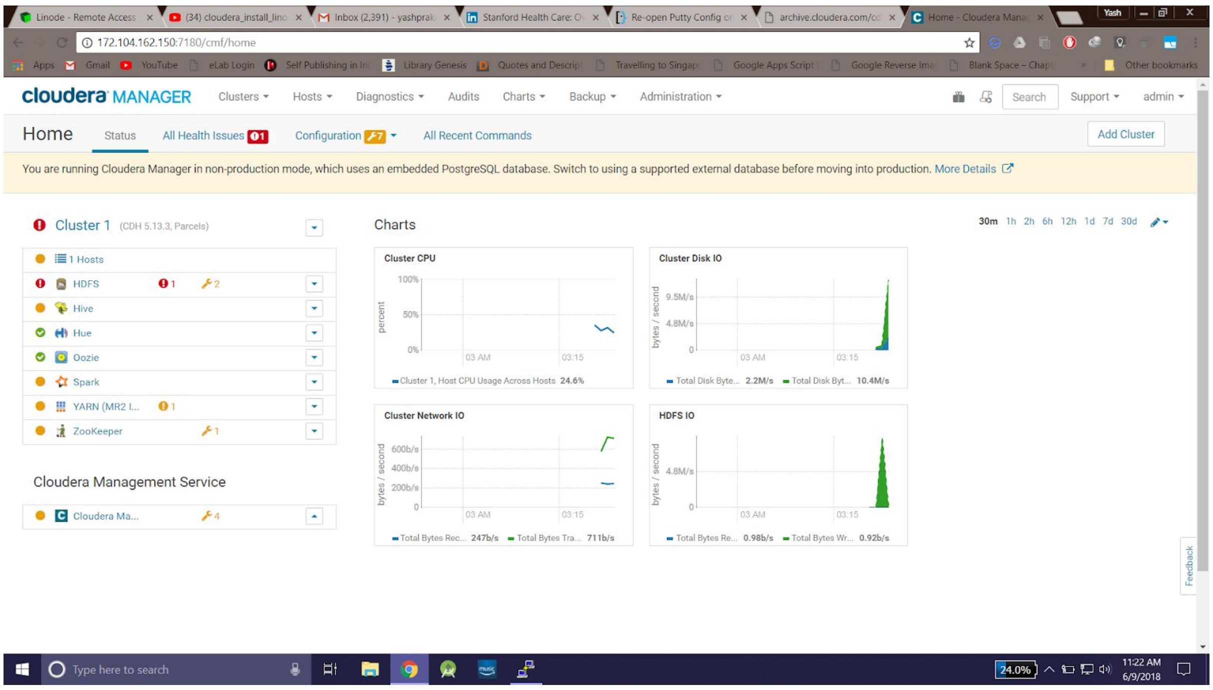
HPE PROJECT

Yash Prakash
Paras Sibal
Akshay Sachdeva
Ankit Sahu
Geet Jethwani

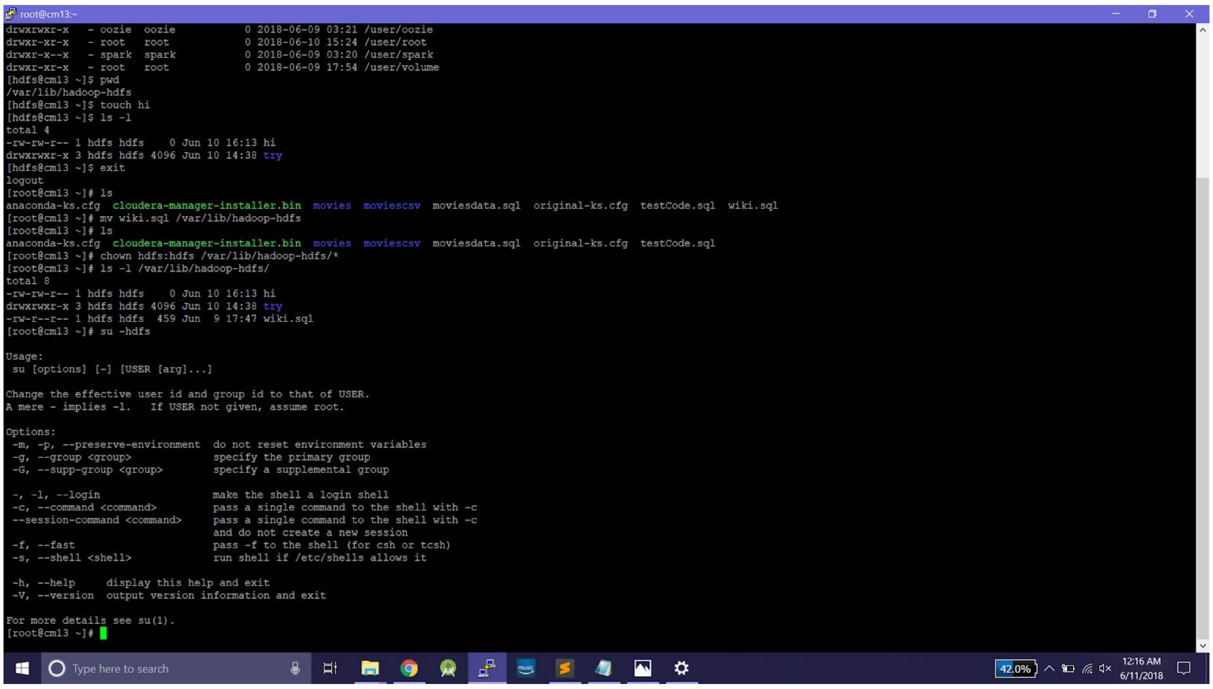
1. Main page of the cloudera manager 5.13

