Homework 3: Data Modeling & SQL

DSCI 551 – Spring 2024

Release: February 26, 2024, Monday

Due: 11:59pm, March 8, 2024, Friday

Points: 100

Consider the following tables:

Movies(id, title, year, length, language)
Actors(id, name, gender)
ActIn(actor_id, movie_id)
Directors(id, name, nationality)
DirectedBy(movie_id, director_id)

Assumptions:

1. All attributes in the above tables are of **string** types (char/varchar/text) except for id (including all types of ids), year, and length attributes which are **integers**.
2. Your code should run without error and satisfy all requirements stated in the question. It's only required for you to define PK, FK and data type for each attribute. It's up to you whether to add **"unique"/ "NOT NULL"/ "CHECK"/ FK CASCADE, etc**.
3. Create your own **sample database** with the above schema and insert some data into the tables for your own testing.
4. The homework assumes that you have created a database CINEMA on your EC2 MySQL, a user dsci551@localhost (with password Dsci-551), and grant all privileges on dsci551.* (i.e., objects in dsci551) to the user. You can log in as root and execute the following to satisfy the assumption:

   ```
   create database CINEMA;
   create user dsci551@localhost identified by "Dsci-551";
   grant all privileges on CINEMA.* to dsci551@localhost;
   ```

1. [20 points] Reverse engineer the above tables into an ER model. Draw the model.

2. [15 points] Create an SQL script that creates the above tables and insert some sample data into the tables. [**DATABASE NAME:** CINEMA]
   1) Inserting sample data: Please make sure to insert all types of data to cover all possible scenarios. Testing will be done on new test data.
   2) Make sure to design your referential integrity constraints appropriately [Foreign keys], Primary keys [Single or composite]. Take the real world cinema industry into account while creating constraints. [Actors can act in multiple movies, A movie casts multiple actors, etc.]

3. [35 points] For each of the following questions, write an SQL query to answer the question.

1) Find titles of the longest movies. Note that there might be more than one such movie.
2) Find out titles of movies that contain "Twilight" and are directed by "Steven Spielberg".
3) Find out how many movies "Tom Hanks" has acted in.
4) Find out which director directed only a single movie.
5) Find titles of movies which have the largest number of actors. Note that there may be multiple such movies.
6) Find names of actors who played in both English (language = "en") and French ("fr") movies.
7) Find names of directors who only directed English movies.

4. [30 points] Write a Python program search.py that takes an actor name and returns titles of movies the actor has acted in. Similar to HW1, your query to mysql server **SHOULD NOT** retrieve the entire table(s). You should query only the required information.

Permitted library: sqlalchemy, pymysql, pandas.

Submission Instructions:

1. Submit only 4 files:
   - one **.pdf file** containing the rendered/hand-drawn ER model called **ERD.pdf**
   - one **.sql file** for create table queries called **Q2.sql**
   - one **.sql file** for Q3 called **Q3.sql**
   - one **.py file** for Q4 called **search.py**
2. **DO NOT** submit a zip file for submission.
3. Points for each query will only be awarded if the query runs on test data or else 0 points.
4. All the 7 queries of Q3 should be present in the Q3.sql file. Any code other than queries in Q3.sql should be in comments.