**CIND 719 – Assignment 1**

Brendan Dagys

6)

hadoop fs -put /root/lab/u.data /user/lab

hadoop fs -put /root/lab/u.item /user/lab

7)

create table ratings (user\_id int, movie\_id int, rating int, unixtime bigint) row format delimited fields terminated by '\t';

load data inpath '/user/lab/u.data' into table ratings;

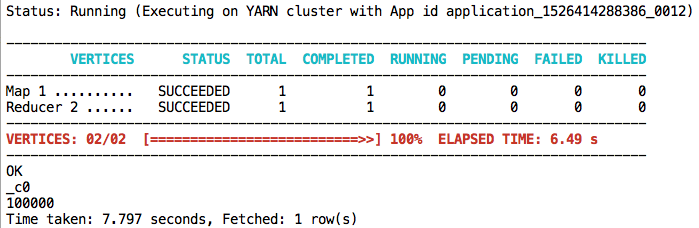
create table movies (movie\_id int, movie\_title string, release\_date string, v\_release\_date string, imdb\_url string, cat\_unknown int, cat\_action int, cat\_adventure int, cat\_animation int, cat\_children int, cat\_comedy int, cat\_crime int, cat\_documentary int, cat\_drama int, cat\_fantasy int, cat\_film\_noir int, cat\_horror int, cat\_musical int, cat\_mystery int, cat\_romance int, cat\_scifi int, cat\_thriller int, cat\_war int, cat\_western int) row format delimited fields terminated by '|';

load data inpath '/user/lab/u.item' into table movies;

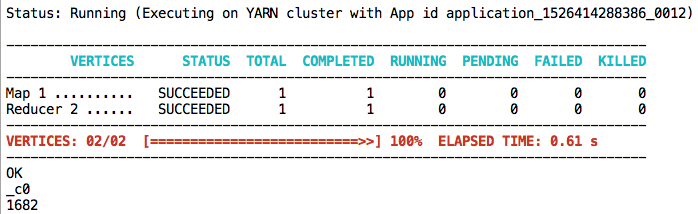
**QUESTIONS**

**1)**

select count(\*) from ratings;



select count(\*) from movies;



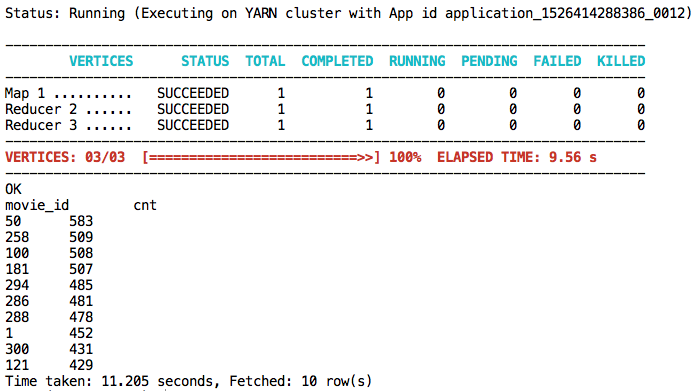
**2)**

select movie\_title from movies where release\_date REGEXP '1990';



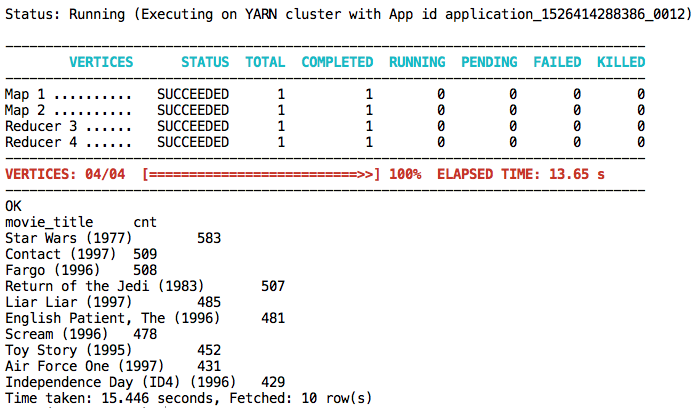
**3)**

select movie\_id, count(rating) cnt from ratings group by movie\_id order by cnt desc limit 10;



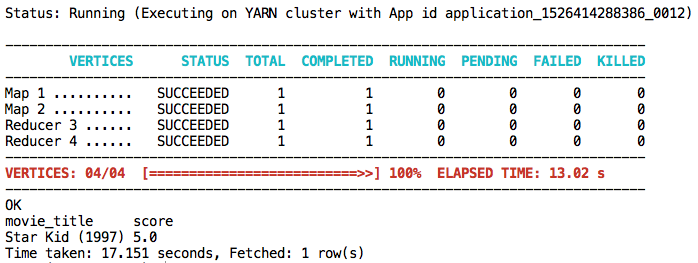
**4)**

select movie\_title, count(rating) cnt from (movies m join ratings r on m.movie\_id = r.movie\_id) group by movie\_title order by cnt desc limit 10;



**5)**

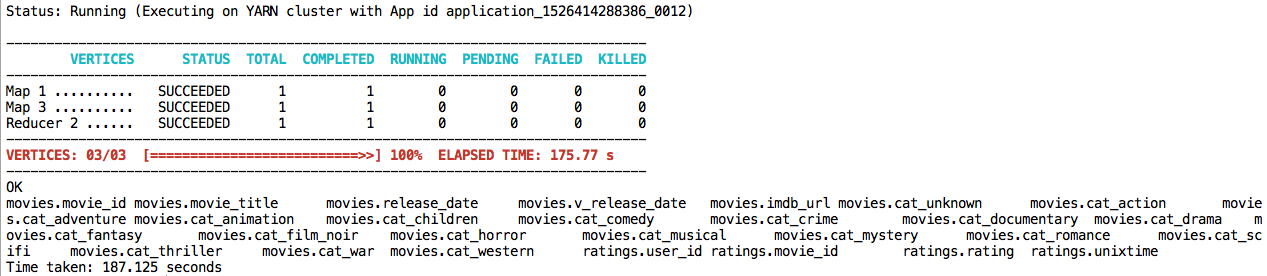
select movie\_title, avg(rating) score from (movies m join ratings r on m.movie\_id = r.movie\_id) where cat\_scifi = 1 group by movie\_title order by score desc limit 1;



The highest-rated movie will be the movie with the highest average score (rating column) amongst all reviewers that reviewed that movie.

**BONUS)**

select \* from (movies full outer join ratings) where rating IS NULL;



There are no movies without a rating. They all have a rating.