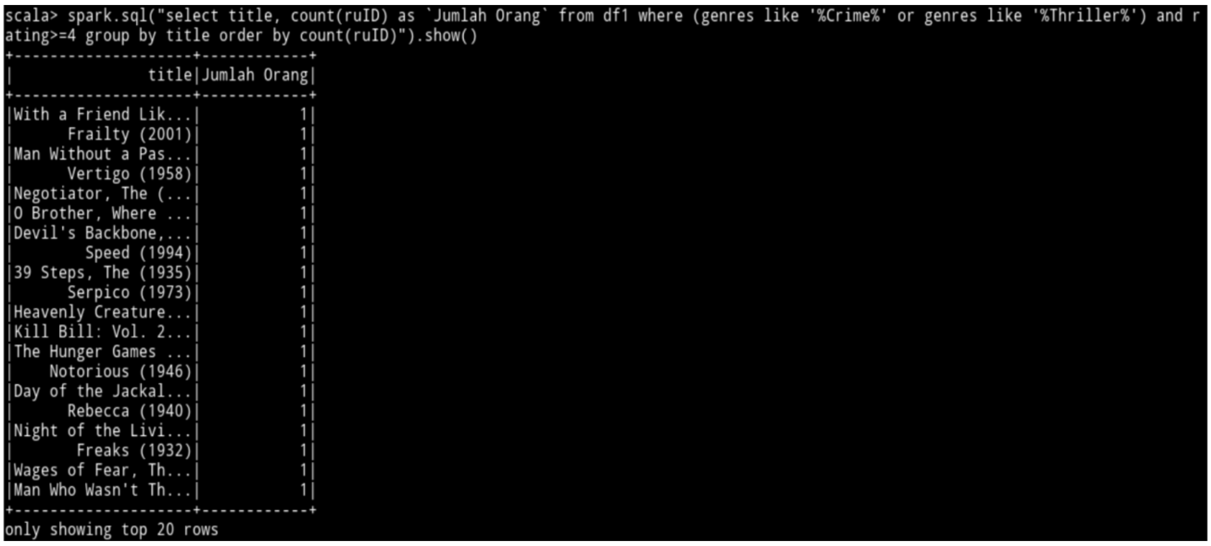
```
scala> spark.sql("select distinct title from df1 where rating=5 and (genres like '%Crime%' or genres like '%Drama%')").show()
+--------------------+
|               title|
+--------------------+
|Man Bites Dog (C'...|
|Eternal Sunshine ...|
|      Rebecca (1940)|
|John Wick: Chapte...|
|    Lady Jane (1986)|
|2001: A Space Ody...|
|Love Me If You Da...|
|(500) Days of Sum...|
|Say Anything... (...|
| Donnie Darko (2001)|
|L?on: The Profess...|
|Lord of the Rings...|
|      Jezebel (1938)|
|Black Mirror: Whi...|
|Dark Knight, The ...|
|Funny Games U.S. ...|
|King and I, The (...|
|Truly, Madly, Dee...|
| Forrest Gump (1994)|
|High Fidelity (2000)|
+--------------------+
only showing top 20 rows
```

**Q14(10 points):** Get the movie titles tagged with 'Disney' with minimum three-star rated in the Adventure genre, the number of ratings for each movie, ordered by the highest rating, using an SQL Query



```
scala> spark.sql("select title, count(rating) as `Jumlah Rating` from df1 where tag like'%Disney%' and rating>=3 and genres like'%Adve
nture%' group by title order by count(rating) desc").show()
+--------------------+-------------+
|               title|Jumlah Rating|
+--------------------+-------------+
|Lion King, The (1...|            3|
|Honey, I Shrunk t...|            1|
| Finding Nemo (2003)|            1|
|Incredibles, The ...|            1|
|  Toy Story 2 (1999)|            1|
|      Aladdin (1992)|            1|
|101 Dalmatians (O...|            1|
+--------------------+-------------+
```

**Q15(10 points):** Get a sorted count of the number of people who gave each Crime/Thriller movie a four-star or more rating, using an SQL Query



```
scala> spark.sql("select title, count(ruID) as `Jumlah Orang` from df1 where (genres like '%Crime%' or genres like '%Thriller%') and r
ating>=4 group by title order by count(ruID)").show()
+--------------------+------------+
|               title|Jumlah Orang|
+--------------------+------------+
|With a Friend Lik...|           1|
|      Frailty (2001)|           1|
|Man Without a Pas...|           1|
|      Vertigo (1958)|           1|
|Negotiator, The (...|           1|
|O Brother, Where ...|           1|
|Devil's Backbone,...|           1|
|        Speed (1994)|           1|
|39 Steps, The (1935)|           1|
|      Serpico (1973)|           1|
|Heavenly Creature...|           1|
|Kill Bill: Vol. 2...|           1|
|The Hunger Games ...|           1|
|    Notorious (1946)|           1|
|Day of the Jackal...|           1|
|      Rebecca (1940)|           1|
|Night of the Livi...|           1|
|       Freaks (1932)|           1|
|Wages of Fear, Th...|           1|
|Man Who Wasn't Th...|           1|
+--------------------+------------+
only showing top 20 rows
```