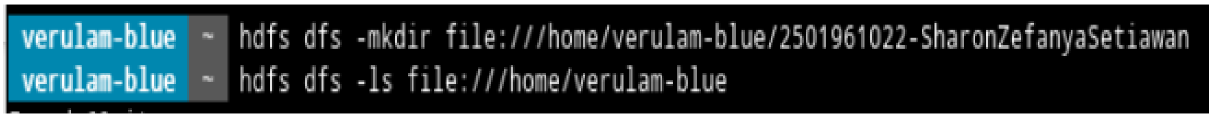
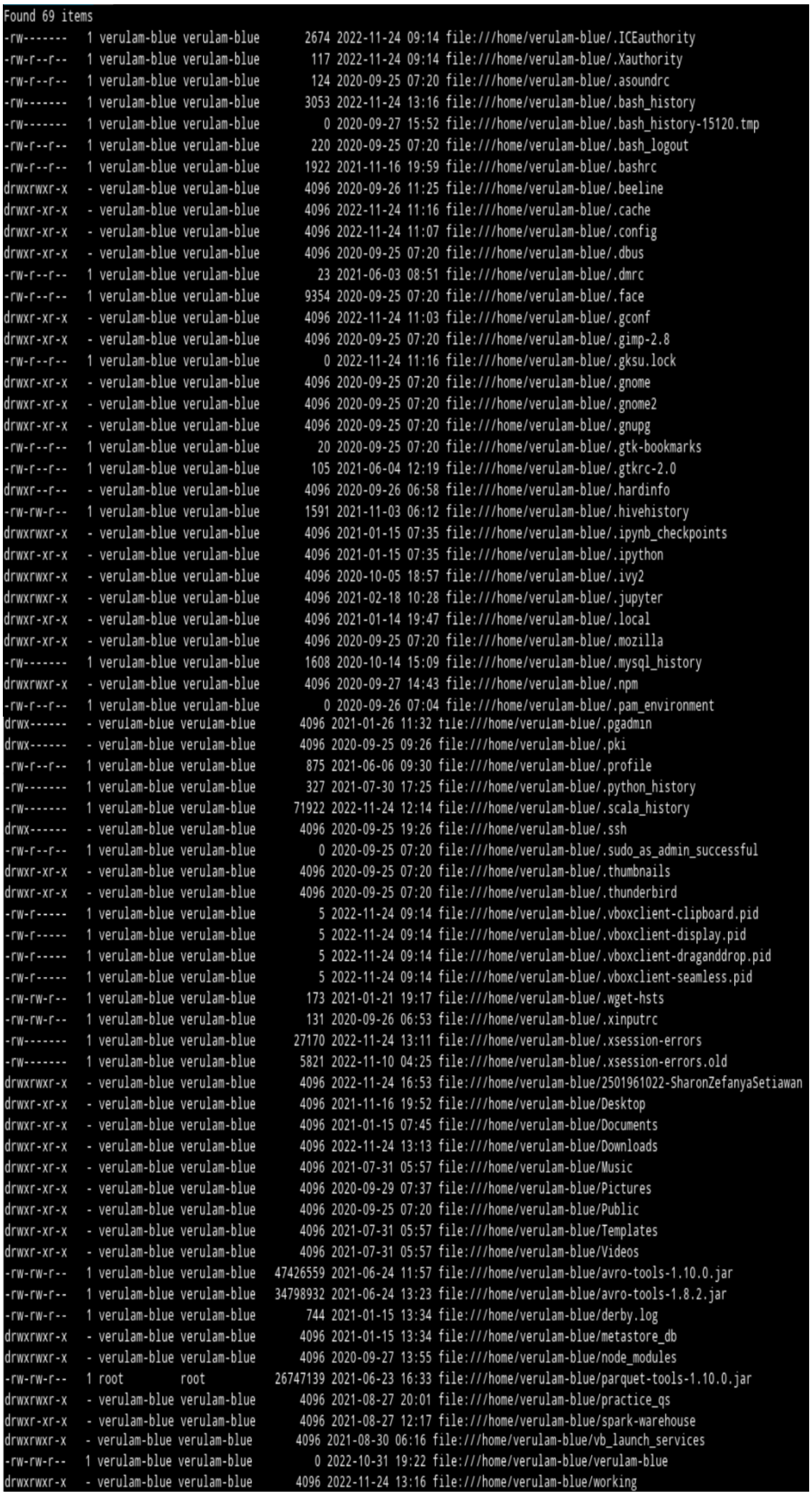
ANSWER Q1:



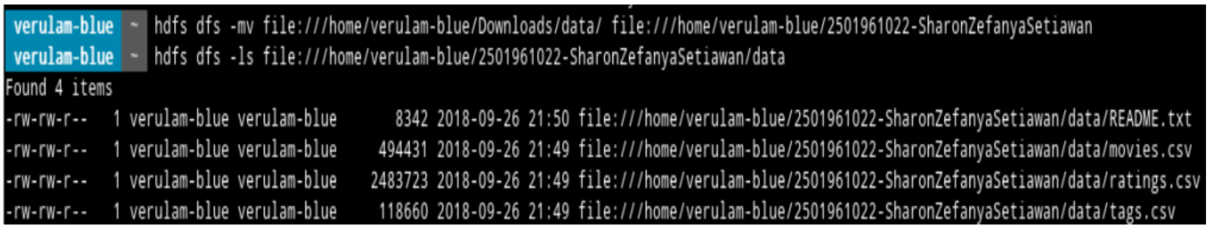
```
verulam-blue  ~  hdfs dfs -mkdir file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan
verulam-blue  ~  hdfs dfs -ls file:///home/verulam-blue
```

- Output: (dipisah agar query di atas dapat terbaca)



