CSE 305 -- Principles of Database Systems

Spring 2017

Requirements Specification for the Database Programming Project

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, for the database server, and Java, Javascript, and JDBC for connectivity between the user interface and database server. The database server is accessible from the PCs in both the undergraduate and graduate Transaction Processing labs (Computer Science Building, Rooms 2114 and 1239), on which you will be given a MySQL account.

Please see the Transaction Processing Lab web site for general information about the Transaction lab and its policies, and the lab's MySQL web site for information on how to access MySQL in the Transaction lab and account information.

If you own a PC, you are encouraged to develop as much of the code as possible on your PC, to ease the congestion in the TP lab. For these purposes, I suggest you use SQL Server or MySQL Server.

SQL server is a relational database server produced by Microsoft, based on the ANSI SQL-92 standard. You can get a **free** copy of the SQL Server Developer Edition, for 2005 and 2008, from the Stony Brook Microsoft Developer Network Academic Alliance (MSDNAA) web site.

For MySQL, you can download a copy of MySQL server from the MySQL web site. MySQL is a good RDBMS for learning SQL, has very low resource requirements, is open source, and has JDBC drivers. Also, it is SQL-99 compliant so it will be more in tune with ANSI SQL standards as opposed to SQL Server, which has some proprietary semantics embedded over the standard query language.

You are to work in **teams of three**. Many of you will form teams of three on your own. If not, do not worry. Please just be sure to attend class on a regular basis, and I will make sure that every student is placed in a team of three. Once you have joined a complete team, you and your teammates should choose a name for your team/company. E-mail the names of your team members and the name of your company to mjavanmard@cs.sunysb.edu as soon as possible. If you are unable to find a team to join, please let me know asap, and I will find one for you.

Getting Started

The system you are to develop is a lot like the on-line movie rental web sites Netflix and Blockbuster. 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 4 types of plans, 1 limited plan and 3 unlimited plans.
6. The limited plan allows one to view 1 movie at-a-time and at most 2 movies per month.
7. The 3 unlimited plans allow one to view 1-3 movies at-a-time, respectively, and place no limit on how many movies you can view per month.
8. The limited plan has the lowest monthly fee, whereas for the 3 unlimited plans, the more movies the user chooses to have access to at-a-time, the higher the monthly fee.
9. 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.
10. Assume a monthly fee of $10/month for the limited plan. For the 3 unlimited plans, assume the following monthly fees: $15/month for 1 movie out at-a-time; $20/month for 2 movies out at-a-time; and $25/month for 3 movies at-a-time.

In reality, Netflix and Blockbuster 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. For this project, your system will furthermore assume that there are only a certain number of copies of any given movie that can be out in circulation at any moment in time. Once that limit is reached, the movie is considered **unavailable**.

A very interesting aspect of online movie rental systems such as Netflix and Blockbuster is that they implement Recommender Systems that make movie suggestions which are likely to be of interest to their customers. Such systems often use a Nearest Neighbor approach to determine the movies they think will be of interest to a particular customer. The well publicized Netflix Prize, a contest with a dataset of over 100 million movie ratings and a grand prize of $1,000,000, has energized the search for new and more accurate algorithms. Please see the Recommender System Wikipedia page for more information.

Please visit the Netflix and Blockbuster 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.

I will also place a Powerpoint tutorial on blackboard on how to connect to Sybase using Java and on how to develop the UI (User Interface) for your online movie rental system course project. You will be able to find the tutorial in the **Project Assignments** subfolder of the **Assignments** folder. (You will also be able to find the tutorial via the link associated with the announcement I will post on blackboard about the tutorial.)

To further help you with the project, a former TA has developed a prototypical database system for a University Registration system that closely resembles the student registration system used as a running example in the course textbook. You should consult this demo system as a guide for your project development. You can access the demo system, and the underlying source code, by following the links I placed in an announcement I recently posted on blackboard. As you experiment with the demo system, you should also start to become familiar with Java, Javascript, HTML, and JDBC on your own. I will lecture about JDBC (Chapter 8 of the course textbook) after the midterm. You might also want to buy the books about these languages recommended on the course homepage.

It is recommended that you regularly check the blackboard **Project Assignments** folder and the PROJECT ASSIGNMENTS web page for information on the course project. This page will be updated from time to time with important information on how to proceed with the project.

CSE 305 Course Project Competition

This semester there will be a **CSE 305 Course Project Competition** in which the top-three finishing teams will receive award certificates. I am trying to find a corporate sponsor for this competition and will let you know when and if I succeed in doing so. So please do your best to produce a professional-looking online movie rental system that offers a full compliment of features and functionality, including at least those specified below.

