Engineering Excellence J1
(Hadoop Cluster Setup Movielens Data)

Spark Use Cases

TABLE OF CONTENT

Spark:

Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. It is based on Hadoop MapReduce and it extends the MapReduce model to efficiently use it for more types of computations, which includes interactive queries and stream processing.
Spark is designed to cover a wide range of workloads such as batch applications, iterative algorithms, interactive queries and streaming. Apart from supporting all these workload in a respective system, it reduces the management burden of maintaining separate tools.

Features of Apache Spark:

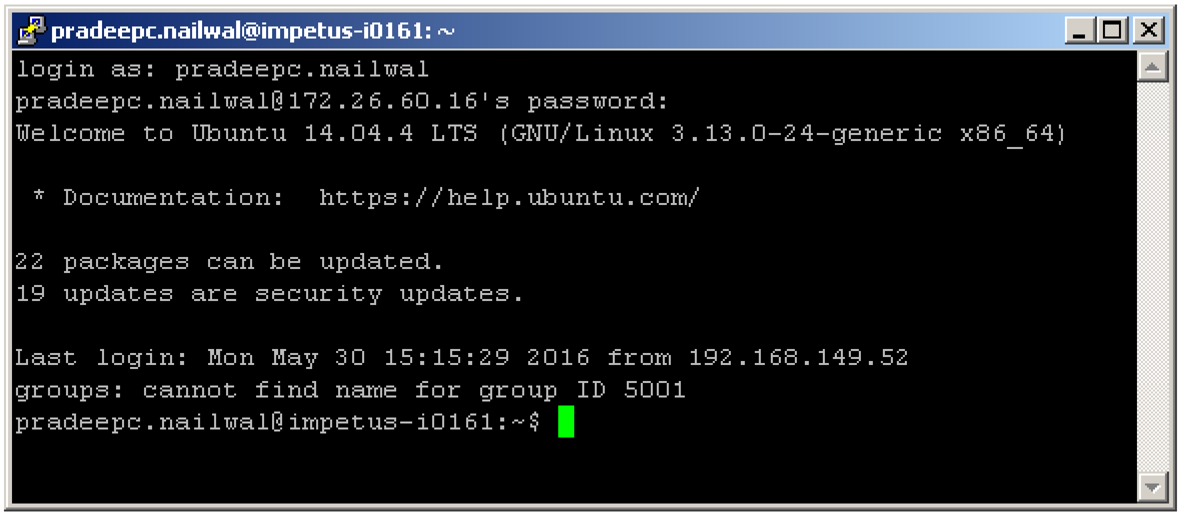
Speed: Spark helps to run an application in Hadoop cluster, up to 100 times faster in memory, and 10 times faster when running on disk. This is possible by reducing number of read/write operations to disk. It stores the intermediate processing data in memory.

Supports multiple languages: Spark provides built-in APIs in Java, Scala, or Python. Therefore, you can write applications in different languages. Spark comes up with 80 high-level operators for interactive querying.

Advanced Analytics: Spark not only supports 'Map' and 'reduce'. It also supports SQL queries, Streaming data, Machine learning (ML), and Graph algorithms.

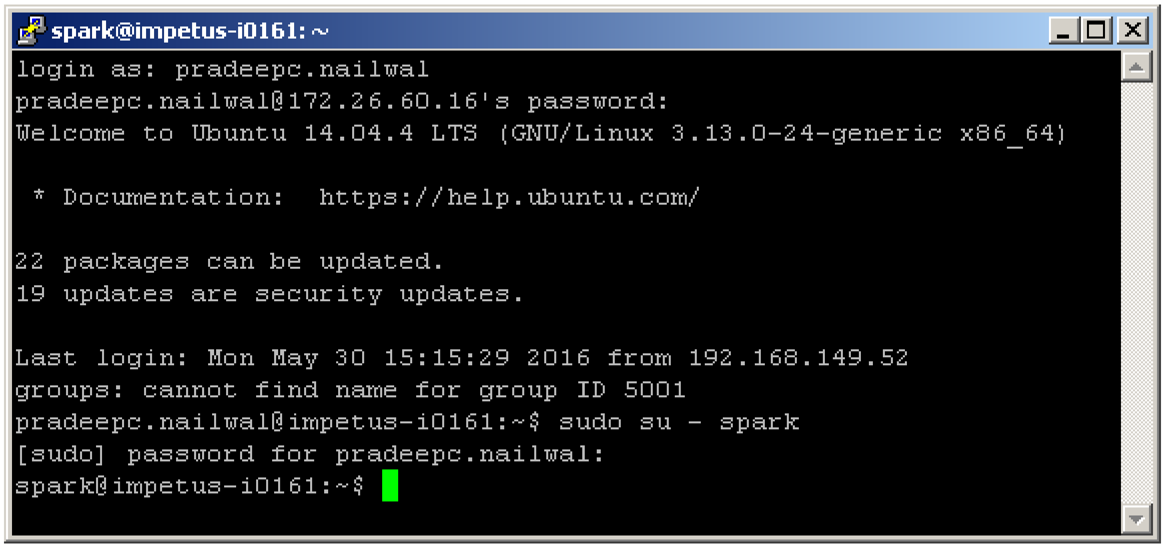
Spark Work Flow for Movie Rating:

(1) Login to Spark machine (172.26.60.16) using given credentials.



```
login as: pradeepc.nailwal
pradeepc.nailwal@172.26.60.16's password:
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

22 packages can be updated.
19 updates are security updates.

Last login: Mon May 30 15:15:29 2016 from 192.168.149.52
groups: cannot find name for group ID 5001
pradeepc.nailwal@impetus-i0161:~$
```

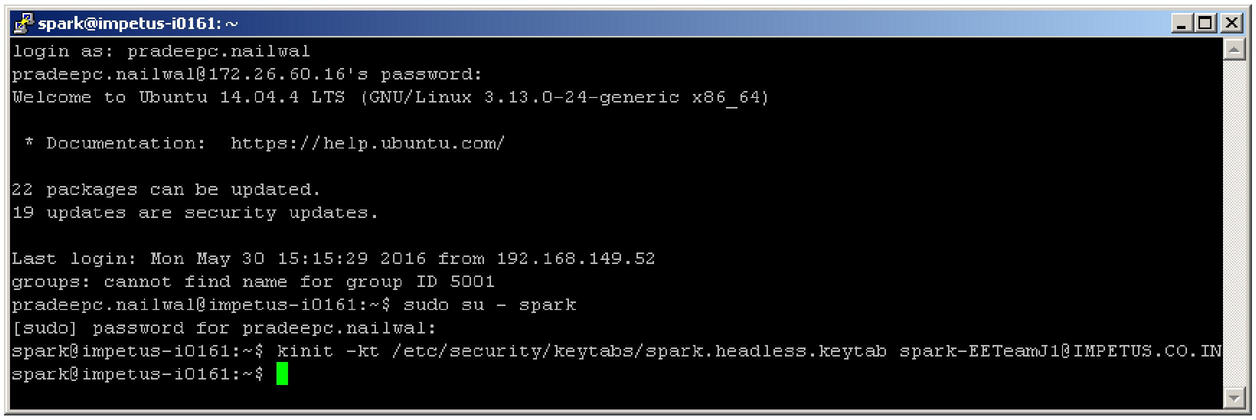
(2) Login to Sudo user as spark as below:


spark@impetus-i0161: ~

```
login as: pradeepc.nailwal
pradeepc.nailwal@172.26.60.16's password:
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

22 packages can be updated.
19 updates are security updates.

Last login: Mon May 30 15:15:29 2016 from 192.168.149.52
groups: cannot find name for group ID 5001
pradeepc.nailwal@impetus-i0161:~$ sudo su - spark
[sudo] password for pradeepc.nailwal:
spark@impetus-i0161:~$
```