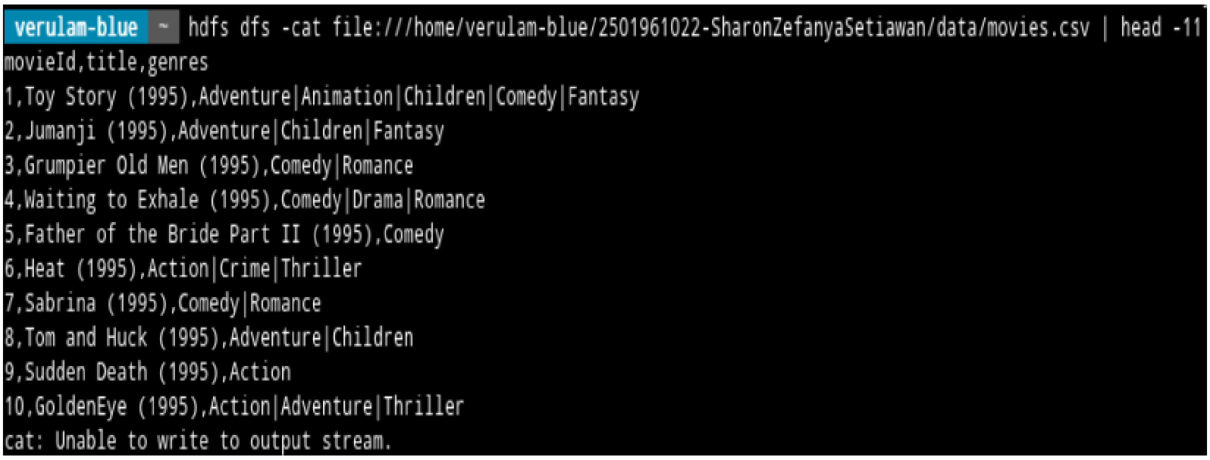
```
Found 69 items
-rw-------   1 verulam-blue verulam-blue       2674 2022-11-24 09:14 file:///home/verulam-blue/.ICEauthority
-rw-r--r--   1 verulam-blue verulam-blue        117 2022-11-24 09:14 file:///home/verulam-blue/.Xauthority
-rw-r--r--   1 verulam-blue verulam-blue        124 2020-09-25 07:20 file:///home/verulam-blue/.asoundrc
-rw-------   1 verulam-blue verulam-blue       3053 2022-11-24 13:16 file:///home/verulam-blue/.bash_history
-rw-------   1 verulam-blue verulam-blue          0 2020-09-27 15:52 file:///home/verulam-blue/.bash_history-15120.tmp
-rw-r--r--   1 verulam-blue verulam-blue        220 2020-09-25 07:20 file:///home/verulam-blue/.bash_logout
-rw-r--r--   1 verulam-blue verulam-blue       1922 2021-11-16 19:59 file:///home/verulam-blue/.bashrc
drwxrwxr-x   - verulam-blue verulam-blue       4096 2020-09-26 11:25 file:///home/verulam-blue/.beeline
drwxr-xr-x   - verulam-blue verulam-blue       4096 2022-11-24 11:16 file:///home/verulam-blue/.cache
drwxr-xr-x   - verulam-blue verulam-blue       4096 2022-11-24 11:07 file:///home/verulam-blue/.config
drwxr-xr-x   - verulam-blue verulam-blue       4096 2020-09-25 07:20 file:///home/verulam-blue/.dbus
-rw-r--r--   1 verulam-blue verulam-blue         23 2021-06-03 08:51 file:///home/verulam-blue/.dmrc
-rw-r--r--   1 verulam-blue verulam-blue       9354 2020-09-25 07:20 file:///home/verulam-blue/.face
drwxr-xr-x   - verulam-blue verulam-blue       4096 2022-11-24 11:03 file:///home/verulam-blue/.gconf
drwxr-xr-x   - verulam-blue verulam-blue       4096 2020-09-25 07:20 file:///home/verulam-blue/.gimp-2.8
-rw-r--r--   1 verulam-blue verulam-blue          0 2022-11-24 11:16 file:///home/verulam-blue/.gksu.lock
drwxr-xr-x   - verulam-blue verulam-blue       4096 2020-09-25 07:20 file:///home/verulam-blue/.gnome
drwxr-xr-x   - verulam-blue verulam-blue       4096 2020-09-25 07:20 file:///home/verulam-blue/.gnome2
drwxr-xr-x   - verulam-blue verulam-blue       4096 2020-09-25 07:20 file:///home/verulam-blue/.gnupg
-rw-r--r--   1 verulam-blue verulam-blue         20 2020-09-25 07:20 file:///home/verulam-blue/.gtk-bookmarks
-rw-r--r--   1 verulam-blue verulam-blue        105 2021-06-04 12:19 file:///home/verulam-blue/.gtkrc-2.0
drwxr--r--   - verulam-blue verulam-blue       4096 2020-09-26 06:58 file:///home/verulam-blue/.hardinfo
-rw-rw-r--   1 verulam-blue verulam-blue       1591 2021-11-03 06:12 file:///home/verulam-blue/.hivehistory
drwxrwxr-x   - verulam-blue verulam-blue       4096 2021-01-15 07:35 file:///home/verulam-blue/.ipynb_checkpoints
drwxr-xr-x   - verulam-blue verulam-blue       4096 2021-01-15 07:35 file:///home/verulam-blue/.ipython
drwxrwxr-x   - verulam-blue verulam-blue       4096 2020-10-05 18:57 file:///home/verulam-blue/.ivy2
drwxrwxr-x   - verulam-blue verulam-blue       4096 2021-02-18 10:28 file:///home/verulam-blue/.jupyter
drwxr-xr-x   - verulam-blue verulam-blue       4096 2021-01-14 19:47 file:///home/verulam-blue/.local
drwxr-xr-x   - verulam-blue verulam-blue       4096 2020-09-25 07:20 file:///home/verulam-blue/.mozilla
-rw-------   1 verulam-blue verulam-blue       1608 2020-10-14 15:09 file:///home/verulam-blue/.mysql_history
drwxrwxr-x   - verulam-blue verulam-blue       4096 2020-09-27 14:43 file:///home/verulam-blue/.npm
-rw-r--r--   1 verulam-blue verulam-blue          0 2020-09-26 07:04 file:///home/verulam-blue/.pam_environment
drwx------   - verulam-blue verulam-blue       4096 2021-01-26 11:32 file:///home/verulam-blue/.pgadmin
drwx------   - verulam-blue verulam-blue       4096 2020-09-25 09:26 file:///home/verulam-blue/.pki
-rw-r--r--   1 verulam-blue verulam-blue        875 2021-06-06 09:30 file:///home/verulam-blue/.profile
-rw-------   1 verulam-blue verulam-blue        327 2021-07-30 17:25 file:///home/verulam-blue/.python_history
-rw-------   1 verulam-blue verulam-blue      71922 2022-11-24 12:14 file:///home/verulam-blue/.scala_history
drwx------   - verulam-blue verulam-blue       4096 2020-09-25 19:26 file:///home/verulam-blue/.ssh
-rw-r--r--   1 verulam-blue verulam-blue          0 2020-09-25 07:20 file:///home/verulam-blue/.sudo_as_admin_successful
drwxr-xr-x   - verulam-blue verulam-blue       4096 2020-09-25 07:20 file:///home/verulam-blue/.thumbnails
drwxr-xr-x   - verulam-blue verulam-blue       4096 2020-09-25 07:20 file:///home/verulam-blue/.thunderbird
-rw-r-----   1 verulam-blue verulam-blue          5 2022-11-24 09:14 file:///home/verulam-blue/.vboxclient-clipboard.pid
-rw-r-----   1 verulam-blue verulam-blue          5 2022-11-24 09:14 file:///home/verulam-blue/.vboxclient-display.pid
-rw-r-----   1 verulam-blue verulam-blue          5 2022-11-24 09:14 file:///home/verulam-blue/.vboxclient-draganddrop.pid
-rw-r-----   1 verulam-blue verulam-blue          5 2022-11-24 09:14 file:///home/verulam-blue/.vboxclient-seamless.pid
-rw-rw-r--   1 verulam-blue verulam-blue        173 2021-01-21 19:17 file:///home/verulam-blue/.wget-hsts
-rw-rw-r--   1 verulam-blue verulam-blue        131 2020-09-26 06:53 file:///home/verulam-blue/.xinputrc
-rw-------   1 verulam-blue verulam-blue      27170 2022-11-24 13:11 file:///home/verulam-blue/.xsession-errors
-rw-------   1 verulam-blue verulam-blue       5821 2022-11-10 04:25 file:///home/verulam-blue/.xsession-errors.old
drwxrwxr-x   - verulam-blue verulam-blue       4096 2022-11-24 16:53 file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan
drwxr-xr-x   - verulam-blue verulam-blue       4096 2021-11-16 19:52 file:///home/verulam-blue/Desktop
drwxr-xr-x   - verulam-blue verulam-blue       4096 2021-01-15 07:45 file:///home/verulam-blue/Documents
drwxr-xr-x   - verulam-blue verulam-blue       4096 2022-11-24 13:13 file:///home/verulam-blue/Downloads
drwxr-xr-x   - verulam-blue verulam-blue       4096 2021-07-31 05:57 file:///home/verulam-blue/Music
drwxr-xr-x   - verulam-blue verulam-blue       4096 2020-09-29 07:37 file:///home/verulam-blue/Pictures
drwxr-xr-x   - verulam-blue verulam-blue       4096 2020-09-25 07:20 file:///home/verulam-blue/Public
drwxr-xr-x   - verulam-blue verulam-blue       4096 2021-07-31 05:57 file:///home/verulam-blue/Templates
drwxr-xr-x   - verulam-blue verulam-blue       4096 2021-07-31 05:57 file:///home/verulam-blue/Videos
-rw-rw-r--   1 verulam-blue verulam-blue   47426559 2021-06-24 11:57 file:///home/verulam-blue/avro-tools-1.10.0.jar
-rw-rw-r--   1 verulam-blue verulam-blue   34798932 2021-06-24 13:23 file:///home/verulam-blue/avro-tools-1.8.2.jar
-rw-rw-r--   1 verulam-blue verulam-blue        744 2021-01-15 13:34 file:///home/verulam-blue/derby.log
drwxrwxr-x   - verulam-blue verulam-blue       4096 2021-01-15 13:34 file:///home/verulam-blue/metastore_db
drwxrwxr-x   - verulam-blue verulam-blue       4096 2020-09-27 13:55 file:///home/verulam-blue/node_modules
-rw-rw-r--   1 root         root           26747139 2021-06-23 16:33 file:///home/verulam-blue/parquet-tools-1.10.0.jar
drwxrwxr-x   - verulam-blue verulam-blue       4096 2021-08-27 20:01 file:///home/verulam-blue/practice_qs
drwxr-xr-x   - verulam-blue verulam-blue       4096 2021-08-27 12:17 file:///home/verulam-blue/spark-warehouse
drwxrwxr-x   - verulam-blue verulam-blue       4096 2021-08-30 06:16 file:///home/verulam-blue/vb_launch_services
-rw-rw-r--   1 verulam-blue verulam-blue          0 2022-10-31 19:22 file:///home/verulam-blue/verulam-blue
drwxrwxr-x   - verulam-blue verulam-blue       4096 2022-11-24 13:16 file:///home/verulam-blue/working
```

ANSWER Q2:



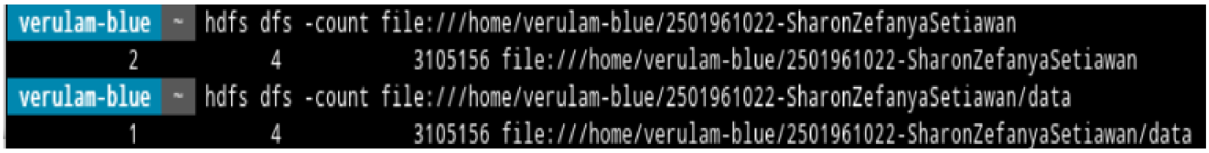
```
verulam-blue ~ hdfs dfs -mv file:///home/verulam-blue/Downloads/data/ file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan
verulam-blue ~ hdfs dfs -ls file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan/data
Found 4 items
-rw-rw-r--   1 verulam-blue verulam-blue       8342 2018-09-26 21:50 file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan/data/README.txt
-rw-rw-r--   1 verulam-blue verulam-blue     494431 2018-09-26 21:49 file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan/data/movies.csv
-rw-rw-r--   1 verulam-blue verulam-blue    2483723 2018-09-26 21:49 file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan/data/ratings.csv
-rw-rw-r--   1 verulam-blue verulam-blue     118660 2018-09-26 21:49 file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan/data/tags.csv
```

ANSWER Q3:



```
verulam-blue ~ hdfs dfs -cat file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan/data/movies.csv | head -11
movieId,title,genres
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
4,Waiting to Exhale (1995),Comedy|Drama|Romance
5,Father of the Bride Part II (1995),Comedy
6,Heat (1995),Action|Crime|Thriller
7,Sabrina (1995),Comedy|Romance
8,Tom and Huck (1995),Adventure|Children
9,Sudden Death (1995),Action
10,GoldenEye (1995),Action|Adventure|Thriller
cat: Unable to write to output stream.
```

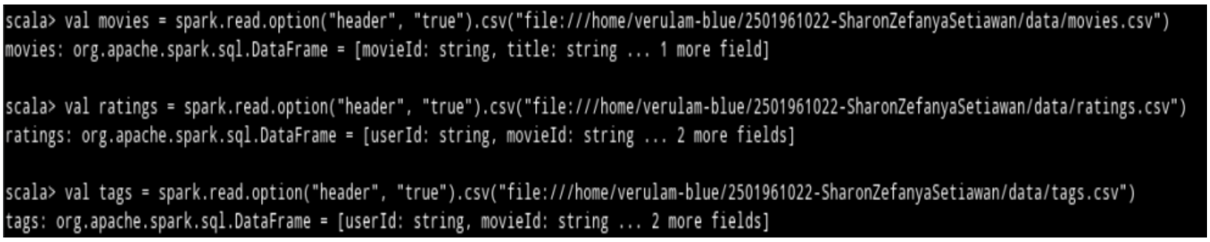
ANSWER Q4:



```
verulam-blue ~ hdfs dfs -count file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan
           2            4            3105156 file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan
verulam-blue ~ hdfs dfs -count file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan/data
           1            4            3105156 file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan/data
```

