**The University of Texas at Dallas**

**NAVEEN JINDAL SCHOOL OF MANAGEMENT**

**MIS 7310: ADVANCED TOPICS IN KNOWLEDGE MANAGEMENT**

**Spring 2016**

**Assignment 5 (Due February 25)**

**1.** (30 points)

* 1. Box Office Mojo (<http://boxofficemojo.com>) is a [website](http://en.wikipedia.org/wiki/Website) that tracks a variety of information on movies released each year (such as genre, studio, [box office](http://en.wikipedia.org/wiki/Box_office) [revenue](http://en.wikipedia.org/wiki/Revenue)s, etc.). You will write scripts that collect data on all 693 movies for the year 2015 that are listed on the site. Information on movies that were released in 2015 is available at: [http://www.boxofficemojo.com/yearly/chart/?yr=2015&p=.htm](http://www.boxofficemojo.com/yearly/chart/?yr=2015&p=.htm%20)
  2. You will also write scripts to collect data from the page People Index (also on the Box Office Mojo site) on all 774 movie actors (ordered by Total Gross) listed on the page:

<http://boxofficemojo.com/people/?view=Actor&pagenum=1&sort=sumgross&order=DESC&&p=.htm>

There will be four main steps involved.

Step 1:

Create a database in MySQL to store the data you will collect. Note that you will need two tables to store the data (one table for the movies, and another table for actors and their rank). Please keep in mind the relevant data type for each attribute in your tables. For example, Movie Name should be defined as a character variable, and Movie ID could be defined as an integer.

Step 2:

Collect the following information for each movie and save in a text file named **movies.txt**:

1. Movie Name.
2. Movie ID. Note that this is not displayed on the webpage, but appears in the source code for the page. For example, for movie “Star Wars: The Force Awakens”, the Movie ID is “starwars7”.
3. Studio that produced the movie.

Step 3:

Collect the following information for each actor from the page People Index (also on the Box Office Mojo site) and save in a separate text file named **actors.txt**:

1. Rank (Row number displayed on the page).
2. Actor Name.
3. Actor ID (it is in the source code of the webpage).

Step 4:

You will load the text files created in Steps 2 and 3 into the corresponding database tables. Note that an alternative approach is to insert the data into the tables directly as you collect data on each movie/actor (i.e., directly in Steps 2 and 3).

Deliverables:

1. Copy of your python code.
2. moves.txt and actors.txt files you collected in Steps 2 and 3.
3. A copy of your database in self-contained file with .sql as the extension name.
4. A data dictionary for your database. The dictionary should list the names of all the tables, and the names and data types of all attributes in each table. It would be helpful to also provide a brief description of each attribute.
5. Which studio produced the most number of movies in 2015? How many movies did the studio produce? Provide a list of all the movies produced by the studio.

Please note that in future assignments you will add to the data collected in this assignment. That will require you to use data collected in this assignment to collect the additional data, and then to conduct analysis on that data. Therefore, it is very important that you ensure all data are correctly captured in your database.

**The University of Texas at Dallas**

**NAVEEN JINDAL SCHOOL OF MANAGEMENT**

**MIS 7310: ADVANCED TOPICS IN KNOWLEDGE MANAGEMENT**

**Spring 2016**

**Assignment 6 (Due March 3rd)**

**1.** (30 points)

In this assignment, you will write scripts to collect additional data on the movies released in 2015. The data items you will need to collect are listed below. For each movie, you will have to use the Movie ID you collected in Assignment 5 to retrieve the page with the necessary information. For example, for the movie Star Wars: The Force Awakens”, the Movie ID is “starwars7”, and the URL corresponding to this particular movie is <http://www.boxofficemojo.com/movies/?id=starwars7.htm>. You will then parse this webpage to collect the following additional information for each movie:

1. Genre.
2. MPAA rating.
3. Budget.
4. Domestic Total Gross revenue for the movie.
5. Foreign Total Gross revenue for the movie.
6. List of actors. Note that not every actor’s name is hyperlinked on this page. When an actor’s name is hyperlinked, an actor id appears in the source code for the webpage (although it is not displayed on the webpage). For example, Harrison Ford’s id is “harrisonford.” For actors for whom no hyperlinks are provided, use their concatenated first and last names to create their ids. Note that you will need another table in your database to store the list of featured actors for each movie.

As mentioned above you will need to add a third table in your database to store the data. The third table will be to store the information on the actors featured in each movie. The other information about the movies should be entered in the table that stores data on the movies (please note that you will need to alter the table structure to store new columns before entering new information about each movie). As before, please keep in mind the relevant data type for each attribute in your tables.

Deliverables:

1. Copy of your Python code.
2. A copy of your database including all three tables.
3. A data dictionary for your revised database. As you did for Assignment 5, the dictionary should list the names of all the tables, and the names and data types of all attributes in each table. It would be helpful to also provide a brief description of each attribute.
4. What is the total Domestic revenue for all the movies in 2015? What is the total foreign revenue?
5. How many movies have available budget information? Among these movies, we assume that movies are considered to be profitable if the gross revenue (domestic plus foreign revenue) is greater than 1.2 times the budget. How many movies were profitable in 2015?
6. Which actor (actor name) starred in the most number of movies in 2015? How many movies did he or she star in? Provide a list of all the movies (movie names) that he or she is featured in.