(3) Run following command to initialize the security settings for spark user

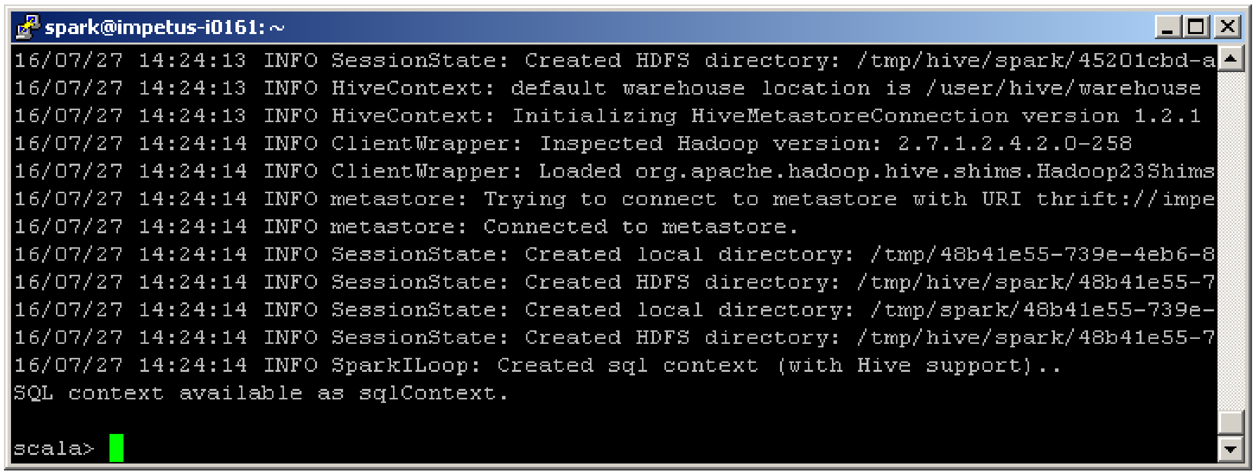
kinit -kt /etc/security/keytabs/spark.headless.keytab spark-EETeamJ1@IMPETUS.CO.IN


spark@impetus-i0161: ~

```
login as: pradeepc.nailwal
pradeepc.nailwal@172.26.60.16's password:
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

22 packages can be updated.
19 updates are security updates.

Last login: Mon May 30 15:15:29 2016 from 192.168.149.52
groups: cannot find name for group ID 5001
pradeepc.nailwal@impetus-i0161:~$ sudo su - spark
[sudo] password for pradeepc.nailwal:
spark@impetus-i0161:~$ kinit -kt /etc/security/keytabs/spark.headless.keytab spark-EETeamJ1@IMPETUS.CO.IN
spark@impetus-i0161:~$
```

(4) Start spark by using below command

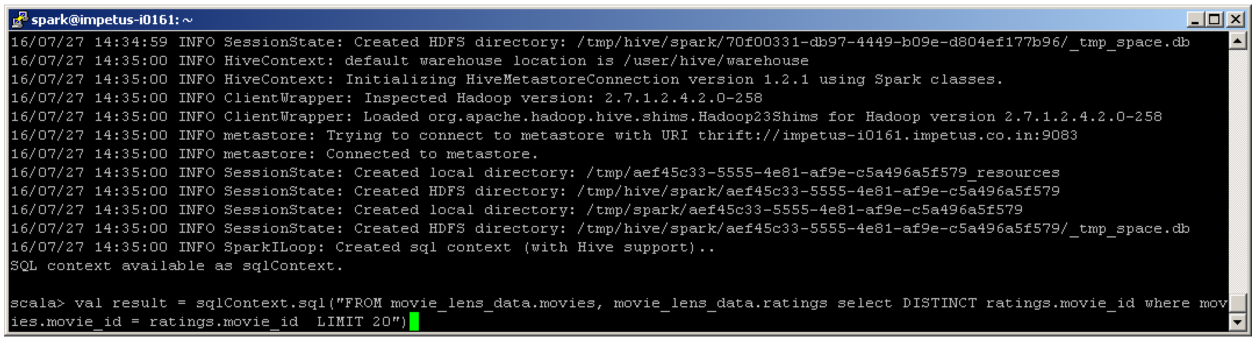
spark-shell



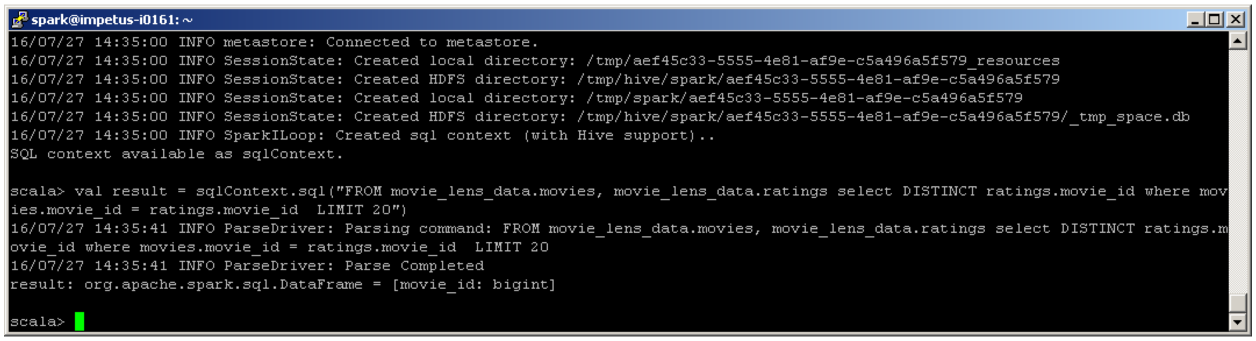
```
16/07/27 14:24:13 INFO SessionState: Created HDFS directory: /tmp/hive/spark/45201cbd-a
16/07/27 14:24:13 INFO HiveContext: default warehouse location is /user/hive/warehouse
16/07/27 14:24:13 INFO HiveContext: Initializing HiveMetastoreConnection version 1.2.1
16/07/27 14:24:14 INFO ClientWrapper: Inspected Hadoop version: 2.7.1.2.4.2.0-258
16/07/27 14:24:14 INFO ClientWrapper: Loaded org.apache.hadoop.hive.shims.Hadoop23Shims
16/07/27 14:24:14 INFO metastore: Trying to connect to metastore with URI thrift://impe
16/07/27 14:24:14 INFO metastore: Connected to metastore.
16/07/27 14:24:14 INFO SessionState: Created local directory: /tmp/48b41e55-739e-4eb6-8
16/07/27 14:24:14 INFO SessionState: Created HDFS directory: /tmp/hive/spark/48b41e55-7
16/07/27 14:24:14 INFO SessionState: Created local directory: /tmp/spark/48b41e55-739e-
16/07/27 14:24:14 INFO SessionState: Created HDFS directory: /tmp/hive/spark/48b41e55-7
16/07/27 14:24:14 INFO SparkILoop: Created sql context (with Hive support)..
SQL context available as sqlContext.

scala>
```

(5) We can fire query on already existing tables related to movie as below

val result = sqlContext.sql("FROM movie_lens_data.movies, movie_lens_data.ratings select DISTINCT ratings.movie_id where movies.movie_id = ratings.movie_id LIMIT 20")