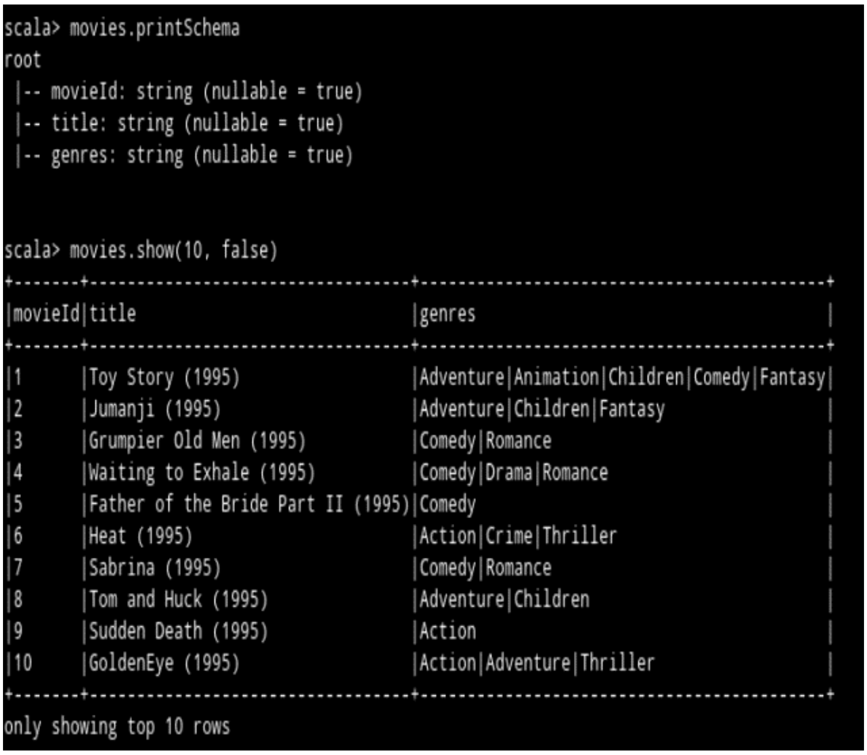
ANSWER Q5:

- Dataframe:



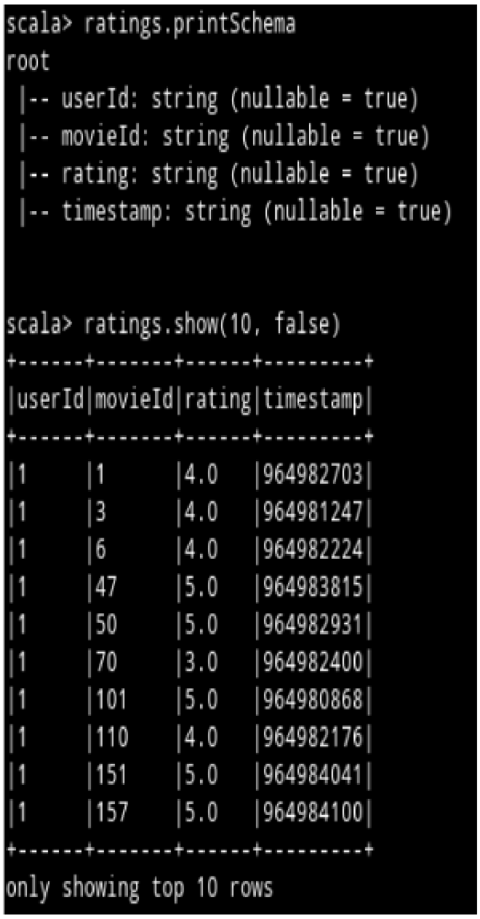
```
scala> val movies = spark.read.option("header", "true").csv("file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan/data/movies.csv")
movies: org.apache.spark.sql.DataFrame = [movieId: string, title: string ... 1 more field]

scala> val ratings = spark.read.option("header", "true").csv("file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan/data/ratings.csv")
ratings: org.apache.spark.sql.DataFrame = [userId: string, movieId: string ... 2 more fields]

scala> val tags = spark.read.option("header", "true").csv("file:///home/verulam-blue/2501961022-SharonZefanyaSetiawan/data/tags.csv")
tags: org.apache.spark.sql.DataFrame = [userId: string, movieId: string ... 2 more fields]
```

- Movies:



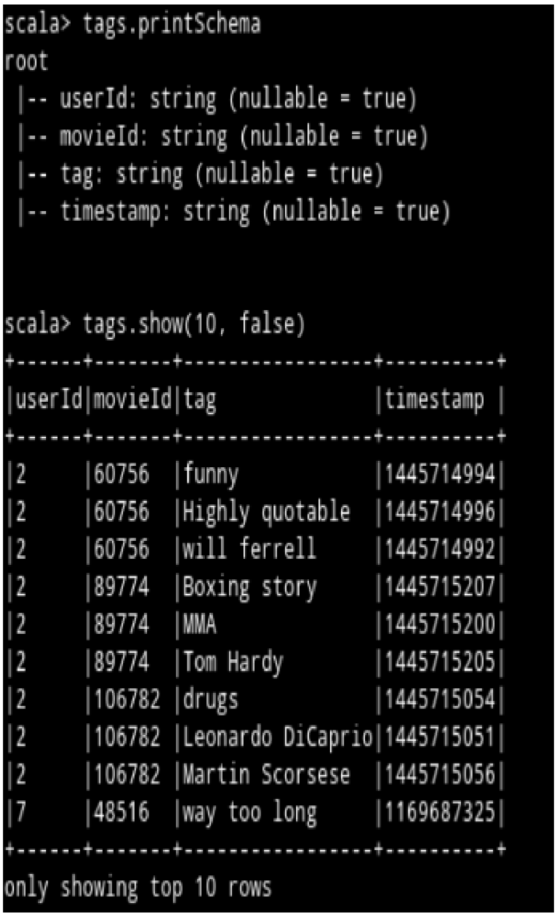
```
scala> movies.printSchema
root
 |-- movieId: string (nullable = true)
 |-- title: string (nullable = true)
 |-- genres: string (nullable = true)


scala> movies.show(10, false)
+-------+----------------------------------+-------------------------------------------+
|movieId|title                             |genres                                     |
+-------+----------------------------------+-------------------------------------------+
|1      |Toy Story (1995)                  |Adventure|Animation|Children|Comedy|Fantasy|
|2      |Jumanji (1995)                    |Adventure|Children|Fantasy                 |
|3      |Grumpier Old Men (1995)           |Comedy|Romance                             |
|4      |Waiting to Exhale (1995)          |Comedy|Drama|Romance                       |
|5      |Father of the Bride Part II (1995)|Comedy                                     |
|6      |Heat (1995)                       |Action|Crime|Thriller                      |
|7      |Sabrina (1995)                    |Comedy|Romance                             |
|8      |Tom and Huck (1995)               |Adventure|Children                         |
|9      |Sudden Death (1995)               |Action                                     |
|10     |GoldenEye (1995)                  |Action|Adventure|Thriller                  |
+-------+----------------------------------+-------------------------------------------+
only showing top 10 rows
```

- Ratings:



```
scala> ratings.printSchema
root
 |-- userId: string (nullable = true)
 |-- movieId: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- timestamp: string (nullable = true)


scala> ratings.show(10, false)
+------+-------+------+---------+
|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|
+------+-------+------+---------+
only showing top 10 rows
```

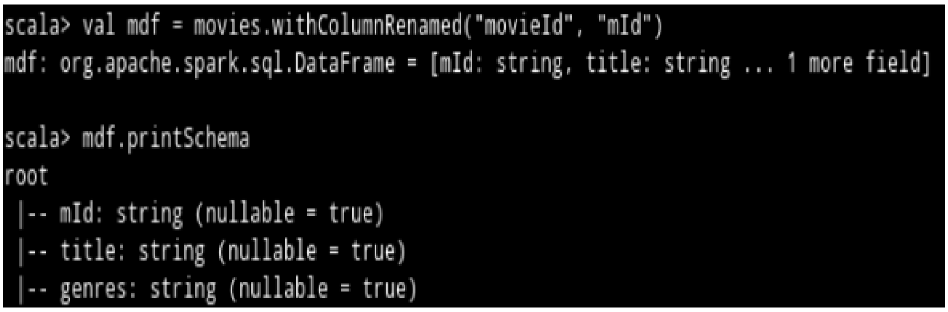
- Tags:



```
scala> tags.printSchema
root
 |-- userId: string (nullable = true)
 |-- movieId: string (nullable = true)
 |-- tag: string (nullable = true)
 |-- timestamp: string (nullable = true)


scala> tags.show(10, false)
+------+-------+-----------------+----------+
|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|
+------+-------+-----------------+----------+
only showing top 10 rows
```

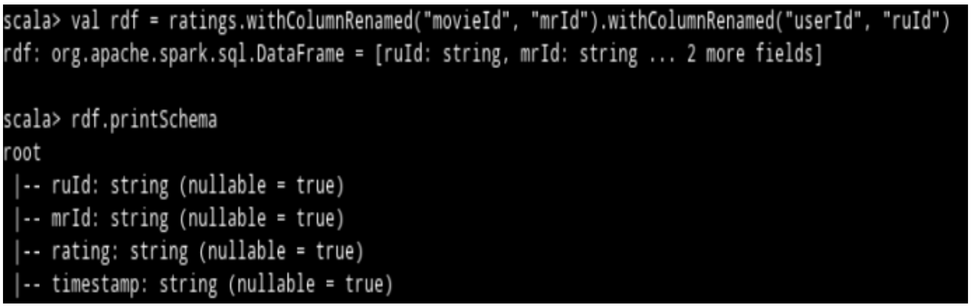
ANSWER Q6:

- Movies:



