SI 601 Fall 2015 Homework 4 (100 points)

Due Date: Tuesday Feb 10 5:00pm

Part 1 (65 points)

The provided ‘movie_actors_data.txt’ file contains a JSON string on each line. For example, the first line is:

{"rating": 9.3, "genres": ["Crime", "Drama"], "rated": "R", "filming_locations": "Ashland, Ohio, USA", "language": ["English"], "title": "The Shawshank Redemption", "runtime": ["142 min"], "poster": "http://img3.douban.com/lpic/s1311361.jpg", "imdb_url": "http://www.imdb.com/title/tt0111161/", "writers": ["Stephen King", "Frank Darabont"], "imdb_id": "tt0111161", "directors": ["Frank Darabont"], "rating_count": 894012, "actors": ["Tim Robbins", "Morgan Freeman", "Bob Gunton", "William Sadler", "Clancy Brown", "Gil Bellows", "Mark Rolston", "James Whitmore", "Jeffrey DeMunn", "Larry Brandenburg", "Neil Giuntoli", "Brian Libby", "David Proval", "Joseph Ragno", "Jude Ciccolella"], "plot_simple": "Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.", "year": 1994, "country": ["USA"], "type": "M", "release_date": 19941014, "also_known_as": ["Die Verurteilten"]}

The fields we are interested in are imdb_id , title , rating, genres, actors, and year. You will parse the JSON strings, and load the data into three tables in SQLite, and then write SQL queries to retrieve the data specified.

You will create three tables:

- The “movie_genre” table, which has two columns: imdb_id and genre. A movie typically has multiple genres, and in this case, there should be one row for each genre. If some movie does not have any genre, ignore that movie.
- The “movies” table, which has four columns: imdb_id, title, year, rating
- The “movie_actor” table, which has two columns imdb_id and actor. A movie typically has multiple actors, and in this case, there should be one row for each actor.

1. (10 points) Parse input file to get needed data for the three tables and load them into appropriate Python data structure.

2. (5 points) Create the movie_genre table and load data into it

3. (5 points) Create the movies table and load data into it

4. (5 points) Create the movie_actor table and load data into it

5. (5 points) Write an SQL query to find top 10 genres with most movies and print out the results

6. (10 points) Write an SQL query to find all fantasy movies order by decreasing rating, then by decreasing year if ratings are the same. Print out the results.

7. (10 points) Write an SQL query for finding top 10 most productive comedy actors, i.e. the actors who played roles in largest numbers of comedy movies. Print out the results.

8. (15 points) Write an SQL query for finding top 10 most frequent pairs of actors who co-stared in the same movie. In each pair of actors you print out, the two actors must be ordered

alphabetically. You will need to join the movie_actor table with itself to get this data. It is a bit tricky. If you cannot do it with SQL statement, you can also write some Python code that works on the Python data structure that you used to create the movie_actor table. That'll mean much more lines of code, and if you do it that way, you'll get 5 points instead of 15 points. You will only get 15 points if you solve it with pure SQL.

When you run your Python code, it should print out EXACTLY such output:

```
Top 10 genres:
Drama
Thriller
Crime
Adventure
Mystery
Comedy
Action
Romance
Fantasy
War

Fantasy movies:
Title, Year, Rating
The Lord of the Rings: The Return of the King, 2003, 8.9
The Lord of the Rings: The Fellowship of the Ring, 2001, 8.8
Star Wars, 1977, 8.8
The Lord of the Rings: The Two Towers, 2002, 8.7
It's a Wonderful Life, 1946, 8.7
Toy Story 3, 2010, 8.5
The Green Mile, 1999, 8.5
Star Wars: Episode VI - Return of the Jedi, 2002, 8.4
Monty Python and the Holy Grail, 1975, 8.4
The Hobbit: An Unexpected Journey, 2012, 8.3
Up, 2009, 8.3
El laberinto del fauno, 2006, 8.3
Toy Story, 1995, 8.3
Det sjunde inseglet, 1957, 8.3
How to Train Your Dragon, 2010, 8.2
V for Vendetta, 2005, 8.2
Tonari no Totoro, 1988, 8.2
The Wizard of Oz, 1939, 8.2
Harry Potter and the Deathly Hallows: Part 2, 2011, 8.1
Hauru no ugoku shiro, 2004, 8.1
Pirates of the Caribbean: The Curse of the Black Pearl, 2003, 8.1
Groundhog Day, 1993, 8.1
The Princess Bride, 1987, 8.1
Tenkû no shiro Rapyuta, 1986, 8.1
Stalker, 1979, 8.1
8½, 1963, 8.1
Ratatouille, 2007, 8.0
Big Fish, 2003, 8.0
Monsters, Inc., 2001, 8.0
Beauty and the Beast, 1991, 8.0

Top 10 most productive comedy actors:
Actor, Movies
Charles Chaplin, 5
John Ratzenberger, 5
```

```
Bob Peterson, 3
Hank Mann, 3
John Goodman, 3
Wallace Shawn, 3
Al Ernest Garcia, 2
Billy Crystal, 2
Brad Garrett, 2
Carol Cleveland, 2

Top 10 most frequent pairs of actors who co-stared in the same movie:
Actor A, Actor B, Co-stared Movies
Christian Bale, Michael Caine, 4
Joe Pesci, Robert De Niro, 4
Al Pacino, John Cazale, 3
Alec Guinness, Anthony Daniels, 3
Alec Guinness, Carrie Fisher, 3
Alec Guinness, David Prowse, 3
Alec Guinness, Harrison Ford, 3
Alec Guinness, Kenny Baker, 3
Alec Guinness, Mark Hamill, 3
Alec Guinness, Peter Mayhew, 3
Anthony Daniels, Carrie Fisher, 3
Anthony Daniels, David Prowse, 3
Anthony Daniels, Harrison Ford, 3
Anthony Daniels, Kenny Baker, 3
Anthony Daniels, Mark Hamill, 3
Anthony Daniels, Peter Mayhew, 3
Benito Stefanelli, Clint Eastwood, 3
Bibi Andersson, Gunnar Björnstrand, 3
Billy Boyd, Cate Blanchett, 3
Billy Boyd, Orlando Bloom, 3
```

Part 2 (35 points)

In this part of the homework, you will use Twitter API to visualize the tweets on your timeline. Here are the steps:

1. Create a twitter account if you don't have one already

2. Following more than 10 people if you haven't done already

3. Sign in twitter dev account at https://dev.twitter.com, and then get your own Twitter consumer_key, consumer_secret, access_token, access_secret by following instructions at https://dev.twitter.com/oauth/overview/application-owner-access-tokens

4. Install Python package oauth2 by typing '[sudo] pip install oauth2' in a terminal.

5. Rename si601_w15_hw4_part2.py to si601_w15_hw4_part2_youruniquename.py. Fill in the empty consumer_key, consumer_secret, access_token, access_secret with your own credentials, and make sure it prints out some interesting JSON response when you run it. (5 points)

6. Add more code where specified to
a) (5 points) Load the response JSON string into a Python data structure
b) (20 points) Create a graph in the DOT language using pydot such that if there is a tweet from user A, and the tweet mentions users B, C, D, then the directed edges A->B, A->C, A->D are added to the graph. Note that self-mentions should not count.

The results should be saved in a file that looks similar to twitter_example_output.dot.

7. (5 points) Open your .dot output in Graphviz and save the graph visualization as a PDF file.

What to submit:

A zip file named si601_w15_hw4_youruniquename.zip that contains:

1. Your python program files for part 1 and part 2
2. Your output files for part 1 and part 2 (.txt, .dot and .pdf files).