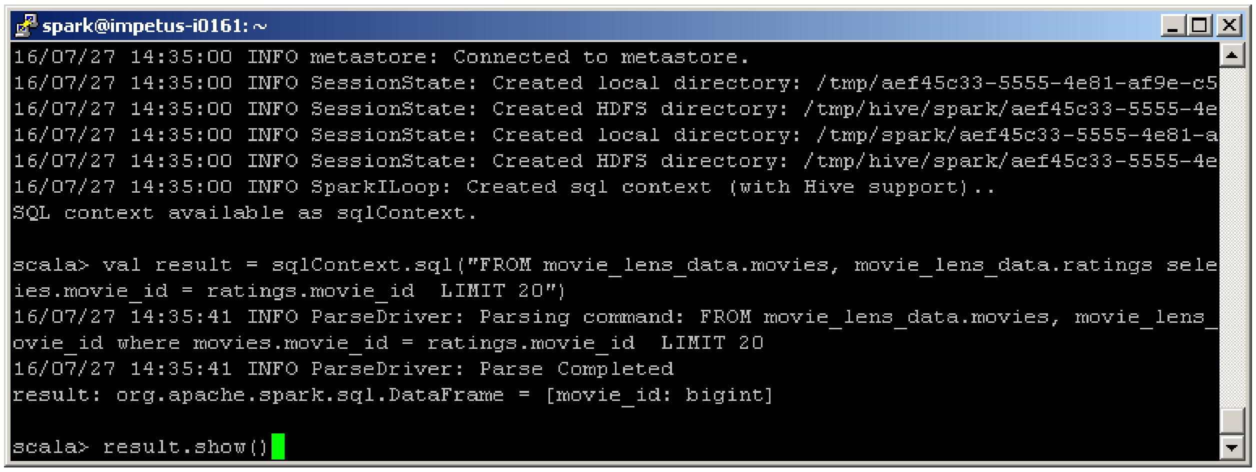
```
16/07/27 14:34:59 INFO SessionState: Created HDFS directory: /tmp/hive/spark/70f00331-db97-4449-b09e-d804ef177b96/_tmp_space.db
16/07/27 14:35:00 INFO HiveContext: default warehouse location is /user/hive/warehouse
16/07/27 14:35:00 INFO HiveContext: Initializing HiveMetastoreConnection version 1.2.1 using Spark classes.
16/07/27 14:35:00 INFO ClientWrapper: Inspected Hadoop version: 2.7.1.2.4.2.0-258
16/07/27 14:35:00 INFO ClientWrapper: Loaded org.apache.hadoop.hive.shims.Hadoop23Shims for Hadoop version 2.7.1.2.4.2.0-258
16/07/27 14:35:00 INFO metastore: Trying to connect to metastore with URI thrift://impetus-i0161.impetus.co.in:9083
16/07/27 14:35:00 INFO metastore: Connected to metastore.
16/07/27 14:35:00 INFO SessionState: Created local directory: /tmp/aef45c33-5555-4e81-af9e-c5a496a5f579_resources
16/07/27 14:35:00 INFO SessionState: Created HDFS directory: /tmp/hive/spark/aef45c33-5555-4e81-af9e-c5a496a5f579
16/07/27 14:35:00 INFO SessionState: Created local directory: /tmp/spark/aef45c33-5555-4e81-af9e-c5a496a5f579
16/07/27 14:35:00 INFO SessionState: Created HDFS directory: /tmp/hive/spark/aef45c33-5555-4e81-af9e-c5a496a5f579/_tmp_space.db
16/07/27 14:35:00 INFO SparkILoop: Created sql context (with Hive support)..
SQL context available as sqlContext.

scala> val result = sqlContext.sql("FROM movie_lens_data.movies, movie_lens_data.ratings select DISTINCT ratings.movie_id where mov
ies.movie_id = ratings.movie_id  LIMIT 20")
```



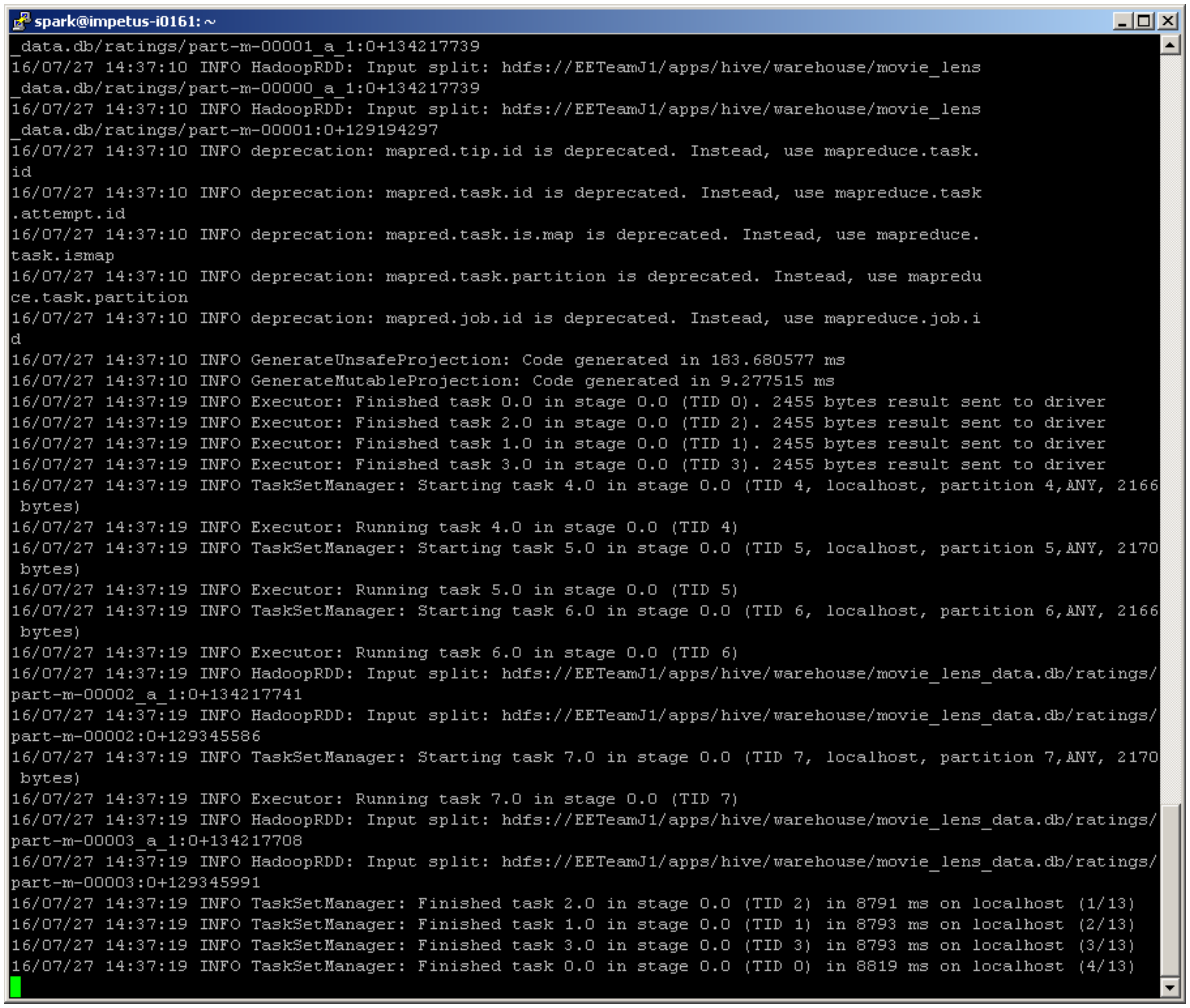
```
16/07/27 14:35:00 INFO metastore: Connected to metastore.
16/07/27 14:35:00 INFO SessionState: Created local directory: /tmp/aef45c33-5555-4e81-af9e-c5a496a5f579_resources
16/07/27 14:35:00 INFO SessionState: Created HDFS directory: /tmp/hive/spark/aef45c33-5555-4e81-af9e-c5a496a5f579
16/07/27 14:35:00 INFO SessionState: Created local directory: /tmp/spark/aef45c33-5555-4e81-af9e-c5a496a5f579
16/07/27 14:35:00 INFO SessionState: Created HDFS directory: /tmp/hive/spark/aef45c33-5555-4e81-af9e-c5a496a5f579/_tmp_space.db
16/07/27 14:35:00 INFO SparkILoop: Created sql context (with Hive support)..
SQL context available as sqlContext.

scala> val result = sqlContext.sql("FROM movie_lens_data.movies, movie_lens_data.ratings select DISTINCT ratings.movie_id where mov
ies.movie_id = ratings.movie_id  LIMIT 20")
16/07/27 14:35:41 INFO ParseDriver: Parsing command: FROM movie_lens_data.movies, movie_lens_data.ratings select DISTINCT ratings.m
ovie_id where movies.movie_id = ratings.movie_id  LIMIT 20
16/07/27 14:35:41 INFO ParseDriver: Parse Completed
result: org.apache.spark.sql.DataFrame = [movie_id: bigint]

scala>
```