In the web-browser type: http://ip:7180



You are running Cloudera Manager in non-production mode, which uses an embedded PostgreSQL database. Switch to using a supported external database before moving into production. More Details

2. HDFS Upload sample file:



```
drwxrwxr-x   - oozie  oozie            0 2018-06-09 03:21 /user/oozie
drwxr-xr-x   - root   root             0 2018-06-10 15:24 /user/root
drwxr-x--x   - spark  spark            0 2018-06-09 03:20 /user/spark
drwxr-xr-x   - root   root             0 2018-06-09 17:54 /user/volume
[hdfs@cm13 ~]$ pwd
/var/lib/hadoop-hdfs
[hdfs@cm13 ~]$ touch hi
[hdfs@cm13 ~]$ ls -l
total 4
-rw-rw-r-- 1 hdfs hdfs    0 Jun 10 16:13 hi
drwxrwxr-x 3 hdfs hdfs 4096 Jun 10 14:38 try
[hdfs@cm13 ~]$ exit
logout
[root@cm13 ~]# ls
anaconda-ks.cfg  cloudera-manager-installer.bin  movies  moviescsv  moviesdata.sql  original-ks.cfg  testCode.sql  wiki.sql
[root@cm13 ~]# mv wiki.sql /var/lib/hadoop-hdfs
[root@cm13 ~]# ls
anaconda-ks.cfg  cloudera-manager-installer.bin  movies  moviescsv  moviesdata.sql  original-ks.cfg  testCode.sql
[root@cm13 ~]# chown hdfs:hdfs /var/lib/hadoop-hdfs/*
[root@cm13 ~]# ls -l /var/lib/hadoop-hdfs/
total 8
-rw-rw-r-- 1 hdfs hdfs    0 Jun 10 16:13 hi
drwxrwxr-x 3 hdfs hdfs 4096 Jun 10 14:38 try
-rw-r--r-- 1 hdfs hdfs  459 Jun  9 17:47 wiki.sql
[root@cm13 ~]# su -hdfs

Usage:
 su [options] [-] [USER [arg]...]

Change the effective user id and group id to that of USER.
A mere - implies -l.   If USER not given, assume root.

Options:
 -m, -p, --preserve-environment  do not reset environment variables
 -g, --group <group>             specify the primary group
 -G, --supp-group <group>        specify a supplemental group

 -, -l, --login                  make the shell a login shell
 -c, --command <command>         pass a single command to the shell with -c
 --session-command <command>     pass a single command to the shell with -c
                                 and do not create a new session
 -f, --fast                      pass -f to the shell (for csh or tcsh)
 -s, --shell <shell>             run shell if /etc/shells allows it

 -h, --help     display this help and exit
 -V, --version  output version information and exit

For more details see su(1).
[root@cm13 ~]#
```

Commands used in HDFS:

1. **hdfs dfs -ls /** *# list the directories*
2. **hdfs dfs -ls /user/** # navigate to user folder
3. **pwd** # view current directory location
4. **touch hi** # create empty file
5. **ls -l** # list all files with their ownership
6. **Exit** # exit user
7. **ls** # list directories
8. **hdfs dfs -mkdir /user/hdfs** # create a directory hdfs inside user
9. **hdfs dfs -put wiki.sql /user/hdfs** # put wiki.sql into hdfs
10. **hdfs dfs -ls /user/hdfs** # show list of directories in hdfs
11. **history**

Commands used in Linux root:

1. **su - hdfs** # switch user to hdfs
2. **mv wiki.sql /var/lib/hadoop-hdfs** # move wiki.sql to hdfs directory
3. **chown hdfs:hdfs /var/lib/hadoop-hdfs/*** # change ownership to hdfs
4. **ls -l /var/lib/hadoop-hdfs/** # list all directories in hdfs

3. Hive commands with outputs:

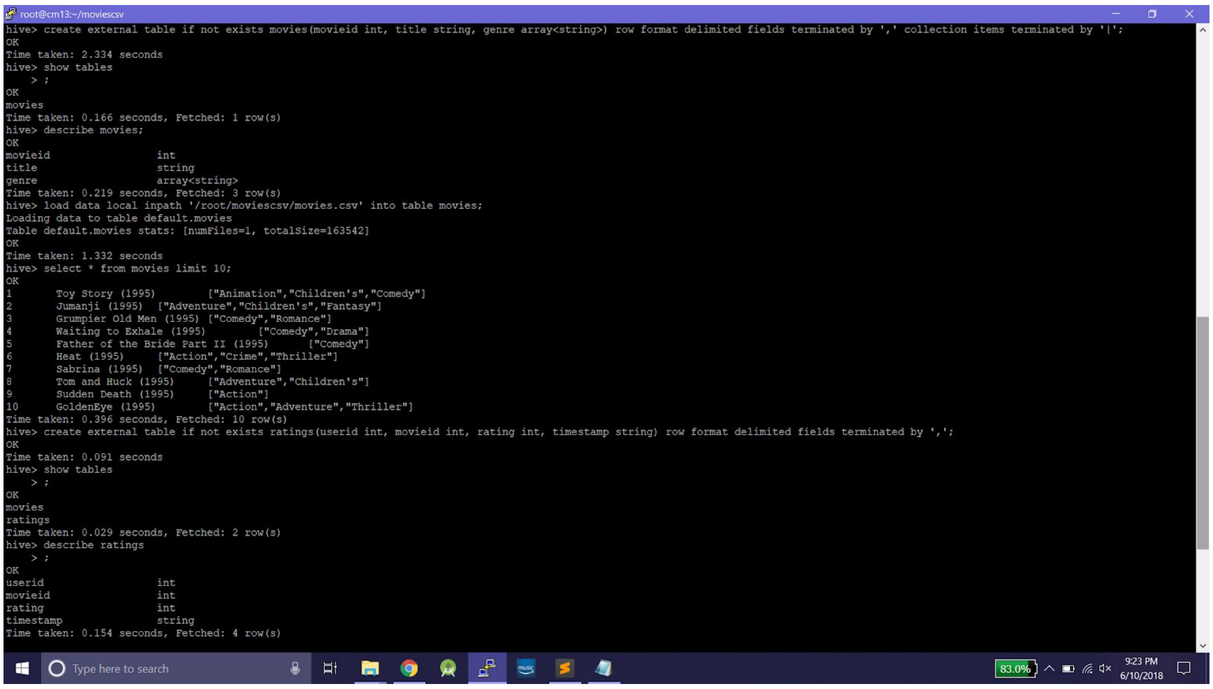
1. **Creating tables:**

I. Movies table for movies.csv
II. Ratings table for ratings.csv
III. Users table for users.csv

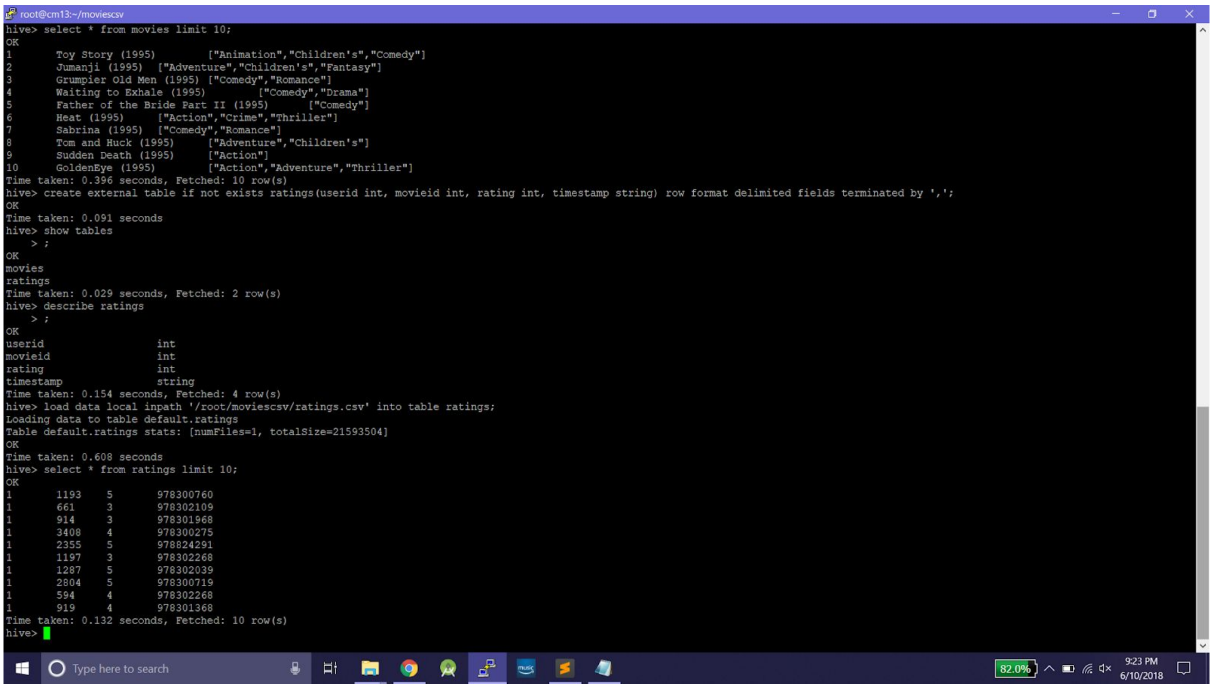
Replace all the ‘::’ delimiters by ‘,’ in the files, to convert them into .csv files.

Creating Tables:
Syntax:
create external table if not exists table_name(identifier_name datatype) row format delimited fields terminated by <delimiter>;
For movies table

1. create external table if not exists movies(movieid int, title string, genre array<string>) row format delimited fields terminated by ‘,’ collection by ‘|’

