

INFS 614 - Database Concepts

Fall 2013

Instructor: Dr. Jessica Lin

Project Assignment

General. Your project is to design and implement a database for an online movie rental company like Netflix (for this project, we will focus on the streaming-only plans). At the back-end, the database manages all of its data in the Oracle database management system. At the front-end, you are required to implement a command-line interface to its users (the database users, not Netflix users) using Java and JDBC.

The following is the list of “basic” information that the database should store. Note the organization of the list does not necessarily imply that this is how you should design the database. For example, for each member, you have a list of items that need to be stored. It is up to you to decide how you should store and organize these items. You will need to come up with a design that makes sense, given the descriptions. *You may work in a team of 2.*

1. Accounts. For each account, record the account owner’s information as follows:
 - a. Unique member ID
 - b. Full name (first name, last name)
 - c. Credit card on file
 - d. Profiles (see below)

2. Movies. For each movie:
 - a. Unique movie ID
 - b. Movie name
 - c. Year it was produced
 - d. Cast
 - e. Producer
 - f. Genre(s)
 - g. Average member rating

3. Actor/Actress. For each actor/actress:
 - a. Unique ID
 - b. Name (first, last)
 - c. Movies that he or she was in

4. Each account can have up to 5 profiles for different members in the household¹. For example, an account owner “bradpitt” may have a profile for himself “Brad”, another profile “Angie” for his wife, and a separate profile “Kids” for his kids. For each profile:
 - a. Profile name (not unique across all members, but unique within an account)
 - b. Preferred movie genre(s)
 - c. Movie Queue (see below)
 - d. Rental history (see below)

5. There is a separate rental history for each “profile” created by a member (account owner). Remember a “profile” does not have a globally unique identifier.
 - a. Movies rented
 - b. Ratings posted by this profile (see below). Rating scale is from 1 to 5, 1 being “Do not like” and 5 being “Like”

6. Each profile has its own movie queue. In the queue, record the movies that he/she wants to see sometime in the future. For each queue,
 - a. Movies (movie ID)
 - b. The profile owner (i.e. the account owner) manually ranks the movies in the queue by assigning priority numbers for the movies. The priority numbers are consecutive numbers between 1 and n , where n is the number of movies in the queue. Movies with smaller priority number will be placed first.

The descriptions above summarize the minimal information your database should contain. As mentioned, it does not necessarily imply the organization of the relations. You may re-arrange the attributes or add more attributes. The design decision is yours, but your design should be reasonably efficient, and well justified. You will be graded on both correctness and quality of the design.

Your interface should have minimal functionality including a variety of queries and browsing capabilities over the movies (e.g. the database user should be able to search by genre, movie name, movie star, etc.), profiles, rental history, etc. The user should be able to update the database by inserting or deleting tuples.

You’ll be graded for each phase. The project accounts for 25% of your total grade.

¹ This is a new feature on Netflix. Separate profiles and rental history allows more accurate recommendation to its members.

Phase 1: Conceptual Design. (5%) You need to submit an ER diagram for your database according to the project description and the assumption you make. (See HW1 assignment for an example).

Phase 2: Schema Design. (5%) After your conceptual design is finished, you need to translate the ER diagram into relation schema. Submit a copy of your relation schema, the SQL script that you use to implement your database, and insert enough tuples in **each** table, e.g. at least 5 accounts (with at least 2 profiles in each account), 10 movies, 20 actors/actresses, etc.

Phase 3: JDBC Implementation. (12%) Write a program in JDBC that will allow users to access and query your database. Command-line menu is acceptable. More information on this phase will be given later in the semester. In-person demos will be scheduled during the last week of class.

Putting it together. (3%) Prepare a final report. Your report should contain the following:

- Materials from all phases
- README describing how to run your program
- Screenshots on program/query execution
- Write-up: Discuss and justify your design decisions and challenges you encountered. List any part(s), if any, that do not work properly or you did not implement.

Timeline:

September 13: project released

October 1: Phase 1 (ER diagram) due

October 15 (online submission): Phase 2 (schema) due

December 3: Demo and final report due

December 10: Final