Complete this homework using you understanding of Hadoop, HDFS, Hive and Apache Spark to perform data management, storage, retrieval, and analysis. Do not answer questions using any other software or method.

## Dataset

Download movielens.rar file. This file contains three data files.

1. u.data: A tab separated list of 100000 ratings by 943 users on 1682 items (movies). Users and items are numbered consecutively from 1. Column names are as following.

user id item id rating timestamp.

1. u.user: A list of 943 user. Values in each row are separated by pipe sign |. Each row contains demographic information about the users under the following column names. user id | age | gender | occupation | zip code
2. u.item: Information about the items (movies); this is a pipe sign separated list of:

movie id | movie title | release date | video release date | IMDb URL | unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |

The last 19 fields are the genres, a 1 indicates the movie is of that genre, a 0 indicates it is not; movies can be in several genres at once.

## Questions

1. Store the data in a Hive database ml as table userratings (u.data), users (u.user)
2. Write HiveQL queries to confirm the number of records in both tables.
3. Extract the list of top 10 items (movies) that received the most ratings (not necessarily highest rating) from female educators.
4. Find the highest rated Fantasy movie.
5. Load the , and u.user files into Apache Spark as DataFrames named df\_udata and df\_uuser. Apply following queries.
   1. How many unique occupations are in the data and what is the frequency of each occupation?
   2. Find the number of recommendations corresponding to each occupation.

## Instructions

For each question, provide the steps, code in both text and image form. Take clear and readable screenshots of the shell commands along with the outputs. Attach or add results after each question.

This screenshot shows the creation of a folder in HDFS,

-mkdr create a folder with the name of “movielie”.

