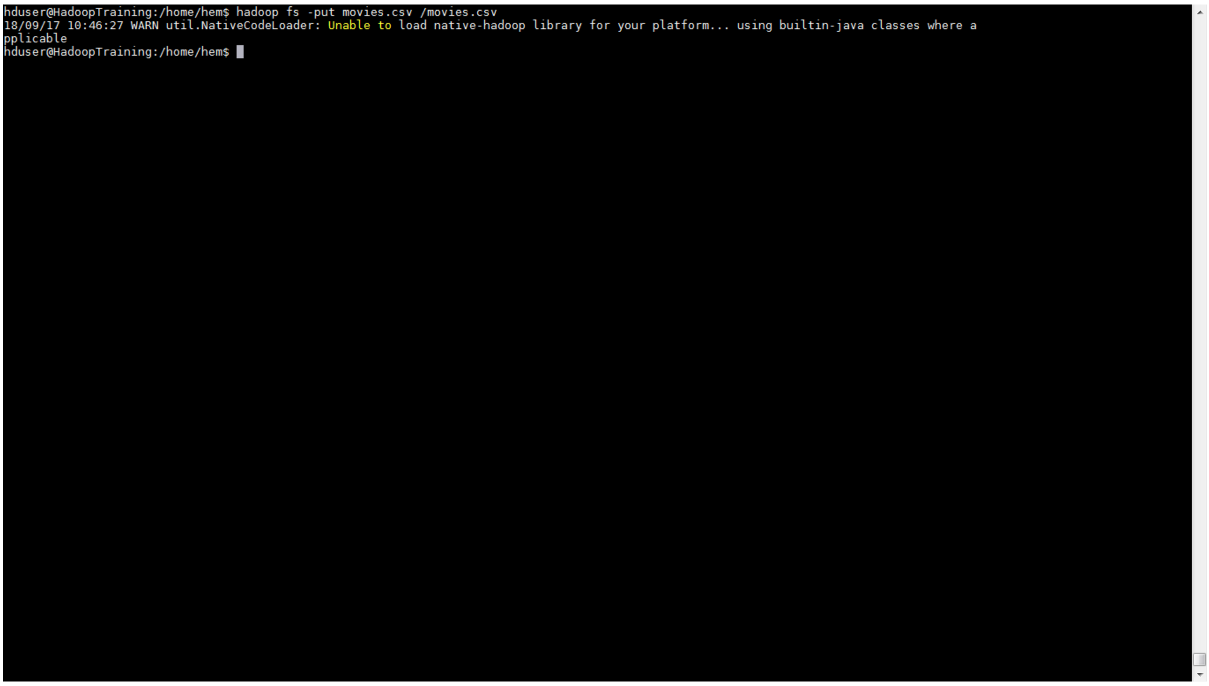
Problem Statement:

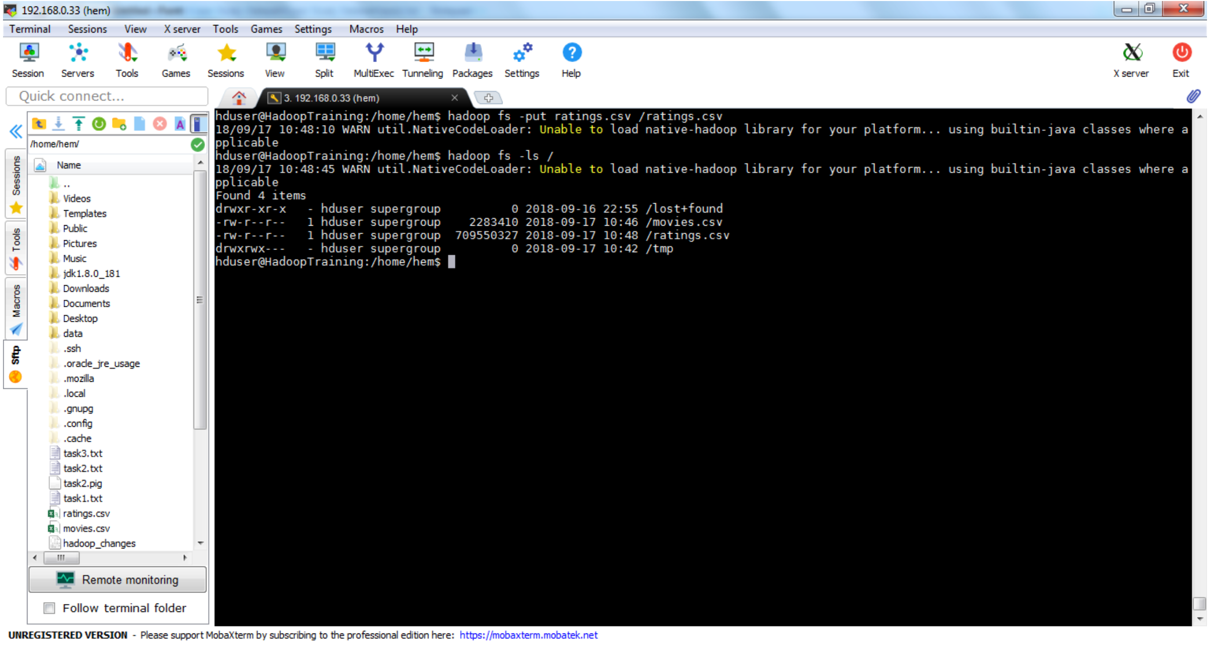
Load Rating.csv and Movies.csv

Loading movies.csv to Hadoop file system



```
hduser@HadoopTraining:/home/hem$ hadoop fs -put movies.csv /movies.csv
18/09/17 10:46:27 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where a
pplicable
hduser@HadoopTraining:/home/hem$
```

Loading ratings.csv to Hadoop file system



```
hduser@HadoopTraining:/home/hem$ hadoop fs -put ratings.csv /ratings.csv
18/09/17 10:48:10 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where a
pplicable
hduser@HadoopTraining:/home/hem$ hadoop fs -ls /
18/09/17 10:48:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where a
pplicable
Found 4 items
drwxr-xr-x   - hduser supergroup          0 2018-09-16 22:55 /lost+found
-rw-r--r--   1 hduser supergroup    2283410 2018-09-17 10:46 /movies.csv
-rw-r--r--   1 hduser supergroup  709550327 2018-09-17 10:48 /ratings.csv
drwxrwx---   - hduser supergroup          0 2018-09-17 10:42 /tmp
hduser@HadoopTraining:/home/hem$
```

What are the movie titles that the user has rated?

