

## Assignment 4 (Project)

---

**Total scores: 100**

**Due Thursday, July 30**

**Summer 1998**

**Instructor: Osmar R. Zaiane**

---

- You must design and implement a database for a video store. The video store acts as a club where members (customers) borrow video tapes, music CDs, and video games. The database is intended to keep track of the current video tapes, CD, and video-games collections as well as the list of the club members.  
The following specifications have been given to you.
  - Members borrow as many video tapes, CDs, and games as they wish.
  - There are different types of movies (drama, fiction, kids, etc.)
  - There are different formats for videos (beta, VHS, and digital)
  - A movie can exist more than once in the collection, in one or different formats.
  - A movie can be borrowed for 1 day, 3 days (week-end) or a week.
  - A music CD or a video game can be borrowed for up to a week.
  - There are three (3) categories of videos: new releases, classic, documentary.
- Implement the video store system with Visual C++ and SQL-Server. The system will be demoed in the CSIL lab.
- Here's a list of some output requirements that your finished design should be able to handle. You don't have to implement any of these things, but you should be able to indicate how they can be handled by your design, and whether application programs will be required to handle some requirements.
  - A list of "bad" members (members who returned the borrowed items late with a given threshold) is printed regularly.
  - A list of members who borrowed more than a given time  $t$  and returned the items on time. (for bonus for example)
  - A report on individual member is available on demand with history and current borrowed item list.
  - A list of new releases is published regularly and mailed to members.
  - A list of most popular CDs, videos, and games.
- Note that the problem definition is somewhat incomplete, so you will have to make some decisions about how information should be managed for the video store. Make reasonable additional assumptions as required. Be sure to state these assumptions clearly. If some of the demands must be handled by application programs communicating with SQL-Server (rather than by SQL queries) state which ones. Some sample queries for an application like this would be
  - Find the club members who borrowed a given movie.
  - Find the telephone numbers of members late in returning an item by more than 2 days, and the specification of the items borrowed.

- List the CDs by the same composer as the sound track of a given movie.
  - List in alphabetical order all movie titles where a given actor is playing.
  - List all members who are currently borrowing a game but are not borrowing a movie.
  - List all the members who are currently borrowing a game and a movie.
  - List the members that never borrowed a video game.
  - List the items currently borrowed by a given member
  - List of items borrowed more than a given time, grouped by item type (video, CD, game).
- Feel free to make up your own -- this just gives you an idea of the level of complexity I'd like to see. These queries will be tested during the demo.

This is a group work. Groups are composed of three (3) students. You will have to form the groups and send me the names and student numbers of members in each group.

Each group will vote a leader. The group leader will assign tasks to all members of the group (example: design, programming, documentation, etc.). Each member will have a responsibility which will be explicitly identified when the assignment is handed in. The overall project grade will be 75% group work + 10% peer evaluation + 15% member task evaluation. This means that the members of the same group may not get the same grade.

Each member of a group has a responsibility. This does not mean that each student works on one and only one task. Collaboration should be on all tasks.

The peer evaluation will be sent to me anonymously by each student by e-mail. Each student will grade his two group-mates between 1 and 10.

Good luck and have fun with the project.

---

*Last update: June 9th, 1998. Page created and maintained by Osmar R. Zaiane*