1.hdfs dfs -mkdir -p /user/root/movielie

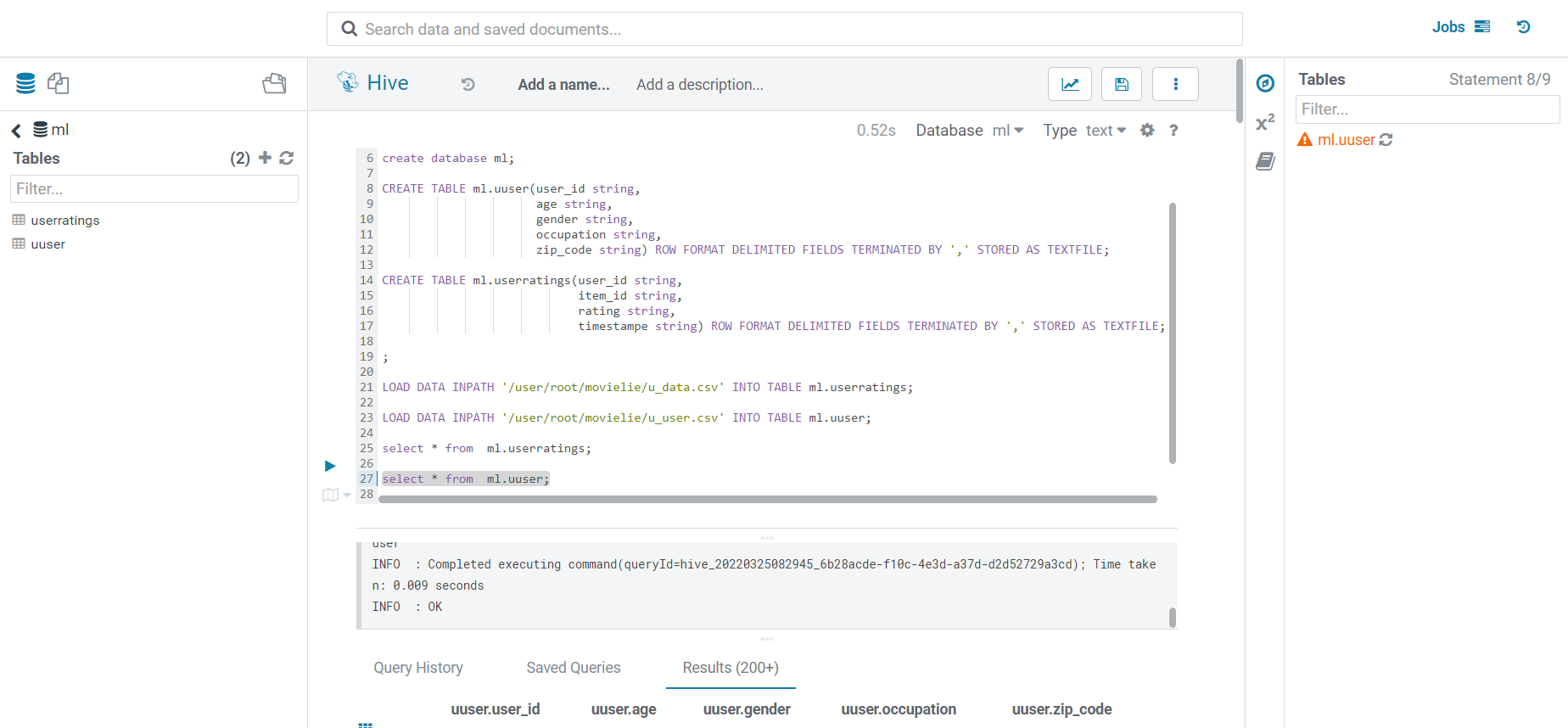
2 hdfs dfs -ls /user/root/

Show that three datasets has been moved to HDFS from local with directory “/user/root/movielie”

3. hdfs dfs -ls /user/root/movielie/

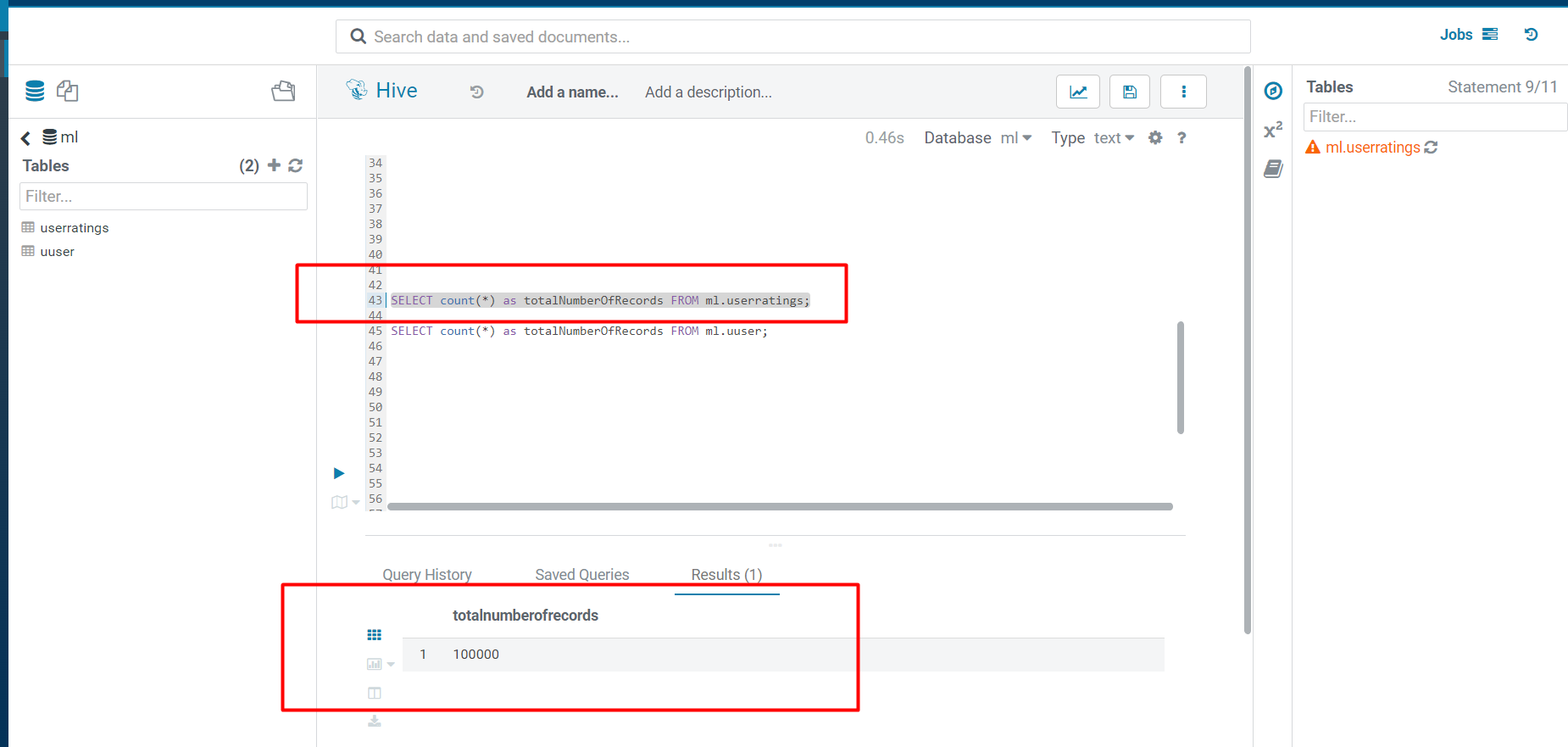
1. Store the data in a Hive database ml as table userratings (u.data), users (u.user)