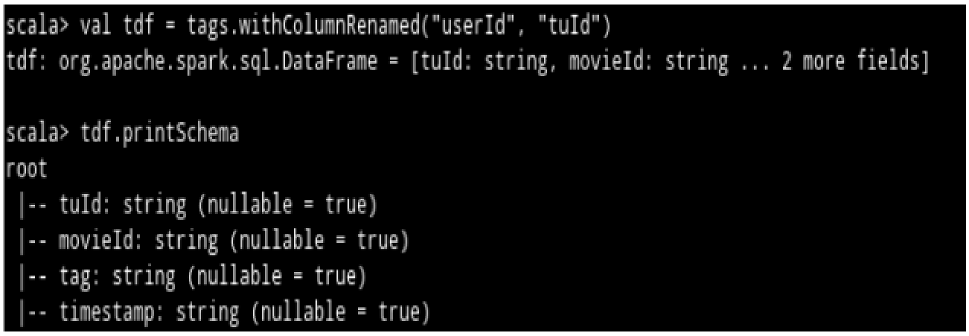
```
scala> val mdf = movies.withColumnRenamed("movieId", "mId")
mdf: org.apache.spark.sql.DataFrame = [mId: string, title: string ... 1 more field]

scala> mdf.printSchema
root
 |-- mId: string (nullable = true)
 |-- title: string (nullable = true)
 |-- genres: string (nullable = true)
```

- Ratings:



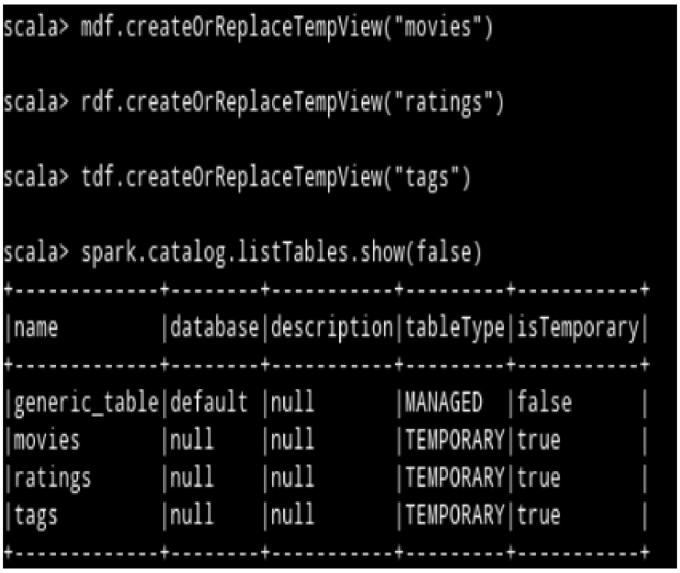
```
scala> val rdf = ratings.withColumnRenamed("movieId", "mrId").withColumnRenamed("userId", "ruId")
rdf: org.apache.spark.sql.DataFrame = [ruId: string, mrId: string ... 2 more fields]

scala> rdf.printSchema
root
 |-- ruId: string (nullable = true)
 |-- mrId: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- timestamp: string (nullable = true)
```

- Tags:



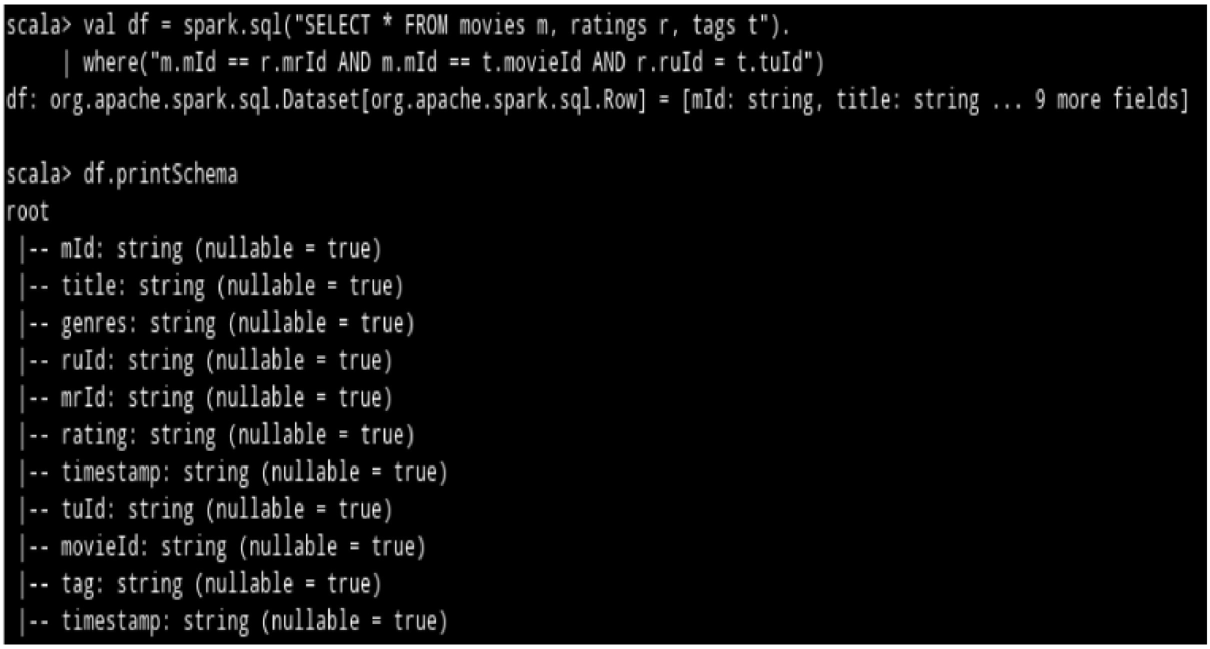
```
scala> val tdf = tags.withColumnRenamed("userId", "tuId")
tdf: org.apache.spark.sql.DataFrame = [tuId: string, movieId: string ... 2 more fields]

scala> tdf.printSchema
root
 |-- tuId: string (nullable = true)
 |-- movieId: string (nullable = true)
 |-- tag: string (nullable = true)
 |-- timestamp: string (nullable = true)
```

ANSWER Q7:

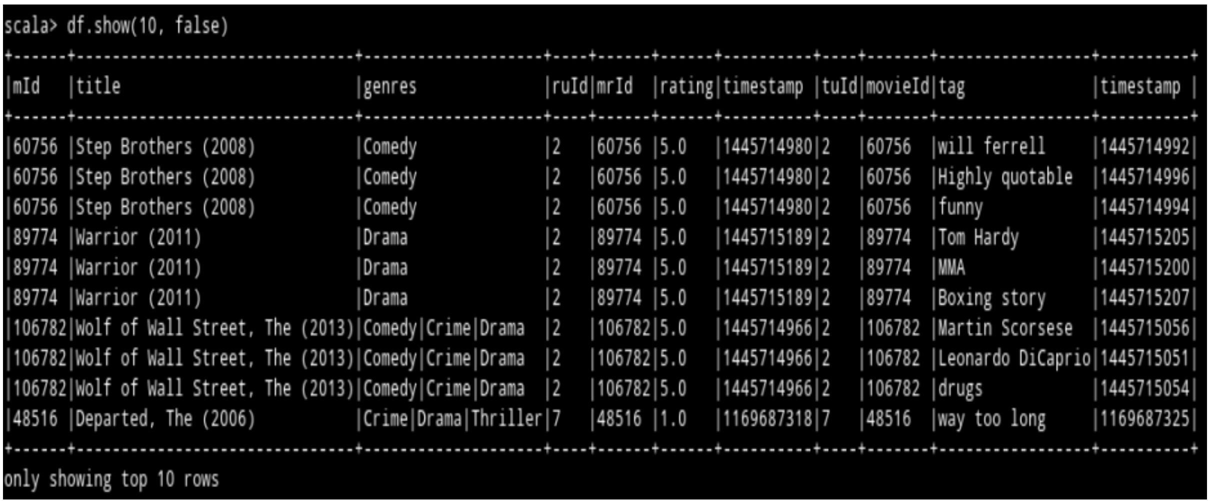


```
scala> mdf.createOrReplaceTempView("movies")

scala> rdf.createOrReplaceTempView("ratings")

scala> tdf.createOrReplaceTempView("tags")

scala> spark.catalog.listTables.show(false)
+-------------+--------+-----------+---------+-----------+
|name         |database|description|tableType|isTemporary|
+-------------+--------+-----------+---------+-----------+
|generic_table|default |null       |MANAGED  |false      |
|movies       |null    |null       |TEMPORARY|true       |
|ratings      |null    |null       |TEMPORARY|true       |
|tags         |null    |null       |TEMPORARY|true       |
+-------------+--------+-----------+---------+-----------+
```

ANSWER Q8:

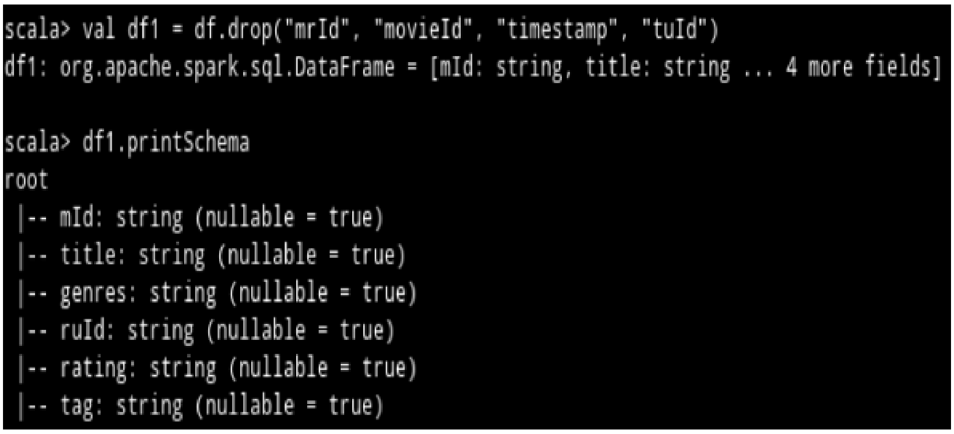


```
scala> val df = spark.sql("SELECT * FROM movies m, ratings r, tags t").
     | where("m.mId == r.mrId AND m.mId == t.movieId AND r.ruId = t.tuId")
df: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [mId: string, title: string ... 9 more fields]

scala> df.printSchema
root
 |-- mId: string (nullable = true)
 |-- title: string (nullable = true)
 |-- genres: string (nullable = true)
 |-- ruId: string (nullable = true)
 |-- mrId: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- timestamp: string (nullable = true)
 |-- tuId: string (nullable = true)
 |-- movieId: string (nullable = true)
 |-- tag: string (nullable = true)
 |-- timestamp: string (nullable = true)
```