(6) Show the result using below command

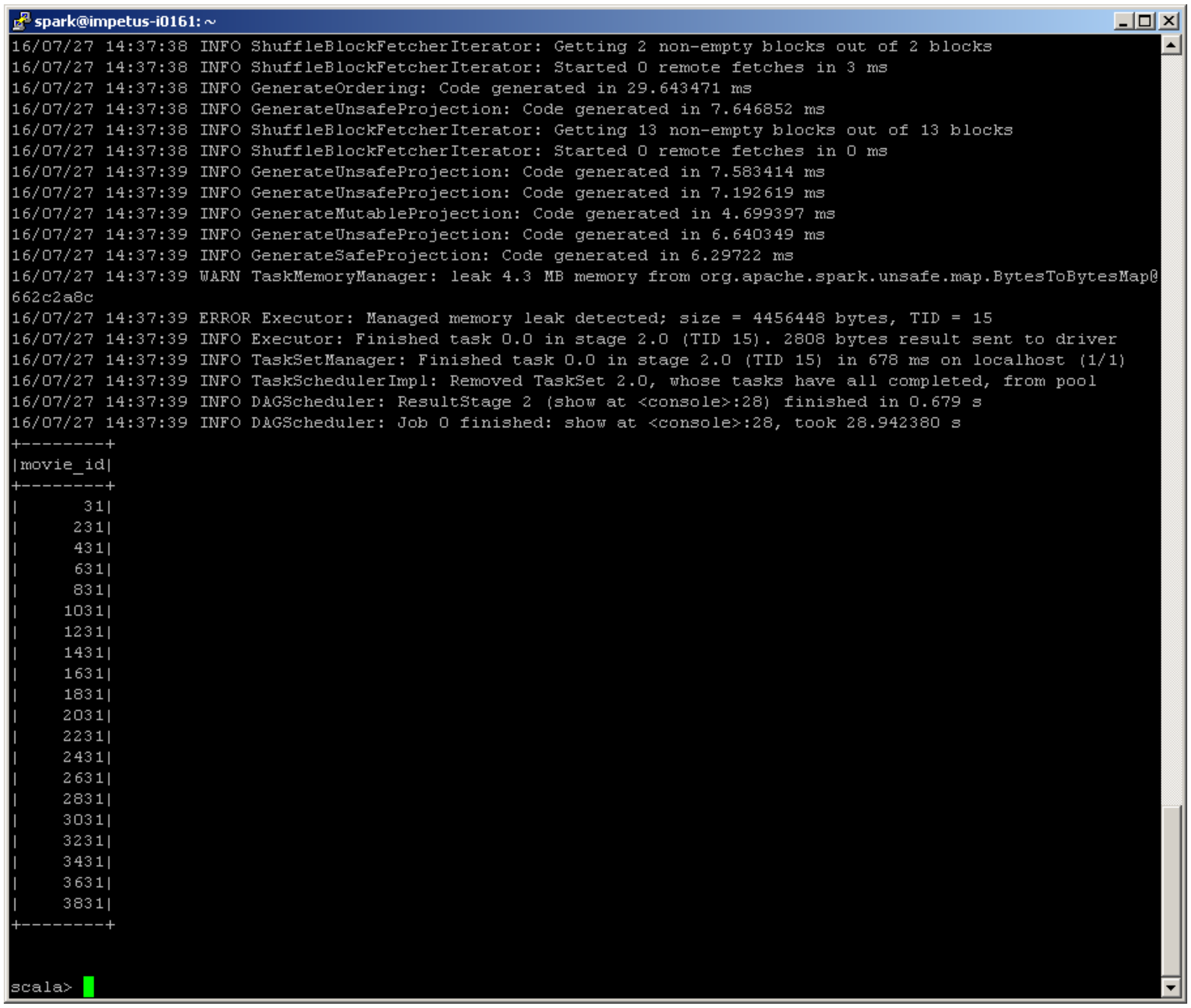
result.show()



```
16/07/27 14:35:00 INFO metastore: Connected to metastore.
16/07/27 14:35:00 INFO SessionState: Created local directory: /tmp/aef45c33-5555-4e81-af9e-c5
16/07/27 14:35:00 INFO SessionState: Created HDFS directory: /tmp/hive/spark/aef45c33-5555-4e
16/07/27 14:35:00 INFO SessionState: Created local directory: /tmp/spark/aef45c33-5555-4e81-a
16/07/27 14:35:00 INFO SessionState: Created HDFS directory: /tmp/hive/spark/aef45c33-5555-4e
16/07/27 14:35:00 INFO SparkILoop: Created sql context (with Hive support)..
SQL context available as sqlContext.

scala> val result = sqlContext.sql("FROM movie_lens_data.movies, movie_lens_data.ratings sele
ies.movie_id = ratings.movie_id  LIMIT 20")
16/07/27 14:35:41 INFO ParseDriver: Parsing command: FROM movie_lens_data.movies, movie_lens_
ovie_id where movies.movie_id = ratings.movie_id  LIMIT 20
16/07/27 14:35:41 INFO ParseDriver: Parse Completed
result: org.apache.spark.sql.DataFrame = [movie_id: bigint]

scala> result.show()
```



```
_data.db/ratings/part-m-00001_a_1:0+134217739
16/07/27 14:37:10 INFO HadoopRDD: Input split: hdfs://EETeamJ1/apps/hive/warehouse/movie_lens
_data.db/ratings/part-m-00000_a_1:0+134217739
16/07/27 14:37:10 INFO HadoopRDD: Input split: hdfs://EETeamJ1/apps/hive/warehouse/movie_lens
_data.db/ratings/part-m-00001:0+129194297
16/07/27 14:37:10 INFO deprecation: mapred.tip.id is deprecated. Instead, use mapreduce.task.
id
16/07/27 14:37:10 INFO deprecation: mapred.task.id is deprecated. Instead, use mapreduce.task
.attempt.id
16/07/27 14:37:10 INFO deprecation: mapred.task.is.map is deprecated. Instead, use mapreduce.
task.ismap
16/07/27 14:37:10 INFO deprecation: mapred.task.partition is deprecated. Instead, use mapredu
ce.task.partition
16/07/27 14:37:10 INFO deprecation: mapred.job.id is deprecated. Instead, use mapreduce.job.i
d
16/07/27 14:37:10 INFO GenerateUnsafeProjection: Code generated in 183.680577 ms
16/07/27 14:37:10 INFO GenerateMutableProjection: Code generated in 9.277515 ms
16/07/27 14:37:19 INFO Executor: Finished task 0.0 in stage 0.0 (TID 0). 2455 bytes result sent to driver
16/07/27 14:37:19 INFO Executor: Finished task 2.0 in stage 0.0 (TID 2). 2455 bytes result sent to driver
16/07/27 14:37:19 INFO Executor: Finished task 1.0 in stage 0.0 (TID 1). 2455 bytes result sent to driver
16/07/27 14:37:19 INFO Executor: Finished task 3.0 in stage 0.0 (TID 3). 2455 bytes result sent to driver
16/07/27 14:37:19 INFO TaskSetManager: Starting task 4.0 in stage 0.0 (TID 4, localhost, partition 4,ANY, 2166
 bytes)
16/07/27 14:37:19 INFO Executor: Running task 4.0 in stage 0.0 (TID 4)
16/07/27 14:37:19 INFO TaskSetManager: Starting task 5.0 in stage 0.0 (TID 5, localhost, partition 5,ANY, 2170
 bytes)
16/07/27 14:37:19 INFO Executor: Running task 5.0 in stage 0.0 (TID 5)
16/07/27 14:37:19 INFO TaskSetManager: Starting task 6.0 in stage 0.0 (TID 6, localhost, partition 6,ANY, 2166
 bytes)
16/07/27 14:37:19 INFO Executor: Running task 6.0 in stage 0.0 (TID 6)
16/07/27 14:37:19 INFO HadoopRDD: Input split: hdfs://EETeamJ1/apps/hive/warehouse/movie_lens_data.db/ratings/
part-m-00002_a_1:0+134217741
16/07/27 14:37:19 INFO HadoopRDD: Input split: hdfs://EETeamJ1/apps/hive/warehouse/movie_lens_data.db/ratings/
part-m-00002:0+129345586
16/07/27 14:37:19 INFO TaskSetManager: Starting task 7.0 in stage 0.0 (TID 7, localhost, partition 7,ANY, 2170
 bytes)
16/07/27 14:37:19 INFO Executor: Running task 7.0 in stage 0.0 (TID 7)
16/07/27 14:37:19 INFO HadoopRDD: Input split: hdfs://EETeamJ1/apps/hive/warehouse/movie_lens_data.db/ratings/
part-m-00003_a_1:0+134217708
16/07/27 14:37:19 INFO HadoopRDD: Input split: hdfs://EETeamJ1/apps/hive/warehouse/movie_lens_data.db/ratings/
part-m-00003:0+129345991
16/07/27 14:37:19 INFO TaskSetManager: Finished task 2.0 in stage 0.0 (TID 2) in 8791 ms on localhost (1/13)
16/07/27 14:37:19 INFO TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 8793 ms on localhost (2/13)
16/07/27 14:37:19 INFO TaskSetManager: Finished task 3.0 in stage 0.0 (TID 3) in 8793 ms on localhost (3/13)
16/07/27 14:37:19 INFO TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 8819 ms on localhost (4/13)
```