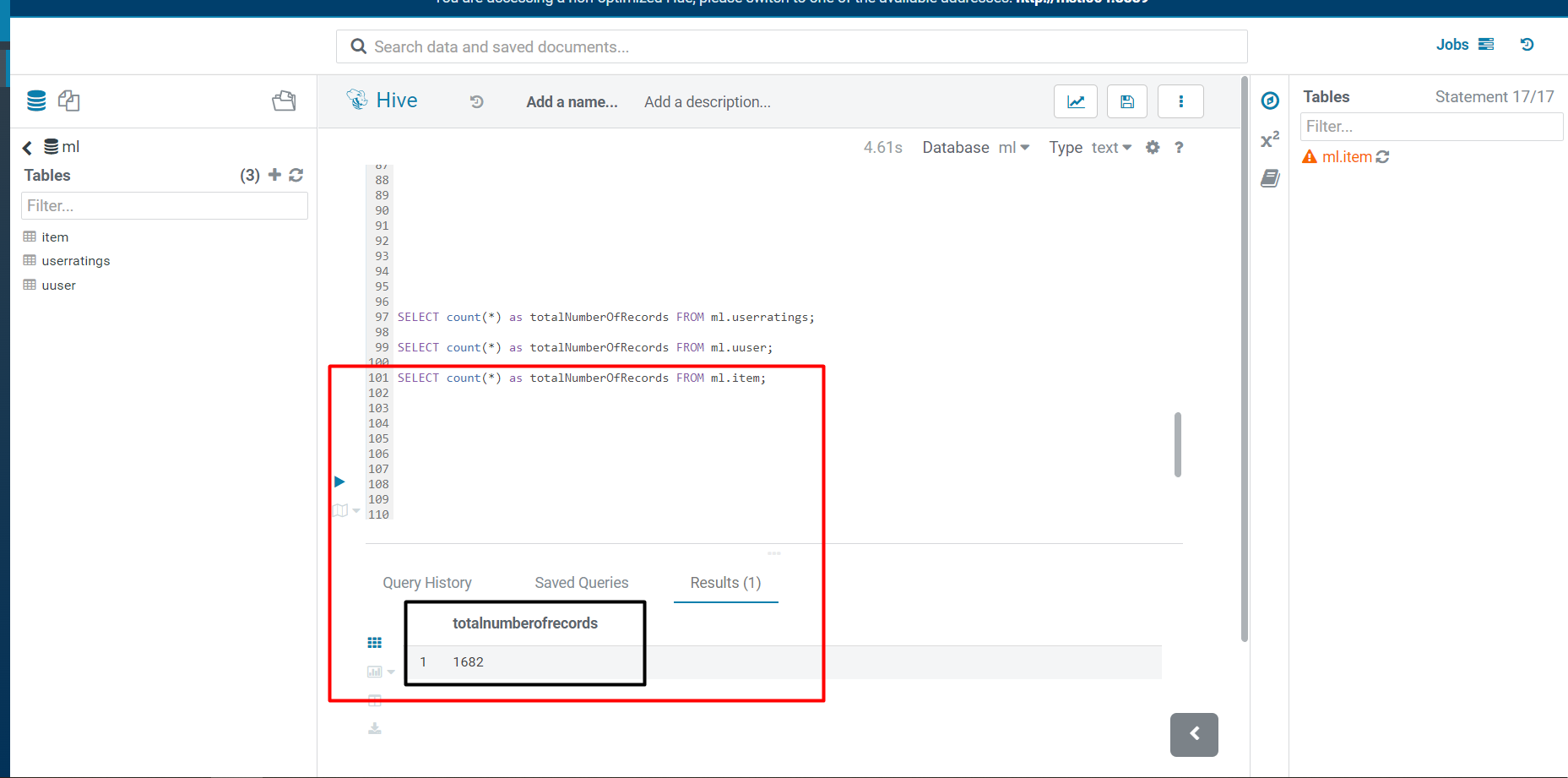
Then created Database with the name of ml, where created three tables in which the three dataset data has been loaded/imported from HDFS to tables



Picture 3.

