CS 450 - Database Systems

Fall 2021

Instructor: Dr. Jessica Lin

Homework 4 Part 2 – Due Nov 12 at 11:59pm

*** Note: This is an INDIVIDUAL assignment ***

Consider the following schemas for the online movie streaming company. This is the same as your last assignment, except that "Member" has been changed to "Members" and "Profile" has been changed to "Profiles" to match the testbed.

Members(member_ID, first_name, last_name)

Profiles(member_ID, profile_name)

Movie(movie_ID, title, movie_year, producer, avg_rating)

Actor(actor_ID, first_name, last_name)

Starred_By(movie_ID, actor_ID)

Watch(member_ID, profile_name, movie_ID, rating)

For each relation, the attribute(s) of the primary key is(are) underlined. In addition, the following foreign key constraints hold:

In *Profiles*,
Foreign Key: member_ID references Members(member_ID)

In *Starred_By*,
Foreign Key: movie_ID references Movie(movie_ID)
Foreign Key: actor_ID references Actor(actor_ID)

In *Watch*,
Foreign Key: (member_ID, profile_name) references Profiles(member_ID, profile_name)
Foreign Key: movie_ID references Movie(movie_ID)

Write the following queries in SQL.

Note: A testbed will be provided for you to test your queries. Run the testbed script to create and populate the relations, and test your queries. You should insert more tuples

into the database to make sure that your queries return correct results, since a different testbed will be used to grade your homework.

1. Find the members who watched *all* of the movies that Hugh Jackman starred in.
2. For each member, print the member's ID and the average rating he/she gave for movies, for members who rated at least 2 movies.
3. For each member, print the member's ID and the number of the movies watched under the account. For example, suppose member David Smith has two profiles "David" and "Kids." Also suppose that, under his account, 3 movies were watched using profile "David", and 5 movies were watched using profile "Kids", then you should print "dsmith 8". If a member has not watched any movie, you should still print the member's ID but with zero as the count.
4. Find actors (actor_ID) who starred in the most movies.
5. Find movies (movie_ID) that have the highest average rating. There might be more than one movie with such highest average rating.

Submission instruction:

Please submit two files: (a) a SQL script (.sql) containing your queries, and (b) a PDF file containing the query results using the testbed provided. Please follow the same instructions from HW2 when you prepare your SQL script. You will only receive full credit for queries that execute successfully (with no error) AND return correct results on both the testbed provided to you and the modified testbed for grading.