spark@impetus-i0161: ~

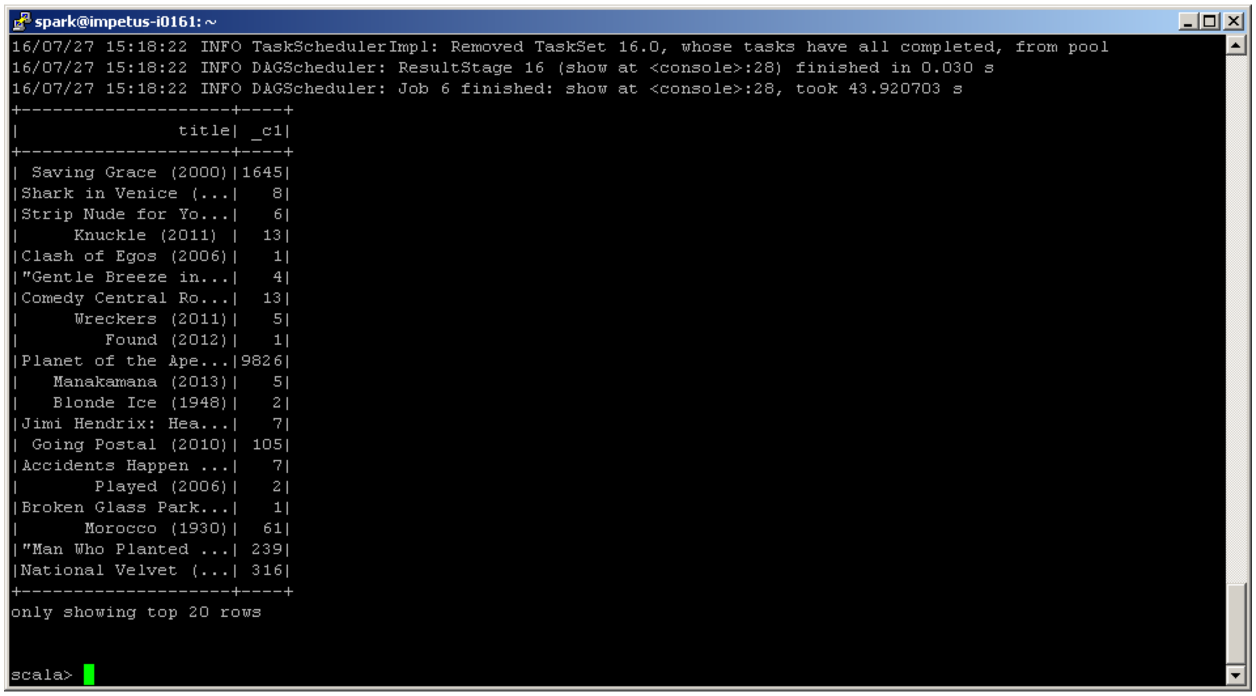
```
16/07/27 14:37:38 INFO ShuffleBlockFetcherIterator: Getting 2 non-empty blocks out of 2 blocks
16/07/27 14:37:38 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 3 ms
16/07/27 14:37:38 INFO GenerateOrdering: Code generated in 29.643471 ms
16/07/27 14:37:38 INFO GenerateUnsafeProjection: Code generated in 7.646852 ms
16/07/27 14:37:38 INFO ShuffleBlockFetcherIterator: Getting 13 non-empty blocks out of 13 blocks
16/07/27 14:37:38 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/07/27 14:37:39 INFO GenerateUnsafeProjection: Code generated in 7.583414 ms
16/07/27 14:37:39 INFO GenerateUnsafeProjection: Code generated in 7.192619 ms
16/07/27 14:37:39 INFO GenerateMutableProjection: Code generated in 4.699397 ms
16/07/27 14:37:39 INFO GenerateUnsafeProjection: Code generated in 6.640349 ms
16/07/27 14:37:39 INFO GenerateSafeProjection: Code generated in 6.29722 ms
16/07/27 14:37:39 WARN TaskMemoryManager: leak 4.3 MB memory from org.apache.spark.unsafe.map.BytesToBytesMap@
662c2a8c
16/07/27 14:37:39 ERROR Executor: Managed memory leak detected; size = 4456448 bytes, TID = 15
16/07/27 14:37:39 INFO Executor: Finished task 0.0 in stage 2.0 (TID 15). 2808 bytes result sent to driver
16/07/27 14:37:39 INFO TaskSetManager: Finished task 0.0 in stage 2.0 (TID 15) in 678 ms on localhost (1/1)
16/07/27 14:37:39 INFO TaskSchedulerImpl: Removed TaskSet 2.0, whose tasks have all completed, from pool
16/07/27 14:37:39 INFO DAGScheduler: ResultStage 2 (show at <console>:28) finished in 0.679 s
16/07/27 14:37:39 INFO DAGScheduler: Job 0 finished: show at <console>:28, took 28.942380 s
+--------+
|movie_id|
+--------+
|      31|
|     231|
|     431|
|     631|
|     831|
|    1031|
|    1231|
|    1431|
|    1631|
|    1831|
|    2031|
|    2231|
|    2431|
|    2631|
|    2831|
|    3031|
|    3231|
|    3431|
|    3631|
|    3831|
+--------+


scala>
```

(7) All distinct movie id are shown in above screenshot as result.