1. Write HiveQL queries to confirm the number of records in both tables.

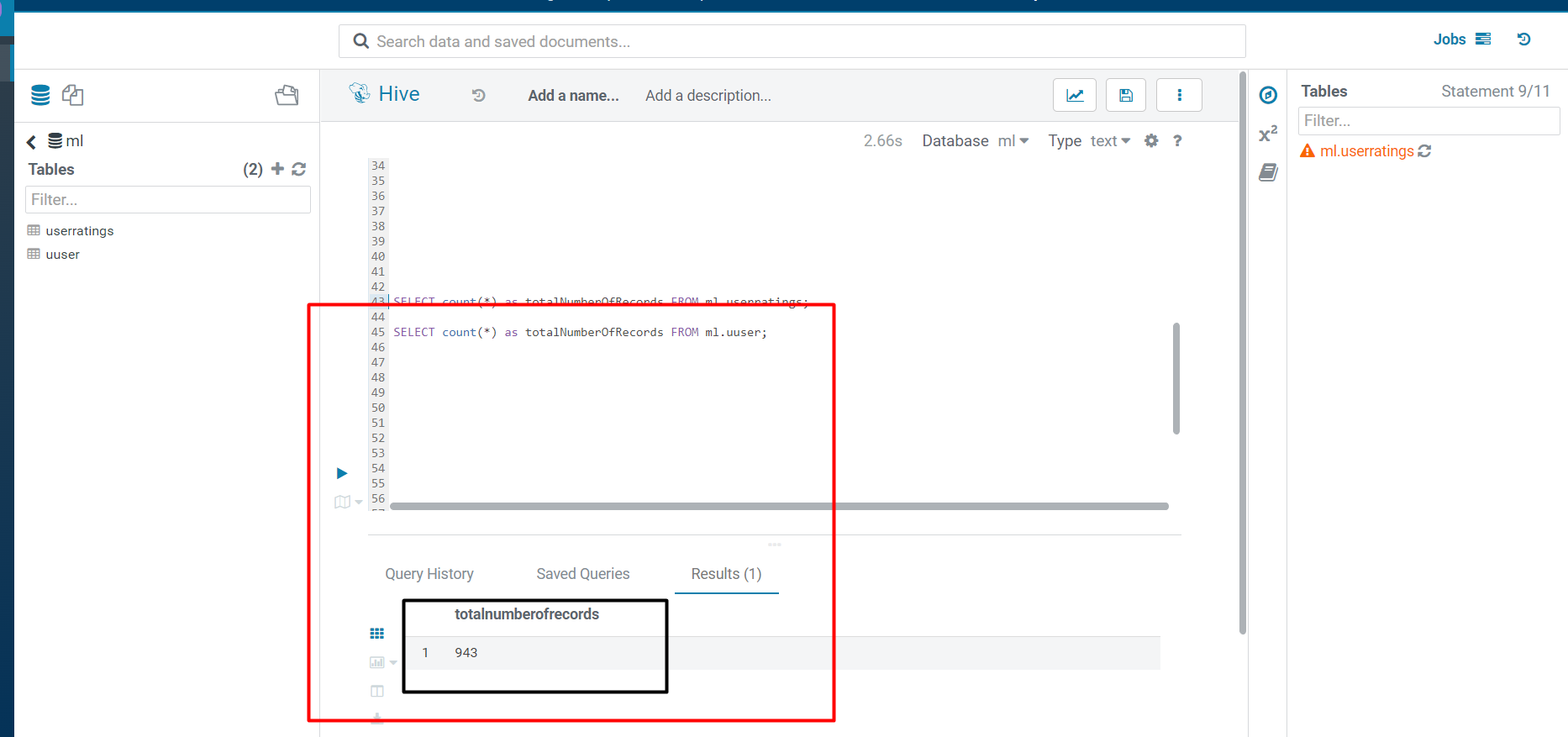




SELECT count(\*) as totalNumberOfRecords FROM ml.userratings;