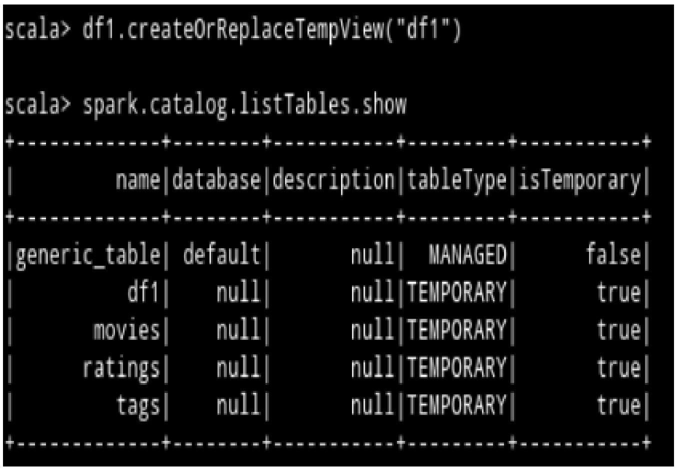


```
scala> df.show(10, false)
+------+-------------------------------+--------------------+----+------+------+----------+----+-------+-----------------+----------+
|mId   |title                          |genres              |ruId|mrId  |rating|timestamp |tuId|movieId|tag              |timestamp |
+------+-------------------------------+--------------------+----+------+------+----------+----+-------+-----------------+----------+
|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 Street, The (2013)|Comedy|Crime|Drama  |2   |106782|5.0   |1445714966|2   |106782 |Martin Scorsese  |1445715056|
|106782|Wolf of Wall Street, The (2013)|Comedy|Crime|Drama  |2   |106782|5.0   |1445714966|2   |106782 |Leonardo DiCaprio|1445715051|
|106782|Wolf of Wall Street, The (2013)|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|
+------+-------------------------------+--------------------+----+------+------+----------+----+-------+-----------------+----------+
only showing top 10 rows
```

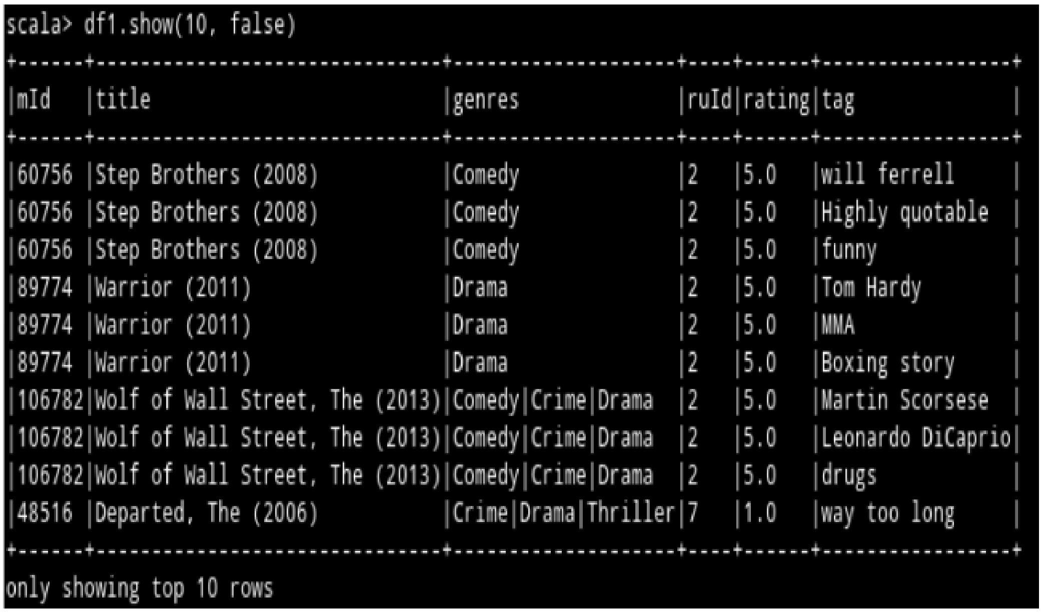
- **ANSWER Q9:**



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

scala> df1.printSchema
root
 |-- mId: string (nullable = true)
 |-- title: string (nullable = true)
 |-- genres: string (nullable = true)
 |-- ruId: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- tag: string (nullable = true)
```

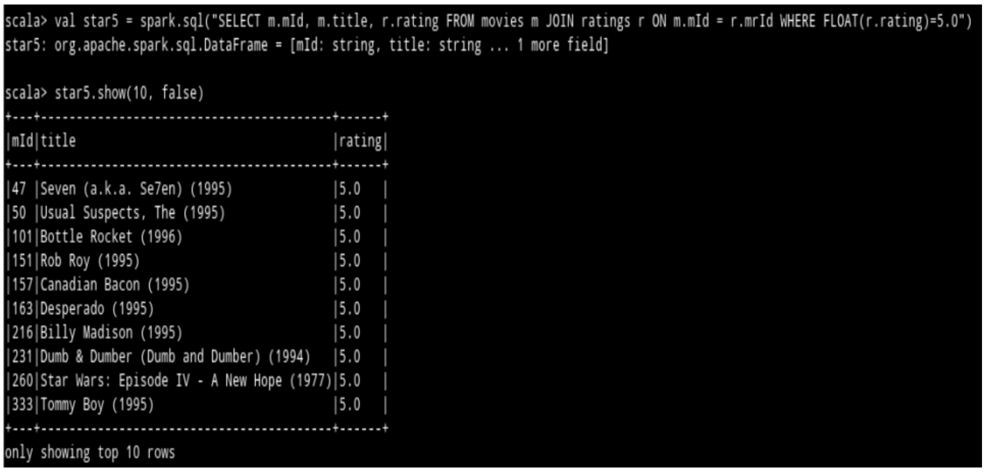


```
scala> df1.createOrReplaceTempView("df1")

scala> spark.catalog.listTables.show
+-------------+--------+-----------+---------+-----------+
|         name|database|description|tableType|isTemporary|
+-------------+--------+-----------+---------+-----------+
|generic_table| default|       null|  MANAGED|      false|
|          df1|    null|       null|TEMPORARY|       true|
|       movies|    null|       null|TEMPORARY|       true|
|      ratings|    null|       null|TEMPORARY|       true|
|         tags|    null|       null|TEMPORARY|       true|
+-------------+--------+-----------+---------+-----------+
```

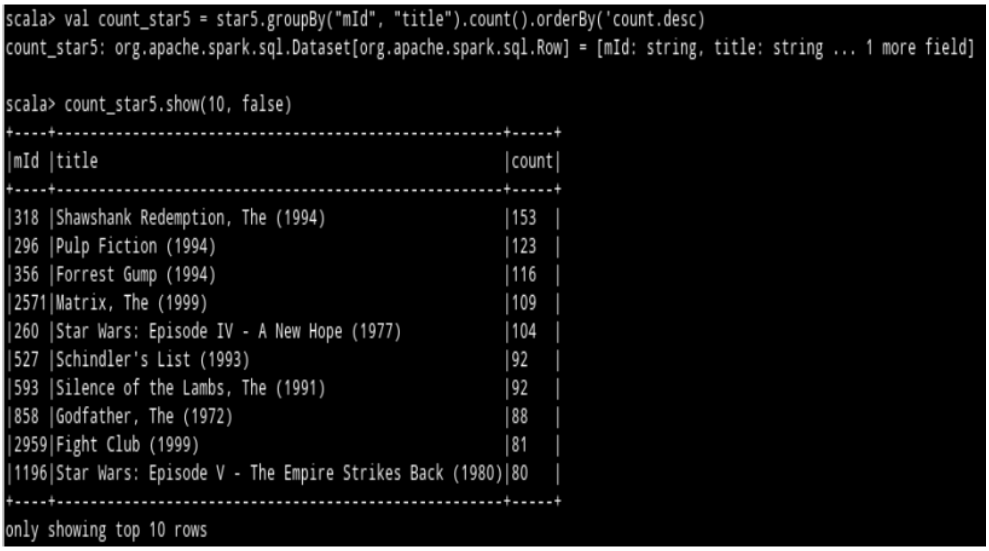