(8) Run below specific use cases

USE CASE 1: List all the movies and the number of ratings

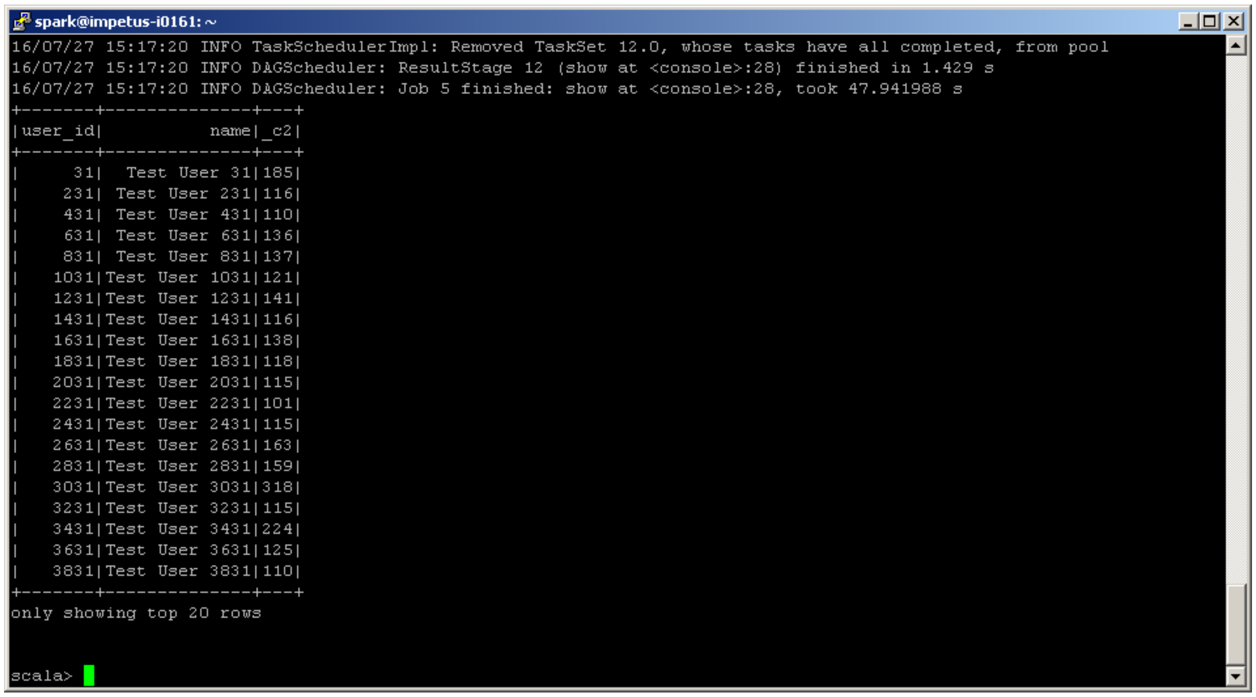
val result = sqlContext.sql("select title, count(*) from movie_lens_data.movies right outer join movie_lens_data.ratings on movies.movie_id=ratings.movie_id group by movies.movie_id, title")


spark@impetus-i0161: ~

```
16/07/27 15:18:22 INFO TaskSchedulerImpl: Removed TaskSet 16.0, whose tasks have all completed, from pool
16/07/27 15:18:22 INFO DAGScheduler: ResultStage 16 (show at <console>:28) finished in 0.030 s
16/07/27 15:18:22 INFO DAGScheduler: Job 6 finished: show at <console>:28, took 43.920703 s
+--------------------+----+
|               title| _c1|
+--------------------+----+
| Saving Grace (2000)|1645|
|Shark in Venice (...|   8|
|Strip Nude for Yo...|   6|
|     Knuckle (2011) |  13|
|Clash of Egos (2006)|   1|
|"Gentle Breeze in...|   4|
|Comedy Central Ro...|  13|
|     Wreckers (2011)|   5|
|        Found (2012)|   1|
|Planet of the Ape...|9826|
|   Manakamana (2013)|   5|
|   Blonde Ice (1948)|   2|
|Jimi Hendrix: Hea...|   7|
| Going Postal (2010)| 105|
|Accidents Happen ...|   7|
|       Played (2006)|   2|
|Broken Glass Park...|   1|
|      Morocco (1930)|  61|
|"Man Who Planted ...| 239|
|National Velvet (...| 316|
+--------------------+----+
only showing top 20 rows

scala> 
```

USE CASE 2: List all the users and the number of ratings they have done for a movie

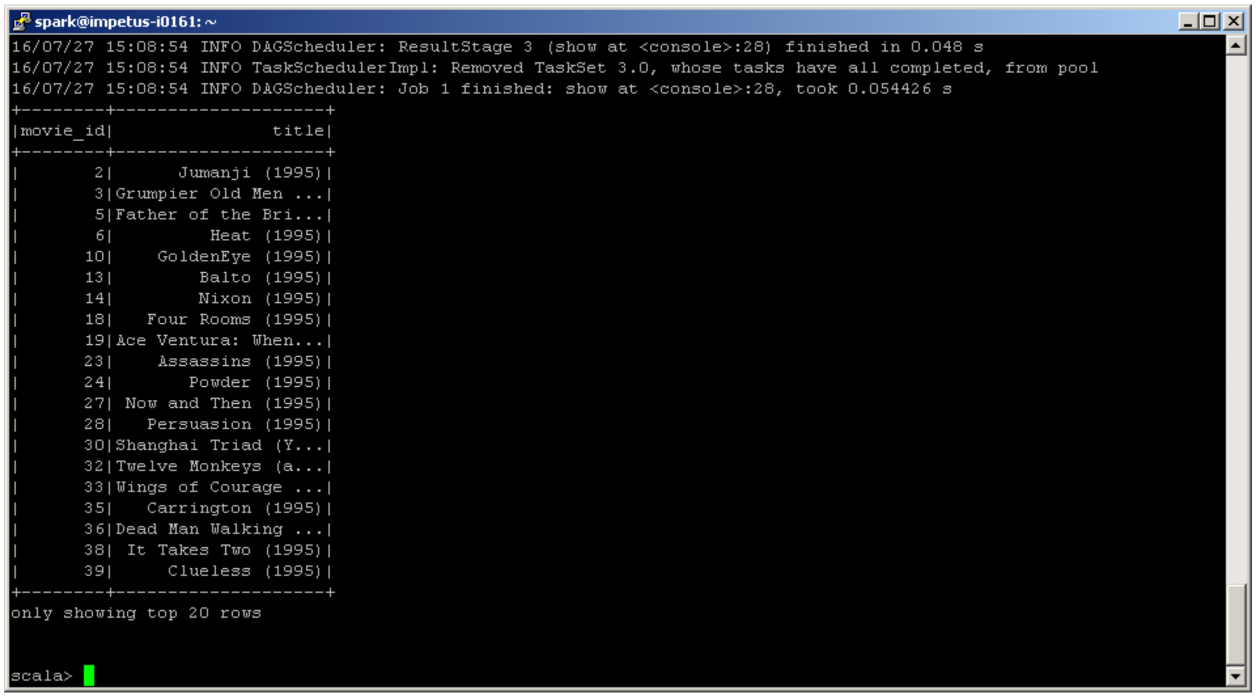
val result = sqlContext.sql("SELECT u.user_id, u.name, COUNT(r.rating) FROM movie_lens_data.users u, movie_lens_data.ratings r WHERE u.user_id=r.user_id GROUP BY u.user_id, u.name")


spark@impetus-i0161: ~

```
16/07/27 15:17:20 INFO TaskSchedulerImpl: Removed TaskSet 12.0, whose tasks have all completed, from pool
16/07/27 15:17:20 INFO DAGScheduler: ResultStage 12 (show at <console>:28) finished in 1.429 s
16/07/27 15:17:20 INFO DAGScheduler: Job 5 finished: show at <console>:28, took 47.941988 s
+-------+--------------+---+
|user_id|          name|_c2|
+-------+--------------+---+
|     31|  Test User 31|185|
|    231| Test User 231|116|
|    431| Test User 431|110|
|    631| Test User 631|136|
|    831| Test User 831|137|
|   1031|Test User 1031|121|
|   1231|Test User 1231|141|
|   1431|Test User 1431|116|
|   1631|Test User 1631|138|
|   1831|Test User 1831|118|
|   2031|Test User 2031|115|
|   2231|Test User 2231|101|
|   2431|Test User 2431|115|
|   2631|Test User 2631|163|
|   2831|Test User 2831|159|
|   3031|Test User 3031|318|
|   3231|Test User 3231|115|
|   3431|Test User 3431|224|
|   3631|Test User 3631|125|
|   3831|Test User 3831|110|
+-------+--------------+---+
only showing top 20 rows

scala> 
```