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 Blockbuster. 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 Users

The users of your system will be the customers that use your system to rent movies and pay monthly fees for doing so, customer representatives who provide customer-related services, and the site's manager. You should assume that the computer knowledge of the users is limited (say, that of a typical AOL subscriber), and thus your system must be easy to access and operate.

2 Required Data

The data items required for the movie-rental database can be classified into five categories: *orders*, *movies*, *actors*, *customers* and *employees*, 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. In addition, you should include indices in your tables to speed up query processing. You should base your choice of indices on the type and expected frequency of the queries outlined in Section 3. Finally, you should specify and enforce integrity constraints on the data, including referential integrity constraints.

As I mentioned in class, you will first create an E-R diagram of your online movie rental system before developing your relational model. Details of this assignment will be forthcoming.

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)
5. Employee ID

A user places an order to rent a particular movie. A customer representative, who oversees the successful completion of the order, is associated with every rental order.

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. Distribution Fee
6. Number of Copies
7. 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. We will provide you with all the movie data you will need to demo your system to us at the end of the semester. The Distribution Fee is the dollar amount your company has to pay in order to obtain the rights to distribute a movie to the general public. 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. Address
4. City
5. State
6. Zip Code
7. Telephone
8. E-mail Address
9. Account Number
10. Account Type
11. Account Creation Date
12. Credit Card Number
13. Movie Queue
14. Rental History
15. Rating

A customer's Account Type indicates which of the four rental plans (the limited plan and the three unlimited plans) 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. The Rental History is a history of the movies the customer has rented since opening his account. The Rental History should also indicate which if any movies the customer currently has out. The customer's Rating should reflect how active a renter he or she is (the more active, the higher the rating). Like a movie rating, a customer rating should be an integer ranging from 1 to 5.

2.5 Employee Data

This category of data should include the following:

1. Social Security #
2. Last Name
3. First Name
4. Address
5. City
6. State
7. Zip Code
8. Telephone
9. Start Date
10. Hourly Rate

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 Manager-Level Transactions

The manager should be able to:

- Add, Edit and Delete movies
- Add, Edit and Delete information for an employee
- Obtain a sales report (i.e. the overall income from all active subscriptions) for a particular month
- Produce a comprehensive listing of all movies
- Produce a list of movie rentals by movie name, movie type or customer name
- Determine which customer representative oversaw the most transactions (rentals)
- Produce a list of most active customers
- Produce a list of most actively rented movies

3.2 Customer-Representative-Level Transactions

Customer Representatives should be able to:

- Record an order
- Add, Edit and Delete information for a customer
- Produce customer mailing lists
- Produce a list of movie suggestions for a given customer (using the recommender system which uses information about the customer's past orders and that of nearest neighbors)

3.3 Customer-Level Transactions

Customers should be 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
- A history of all current and past orders a customer has placed
- 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

Your database system should provide controlled access to the data by distinguishing between the different types of users: manager, customer representatives, and customers.

- Customer Representatives should not be able to perform manager-level transactions; however, they should be able to read employee information, except for the hourly rate.
- Customer Representatives should be able to record the receipt of an order from a customer.
- A customer should not be allowed access to other customers' account information, or to any employee information.

5 Utilities

In addition to the transactions described above, the system should provide facilities for:

- Allowing the manager to add and delete users and movies
- Backing up the database files
- A comprehensive **Help** facility, including a topic-driven pull-down Help menu

6 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.

7 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.

8 Grading

You will be given three assignments: 1) produce an E-R and relational model of your system; 2) implement (in SQL) and execute all transactions described in the above project specification; and 3) implement (using Java, Javascript and JDBC) the final interactive system to support your online movie rental system. The due date for the first assignment will be announced shortly; as for the the other two assignments, the due dates for all three assignments will be spaced roughly three weeks apart.

All documentation should be on-line. You will also be asked to hand-in hardcopies when assignments are due.

In order to evaluate your final system, you will be asked to present a short (15-25 minutes) demo to myself or the TA. This will most likely happen during the last week of classes or shortly thereafter.

9 Collaboration Plan

As stated above, you will be working on the course project in in teams of three. A rough, three-way division of labor for the first project assignment is as follows:

- Teammate 1 will focus movie and actor data.
- Teammate 2 will focus on account and order (rental) data.
- Teammate 3 will focus on customer and employee data.

About this document ...

Scott Smolka
Last Modified: Wed Feb 13 11:20:11 EDT 2013