Progress Report

1. change of tables and E/R diagram

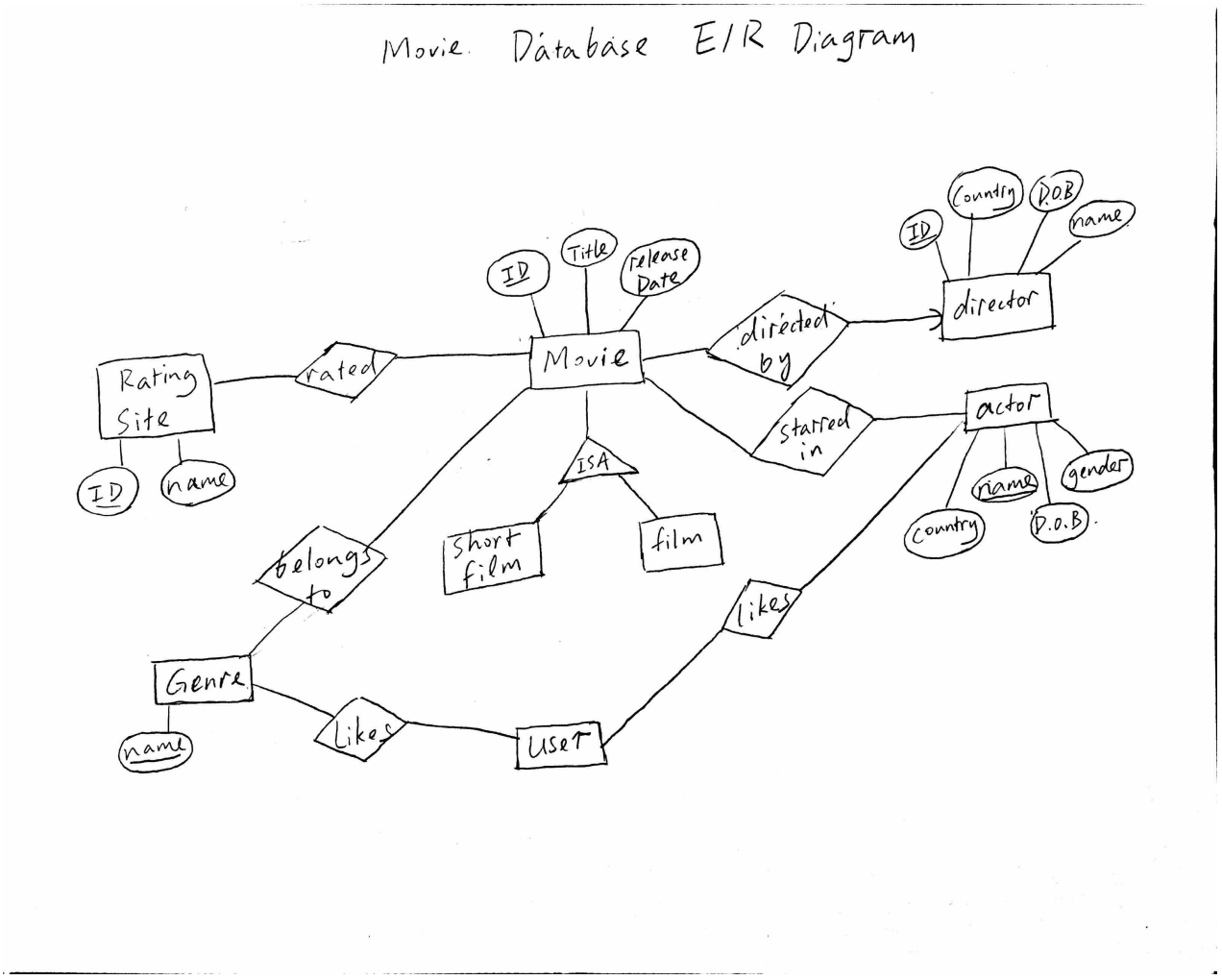
   Since we got more columns from the different online sites so the attributes on the table has changed a bit. The following table is what's currently in our database.

   Director(id, first_name, last_name)

   Movie(id, title, genre, drector, date_released)

   Rating(id, source_name, movie_id, max_rating, rating_score)

   GrossingInfo(movie_id, data_timestamp, amount_grossed)

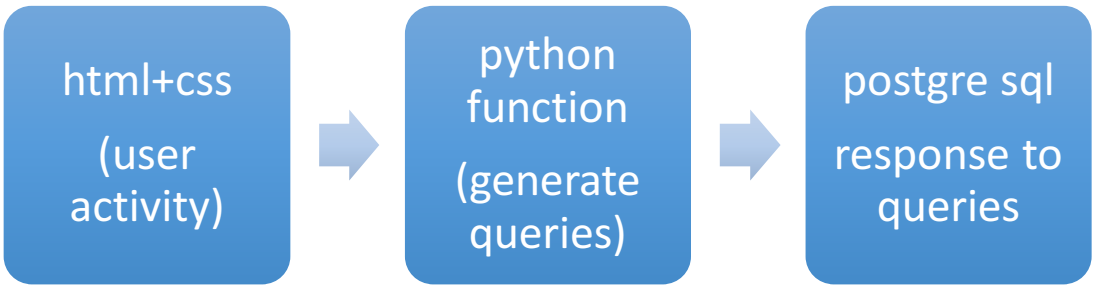


2. new assumptions

- we are planning to make our own scaling function for the rating which will take in the number of mentions from reddit, and twitter as a major factor and also take into the rating from other sites into consideration. For now, we do not have a scaling function, so we are simply scaling every rating number to be an integer out of 100 and have an even weight for all the rating site.
- We are now assuming all the sites have the rating for all the movies in our database. This is a fair assumption since we are only doing the top 250 movie from imdb, however, as we expand the size of our database, we may have to create a more complicated scaling function since some rating data will be missing.
- We also assume all the info from different online movie site matches. For example, the name and director and title can fix a unique movie. In reality, as we have larger set of movies, info can easily mismatch. We might have to do some data cleaning or data matching.

3. Web platform

   We are using heroku to host hour website. The url to our site is the following:
   https://movie-suggestion-316.herokuapp.com/
   Our git repo is the following:
   git@github.com:yuminz/316Spring2017.git
   We are using postgre sql as our database, and it is hosted on heroku. On back end, we are using python to connect to the database. On the front end we are using html, css and JavaScript to construct the user interface. Based on the user and user's activity, we are calling python functions to generate the queries, get data from the database and pass data back to display on the user interface. Thus the basic workflow is like the following:



So far, we have a very basic but all linked up website. Our front end, back end and database already hooked up. We can send basic queries from web interface to get the data from the database. For the next stage of the project, all we need to do is making more functions on the front end and expand our database to include more data from more sources.

4. Changes made to database

   We have now accomplished scraping the top 250 rated movie data from imdb and getting the reddit comment counts for each of the 250 movies. We are planning to get more movie data, from guardian, rotton tomatoes, yahoo movies and twitter in the next stage.