```
scala> df1.show(10, false)
+------+-------------------------------+--------------------+----+------+-----------------+
|mId   |title                          |genres              |ruId|rating|tag              |
+------+-------------------------------+--------------------+----+------+-----------------+
|60756 |Step Brothers (2008)           |Comedy              |2   |5.0   |will ferrell     |
|60756 |Step Brothers (2008)           |Comedy              |2   |5.0   |Highly quotable  |
|60756 |Step Brothers (2008)           |Comedy              |2   |5.0   |funny            |
|89774 |Warrior (2011)                 |Drama               |2   |5.0   |Tom Hardy        |
|89774 |Warrior (2011)                 |Drama               |2   |5.0   |MMA              |
|89774 |Warrior (2011)                 |Drama               |2   |5.0   |Boxing story     |
|106782|Wolf of Wall Street, The (2013)|Comedy|Crime|Drama  |2   |5.0   |Martin Scorsese  |
|106782|Wolf of Wall Street, The (2013)|Comedy|Crime|Drama  |2   |5.0   |Leonardo DiCaprio|
|106782|Wolf of Wall Street, The (2013)|Comedy|Crime|Drama  |2   |5.0   |drugs            |
|48516 |Departed, The (2006)           |Crime|Drama|Thriller|7   |1.0   |way too long     |
+------+-------------------------------+--------------------+----+------+-----------------+
only showing top 10 rows
```

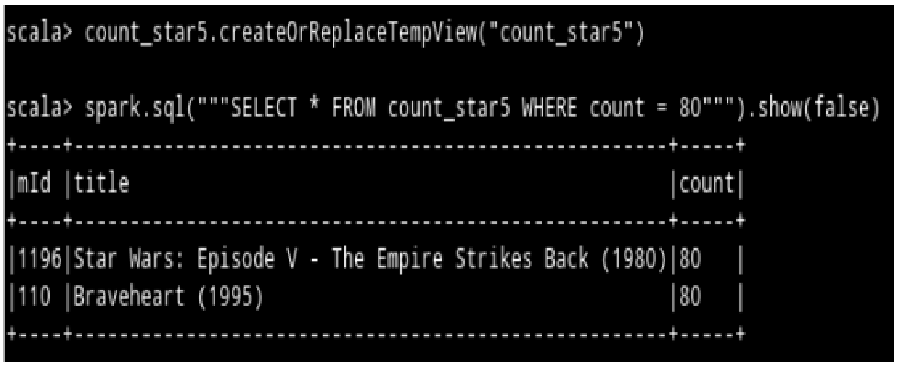
ANSWER Q10:



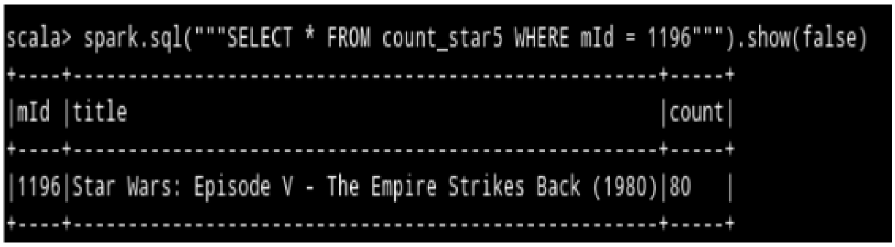
```
scala> val star5 = spark.sql("SELECT m.mId, m.title, r.rating FROM movies m JOIN ratings r ON m.mId = r.mrId WHERE FLOAT(r.rating)=5.0")
star5: org.apache.spark.sql.DataFrame = [mId: string, title: string ... 1 more field]

scala> star5.show(10, false)
+---+-----------------------------------------+------+
|mId|title                                    |rating|
+---+-----------------------------------------+------+
|47 |Seven (a.k.a. Se7en) (1995)              |5.0   |
|50 |Usual Suspects, The (1995)               |5.0   |
|101|Bottle Rocket (1996)                     |5.0   |
|151|Rob Roy (1995)                           |5.0   |
|157|Canadian Bacon (1995)                    |5.0   |
|163|Desperado (1995)                         |5.0   |
|216|Billy Madison (1995)                     |5.0   |
|231|Dumb & Dumber (Dumb and Dumber) (1994)   |5.0   |
|260|Star Wars: Episode IV - A New Hope (1977)|5.0   |
|333|Tommy Boy (1995)                         |5.0   |
+---+-----------------------------------------+------+
only showing top 10 rows
```



```
scala> val count_star5 = star5.groupBy("mId", "title").count().orderBy('count.desc)
count_star5: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [mId: string, title: string ... 1 more field]

scala> count_star5.show(10, false)
+----+-----------------------------------------------------+-----+
|mId |title                                                |count|
+----+-----------------------------------------------------+-----+
|318 |Shawshank Redemption, The (1994)                     |153  |
|296 |Pulp Fiction (1994)                                  |123  |
|356 |Forrest Gump (1994)                                  |116  |
|2571|Matrix, The (1999)                                   |109  |
|260 |Star Wars: Episode IV - A New Hope (1977)            |104  |
|527 |Schindler's List (1993)                              |92   |
|593 |Silence of the Lambs, The (1991)                     |92   |
|858 |Godfather, The (1972)                                |88   |
|2959|Fight Club (1999)                                    |81   |
|1196|Star Wars: Episode V - The Empire Strikes Back (1980)|80   |
+----+-----------------------------------------------------+-----+
only showing top 10 rows
```

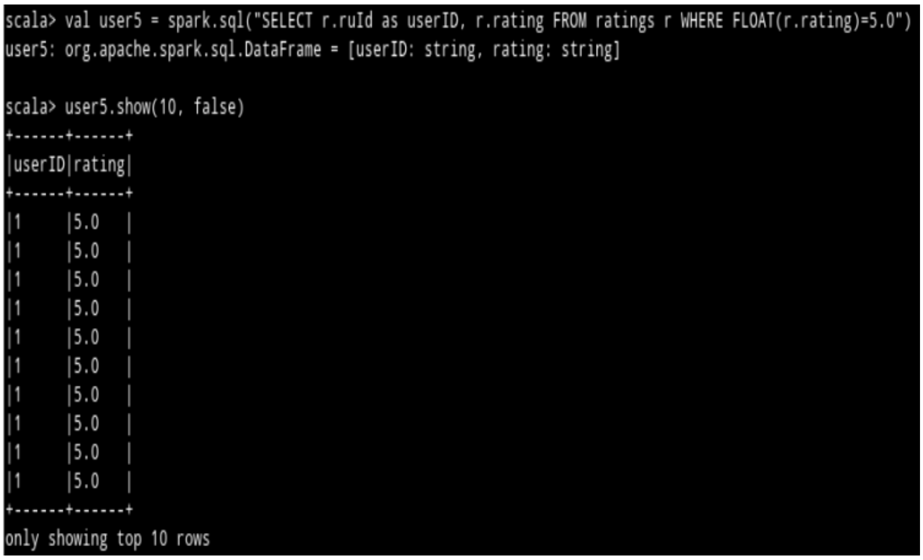


```
scala> count_star5.createOrReplaceTempView("count_star5")

scala> spark.sql("""SELECT * FROM count_star5 WHERE count = 80""").show(false)
+----+-----------------------------------------------------+-----+
|mId |title                                                |count|
+----+-----------------------------------------------------+-----+
|1196|Star Wars: Episode V - The Empire Strikes Back (1980)|80   |
|110 |Braveheart (1995)                                    |80   |
+----+-----------------------------------------------------+-----+
```

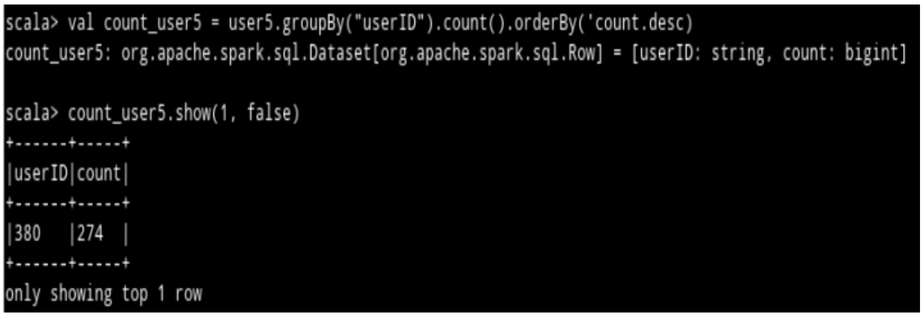


```
scala> spark.sql("""SELECT * FROM count_star5 WHERE mId = 1196""").show(false)
+----+-----------------------------------------------------+-----+
|mId |title                                                |count|
+----+-----------------------------------------------------+-----+
|1196|Star Wars: Episode V - The Empire Strikes Back (1980)|80   |
+----+-----------------------------------------------------+-----+
```

ANSWER Q11:

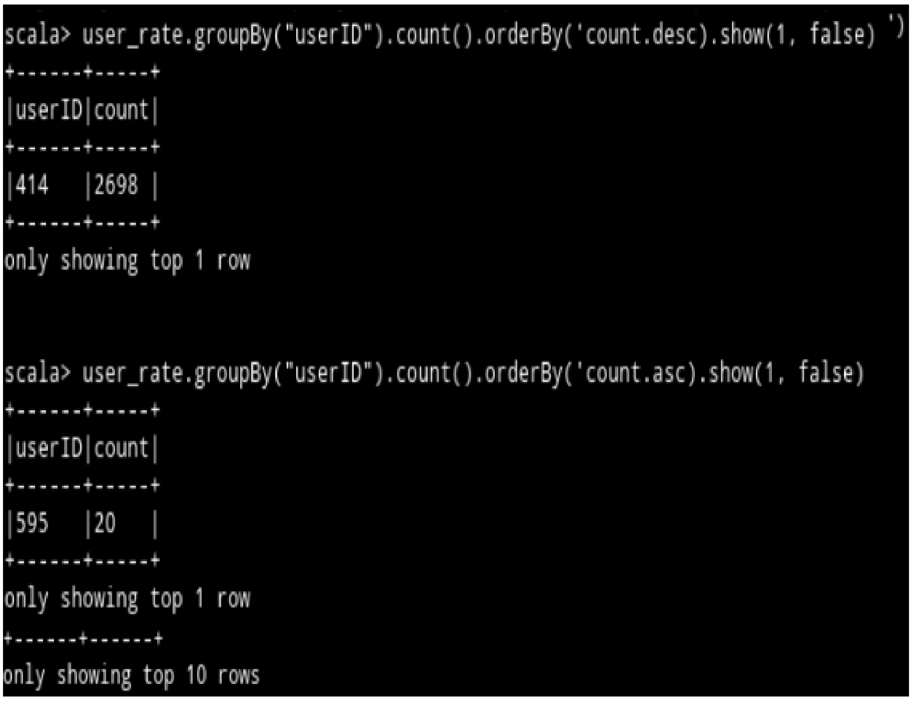


