```
CREATE TABLE business (businessid STRING,  fullAddress STRING,
categories STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '^';

CREATE TABLE review (reviewid STRING, userid STRING, businessid
STRING, stars FLOAT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '^';

CREATE TABLE users (userid STRING, name STRING, url STRING) ROW FORMAT
DELIMITED FIELDS TERMINATED BY '^';

hdfs dfs -cp /yelp/business/business.csv business.csv
hdfs dfs -cp /yelp/review/review.csv review.csv
hdfs dfs -cp /yelp/user/user.csv user.csv


LOAD DATA  INPATH 'business.csv' OVERWRITE INTO TABLE business;
LOAD DATA  INPATH 'review.csv' OVERWRITE INTO TABLE review;
LOAD DATA  INPATH 'user.csv' OVERWRITE INTO TABLE users;



1. List the 'user id' and 'stars' of users that reviewed businesses
located in La Jolla, CA.

   select count(*) from business;
   describe business;
   select * from business where fulladdress like '%La Jolla%';
   select r.userid, r.stars from review r inner join business b on
r.businessid = b.businessid where b.fulladdress like '%La Jolla, CA%';


2. List businessid and average stars of businesses that are located in
La Jolla, CA.

   select r.businessid, AVG (r.stars) from review r inner join
business b on r.businessid = b.businessid group by r.businessid having
b.fulladdress like '%La Jolla, CA%as’;

   ** (above first step, below final code)
   select r.businessid, AVG(r.stars) from review r inner join (SELECT
businessid from business where fulladdress like '%La Jolla, CA%’) as b
ON r.businessid = b.businessid group by r.businessid;


3. List the business_id , full address and categories of the Top 10
highest rated businesses using the average ratings.

   select b.businessid, AVG(r.stars) from business b inner join review
r on b.businessid = r.businessid group by b.businessid;
```

**

select b.businessid, AVG(r.stars) as avgStars from business b inner join review r on b.businessid = r.businessid group by b.businessid ORDER BY avgStars DESC LIMIT 10;

4. List the user_id , and name of the top 5th user who has written the most reviews.

select userid, count(reviewid) as cntReview from review group by userid order by cntReview limit 8;

**

select u.name, u.userid, i.cntReview as countOfReview from user u inner join (select userid, count(reviewid) as cntReview from review group by userid order by cntReview limit 8) as i on u.userid = i.userid order by i.cntReview limit 1;

5. Find out the userid of users who have not written any reviews.

Select userid, count(reviewid) as cntReview from review group by userid having cntReview = 0;

6. List the business_id, and count of each business's ratings for the businesses that are located in the state of TX.

Select r.businessid, count(r.reviewid) as cntReview from review r inner join business b on r.businessid = b.businessid group by r.businessid having b.fulladdress like '%TX%';

For the next questions, download the movielens dataset from: http://files.grouplens.org/datasets/movielens/ml-100k.zip
Note that the files are tab separated. The list of fields is available in the README file: http://files.grouplens.org/datasets/movielens/ml-100k-README.txt
You should be able to answer the queries below using the following tables:
u.data, u.user, and u.item.
Note that item id and movie id are used interchangeably.

7. Find the age, gender, and occupation of the user that has written the most reviews. If there is a tie, you can randomly select any.

select m.age, m.userid, i.cntReview as countOfReview from movieusers m inner join(select userid, count(itemid) as cntReview from rating group by userid order by cntReview desc) as i on m.userid = i.userid order by countOfReview limit 1;

8. Create a list of occupations and the number of reviews written by them sorted by the count of reviews in descending order.
Example:

Student 200000
Programmer 18000
....

Select m.occupation, SUM(i.cntReview) as countOfReview from movieusers m inner join
(select userid, count(itemid) as cntReview from rating group by userid order by cntReview desc) as i on m.userid = i.userid group by m.occupation order by countOfReview;

9. Using the u.item table, find the total number of movies of each category i.e. something like:
TotalOfAdventure TotalOfComedy ....

Select SUM(unknown), SUM(Action), SUM(Adventure), SUM(Animation), SUM(Childrens), SUM(Comedy), SUM(Crime), SUM(Documentary), SUM(Drama), SUM(Fantasy), SUM(FilmNoir), SUM(Horror), SUM(Musical), SUM(Mystery), SUM(Romance), SUM(SciFi), SUM(Thriller), SUM(War), SUM(Western) from item;

10. From the user table, find the number and average age of those with occupation of "scientist"

Select count(userid), AVG(age) from movieusers group by occupation having occupation = “scientist”;