Programming Assignment 1.2

Due: Friday 21st March, 2025 at 23:59 on **Gradescope**.

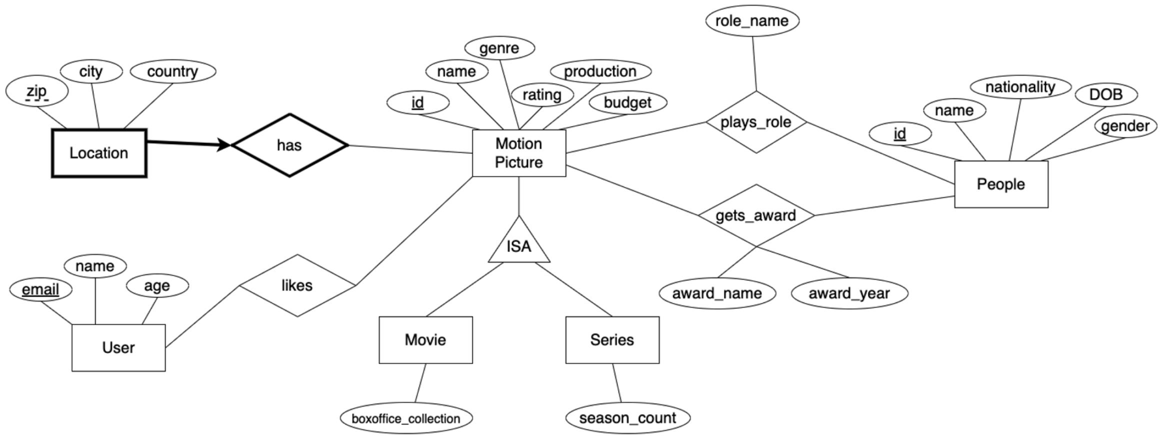
Quick recap of the first deliverable

In Problem 3 of Written Assignment 1, you have designed an Entity-Relationship Diagram (ER Diagram) for an IMDb-like movie database. Project 1 is an extension to that problem, whereas in Project 1.1, you worked on all database-related stuff and built a functional website.

We recommend that you start as early as possible on this project.

1 Existing Database Setup

In part one of the programming assignment, PA 1.1, you created ten relations for your movie database, had some dummy data inserted into the relations, and created a functioning website to project the query results. For your ready reference, here are the ER diagram and the schema of the relational database.



MotionPicture (id, name, rating, production, budget)
User (email, name, age)
Likes (uemail, mpid)
Movie (mpid, boxoffice_collection)
Series (mpid, season_count)
People (id, name, nationality, dob, gender)
Role (mpid, pid, role_name)
Award (mpid, pid, award_name, award_year)
Genre (mpid, genre_name)
Location (mpid, zip, city, country)

Note: Primary keys are underlined and foreign keys are in blue.

In PA 1.2, you will write a set of simple and complex **SQL queries** that would enable the website users to visualize the results of a diverse range of queries.

2 Data Loading for PA 1.2

For PA 1.1, you loaded the database using the `data.sql` file. For PA 1.2, you will use the `data2.sql` file. You will find this file in the `data` directory of the project. The lines beginning with `'--'` are comments for the PA 1.1 data and should already be in your database. The same data loading approach that you used in PA 1.1 can be used to load the data for PA 1.2. This means after selecting the correct database, i.e., `moviedb`, you can copy and paste all content of the `data2.sql` file into the `MySQL` shell.

3 Task: Queries

Implement these queries in `queries.py` file, more information in Section 4

1. List all the tables in the database.
2. Search for a motion picture by its name (parameterized). List the movie name, rating, production, and budget.
3. Find the movies that have been liked by a specific user's email (parameterized). List the movie name, rating, production, and budget.
4. Search motion pictures by their shooting location country (parameterized). List only the motion picture names without any duplicates.
5. List all directors who have directed TV series shot in a specific zip code (parameterized). List the director's name and TV series name only without duplicates.
6. Find the people who have received more than "k" (parameterized) awards for a single motion picture in the same year. List the person name, the motion picture name, the award year, and the award count.
7. Find the youngest and oldest actors to win at least one award. List the actor names and their age (at the time they received the award). The age should be computed from the person's date of birth to the award-winning year only. If there is a tie, please list all of them.
8. Find the American Producers who had a box office collection of more than or equal to "X" (parameterized) with a budget less than or equal to "Y" (parameterized). List the producer name, movie name, box office collection, and budget.
9. List the people who have played multiple roles in a motion picture where the rating is more than "X" (parameterized). List the person's and motion picture names, and count the number of roles for that particular picture.
10. Find the top 2 rates thriller movies (genre is thriller) that were shot exclusively in Boston. This means that the movie cannot have any other shooting location. List the movie names and their ratings.
11. Find all the movies with more than "X" (parameterized) likes by users of age less than "Y" (parameterized). List the movie names and the number of likes by those age-group users.
12. Find the actors who have played a role in both "Marvel" and "Warner Bros" productions. List the actor names and the corresponding motion picture names.

13. Find the motion pictures with a higher rating than the average rating of all comedy (genre) motion pictures. Show the names and ratings in descending order of ratings.
14. Find the top 5 movies with the most people playing a role in that movie. Show the movie name, people count, and role count for the movies.
15. Find actors who share the same birthday. List the actors' names (actors 1 and 2) and their common birthday.

The baseline requirement for all parameterized queries is that they at least carry text boxes for us to enter in the parameters required to execute the query. Please visit during office hours if you need any help.

4 Updating Your Code for PA 1.2

Before you start PA 1.2, you need to update your existing code with the latest changes. You can do this in one of two ways:

1. **Using Git:**

   (a) Open your terminal.

   (b) Navigate to your project directory using the command:

   ```
   cd <path to your project directory>
   ```

   (c) Run the following command to pull the latest changes:

   ```
   git pull
   ```

2. **Manually Downloading the Update:**

   (a) Visit the YourOwnIMDb GitHub repository.

   (b) Click the green **"Code"** button.

   (c) Select **"Download ZIP"** to get the updated code.

5 Logistics

5.1 Collobaration

You continue to work with the same project partner as in PA 1.2.

5.2 Submitting your assignment

You will need to submit your entire code base. You have two options to submit your assignment:

(a) Upload your code base to a Github repository and upload the link to **Gradescope**.

(b) Place all relevant files into a single folder, zip it and submit the zipped file to Gradescope. (Make sure you discard all irrelevant files (like .DS_Store) before clicking on upload.)