**SI 618 Fall 2019 Homework 2 (100 points)**

**Part 1 (75 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. Write a python program named si618\_f17\_hw1\_part1\_youruniquename.py. 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. (10 points) Write a SQL query to find top 10 actors starring in the most number of movies and print out the results.

6. (10 points) Write a SQL query to find the number of movies broken down by year in chronological order

7. (10 points) Write a SQL query to find all Mystery movies from the U.S. ordered by increasing rating, then by decreasing year if ratings are the same.

8. (10 points) Write a SQL query to find the top 10 actors that played in the highest rated movies (on average) after year 2000. For each actor, give their name, average rating of the movies they played in, and the number of movies. Sort the result from descending order based on average movie rating. In case of ties, sort the rows by actor name.

9. (10 points) Write a SQL query for finding pairs of actors who co-starred in at least 3 highly rated (rating > 8.5) movies together. The pairs of names must be unique. This means that ‘actor A, actor B’ and ‘actor B, actor A’ are the same pair, so only one of them should appear.

For each pair of actors you print out, the two actors must be ordered alphabetically. The pairs are ordered in decreasing number of movies they co-stared in. In case of ties, the rows are ordered by actors’ names.

You will need to join the movie\_actor table with itself to get this data (in addition to another required join). 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 10 points. You will only get 10 points if you solve it with pure SQL.

When you run your Python code, it should print out EXACTLY such output in your terminal: (See below)

(this entire output will change given that the questions changed—we also will need to give this in a separate file so people can check directly).

Top 10 actors:Actor, Movies

Robert De Niro,10

James Stewart,8

Morgan Freeman,7

Harrison Ford,7

Michael Caine,6

John Ratzenberger,6

Clint Eastwood,6

William Holden,5

Tom Hanks,5

Steve Buscemi,5

Movies broken down by year:

Year, Movies

1921, 1

1922, 1

1925, 1

1926, 1

1927, 1

1930, 1

1931, 2

1934, 1

1936, 1

1939, 3

1940, 3

1941, 2

1942, 1

1943, 1

1944, 2

1946, 3

1948, 3

1949, 1

1950, 4

1951, 2

1952, 2

1953, 2

1954, 5

1955, 1

1956, 1

1957, 6

1958, 2

1959, 5

1960, 2

1961, 2

1962, 4

1963, 2

1964, 2

1965, 1

1966, 3

1967, 2

1968, 3

1969, 2

1971, 1

1972, 2

1973, 3

1974, 2

1975, 5

1976, 3

1977, 2

1978, 1

1979, 5

1980, 3

1981, 2

1982, 3

1983, 1

1984, 3

1985, 2

1986, 3

1987, 3

1988, 5

1989, 1

1990, 1

1991, 3

1992, 2

1993, 4

1994, 5

1995, 8

1996, 2

1997, 3

1998, 5

1999, 5

2000, 5

2001, 5

2002, 5

2003, 9

2004, 5

2005, 3

2006, 4

2007, 7

2008, 7

2009, 7

2010, 7

2011, 6

2012, 8

Mystery movies:Title, Year, Rating

Shutter Island, 2010, 8.0

Mystic River, 2003, 8.0

Rosemary's Baby, 1968, 8.0

Harry Potter and the Deathly Hallows: Part 2, 2011, 8.1

Black Swan, 2010, 8.1

Twelve Monkeys, 1995, 8.1

Sleuth, 1972, 8.1

The Manchurian Candidate, 1962, 8.1

Anatomy of a Murder, 1959, 8.1

Rope, 1948, 8.1

Donnie Darko, 2001, 8.2

The Sixth Sense, 1999, 8.2

The Thing, 1982, 8.2

Dial M for Murder, 1954, 8.2

Strangers on a Train, 1951, 8.2

The Big Sleep, 1946, 8.2

The Maltese Falcon, 1941, 8.3

Rebecca, 1940, 8.3

L.A. Confidential, 1997, 8.4

Chinatown, 1974, 8.4

2001: A Space Odyssey, 1968, 8.4

Witness for the Prosecution, 1957, 8.4

The Prestige, 2006, 8.5

The Green Mile, 1999, 8.5

To Kill a Mockingbird, 1962, 8.5

Vertigo, 1958, 8.5

Citizen Kane, 1941, 8.5

Memento, 2000, 8.6

Psycho, 1960, 8.6

Se7en, 1995, 8.7

The Usual Suspects, 1995, 8.7

Rear Window, 1954, 8.7

Inception, 2010, 8.8

12 Angry Men, 1957, 8.9

In and after year 2000, top 10 actors who played in the highest rated movies: Actor, Movies

Aaron Eckhart, 1, 9.0

Anthony Michael Hall, 1, 9.0

Chin Han, 1, 9.0

Eric Roberts, 1, 9.0

Heath Ledger, 1, 9.0

Monique Gabriela Curnen, 1, 9.0

Nestor Carbonell, 1, 9.0

Ritchie Coster, 1, 9.0

Ron Dean, 1, 9.0

Alexandra Astin, 1, 8.9

Pairs of actors who co-stared in 3 or more highly rated movies: Actor A, Actor B, Co-stared Movies

Billy Boyd, Cate Blanchett, 3

Billy Boyd, Orlando Bloom, 3

Billy Boyd, Sean Astin, 3

Cate Blanchett, Orlando Bloom, 3

Cate Blanchett, Sean Astin, 3

Orlando Bloom, Sean Astin, 3

**Part 2 (25 points)**

The program should print out the top k actors who played roles in the highest rated movies (on average) in the provided genre.

You should use the sqlite3 database file you created in Part 1.

Top 5 actors who played in highest rated Drama movies:

Actor, Average Drama movie rating

William Sadler, 9.3

Neil Giuntoli, 9.3

Mark Rolston, 9.3

Larry Brandenburg, 9.3

Joseph Ragno, 9.3

Top 10 actors who played in highest rated Action movies:

Actor, Average Action movie rating

Ron Dean, 9.0

Ritchie Coster, 9.0

Nestor Carbonell, 9.0

Monique Gabriela Curnen, 9.0

Maggie Gyllenhaal, 9.0

Heath Ledger, 9.0

Eric Roberts, 9.0

Chin Han, 9.0

Anthony Michael Hall, 9.0

Aaron Eckhart, 9.0

Top 10 actors who played in highest rated Comedy movies:

Actor, Average Comedy movie rating

Wilfred Lucas, 8.6

Virginia Cherrill, 8.6

Tracy Reed, 8.6

Tiny Sandford, 8.6

Thomas Solivéres, 8.6

Sterling Hayden, 8.6

Stanley Blystone, 8.6

Slim Pickens, 8.6

Shane Rimmer, 8.6

Roy Stephens, 8.6

**What to submit:**

* uniqname\_si618\_hw3.db
* uniqname\_si618\_hw3.ipynb
* uniqname\_si618\_hw3.html

All your python program files should be able to be run without any error