```
scala> val user5 = spark.sql("SELECT r.ruId as userID, r.rating FROM ratings r WHERE FLOAT(r.rating)=5.0")
user5: org.apache.spark.sql.DataFrame = [userID: string, rating: string]

scala> user5.show(10, false)
+------+------+
|userID|rating|
+------+------+
|1     |5.0   |
|1     |5.0   |
|1     |5.0   |
|1     |5.0   |
|1     |5.0   |
|1     |5.0   |
|1     |5.0   |
|1     |5.0   |
|1     |5.0   |
|1     |5.0   |
+------+------+
only showing top 10 rows
```

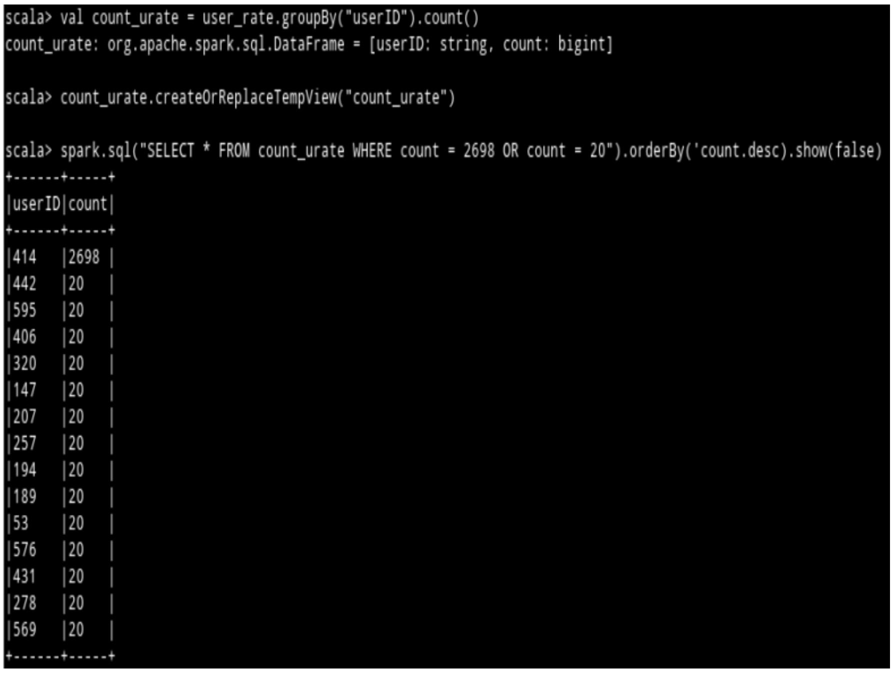


```
scala> val count_user5 = user5.groupBy("userID").count().orderBy('count.desc)
count_user5: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [userID: string, count: bigint]

scala> count_user5.show(1, false)
+------+-----+
|userID|count|
+------+-----+
|380   |274  |
+------+-----+
only showing top 1 row
```

ANSWER Q12:

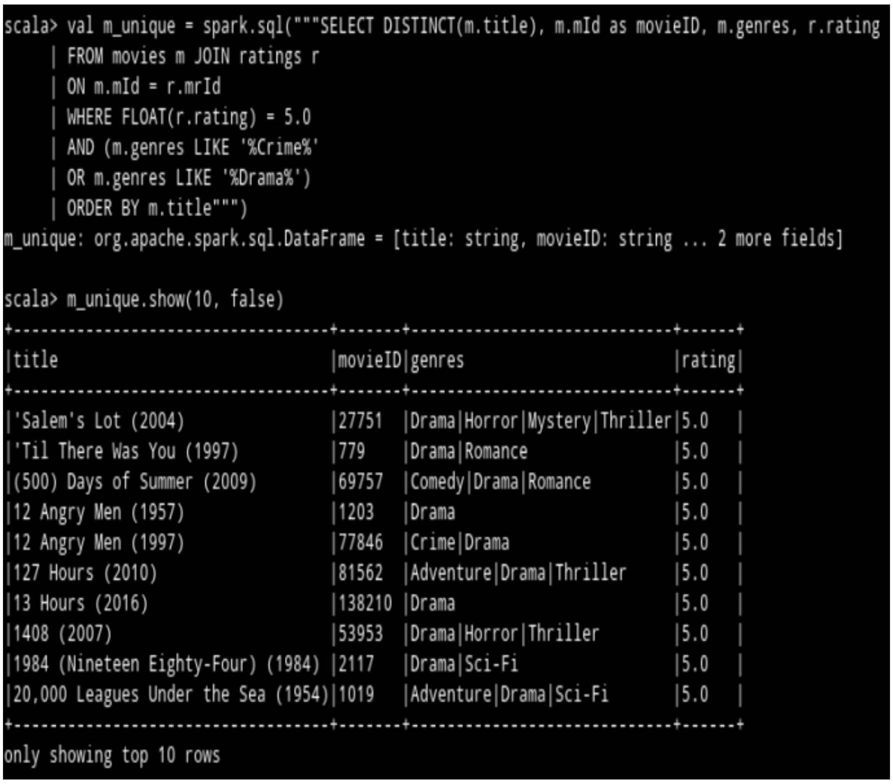


```
scala> user_rate.groupBy("userID").count().orderBy('count.desc).show(1, false) ')
+------+-----+
|userID|count|
+------+-----+
|414   |2698 |
+------+-----+
only showing top 1 row


scala> user_rate.groupBy("userID").count().orderBy('count.asc).show(1, false)
+------+-----+
|userID|count|
+------+-----+
|595   |20   |
+------+-----+
only showing top 1 row
+------+------+
only showing top 10 rows
```



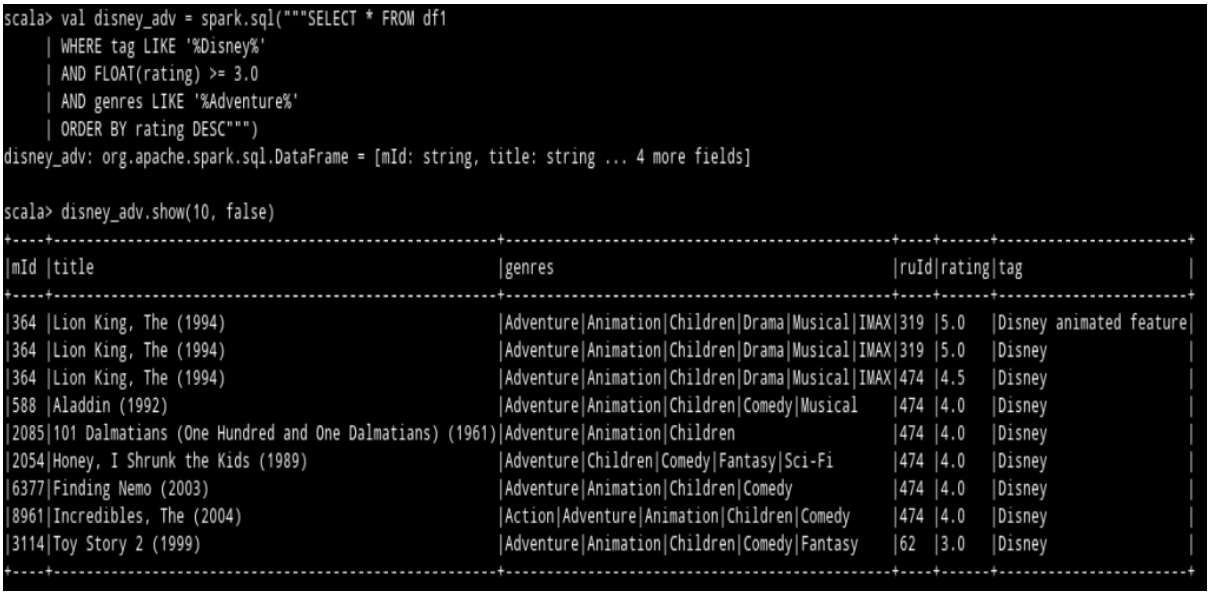
```
scala> val count_urate = user_rate.groupBy("userID").count()
count_urate: org.apache.spark.sql.DataFrame = [userID: string, count: bigint]

scala> count_urate.createOrReplaceTempView("count_urate")

scala> spark.sql("SELECT * FROM count_urate WHERE count = 2698 OR count = 20").orderBy('count.desc).show(false)
+------+-----+
|userID|count|
+------+-----+
|414   |2698 |
|442   |20   |
|595   |20   |
|406   |20   |
|320   |20   |
|147   |20   |
|207   |20   |
|257   |20   |
|194   |20   |
|189   |20   |
|53    |20   |
|576   |20   |
|431   |20   |
|278   |20   |
|569   |20   |
+------+-----+
```

ANSWER Q13:



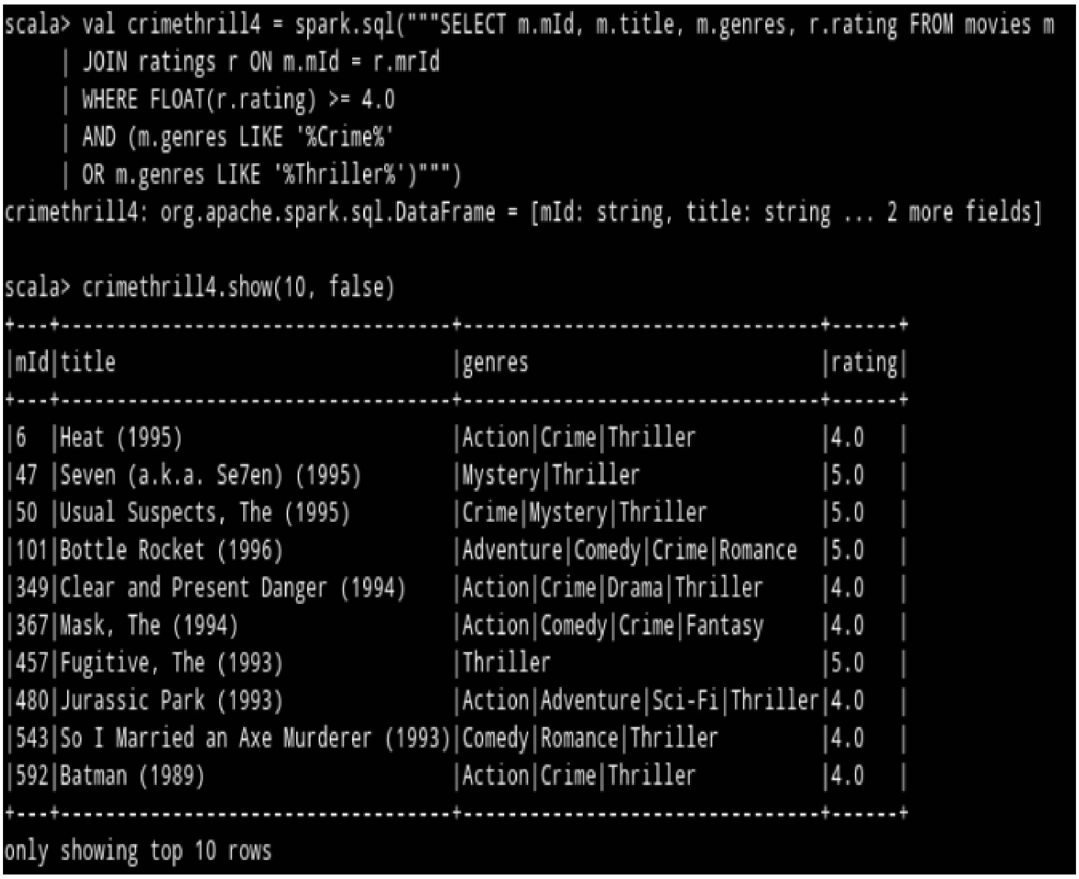
```
scala> val m_unique = spark.sql("""SELECT DISTINCT(m.title), m.mId as movieID, m.genres, r.rating
     | FROM movies m JOIN ratings r
     | ON m.mId = r.mrId
     | WHERE FLOAT(r.rating) = 5.0
     | AND (m.genres LIKE '%Crime%'
     | OR m.genres LIKE '%Drama%')
     | ORDER BY m.title""")
m_unique: org.apache.spark.sql.DataFrame = [title: string, movieID: string ... 2 more fields]