SELECT count(\*) as totalNumberOfRecords FROM ml.uuser;

SELECT count(\*) as totalNumberOfRecords FROM ml.item;



create database ml;

CREATE TABLE ml.uuser(user\_id string,

age string,

gender string,

occupation string,

zip\_code string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;

CREATE TABLE ml.userratings(user\_id string,

item\_id string,

rating string,

timestampe string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;

CREATE TABLE ml.item(movieid string,

movietitle string,

releasedate string,

videoreleasedate string,

imdburl string,

UNKNOWN string,

Action string,

Adventure string,

Animation string,

Childrens string,

Comedy string,

Crime string,

Documentary string,

Drama string,

Fantasy string,

FilmNoir string,

Horror string,

Musical string,

Mystery string,

Romance string,

SciFi string,

Thriller string,

War string,

Western string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;

LOAD DATA INPATH '/user/root/movielie/u\_item.csv' INTO TABLE ml.item;

LOAD DATA INPATH '/user/root/movielie/u\_data.csv' INTO TABLE ml.userratings;

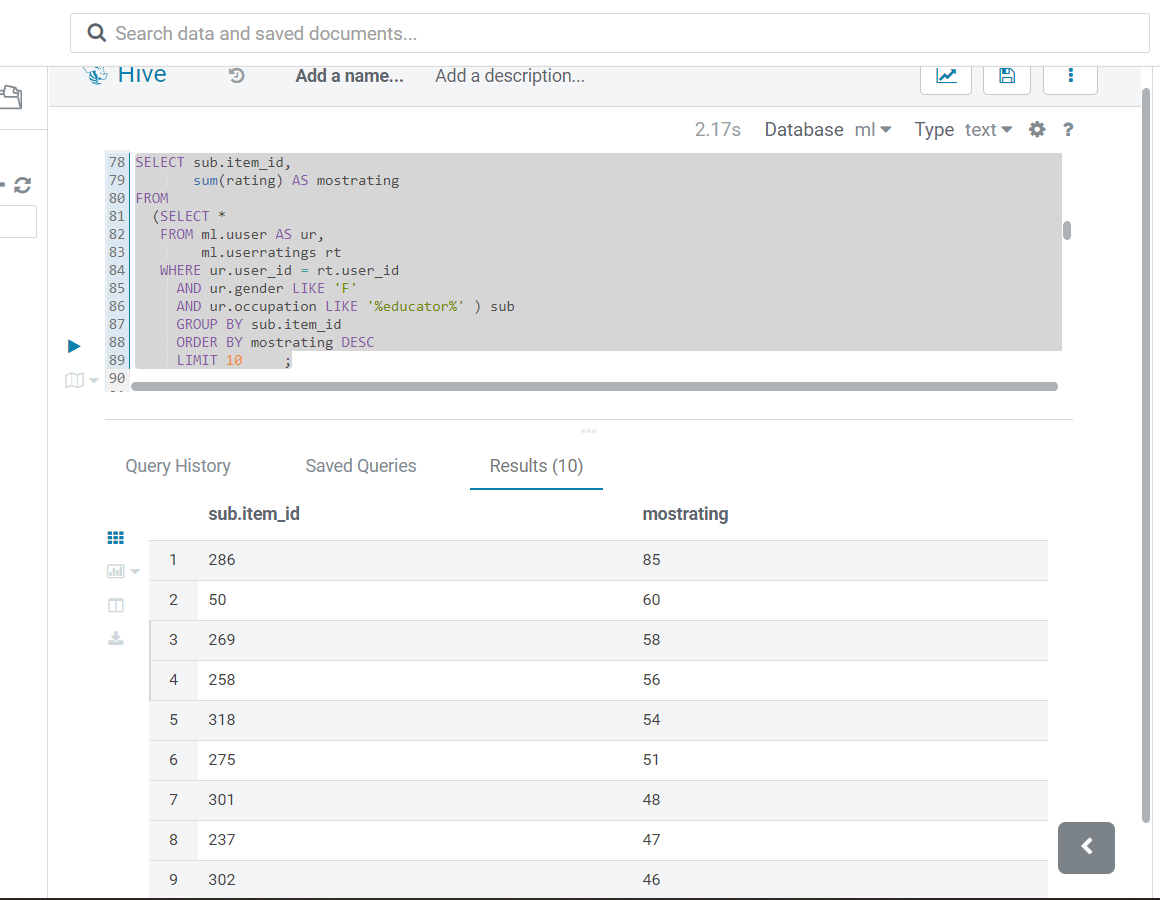
LOAD DATA INPATH '/user/root/movielie/u\_user.csv' INTO TABLE ml.uuser;

