Escuela Politécnica Superior, UAM

Data Structures 2019-2020

Assignment 2: Databases and Programming Languages.

Upload the completed assignment to Moodle.
Late submissions carry a penalty of 1 point for each natural day after the deadline.

1 Objective

The end user never interacts directly with a database through SQL, but rather through some user interface, which in turn uses a "logical layer" of software to build the SQL queries and run them on the database. Thus, once we have our data model implemented in a relational database, we need our software to access it and work with it.

In this assignment, you will experiment with database access and manipulation using the C programming language. In particular, we will use the ODBC (Open Database Connectivity) library to create a number of programs using the *dvdrental* database provided in the first assignment.

Everything you learn in this assignment applies to any programming language having an implementation similar to ODBC, JDBC, etc. That includes most programming languages.

2 Submission format

Upload your completed assignment with a single .zip or .tgz file to Moodle, using the link provided in the course page. Name your file EDAT1920_P2_GGGG_XX.zip (or .tgz), where:
- GGGG is your lab group (1201, etc.)
- XX is your team's number in the lab (01, etc.; these are listed in Moodle)

Only ONE team member should upload. Do not forget to include your names in the assignment report.

3 Submission content

- Source code of all programs.
- Makefile needed to compile the code.

- A report (in pdf) which must include:
    - Names of both team members
    - A section for each of the programs, containing:
        - A brief explanation of the implemented logic and the SQL queries used.
        - A sample run of each program

Very important:

- The database *must* be named "dvdrental".
- Use *only* the provided database.
- We will grade the assignment by loading this database from scratch for each submission. DO NOT USE ANY OTHER DATABASE.

4 Resources

Use the database dump "dvdrental.sql" provided in Assignment 1.

The course page provides a few ODBC code samples, including two **VERY USEFUL** files, "odbc.h" and "odbc.c", with the most commonly used functions to connect to and disconnect from the database, and to print errors encountered in the execution of queries. You are encouraged to use these functions in your code.

The main resource for information about ODBC is http://www.unixodbc.org. The "Manuals" section in this site contains a brief Programming Manual tutorial, including compilation, database connection, and retrieval of results from queries.

You can find much more detailed tutorials and examples in the Easysoft tutorials on using ODBC from C, including:

- ODBC from C Tutorial Part 1
- ODBC from C Tutorial Part 2 - Fetching Results
- ODBC Code Samples

5 Accessing a database with C

In this section, you will use ODBC to create C programs to perform a number of operations on the DB. ODBC is a library for connecting to a database and retrieving the results. We will see in class the fundamental concepts of ODBC and how to use it to interact with the DB.

5.1 Your task

Implement the programs described below in C using ODBC, using the provided database.

- The programs should check for errors in the number of arguments and their values, but do not need to check for format problems in the arguments.
- The programs should also control for errors in their interaction with the database. If the result of running a query is unexpected, the user should get a clear explanation of the source of the problem.
- Deletions should propagate to all entries related through foreign keys to the deleted entry.
- The programs should connect to the “dvdrental” database with user “alumnodb” and password “alumnodb”

5.2 Programs

The main task of this assignment is to implement the following programs:

5.2.1 dvdreq

This program allows the user to perform a number of queries on the database.

dvdreq customer -n <First Name> -a <Last Name>

Print a list of customers with matching first name and/or last name. For each customer, print:

- customer id
- customer’s first name
- customer’s last name
- registration date
- complete address

dvdreq film <title>

Print, for each movie whose title matches, fully or partially, the title argument:

- film id
- film title
- release year
- length
- language
- description
- a listing of actors with their first and last name

dvdreq rent <customer_id> <init date> <end date>

Print the rentals carried out by a customer between init date and end date, sorted by rental date.

Include, for each rental:

- rental id
- rental date
- film id
- film title
- staff id of the employee who rented the movie to the customer
- employee name
- store id
- store name
- amount paid for the rental

dvdreq recommend <customer Id>

Print a list of the three most rented movies that belong to the customer's favorite (most often rented) category, but which he or she has not rented before. For each of the three recommended movies, print:

- film id
- film title
- film category name

5.2.2 dvdrent

This program manages rentals.

dvdrent new <customer Id> <film id> <staff id> <store id> <amount>

Add a new rental, after making sure that all input arguments are correct (exist in the database). This requires finding an unrented DVD for the chosen film, using the inventory and rental tables.

dvdrent remove <rent Id>

Remove a rental from the database, as well as the associated payment.

5.2.3 dvdfilm

This program removes a film from the database.

dvdfilm remove <film id>

6 Grading criteria

- Assignment Report (2pts)
    - Includes all required content: 50%
    - Quality and clarity of content: 30%
    - Presentation: 20%

- Programs (8pts)
    - Criteria:
        - Functionality: 50%
        - Error control: 30%
        - Quality of solution, code structure and code readability: 20%
    - dvdreq (4pts)
    - dvdrent (2pts)
    - dvdfilm (2 pts)