

CSE305-532 Project Description

The Contents are modified from Scott A. Smolka's Spring 17 CSE305 "Requirements Specification for the Database Programming Project" for SUNY K CSE305-532

URL Smolka's page: <http://www3.cs.stonybrook.edu/~sas/courses/cse305/spr17/project/index.html>

Introduction

In this project, you will design and implement a relational database system to support the operations of an **on-line movie rental system**, along the lines of netflix.com. You will use HTML for the user interface, MySQL for the database server, and Python, mysql-python, flask, flask-mysql for connectivity between the user interface and database server.

Getting Started

The system you are to develop is a lot like the on-line movie rental web sites [Netflix](http://netflix.com) and [Amazon Video](http://amazon.com/video). These systems work basically as follows:

1. A user maintains a queue of movies he or she would like to see.
2. The user selects movies from a movie database to place in his queue.
3. The order of the movies in the queue is the order in which he would like to view the movies.
4. The user signs up for a monthly plan that allows him to have viewing access to anywhere from 1 to 3 movies at-a-time.
5. There are 2 types of plans, 1 limited plan and 1 unlimited plans.
6. The limited plan allows one to view 1 movie at-a-time and at most 2 movies per month.
7. The 1 unlimited plan allow one to view 1 movies at-a-time and place no limit on how many movies you can view per month.
8. The user "returns" the movie when he is done viewing it, which then allows him to get access to the next **available** movie from his queue.
9. Assume a monthly fee of \$10/month for the limited plan. For the unlimited plans, assume the following monthly fees of \$20/month.

In reality, Netflix and Amazon use the postal system to mail users a DVD along with a pre-paid envelope for returning it. They also allow you to stream the movie to your PC, laptop, or mobile device. You "return" the movie by hitting a return button linked to your account.

Please visit the [Netflix](http://netflix.com) and [Amazon Video](http://amazon.com/video) web sites in order to get a better idea of how they work and to obtain an understanding of the **look and feel** of a movie rental web site.

Project Specification

The basic idea behind your on-line movie rental system is that it will allow customers to use the web to browse/search the contents of your database (at least that part you want the customer to see) and to rent movies over the web. In this regard, it is a lot like the on-line movie rental web sites [Netflix](#) and [Amazon Video](#). So visit these sites to get an idea as to what your system should look like.

Your database system must be based on the specifications and requirements that follow.

1 System User

The users of your system will be the customers that use your system to rent movies and pay monthly fees for doing so.

2 Required Data

The data items required for the movie-rental database can be classified into five categories: orders, movies, actors, and customers where an *order* is an order to rent a movie.

This classification does not imply any particular table arrangement. You are responsible for arranging the data items into tables, determining the relationships among tables and identifying the key attributes. Finally, you should specify and enforce integrity constraints on the data, including referential integrity constraints.

You will first create an E-R diagram of your online movie rental system before developing your relational model.

2.1 Order Data

This category of data should include the following items:

1. Order ID
2. Movie ID
3. Customer ID
4. Date/Time (the order was placed)

A user places an order to rent a particular movie.

2.2 Movie Data

This category of data should include the following items:

1. Movie ID
2. Movie Name
3. Movie Type
4. Actors
5. Number of Copies
6. Rating

Movies are of a certain type: **Comedy, Drama, Action, or Foreign**. For testing purposes, you can populate your database with any movies you like. A movie's rating is a measure of its popularity. You can assume it is a integer ranging from 1 to 5, with 5 indicating "must see at all cost" and 1 indicating "a clunker".

2.3 Actor Data

This category of data should include the following items:

1. Actor ID
2. Actor Name
3. Male/Female
4. Age
5. Movies appeared in
6. Rating

An actor's rating is a measure of his or her popularity and like a movie rating is an integer from 1 to 5.

2.4 Customer Data

The items required for this category include:

1. Last Name
2. First Name
3. E-mail Address
4. Account Number
5. Account Type
6. Movie Queue

A customer's Account Type indicates which of the 2 rental plans (the limited plan and the unlimited plan) the customer has signed up for. The Account Type also functionally determines the customer's monthly account fee. The Movie Queue is the queue of movies the customer would like to see.

3 User-Level Transactions

A database *transaction* can be viewed as a small program (written in the DML) that either updates or queries the database. Transactions that change the contents of the database must do so in a consistent manner. Moreover, transactions should not interfere with one another when running concurrently.

What follows is a breakdown of the user-level transactions that your database system should support. To make sure transactions maintain the integrity of the database, you must write them using the SQL transaction structuring capabilities (i.e., begin transaction, commit transaction, etc.).

3.1 Customer-Level Transactions

Customers should be easily able to browse your online movie rental system on the web and place orders to rent movies. In particular, they should be able to readily able to maintain a queue of movies they would like to see. While they will not be permitted to access the database directly, they should be able to retrieve the following information:

- A customer's currently held movies
- A customer's queue of movies it would like to see
- A customer's account settings
- Movies available of a particular type
- Movies available with a particular keyword or set of keywords in the movie name
- Movies available starring a particular actor or group of actors
- Best-Seller list of movies
- Personalized movie suggestion list

Customers should also be able to:

- Rate the movies they have rented

4 User Access Control.

- A customer should not be allowed access to other customers' account information

5 User Interface

HTML and its successors provide facilities for creating pop-up and pull-down menus, value lists, input/output forms, labels and customized reports. You should make use of all of these capabilities, and in the process come up with a system that caters to users with only limited computer knowledge. The information you provide to customers should look professional and inviting.

6 Documentation

You will be required to supplement your completed database implementation with a design document that contains information concerning your design criteria and decisions. The following is a list of some of the information you should include:

- Entity-Relationship (E-R) Diagram of the complete database scheme
- Lucid description of the relational database scheme for your online movie rental system, including a discussion of the reasoning behind your design decisions. Make clear how your design supports efficient query processing.
- A list of all functional dependencies in the relational database scheme
- Description of integrity constraints including referential integrity

You will also be required to submit a *Users Guide* that carefully explains how to use all aspects of the system. It should be understandable by non-computer experts. Be sure that the user interface (screen design, menu structure, etc.) is clearly explained.