select \* from ml.userratings;

select \* from ml.uuser;

SELECT \* from ml.item;

1. Extract the list of top 10 items (movies) that received the most ratings (not necessarily highest rating) from female educators.



SELECT sub.item\_id,

sum(rating) AS mostrating

FROM

(SELECT \* FROM ml.uuser AS ur,

ml.userratings rt

WHERE ur.user\_id = rt.user\_id

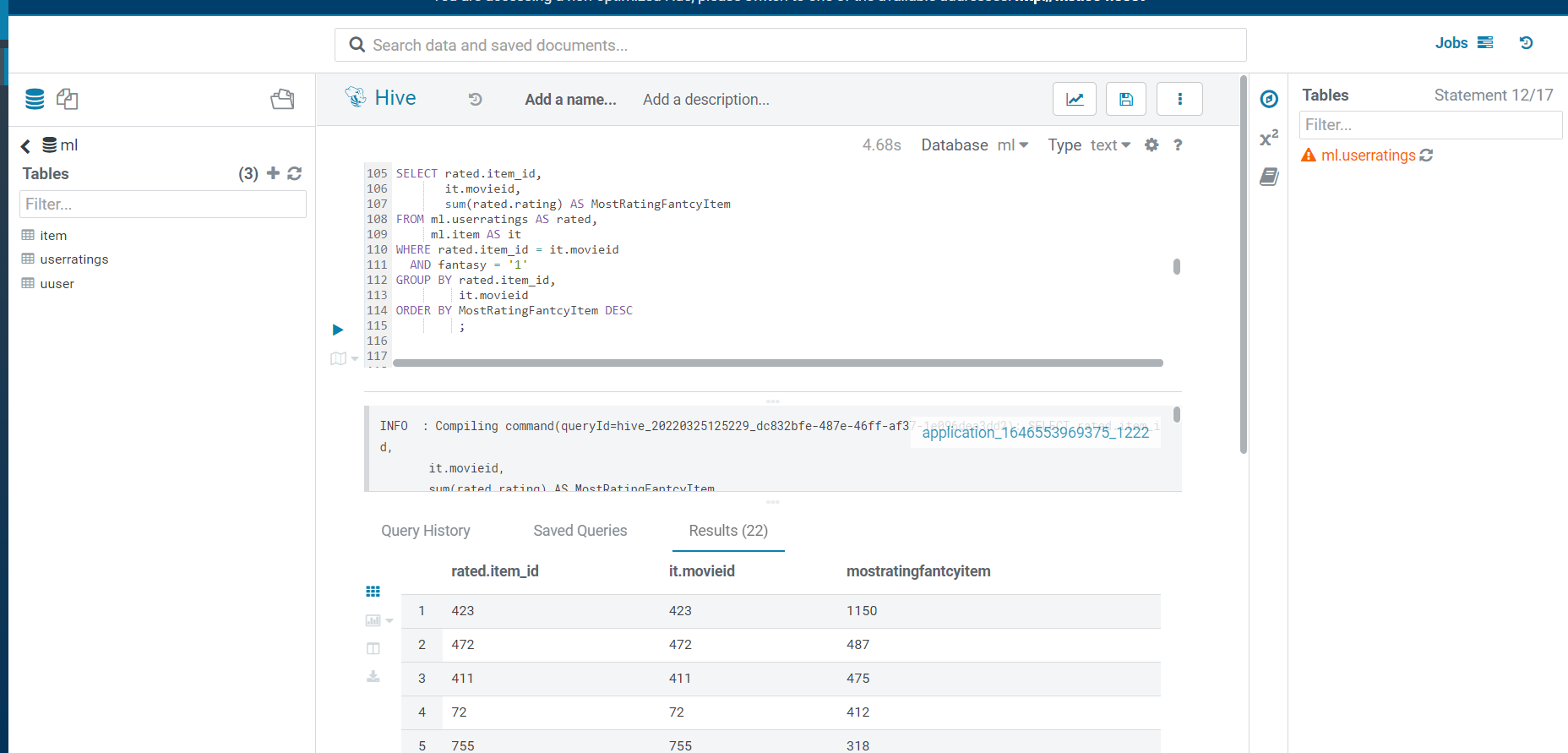
AND ur.gender LIKE 'F'