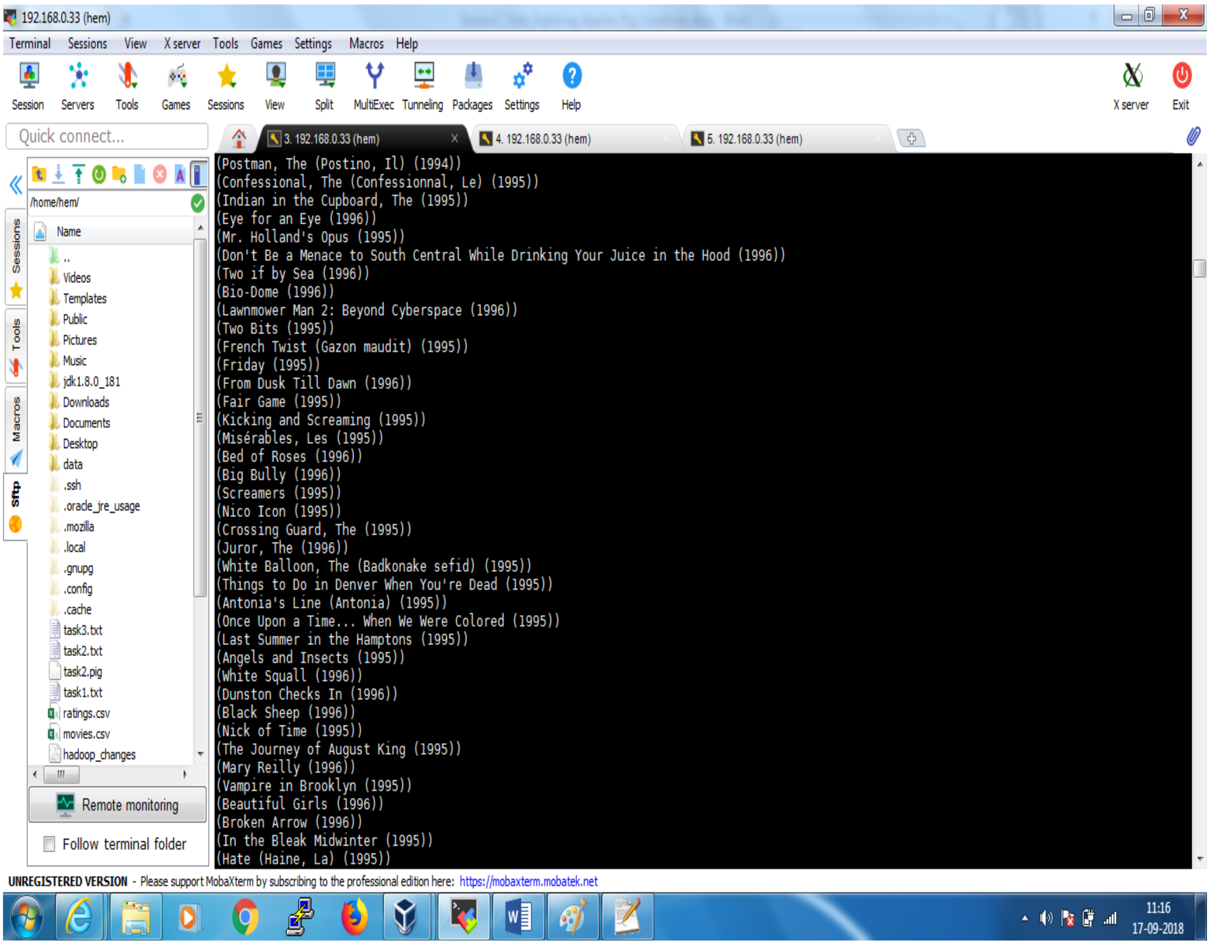
Pig Script-:

```
REGISTER '/home/acadgild/piggybank.jar';
A = LOAD '/movies.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'NO_MULTILINE', 'UNIX',
'SKIP_INPUT_HEADER');
B = foreach A generate (int)$0 as movieid, (chararray)$1 as title;
C = filter B by title is not null;
D = order C by movieid asc;
E = LOAD '/ratings.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'NO_MULTILINE', 'UNIX',
'SKIP_INPUT_HEADER');
F = foreach E generate (int)$1 as movieid,(float)$2 as rating;
G = filter F by (rating > 0 );
H = order G by movieid asc;
I = COGROUP D by movieid  , H by movieid;
J = order I by $0 asc;
K = distinct J;
L = join C by $0 , K by $0 ;
M = foreach L generate $1;
dump M;
```

Script Ilustration-:

In Line 1: We are registering the *piggybank* jar in order to use the CSVExcelStorage class.
In relation **A**, we are loading the dataset using CSVExcelStorage because of its effective technique to handle double quotes and headers.
In relation **B**, we are generating the columns that are required for processing and explicitly typecasting each of them.
In relation **C**, we are filtering the null values from the “title” column.
In relation **D**, we are ordering result by “movieid” column
In relation **E**, we are loading the dataset using CSVExcelStorage because of its effective technique to handle double quotes and headers
In relation **F** we are generating the columns that are required for processing and explicitly typecasting each of them.
Relation **G** and **H** is used to filter rating is greater than 0 and order by movie id in asc order and co grouping D by movieid and H by movieid.
Relation **J** is ordering group by movie id in asc.
Relation **K** is removing duplicates.
In final steps **L**, **M** we are joining both group and **C** by movie id and generate the required column