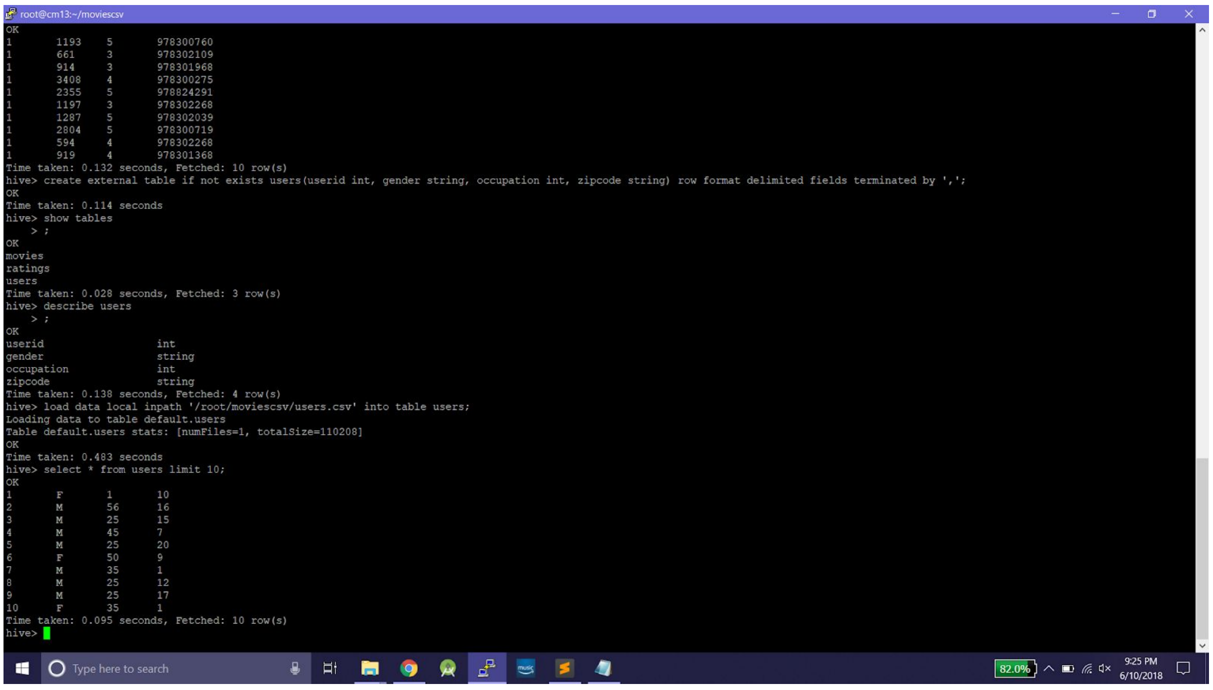


```
hive> create external table if not exists movies(movieid int, title string, genre array<string>) row format delimited fields terminated by ',' collection items terminated by '|';
OK
Time taken: 2.334 seconds
hive> show tables
    > ;
OK
movies
Time taken: 0.166 seconds, Fetched: 1 row(s)
hive> describe movies;
OK
movieid                 int
title                   string
genre                   array<string>
Time taken: 0.219 seconds, Fetched: 3 row(s)
hive> load data local inpath '/root/moviescsv/movies.csv' into table movies;
Loading data to table default.movies
Table default.movies stats: [numFiles=1, totalSize=163542]
OK
Time taken: 1.332 seconds
hive> select * from movies limit 10;
OK
1       Toy Story (1995)        ["Animation","Children's","Comedy"]
2       Jumanji (1995)  ["Adventure","Children's","Fantasy"]
3       Grumpier Old Men (1995) ["Comedy","Romance"]
4       Waiting to Exhale (1995)        ["Comedy","Drama"]
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's"]
9       Sudden Death (1995)     ["Action"]
10      GoldenEye (1995)        ["Action","Adventure","Thriller"]
Time taken: 0.396 seconds, Fetched: 10 row(s)
hive> create external table if not exists ratings(userid int, movieid int, rating int, timestamp string) row format delimited fields terminated by ',';
OK
Time taken: 0.091 seconds
hive> show tables
    > ;
OK
movies
ratings
Time taken: 0.029 seconds, Fetched: 2 row(s)
hive> describe ratings
    > ;
OK
userid                  int
movieid                 int
rating                  int
timestamp               string
Time taken: 0.154 seconds, Fetched: 4 row(s)
```



```
hive> select * from movies limit 10;
OK
1       Toy Story (1995)        ["Animation","Children's","Comedy"]
2       Jumanji (1995)  ["Adventure","Children's","Fantasy"]
3       Grumpier Old Men (1995) ["Comedy","Romance"]
4       Waiting to Exhale (1995)        ["Comedy","Drama"]
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's"]
9       Sudden Death (1995)     ["Action"]
10      GoldenEye (1995)        ["Action","Adventure","Thriller"]
Time taken: 0.396 seconds, Fetched: 10 row(s)
hive> create external table if not exists ratings(userid int, movieid int, rating int, timestamp string) row format delimited fields terminated by ',';
OK
Time taken: 0.091 seconds
hive> show tables
    > ;
OK
movies
ratings
Time taken: 0.029 seconds, Fetched: 2 row(s)
hive> describe ratings
    > ;
OK
userid                  int
movieid                 int
rating                  int
timestamp               string
Time taken: 0.154 seconds, Fetched: 4 row(s)
```

```
hive> load data local inpath '/root/moviescsv/ratings.csv' into table ratings;
Loading data to table default.ratings
Table default.ratings stats: [numFiles=1, totalSize=21593504]
OK
Time taken: 0.608 seconds
hive> select * from ratings limit 10;
OK
1       1193    5       978300760
1       661     3       978302109
1       914     3       978301968
1       3408    4       978300275
1       2355    5       978824291
1       1197    3       978302268
1       1287    5       978302039
1       2804    5       978300719
1       594     4       978302268
1       919     4       978301368
Time taken: 0.132 seconds, Fetched: 10 row(s)
hive>
```