USE CASE 3: List all the Movie IDs which have been rated (Movie Id with at least one
user rating it)

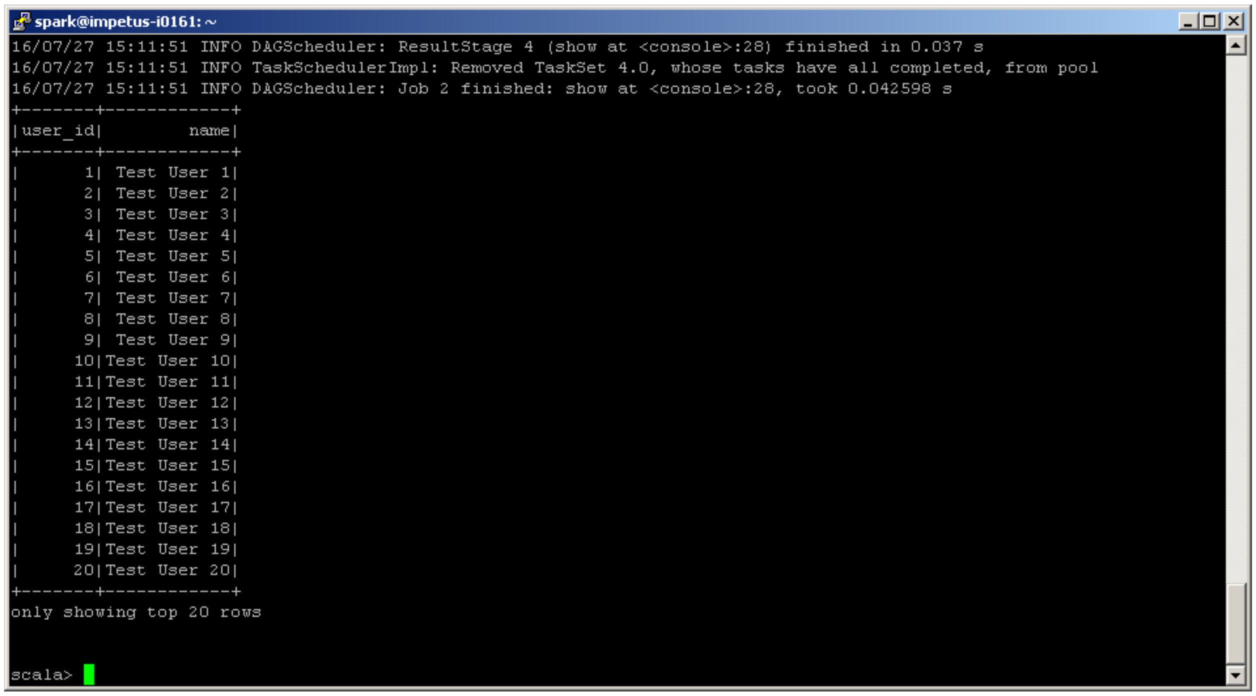
val result = sqlContext.sql("select movie_id, title from movie_lens_data.mov_rating_count")



```
16/07/27 15:08:54 INFO DAGScheduler: ResultStage 3 (show at <console>:28) finished in 0.048 s
16/07/27 15:08:54 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
16/07/27 15:08:54 INFO DAGScheduler: Job 1 finished: show at <console>:28, took 0.054426 s
+--------+--------------------+
|movie_id|               title|
+--------+--------------------+
|       2|      Jumanji (1995)|
|       3|Grumpier Old Men ...|
|       5|Father of the Bri...|
|       6|         Heat (1995)|
|      10|    GoldenEye (1995)|
|      13|        Balto (1995)|
|      14|        Nixon (1995)|
|      18|   Four Rooms (1995)|
|      19|Ace Ventura: When...|
|      23|    Assassins (1995)|
|      24|       Powder (1995)|
|      27| Now and Then (1995)|
|      28|   Persuasion (1995)|
|      30|Shanghai Triad (Y...|
|      32|Twelve Monkeys (a...|
|      33|Wings of Courage ...|
|      35|   Carrington (1995)|
|      36|Dead Man Walking ...|
|      38| It Takes Two (1995)|
|      39|     Clueless (1995)|
+--------+--------------------+
only showing top 20 rows

scala> 
```

USE CASE 4: List all the Users who have rated the movies (Users who have rated
At least one movie)

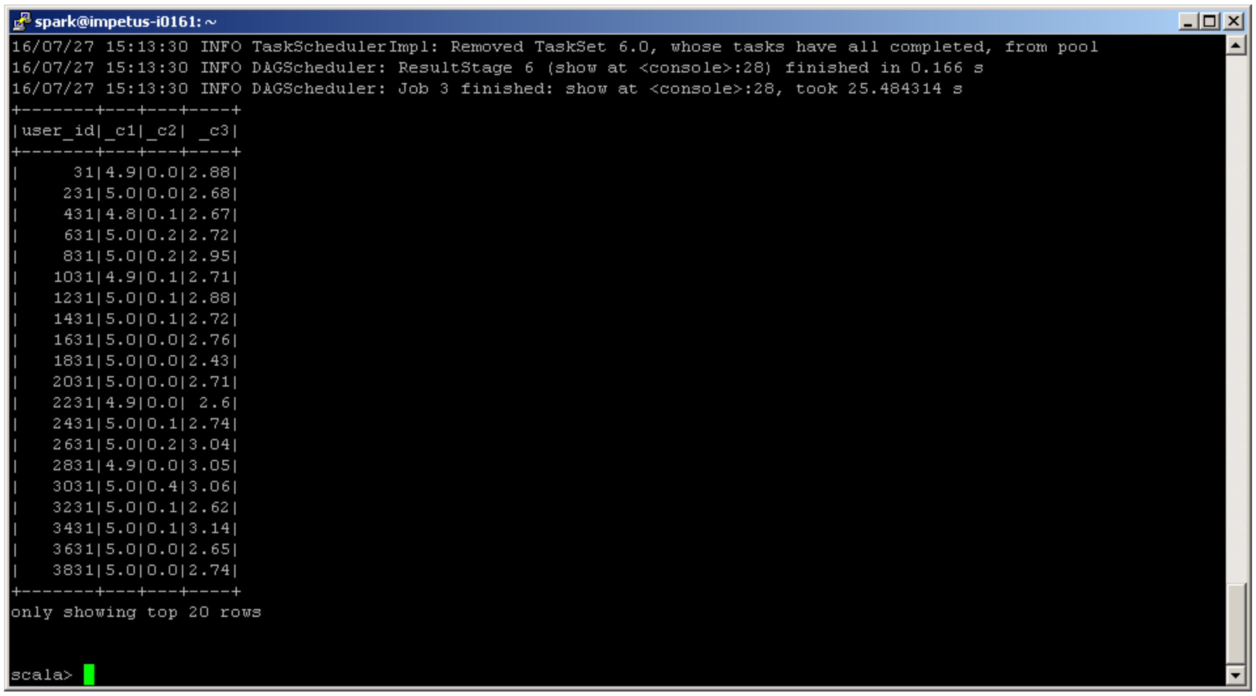
val result = sqlContext.sql("select user_id, name from movie_lens_data.user_rating_count")



```
16/07/27 15:11:51 INFO DAGScheduler: ResultStage 4 (show at <console>:28) finished in 0.037 s
16/07/27 15:11:51 INFO TaskSchedulerImpl: Removed TaskSet 4.0, whose tasks have all completed, from pool
16/07/27 15:11:51 INFO DAGScheduler: Job 2 finished: show at <console>:28, took 0.042598 s
+-------+------------+
|user_id|        name|
+-------+------------+
|      1| Test User 1|
|      2| Test User 2|
|      3| Test User 3|
|      4| Test User 4|
|      5| Test User 5|
|      6| Test User 6|
|      7| Test User 7|
|      8| Test User 8|
|      9| Test User 9|
|     10|Test User 10|
|     11|Test User 11|
|     12|Test User 12|
|     13|Test User 13|
|     14|Test User 14|
|     15|Test User 15|
|     16|Test User 16|
|     17|Test User 17|
|     18|Test User 18|
|     19|Test User 19|
|     20|Test User 20|
+-------+------------+
only showing top 20 rows

scala>
```