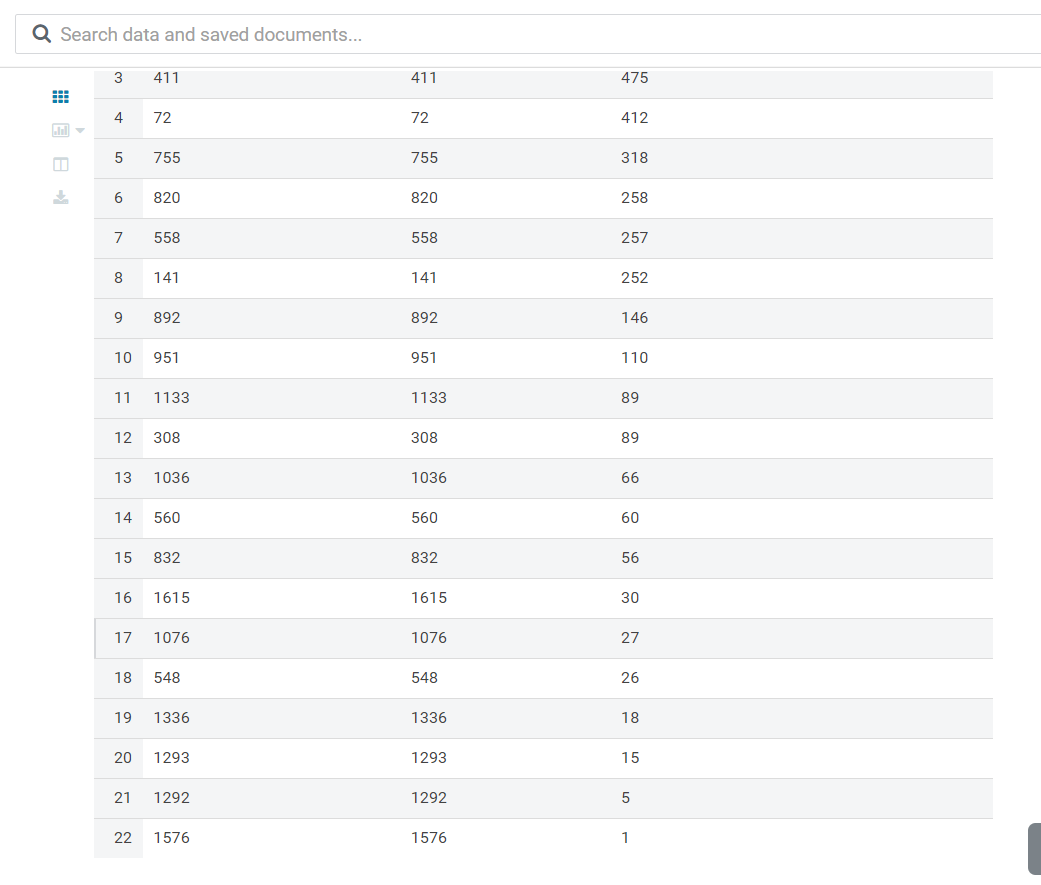
AND ur.occupation LIKE '%educator%' ) sub

GROUP BY sub.item\_id

ORDER BY mostrating DESC LIMIT 10 ;

1. **Find the highest rated Fantasy movie.**





SELECT rated.item\_id,

it.movieid,

sum(rated.rating) AS MostRatingFantcyItem

FROM ml.userratings AS rated,

ml.item AS it

WHERE rated.item\_id = it.movieid

AND fantasy = '1'

GROUP BY rated.item\_id,

it.movieid

ORDER BY MostRatingFantcyItem DESC

1. Load the , and u.user files into Apache Spark as DataFrames named df\_udata and df\_uuser. Apply following queries.