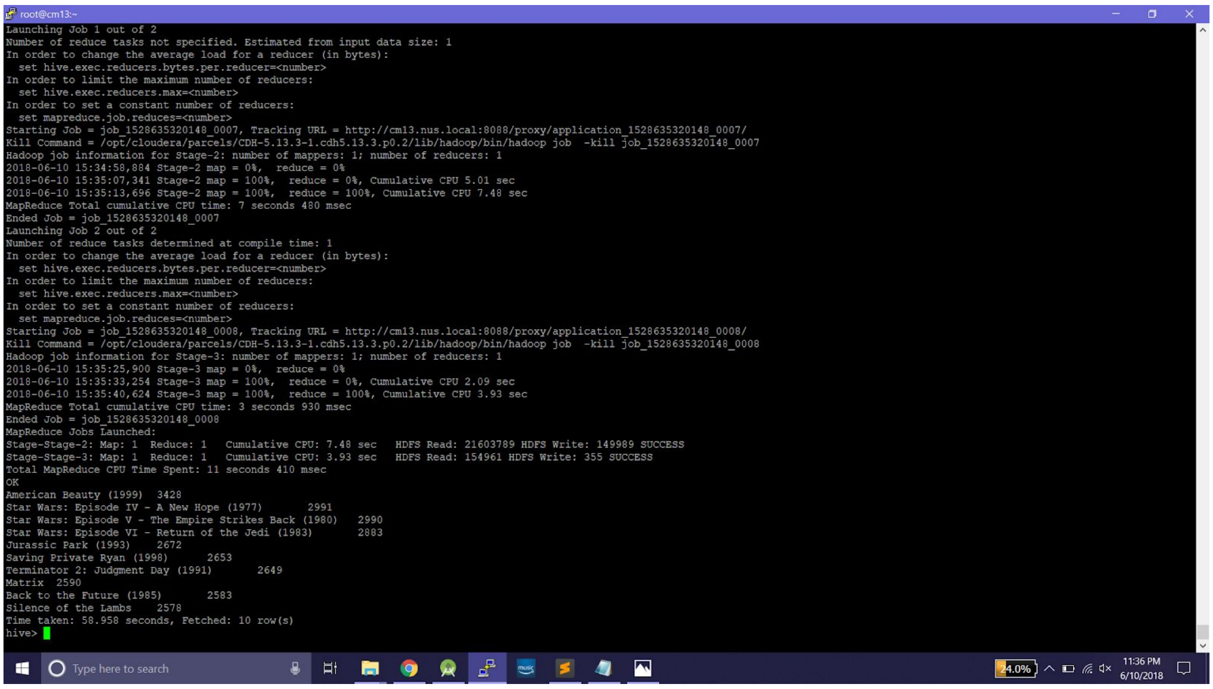
```
OK
1       1193    5       978300760
1       661     3       978302109
1       914     3       978301968
1       3408    4       978300275
1       2355    5       978824291
1       1197    3       978302268
1       1287    5       978302039
1       2804    5       978300719
1       594     4       978302268
1       919     4       978301368
Time taken: 0.132 seconds, Fetched: 10 row(s)
hive> create external table if not exists users(userid int, gender string, occupation int, zipcode string) row format delimited fields terminated by ',';
OK
Time taken: 0.114 seconds
hive> show tables
    > ;
OK
movies
ratings
users
Time taken: 0.028 seconds, Fetched: 3 row(s)
hive> describe users
    > ;
OK
userid                  int
gender                  string
occupation              int
zipcode                 string
Time taken: 0.138 seconds, Fetched: 4 row(s)
hive> load data local inpath '/root/moviescsv/users.csv' into table users;
Loading data to table default.users
Table default.users stats: [numFiles=1, totalSize=110208]
OK
Time taken: 0.483 seconds
hive> select * from users limit 10;
OK
1       F       1       10
2       M       56      16
3       M       25      15
4       M       45      7
5       M       25      20
6       F       50      9
7       M       35      1
8       M       25      12
9       M       25      17
10      F       35      1
Time taken: 0.095 seconds, Fetched: 10 row(s)
hive>
```

Part 1:

select movies.title, count(*) as count from
movies join ratings on(movies.movieid = ratings.movieid)
group by movies.title
order by count DESC
limit 10

