
EECS-3421, Winter 2017

Projects II: APPs

In this project, you will work again with an existing database, the **YRB** database from project I.

Project II, APPs, is to write an application program for the **YRB**. Application programs are often called "APPs" in jargon. As your APP will modify the **YRB**, this is the reason everyone has his/her own copy of **YRB** for the project.

APPs (Project II)

Write an application program in Java to do the following. Go to [JDBC on PRISM at CSE York](#) and check out the instructions on how to write and compile SQL application programs. Provided are some on-line examples.

Search and Purchase application:

- The program starts by finding a customer, that is, it looks for a customer to see if the customer with a given id exists.

If the customer does not exist, the program displays an error message and requests the customer id again.

If the customer exists, the query returns and displays the customer information (the customer id, name, and city).

(If you write a stored procedure to find the customer, name your stored procedure "find_customer". This stored procedure gets a number as the customer id and returns the customer information (cid, name, city) if there is such a customer.)

- If the customer exists:

Write a query in your program to return all the categories (cat) in your database. Then, display all the categories (in a drop-down list), so the customer can choose a category from the list. (If you write a procedure to return all the categories' name in your database, name it "fetch_categories".)

- After choosing a category by the customer, customer can enter the title of the book. You need to write a query that looks for the book with the given title and the selected category. If the given title with the selected category exists, return the book information (title, year, language, weight). (If you write a stored procedure for this part of the code, name it "find_book". It gets the category and the book title, looks for the book, if the book exist it will return title, year, language, and weight.) The query for this part may return more than one book. So, display all the books in a list and let the user choose a book from the list.

If the book with the given title and the category does not exist, the program lets the user choose another category and enter another title.

- If the book exists:

Then, the user selects a book from the result of the previous query (or stored procedure) to buy.

After, the user selects a book to buy, the minimum price for that book will be retrieved from the database (yrb_offer). You need to write a query that returns the minimum price for the book has been offered and display the price to the user. (If you write a stored procedure to find the minimum price name it "min_price". This procedure gets the category, book title, the year and the customer id and returns the minimum price for the book.) Remember, that each customer is a member of a club that offers books at particular prices.

The minimum price will be displayed to the user.

Ask the user to enter the number of books (the quantity) to buy. After, the user enters the quantity, the total price is calculated and displayed to the user (quantity * minimum price)

- If the user approves (ask if the user wants to purchase the book/books?), the purchase information will be stored in the purchase table with current date and time. You need to write a query to insert the purchase into the database. (If you write a stored procedure, name it "insert_purchase", it gets the purchase information (cid, club, title, year, quantity) and adds a new tuple(s) for the new purchase in the yrb_purchase table).

The name of the stored procedures must be exactly the same as the names given in the project description.

You have two options in your implementation:

- Write queries/updates in java code with the same format and returning the columns as described above.
- Or, write stored procedures and call them in your code. (The advantage of having stored procedures is that in case your program does not work properly or does not get compiled you may get partial mark for the project.)

Deliverables

Due by Wednesday, April 5, 11:59pm.

When you are finished, hand in the following.

1. **hard copy**

- A. Your source program.
- B. A recorded session of executing your program.

This should include a variety of scenarios to demonstrate that your program can handle different situations. Explain each of the scenarios in a sentence or two.

Use standard size (8×11) paper and stapled together. **Do not** use binders, folders, paper clips etc; just pure standard size paper.

2. **electronic copy (same content as hard copy)**

- Use the submit command to turn in your work. The directory is a3.

Grading

Marking Scheme (Total is 10 points.)

- a. Requirements working: 8 points
- b. Interface design: 2 points

If your program does not compile, you may get at most 3 points for the project.