USE CASE 5: List of all the User with the max, min, average ratings they have given
against any movie

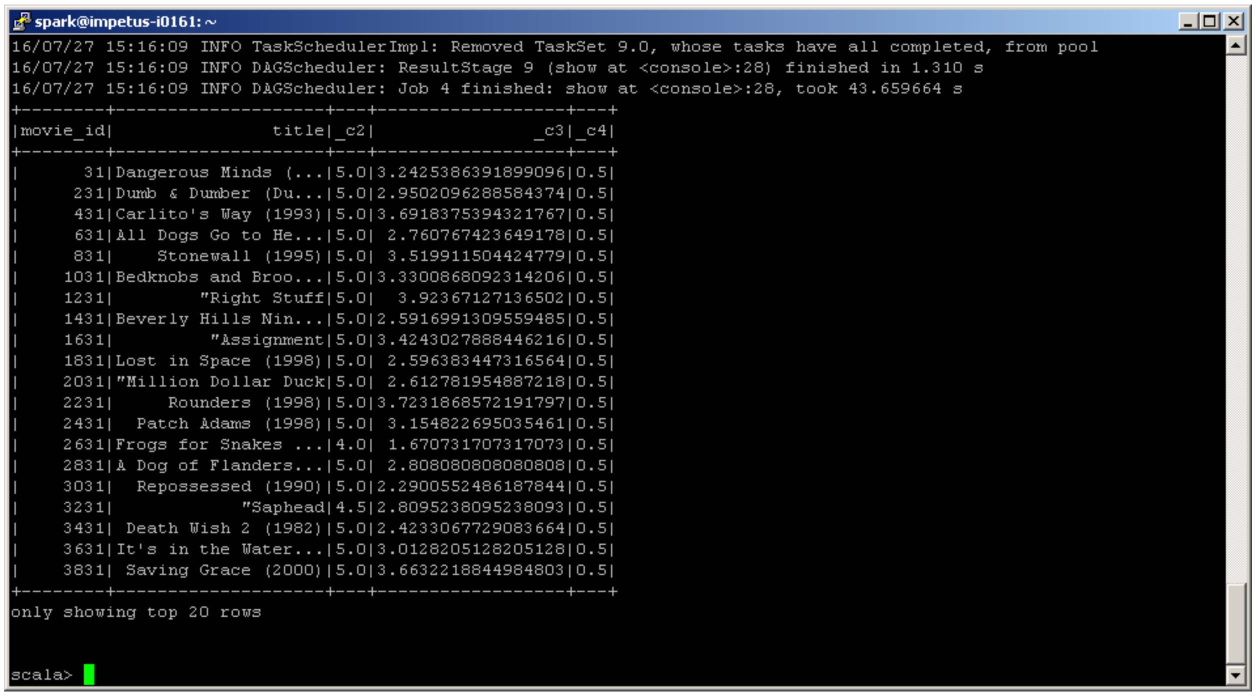
val result = sqlContext.sql("select user_id, max(rating), min(rating), round(avg(rating),2) from movie_lens_data.ratings group by user_id")



```
16/07/27 15:13:30 INFO TaskSchedulerImpl: Removed TaskSet 6.0, whose tasks have all completed, from pool
16/07/27 15:13:30 INFO DAGScheduler: ResultStage 6 (show at <console>:28) finished in 0.166 s
16/07/27 15:13:30 INFO DAGScheduler: Job 3 finished: show at <console>:28, took 25.484314 s
+-------+---+---+----+
|user_id|_c1|_c2| _c3|
+-------+---+---+----+
|     31|4.9|0.0|2.88|
|    231|5.0|0.0|2.68|
|    431|4.8|0.1|2.67|
|    631|5.0|0.2|2.72|
|    831|5.0|0.2|2.95|
|   1031|4.9|0.1|2.71|
|   1231|5.0|0.1|2.88|
|   1431|5.0|0.1|2.72|
|   1631|5.0|0.0|2.76|
|   1831|5.0|0.0|2.43|
|   2031|5.0|0.0|2.71|
|   2231|4.9|0.0| 2.6|
|   2431|5.0|0.1|2.74|
|   2631|5.0|0.2|3.04|
|   2831|4.9|0.0|3.05|
|   3031|5.0|0.4|3.06|
|   3231|5.0|0.1|2.62|
|   3431|5.0|0.1|3.14|
|   3631|5.0|0.0|2.65|
|   3831|5.0|0.0|2.74|
+-------+---+---+----+
only showing top 20 rows

scala>
```

USE CASE 6: List all the Movies with the max, min, average ratings given by any user

val result = sqlContext.sql("SELECT m.movie_id, m.title, MAX(r.rating), AVG(r.rating), MIN(r.rating) FROM movie_lens_data.movies m, movie_lens_data.ratings r WHERE m.movie_id=r.movie_id GROUP BY m.movie_id,m.title")



```
16/07/27 15:16:09 INFO TaskSchedulerImpl: Removed TaskSet 9.0, whose tasks have all completed, from pool
16/07/27 15:16:09 INFO DAGScheduler: ResultStage 9 (show at <console>:28) finished in 1.310 s
16/07/27 15:16:09 INFO DAGScheduler: Job 4 finished: show at <console>:28, took 43.659664 s
+--------+--------------------+---+------------------+---+
|movie_id|               title|_c2|               _c3|_c4|
+--------+--------------------+---+------------------+---+
|      31|Dangerous Minds (...|5.0|3.2425386391899096|0.5|
|     231|Dumb & Dumber (Du...|5.0|2.9502096288584374|0.5|
|     431|Carlito's Way (1993)|5.0|3.6918375394321767|0.5|
|     631|All Dogs Go to He...|5.0| 2.760767423649178|0.5|
|     831|    Stonewall (1995)|5.0| 3.519911504424779|0.5|
|    1031|Bedknobs and Broo...|5.0|3.3300868092314206|0.5|
|    1231|         "Right Stuff|5.0|  3.92367127136502|0.5|
|    1431|Beverly Hills Nin...|5.0|2.5916991309559485|0.5|
|    1631|          "Assignment|5.0|3.4243027888446216|0.5|
|    1831|Lost in Space (1998)|5.0| 2.596383447316564|0.5|
|    2031|"Million Dollar Duck|5.0| 2.612781954887218|0.5|
|    2231|     Rounders (1998)|5.0|3.7231868572191797|0.5|
|    2431|  Patch Adams (1998)|5.0| 3.154822695035461|0.5|
|    2631|Frogs for Snakes ...|4.0| 1.670731707317073|0.5|
|    2831|A Dog of Flanders...|5.0| 2.808080808080808|0.5|
|    3031|  Repossessed (1990)|5.0|2.2900552486187844|0.5|
|    3231|             "Saphead|4.5|2.8095238095238093|0.5|
|    3431| Death Wish 2 (1982)|5.0|2.4233067729083664|0.5|
|    3631|It's in the Water...|5.0|3.0128205128205128|0.5|
|    3831| Saving Grace (2000)|5.0|3.6632218844984803|0.5|
+--------+--------------------+---+------------------+---+
only showing top 20 rows


scala>
```