**UNIVERSITY OF REGINA**
**Department of Computer Science**
**CS 215 – Web & Database Programming**
**Fall 2019**

**Assignment #4: Movie Database Website – Database Schema Design & Querying**
**Posted: Sunday, October 27, 2019 at 6 AM**
**Due: Wednesday November 13, 2019 by 11:55 PM**

---

This is the third in the series of assignments for building movie database website. In the first of these assignments, you built an interface mock-up of the site. For the second one, you used JavaScript to implement the form validation and to support other types of interaction. **In this assignment, you will design and implement the database in MySQL (Assignment #4)**. In the subsequent assignments, you will implement the back-end programming in PHP (Assignment #5), and implement AJAX-based updating (Assignment #6). Because the assignments build upon each other, you may wish to make modifications to your previous assignment submission before starting this assignment.

---

A critical element of many Web-based applications is the ability to store and query data. As a result, it is necessary to design an appropriate database for storing the data, construct the database in MySQL, and write the queries to store and retrieve the data.

This assignment is divided into three different components:

## 1. ERD and database specification

Your task here is to create an appropriate Entity Relationship Diagram (ERD) for the data associated with your movie database system. This ERD must document the data model for the entire system, and should conform to the **second normal form (2nf)** specification for the data model. Your ERD diagram should also show relationships between entities.

Based on this ERD, provide the SQL statements that build the table structure for your database.

Note: You should **think carefully what entities and attributes** the Movie Database system will contain. Read through the previous assignment and **the next two sections of this assignment** so that you fully understand what data is to be stored and retrieved. This will help you to make important design decisions for the database.

**Note:** There should be **no many-to-many relationship** in your ERD. If there is one, add a new table and break the relationship into one-to-many and many-to-one relationships**.**

## 2. Data Storage Queries

There are a number of forms in this web application that will require that data be stored in the database. Your task here is to write sample queries to verify that the data storage is working correctly.

a. Login / logout status
   - Save the fact that the **user** has logged in
   - Save the fact that the user has logged out

b. Login page
   - Signup form collects details about the user. Insert them in the database. You can assume the URL for the avatar is known at the point of executing this query.

c. Movie details page
   - Save the user's **rating** for a movie. (rating and date-time rated)
   - Save the fact the user **viewed** the movie. (time viewed)
     *Note: Viewed a movie means the user saw the movie's detail page, not watching the movie.*

d. Watchlist page
   - When a new **watchlist** is created, you need to save the watchlist's name in the database. Write a query to save the watchlist's name and date-time created in the database.

e. Watchlist details page
   - Save a movie as an **entry** to a watchlist (movie and date-time added).

## 3. Data Retrieval Queries

There are a number of pages in this web application that require that data be extracted from the database. Your task here is to write sample queries to verify that the data is being extracted correctly. Note that some queries require you perform a **join operation.**

a. Login page
   - Given a username and password pair, retrieve the user ID, matching handle (username), and avatar image for the record; if the username and password pair do not match any records in the database, an empty result set will be returned.

b. Home page
   - Retrieve top 15 rated movies.
   - Retrieve 15 latest movies. Sort them based on release date (descending).
   - Retrieve 15 movies and sort them based on movie title (ascending).
   - Given a movie title retrieve matching movies (for the search bar). (Hint: use LIKE %movie_name%)
   - Given a movie genre retrieve a top 15 rated movies based on that genre. (Hint: use LIKE %movie_name%)
   - Given a movie origin retrieve a top 15 rated movies based on that origin. (Hint: use LIKE %movie_name%)

   *Note: You need to return the movie title, id, poster, and rating only.*

c. Movie details page
   - Given a movie id retrieve the title, poster, genre, origin, rating, cast, director, and Wikipedia link.

d. Watchlist page
   - Retrieve all the watchlist names. Sort them by date created (descending).

e. Watchlist detail page
   - For a watchlist id retrieve all the **entries** (movies) that are in that watchlist. Sort them by date added (descending).

   Note: Retrieve only movie title, id, and rating of a movie.

**Grading Scheme**

This assignment will be graded out of 10 marks, based on the following criteria:

2 marks: Entity Relationship Diagram
2 marks: SQL statements to build table structure
3 marks: Data storage queries
3 marks: Data retrieval queries

**Submissions**

Your database on Hercules should include all of the tables for this assignment. You should provide the ERD in a single PDF file, and the SQL statements in a **simple text file (.txt),** numbered accordingly. These should be zipped together and uploaded to UR Courses.

A simple submission log file should be provided that includes your name, student number, class number, and the username/password for your MySQL account on Hercules. **Failure to provide these support documents will result in delays in the grading of your assignment and possibly a deduction in your grade.**

Late submissions will not be accepted. If there are exceptional circumstances that kept you from submitting your assignment on-time, you should consult with your instructor as soon as you are able to do so. See the syllabus for more details on the late policy for this class.