Output-:



```
(Postman, The (Postino, Il) (1994))
(Confessional, The (Confessionnal, Le) (1995))
(Indian in the Cupboard, The (1995))
(Eye for an Eye (1996))
(Mr. Holland's Opus (1995))
(Don't Be a Menace to South Central While Drinking Your Juice in the Hood (1996))
(Two if by Sea (1996))
(Bio-Dome (1996))
(Lawnmower Man 2: Beyond Cyberspace (1996))
(Two Bits (1995))
(French Twist (Gazon maudit) (1995))
(Friday (1995))
(From Dusk Till Dawn (1996))
(Fair Game (1995))
(Kicking and Screaming (1995))
(Misérables, Les (1995))
(Bed of Roses (1996))
(Big Bully (1996))
(Screamers (1995))
(Nico Icon (1995))
(Crossing Guard, The (1995))
(Juror, The (1996))
(White Balloon, The (Badkonake sefid) (1995))
(Things to Do in Denver When You're Dead (1995))
(Antonia's Line (Antonia) (1995))
(Once Upon a Time... When We Were Colored (1995))
(Last Summer in the Hamptons (1995))
(Angels and Insects (1995))
(White Squall (1996))
(Dunston Checks In (1996))
(Black Sheep (1996))
(Nick of Time (1995))
(The Journey of August King (1995))
(Mary Reilly (1996))
(Vampire in Brooklyn (1995))
(Beautiful Girls (1996))
(Broken Arrow (1996))
(In the Bleak Midwinter (1995))
(Hate (Haine, La) (1995))
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

How many times a movie has been rated by the user?

Pig Script-:

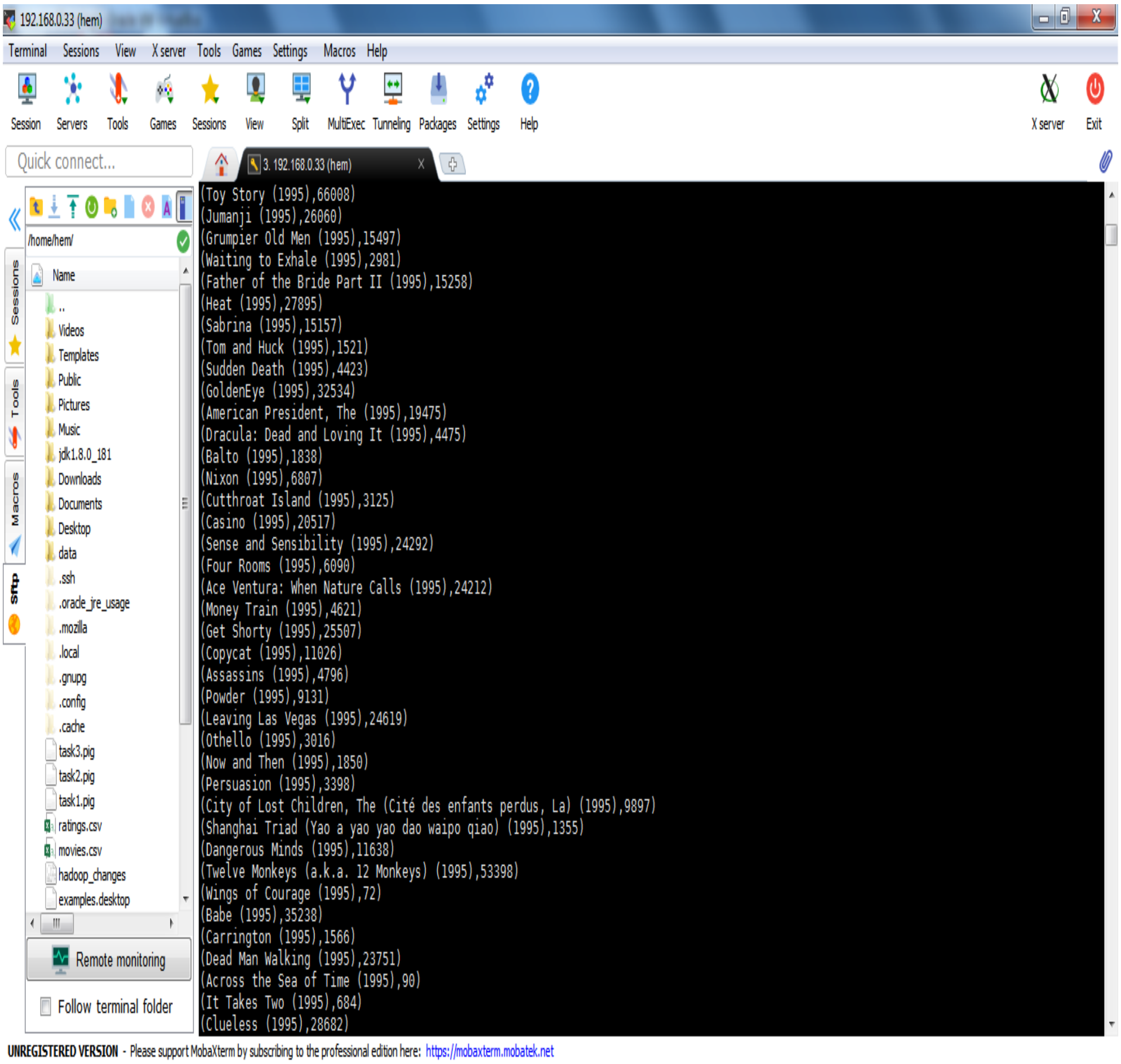
```
REGISTER '/usr/local/pig/lib/piggybank.jar';