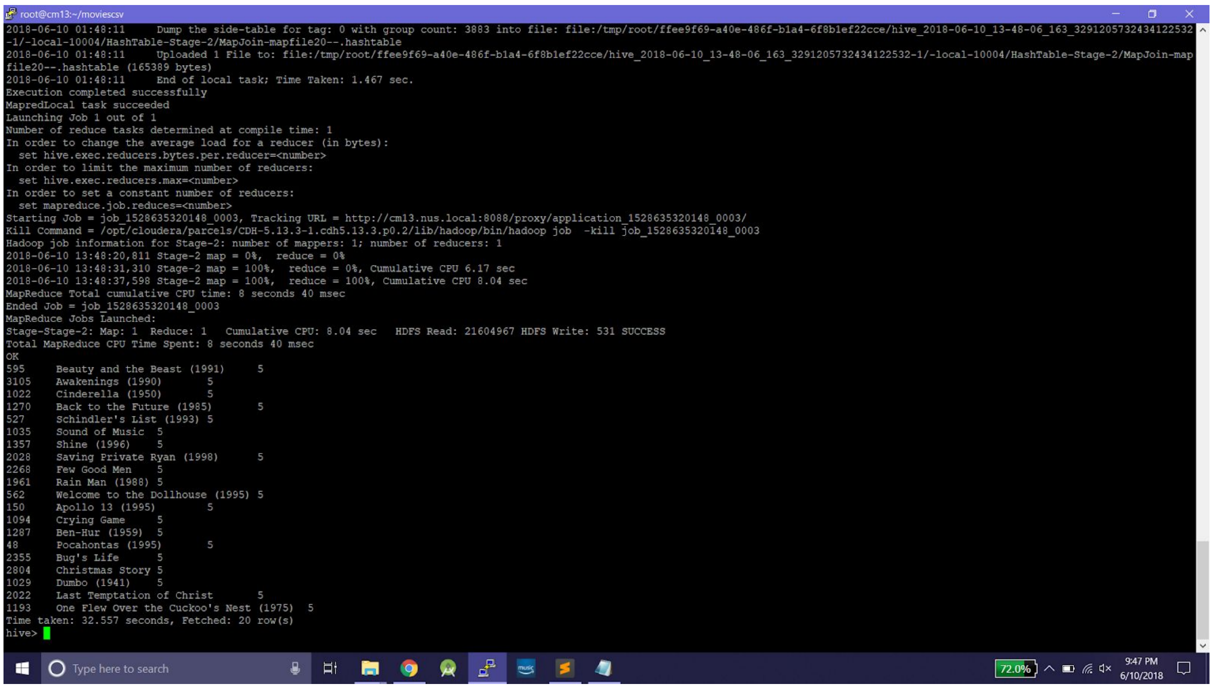
# To display the movie title of the top 10 viewed movies, we use the 'join' command. We use 'movieid' as the primary key from the movies table and combine it with the 'movieid' from the ratings table to get the title of the movies in the output produced.

# We group the movies via title, ordered in descending order and limit the queries to 10.



```
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1528635320148_0007, Tracking URL = http://cm13.nus.local:8088/proxy/application_1528635320148_0007/
Kill Command = /opt/cloudera/parcels/CDH-5.13.3-1.cdh5.13.3.p0.2/lib/hadoop/bin/hadoop job  -kill job_1528635320148_0007
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-06-10 15:34:58,884 Stage-2 map = 0%,  reduce = 0%
2018-06-10 15:35:07,341 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 5.01 sec
2018-06-10 15:35:13,696 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 7.48 sec
MapReduce Total cumulative CPU time: 7 seconds 480 msec
Ended Job = job_1528635320148_0007
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1528635320148_0008, Tracking URL = http://cm13.nus.local:8088/proxy/application_1528635320148_0008/
Kill Command = /opt/cloudera/parcels/CDH-5.13.3-1.cdh5.13.3.p0.2/lib/hadoop/bin/hadoop job  -kill job_1528635320148_0008
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2018-06-10 15:35:25,900 Stage-3 map = 0%,  reduce = 0%
2018-06-10 15:35:33,254 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 2.09 sec
2018-06-10 15:35:40,624 Stage-3 map = 100%,  reduce = 100%, Cumulative CPU 3.93 sec
MapReduce Total cumulative CPU time: 3 seconds 930 msec
Ended Job = job_1528635320148_0008
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 7.48 sec   HDFS Read: 21603789 HDFS Write: 149989 SUCCESS
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 3.93 sec   HDFS Read: 154961 HDFS Write: 355 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 410 msec
OK
American Beauty (1999)  3428
Star Wars: Episode IV - A New Hope (1977)       2991
Star Wars: Episode V - The Empire Strikes Back (1980)   2990
Star Wars: Episode VI - Return of the Jedi (1983)       2883
Jurassic Park (1993)    2672
Saving Private Ryan (1998)      2653
Terminator 2: Judgment Day (1991)       2649
Matrix  2590
Back to the Future (1985)       2583
Silence of the Lambs    2578
Time taken: 58.958 seconds, Fetched: 10 row(s)
hive>
```