scala> m_unique.show(10, false)
+-----------------------------------+-------+-----------------------------+------+
|title                              |movieID|genres                       |rating|
+-----------------------------------+-------+-----------------------------+------+
|'Salem's Lot (2004)                |27751  |Drama|Horror|Mystery|Thriller|5.0   |
|'Til There Was You (1997)          |779    |Drama|Romance                |5.0   |
|(500) Days of Summer (2009)        |69757  |Comedy|Drama|Romance         |5.0   |
|12 Angry Men (1957)                |1203   |Drama                        |5.0   |
|12 Angry Men (1997)                |77846  |Crime|Drama                  |5.0   |
|127 Hours (2010)                   |81562  |Adventure|Drama|Thriller     |5.0   |
|13 Hours (2016)                    |138210 |Drama                        |5.0   |
|1408 (2007)                        |53953  |Drama|Horror|Thriller        |5.0   |
|1984 (Nineteen Eighty-Four) (1984) |2117   |Drama|Sci-Fi                 |5.0   |
|20,000 Leagues Under the Sea (1954)|1019   |Adventure|Drama|Sci-Fi       |5.0   |
+-----------------------------------+-------+-----------------------------+------+
only showing top 10 rows
```

ANSWER Q14:

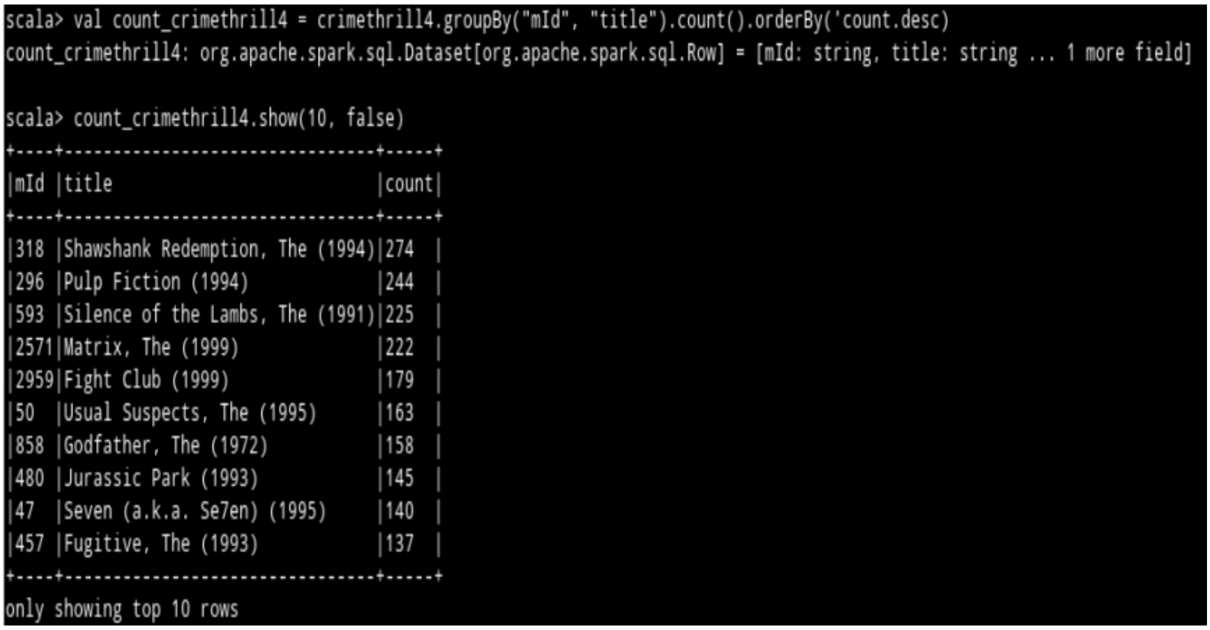


```
scala> val disney_adv = spark.sql("""SELECT * FROM df1
     | WHERE tag LIKE '%Disney%'
     | AND FLOAT(rating) >= 3.0
     | AND genres LIKE '%Adventure%'
     | ORDER BY rating DESC""")
disney_adv: org.apache.spark.sql.DataFrame = [mId: string, title: string ... 4 more fields]

scala> disney_adv.show(10, false)
+----+------------------------------------------------------+-----------------------------------------------+----+------+-----------------------+
|mId |title                                                 |genres                                         |ruId|rating|tag                    |
+----+------------------------------------------------------+-----------------------------------------------+----+------+-----------------------+
|364 |Lion King, The (1994)                                 |Adventure|Animation|Children|Drama|Musical|IMAX|319 |5.0   |Disney animated feature|
|364 |Lion King, The (1994)                                 |Adventure|Animation|Children|Drama|Musical|IMAX|319 |5.0   |Disney                 |
|364 |Lion King, The (1994)                                 |Adventure|Animation|Children|Drama|Musical|IMAX|474 |4.5   |Disney                 |
|588 |Aladdin (1992)                                        |Adventure|Animation|Children|Comedy|Musical    |474 |4.0   |Disney                 |
|2085|101 Dalmatians (One Hundred and One Dalmatians) (1961)|Adventure|Animation|Children                    |474 |4.0   |Disney                 |
|2054|Honey, I Shrunk the Kids (1989)                       |Adventure|Children|Comedy|Fantasy|Sci-Fi       |474 |4.0   |Disney                 |
|6377|Finding Nemo (2003)                                   |Adventure|Animation|Children|Comedy            |474 |4.0   |Disney                 |
|8961|Incredibles, The (2004)                               |Action|Adventure|Animation|Children|Comedy     |474 |4.0   |Disney                 |
|3114|Toy Story 2 (1999)                                    |Adventure|Animation|Children|Comedy|Fantasy    |62  |3.0   |Disney                 |
+----+------------------------------------------------------+-----------------------------------------------+----+------+-----------------------+
```

ANSWER Q15:



```
scala> val crimethrill4 = spark.sql("""SELECT m.mId, m.title, m.genres, r.rating FROM movies m
     | JOIN ratings r ON m.mId = r.mrId
     | WHERE FLOAT(r.rating) >= 4.0
     | AND (m.genres LIKE '%Crime%'
     | OR m.genres LIKE '%Thriller%')""")
crimethrill4: org.apache.spark.sql.DataFrame = [mId: string, title: string ... 2 more fields]

scala> crimethrill4.show(10, false)
+---+-----------------------------------+--------------------------------+------+
|mId|title                              |genres                          |rating|
+---+-----------------------------------+--------------------------------+------+
|6  |Heat (1995)                        |Action|Crime|Thriller           |4.0   |
|47 |Seven (a.k.a. Se7en) (1995)        |Mystery|Thriller                |5.0   |
|50 |Usual Suspects, The (1995)         |Crime|Mystery|Thriller          |5.0   |
|101|Bottle Rocket (1996)               |Adventure|Comedy|Crime|Romance  |5.0   |
|349|Clear and Present Danger (1994)    |Action|Crime|Drama|Thriller     |4.0   |
|367|Mask, The (1994)                   |Action|Comedy|Crime|Fantasy     |4.0   |
|457|Fugitive, The (1993)               |Thriller                        |5.0   |
|480|Jurassic Park (1993)               |Action|Adventure|Sci-Fi|Thriller|4.0   |
|543|So I Married an Axe Murderer (1993)|Comedy|Romance|Thriller         |4.0   |
|592|Batman (1989)                      |Action|Crime|Thriller           |4.0   |
+---+-----------------------------------+--------------------------------+------+
only showing top 10 rows
```



```
scala> val count_crimethrill4 = crimethrill4.groupBy("mId", "title").count().orderBy('count.desc)
count_crimethrill4: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [mId: string, title: string ... 1 more field]

scala> count_crimethrill4.show(10, false)
+----+--------------------------------+-----+
|mId |title                           |count|
+----+--------------------------------+-----+
|318 |Shawshank Redemption, The (1994)|274  |
|296 |Pulp Fiction (1994)             |244  |
|593 |Silence of the Lambs, The (1991)|225  |
|2571|Matrix, The (1999)              |222  |
|2959|Fight Club (1999)               |179  |
|50  |Usual Suspects, The (1995)      |163  |
|858 |Godfather, The (1972)           |158  |
|480 |Jurassic Park (1993)            |145  |
|47  |Seven (a.k.a. Se7en) (1995)     |140  |
|457 |Fugitive, The (1993)            |137  |
+----+--------------------------------+-----+
only showing top 10 rows
```