A = LOAD '/movies.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'NO_MULTILINE', 'UNIX',
'SKIP_INPUT_HEADER');

B = foreach A generate (int)$0 as movieid, (chararray)$1 as title;

C = filter B by title is not null;

D = order C by movieid asc;

E = LOAD '/ratings.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'NO_MULTILINE', 'UNIX',
'SKIP_INPUT_HEADER');

F = foreach E generate (int)$1 as movieid,(float)$2 as rating;

G = filter F by (rating > 0 );

H = order G by movieid asc;

I = COGROUP D by movieid  , H by movieid;

J = order I by $0 asc;

K = foreach J generate group, COUNT(H.$1) as cnt;

L = join C by $0, K by $0;

M = foreach L generate $1 , $3;

dump M;
```

Script Ilustration-:

In Line 1: We are registering the *piggybank* jar in order to use the CSVExcelStorage class.
In relation **A**, we are loading the dataset using CSVExcelStorage because of its effective technique to handle double quotes and headers.
In relation **B**, we are generating the columns that are required for processing and explicitly typecasting each of them.
In relation **C**, we are filtering the null values from the “title” column.
In relation **D**, we are ordering result by “movieid” column
In relation **E**, we are loading the dataset using CSVExcelStorage because of its effective technique to handle double quotes and headers
In relation **F** we are generating the columns that are required for processing and explicitly typecasting each of them.
Relation **G** and **H** is used to filter rating is greater than 0 and order by movie id in asc

order and co grouping D by movieid and H by movieid.
Relation **J** is ordering group by movie id in asc.
Relation **K** is used to generate group and count.
In final steps **L**, **M** we are joining both group and **C** by movie id and generate the required columns

Output-:



```
(Toy Story (1995),66008)
(Jumanji (1995),26060)
(Grumpier Old Men (1995),15497)
(Waiting to Exhale (1995),2981)
(Father of the Bride Part II (1995),15258)
(Heat (1995),27895)
(Sabrina (1995),15157)
(Tom and Huck (1995),1521)
(Sudden Death (1995),4423)
(GoldenEye (1995),32534)
(American President, The (1995),19475)
(Dracula: Dead and Loving It (1995),4475)
(Balto (1995),1838)
(Nixon (1995),6807)
(Cutthroat Island (1995),3125)
(Casino (1995),20517)
(Sense and Sensibility (1995),24292)
(Four Rooms (1995),6090)
(Ace Ventura: When Nature Calls (1995),24212)
(Money Train (1995),4621)
(Get Shorty (1995),25507)
(Copycat (1995),11026)
(Assassins (1995),4796)
(Powder (1995),9131)
(Leaving Las Vegas (1995),24619)
(Othello (1995),3016)
(Now and Then (1995),1850)
(Persuasion (1995),3398)
(City of Lost Children, The (Cité des enfants perdus, La) (1995),9897)
(Shanghai Triad (Yao a yao yao dao waipo qiao) (1995),1355)
(Dangerous Minds (1995),11638)
(Twelve Monkeys (a.k.a. 12 Monkeys) (1995),53398)
(Wings of Courage (1995),72)
(Babe (1995),35238)
(Carrington (1995),1566)
(Dead Man Walking (1995),23751)
(Across the Sea of Time (1995),90)
(It Takes Two (1995),684)
(Clueless (1995),28682)
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

In question 2 above, what is the average rating given for a movie?

Pig Script-:

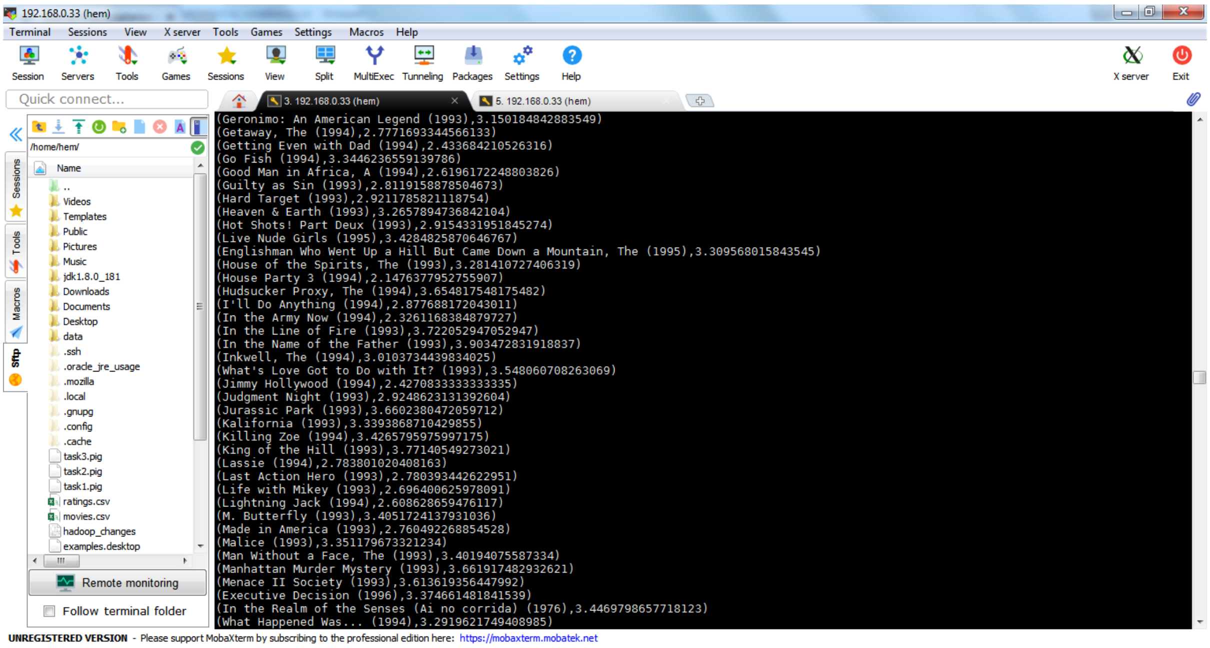
```
REGISTER '/usr/local/pig/lib/piggybank.jar';

A = LOAD '/movies.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',',
'NO_MULTILINE', 'UNIX', 'SKIP_INPUT_HEADER');

B = foreach A generate (int)$0 as movieid, (chararray)$1 as title;

C = filter B by title is not null;

D = order C by movieid asc;

E = LOAD '/ratings.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',',
'NO_MULTILINE', 'UNIX', 'SKIP_INPUT_HEADER');

F = foreach E generate (int)$1 as movieid,(float)$2 as rating;

G = filter F by (rating > 0 );

H = order G by movieid asc;

I = COGROUP D by movieid  , H by movieid;

J = order I by $0 asc;

K = foreach J generate group, AVG(H.$1) as cnt;

L = join C by $0, K by $0;

M = foreach L generate $1 , $3;

dump M;
```