Part 2:



```
2018-06-10 01:48:11     Dump the side-table for tag: 0 with group count: 3883 into file: file:/tmp/root/ffee9f69-a40e-486f-b1a4-6f8b1ef22cce/hive_2018-06-10_13-48-06_163_3291205732434122532
-1/-local-10004/HashTable-Stage-2/MapJoin-mapfile20--.hashtable
2018-06-10 01:48:11     Uploaded 1 File to: file:/tmp/root/ffee9f69-a40e-486f-b1a4-6f8b1ef22cce/hive_2018-06-10_13-48-06_163_3291205732434122532-1/-local-10004/HashTable-Stage-2/MapJoin-map
file20--.hashtable (165389 bytes)
2018-06-10 01:48:11     End of local task; Time Taken: 1.467 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1528635320148_0003, Tracking URL = http://cm13.nus.local:8088/proxy/application_1528635320148_0003/
Kill Command = /opt/cloudera/parcels/CDH-5.13.3-1.cdh5.13.3.p0.2/lib/hadoop/bin/hadoop job  -kill job_1528635320148_0003
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-06-10 13:48:20,811 Stage-2 map = 0%,  reduce = 0%
2018-06-10 13:48:31,310 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 6.17 sec
2018-06-10 13:48:37,598 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 8.04 sec
MapReduce Total cumulative CPU time: 8 seconds 40 msec
Ended Job = job_1528635320148_0003
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 8.04 sec   HDFS Read: 21604967 HDFS Write: 531 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 40 msec
OK
595     Beauty and the Beast (1991)     5
3105    Awakenings (1990)       5
1022    Cinderella (1950)       5
1270    Back to the Future (1985)       5
527     Schindler's List (1993) 5
1035    Sound of Music  5
1357    Shine (1996)    5
2028    Saving Private Ryan (1998)      5
2268    Few Good Men    5
1961    Rain Man (1988) 5
562     Welcome to the Dollhouse (1995) 5
150     Apollo 13 (1995)        5
1094    Crying Game     5
1287    Ben-Hur (1959)  5
48      Pocahontas (1995)       5
2355    Bug's Life      5
2804    Christmas Story 5
1029    Dumbo (1941)    5
2022    Last Temptation of Christ       5
1193    One Flew Over the Cuckoo's Nest (1975)  5
Time taken: 32.557 seconds, Fetched: 20 row(s)
hive>
```

```
SELECT m.movieid, m.title, r.rating FROM
movies m JOIN rating r ON (m.movieid = r.movieid)
ORDER BY  r.rating DESC
LIMIT 20
```

# This command displays movie id, movie title and rating by joining rating table and movies table via primary key 'movieid' in both tables.
# We order it by ratings in the descending order and limit it to 20 queries.

ANOTHER APPROACH:

SELECT movies.title , count(*) as count FROM
movies JOIN ratings ON (movies.movieid = ratings.movieid) WHERE movieid IN (SELECT movieid FROM ratings WHERE count(rating)>40) GROUP BY movieid ORDER BY count desc limit 20;

# This command displays movie id, movie title and rating by joining ratings table and movies table via primary key(common key) 'movieid' in both tables.

# We select the 'movieid' from ratings table where the count of rating is greater than 40 and then we group them by 'movieid' using nested SELECT command.

# We order it by ratings in the descending order and limit it to 20 queries.

Part 3:

Approach: In this question, we can use the 'select' command to calculate the average ratings provided by users of each occupation and then we can 'group' them by age, and then 'order' them by occupation with their corresponding age group.

We can use the 'userid' as the primary key for joining the 'ratings' table and the 'users' table using the 'JOIN' command.

We can then order them by the 'users.occupation' key, and also the 'users.age' key using nested select command, sort them in the descending order using 'desc' and hence limit the outputs to '10'.

****