Run on command line:
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
****
Note: If the above commands don't work, try the following:
curl -o business.csv
http://www.utdallas.edu/~axn112530/cs6350/yelp/business/business.csv
curl -o review.csv http://www.utdallas.edu/~axn112530/cs6350/yelp/review/review.csv
curl -o user.csv http://www.utdallas.edu/~axn112530/cs6350/yelp/user/user.csv
hdfs dfs -put business.csv
hdfs dfs -put review.csv
hdfs dfs -put user.csv
****
Start Hive CLI client as:
beeline
and then
!connect jdbc:hive2:// scott tiger

Run on Hive CLI:

```
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 '^';

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.

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

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

4. List the userid , and name of the top 8th user who has written the most reviews.

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

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

Run on command line:
wget http://files.grouplens.org/datasets/movielens/ml-100k.zip
unzip ml-100k.zip

CLI commands:

```
CREATE TABLE rating (userid STRING,  itemid STRING, rating FLOAT, timest STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';

LOAD DATA LOCAL INPATH 'ml-100k/u.data' OVERWRITE INTO TABLE rating;


CREATE TABLE item (movieid STRING,  movie STRING, releaseDate STRING,
videoReleaseDate STRING,
              IMDbURL STRING, unknown INT, Action INT, Adventure INT,  Animation INT,
              Childrens INT, Comedy INT, Crime INT, Documentary INT, Drama INT,
Fantasy INT,
              FilmNoir INT, Horror INT, Musical INT, Mystery INT, Romance INT, SciFi
INT,
              Thriller INT, War INT, Western INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '|';

LOAD DATA LOCAL INPATH 'ml-100k/u.item' OVERWRITE INTO TABLE item;


CREATE TABLE movieusers (userid STRING,  age STRING, gener STRING, occupation STRING,
zipcode STRING )
ROW FORMAT DELIMITED FIELDS TERMINATED BY '|';

LOAD DATA LOCAL INPATH 'ml-100k/u.user' OVERWRITE INTO TABLE movieusers;
```

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.

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
....

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

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