Script Ilustration-:

In Line 1: We are registering the *piggybank* jar in order to use the CSVExcelStorage class.
In relation **A**, we are loading the dataset using CSVExcelStorage because of its effective technique to handle double quotes and headers.
In relation **B**, we are generating the columns that are required for processing and explicitly typecasting each of them.
In relation **C**, we are filtering the null values from the “title” column.
In relation **D**, we are ordering result by “movieid” column
In relation **E**, we are loading the dataset using CSVExcelStorage because of its effective technique to handle double quotes and headers
In relation **F** we are generating the columns that are required for processing and explicitly typecasting each of them.
Relation **G** and **H** is used to filter rating is greater than 0 and order by movie id in asc order and co grouping D by movieid and H by movieid.
Relation **J** is ordering group by movie id in asc.
Relation **K** is used to generate group and avg

In final steps **L**, **M** we are joining both group and **C** by movie id and generate the required columns

Output-:



```
(Geronimo: An American Legend (1993),3.150184842883549)
(Getaway, The (1994),2.7771693344566133)
(Getting Even with Dad (1994),2.433684210526316)
(Go Fish (1994),3.3446236559139786)
(Good Man in Africa, A (1994),2.6196172248803826)
(Guilty as Sin (1993),2.8119158878504673)
(Hard Target (1993),2.9211785821118754)
(Heaven & Earth (1993),3.2657894736842104)
(Hot Shots! Part Deux (1993),2.9154331951845274)
(Live Nude Girls (1995),3.4284825870646767)
(Englishman Who Went Up a Hill But Came Down a Mountain, The (1995),3.309568015843545)
(House of the Spirits, The (1993),3.281410727406319)
(House Party 3 (1994),2.1476377952755907)
(Hudsucker Proxy, The (1994),3.654817548175482)
(I'll Do Anything (1994),2.877688172043011)
(In the Army Now (1994),2.3261168384879727)
(In the Line of Fire (1993),3.722052947052947)
(In the Name of the Father (1993),3.903472831918837)
(Inkwell, The (1994),3.0103734439834025)
(What's Love Got to Do with It? (1993),3.548060708263069)
(Jimmy Hollywood (1994),2.4270833333333335)
(Judgment Night (1993),2.9248623131392604)
(Jurassic Park (1993),3.6602380472059712)
(Kalifornia (1993),3.3393868710429855)
(Killing Zoe (1994),3.4265795975997175)
(King of the Hill (1993),3.77140549273021)
(Lassie (1994),2.783801020408163)
(Last Action Hero (1993),2.780393442622951)
(Life with Mikey (1993),2.696400625978091)
(Lightning Jack (1994),2.608628659476117)
(M. Butterfly (1993),3.4051724137931036)
(Made in America (1993),2.760492268854528)
(Malice (1993),3.351179673321234)
(Man Without a Face, The (1993),3.40194075587334)
(Manhattan Murder Mystery (1993),3.661917482932621)
(Menace II Society (1993),3.613619356447992)
(Executive Decision (1996),3.374661481841539)
(In the Realm of the Senses (Ai no corrida) (1976),3.4469798657718123)
(What Happened Was... (1994),3.2919621749408985)
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net