CIS560 – Database System Concepts **Name:_____**

SQL Assignment 1 (20 points) – due Friday, September 6th at 11:59PM

***Collaboration policy**: This is an individual assignment. You are allowed to discuss the assignment with your colleagues, but your submission should reflect your own work. Sharing or copying code is not permitted.*

In this assignment, you are asked to create a relational **movie** database, populate the database and practice SQL queries on this database. The **movie** database consists of six tables:

movie_info (movie_id, movie_name, year, rating)
actor_ids (actor_id, actor_name, gender)
actor_movies (actor_id, movie_id)
producer_ids (producer_id, producer_name)
producer_movies (producer_id, movie_id)
genre(movie_id, genre)

actor_movies.actor_id refers to **actor_ids.actor_id**
actor_movies.movie_id refers to **movie_info.movie_id**

producer_movies.producer_id refers to **producer_ids.producer_id**
producer_movies.movie_id refers to **movie_info.movie_id**

genre.movie_id refers to **movie_info.movie_id**

We will use the following syntax to create tables in MySQL:

```
CREATE TABLE IF NOT EXISTS `example` (
`example_index` varchar(100) NOT NULL,
`sample_id` varchar(100) NOT NULL,
`examples` varchar(100) NOT NULL,
PRIMARY KEY (`example_index`),
FOREIGN KEY (`sample_id`) references sample_info(`sample_id`) ON
DELETE CASCADE
) ENGINE=InnoDB;
```

Note: There are several engines that MySQL can use and each has different properties. We will be using ENGINE=InnoDB in this class, as this type of engine supports transactions, row-level locking and foreign keys. You can read about other engines in MySQL at: http://dev.mysql.com/doc/refman/5.0/en/storage-engines.html

SQL scripts for creating and populating the database are provided in the files `TableCreation.sql` and `MovieData.sql`, available on KSOL.

1. (2p) Log-in to `phpmyadmin`. Select your database on the left hand side (your database has your account name). Use the `Import` tab to import the `TableCreation.sql` and `MovieData.sql` files, for creating tables and importing data. (*Note: tables must be created before data can be imported.*) Run a COUNT query for each relation to show the total number of tuples loaded.
2. (1p) Find all movies with your favorite star (actor/actress). Your query should return movie name and rating.
3. (1p) List all the producers who produced a *'Film-Noir'* movie in a leap year. (You need to check that the genre is *'Film-Noir'* and year is divisible by 4.) Your query should return the producer name, movie name, and year.
4. (1p) List, in alphabetical order, the names of all the actors who played in the movie 'alien: resurrection'.
5. (2p) List all producers who produced 10 movies or more, in descending order of the number of movies they produced. Return the producers' names and the number of movies each of them produced.
6. (2p) Find the movies which had both 'spacey,kevin' and 'zachary,emily' in their cast.
7. (3p) How many movies had 'spacey,kevin' but not 'zachary,emily' in their cast?
8. (2p) Find all actors that played in at least two different movies during the year 2005.
9. (2p) Find the number of movies by genre, for movies released in between 2005 and 2010.
10. (4p) (a) For each year, count the number of movies in that year that had only female actors. (b) Now make a small change: for each year, report the percentage of movies with only female actors made that year, and also the total number of movies made that year. For example, one answer can be:

    2011 10.81 135

    meaning that in 2011 there were 135 movies, and 10.81% had only female actors.

What to turn it: A *.txt* file with all SQL queries and the result of each query (if there are too many tuples in a result, show only the first result page). Submit this file using the course Dropbox.