Introduction to Database Systems

EECS3421-B

York University

Fall 2019

Project 4

JAVA Console Application

In this project, you will work again with the same database, the **YRB** database from project 3.

In this project, you need to some constraints/triggers and a **JAVA** (***console***) application program for the **YRB** (do not use GUIs in your program). Your application will modify the **YRB** in your own copy of **YRB** database.

For projects, you are permitted to confer with others, seek advice, and (to a reasonable extent) help. However, remember that copying someone else's queries and claiming them as your own work is plagiarism. You must do your own work.

Part A (Constraints and Triggers)

- When a book is not offered by YRB online bookstore, we do not want to have the book in the available offers. Write a constraint or a trigger to ensure that when a book is removed from our database, all relevant offers of that book will also be deleted. (2 points)
    - Write a sample query (delete) to test your constraint/trigger. The query violates the constraint/trigger.

- We want to ensure that only members of a club can purchase the books offered by that club. Make a change to the relational schema to ensure this. If a customer wants to buy a book offered from a club which is not the customer club, the insertion of that purchase will be rejected. (2 points)
    - Write a sample query (insert) to test your constraint/trigger. The query violates the constraint/trigger.

- We know that every customer is a member of at least the club ***basic***. We want to make sure that when a new customer is added to the database, the customer becomes a member of the club ***basic***. Write a trigger to ensure that. (2 points)
    - Write a sample query (insert) to test your constraint/trigger. The query violates the constraint/trigger.

Include your constraints/triggers definition and the sample queries to test them in a text file named const_trig.sql.

Part B (The Application)

Write an application program in Java to do the following. Go to JDBC on PRISM at EECS York and check out the instructions on how to write and compile SQL application programs.

YRB Online Bookstore application:

- Display the following title:

  ********** YRB Online Bookstore **********

Fetch customer information (2 points)

- Prompt the user to enter a customer number.

  Customer Id: ##

- The program starts by finding a customer, that is, it looks for a customer to see if the customer with the 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) and asks the user whether he/she would like to update the customer information.

  Update the customer information (2 points)

  Display the following message:

  ```
  Would you like to update the customer information? (Y/N)
  ```

  If the user enters ‘y’ or ‘Y’, asks the new values for the following customer’s information:

  ```
  Customer Name:

  Customer City:
  ```

  Store the new values of name and city for the given customer.

  If the user enters ‘n’ or ‘N’, go to the next step.

Display and select Categories (2 points)

- If the customer exists:

  Write a query in your program to return all the categories (cat) in your database. Number the categories from $1 \ldots n$ and ask the user to enter a number representing a category.

```
************* Book Categories *************

  1.  children
  2.  cooking
  3.  drama
  4.  guide
  5.  history
  6.  horror
  7.  humor
  8.  mystery
  9.  phil
 10.  romance
 11.  science
 12.  travel

Choose a category: 10

Category romance is selected.
```

Display and select books (2 points)

- 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). 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.

  Title: I don't think so

```
TITLE                     YEAR   LANGUAGE   CAT        WEIGHT
------------------------- ------ ---------- ---------- ------
I don't think so       1997  English    romance       139
```

  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:

  The user selects a book from the result of the previous query to buy.

```
Select a book to purchase: 1
```

Display the price and calculate the total price (2 points)

After, the user selects a book to buy, the minimum price for that book will be retrieved from the database. You need to write a query that returns the minimum price for the book that has been offered and display the price to the user. 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)

- Ask the user to confirm the purchase:

```
Would you like to purchase the book/books? (Y/N)
```

  - If the user approves (the user enters 'y' or 'Y'), the purchase information will be stored in the purchase table with the current date and time. Display:

```
Thank you for your purchase.
```

  - If the user enters 'n' or 'N', program terminates. Display:

```
Would you like to continue? (Y/N)
```

    - If yes, let the user choose a category and books.
    - If no, terminate the program and display.

```
Good bye!
```

Note: The user should be able to exit the program in any step if they do not want to continue with their purchase.

Marking Scheme

1) Report (If your program works properly) (3 points)
2) Constraints/Triggers (6 points)
3) Application (Working properly and error free) (10 points)
4) Readability of the program (If your program works properly): 1 points
   Your code should be organized, clean, and commented.

Projects with compilation errors will get zero. Make sure your code executes successfully before submitting it.

Deliverable

1. **Hard Copy**
   A. A printed version of your constraints and triggers.
   B. A project report (***project4.pdf***). The report should include a user manual of your program, tools used in the project, some screenshots of the 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. Last, you should explain how a user should set up their program on a system (server). (If we follow your instructions and cannot install your program, you get no credit for it!)

2. **Electronic Copy**
   - Your source program.
   - The file ***const_trig.sql*** containing the SQL commands and statements to create constraint and triggers from Part A.
   - Your project report ***project4.pdf***.

Submit

```
% submit 3421B p4a const_trig.sql
% submit 3421B p4b your_source_codes project4.pdf
```

Online Submission Due: by *10:00am Friday 29 November 2019*.

In Class Submission Due: by *10:00am Friday 29 November 2019.*

Your project report should include the following.

- **Cover Page**
  A *cover page* should have your name and student#, and should indicate it is for the *Application project* of **EECS-3421B** for *Fall 2019*.
- **Documentation**
  Any clarifications about your submission.

Your project report must be *typeset*; that is, no *hand-writing* submission. The cover page for submitting your work should look something as follows.

Student#:	
Sur (Family) Name:	
Given Name:	
Class:	EECS-3421B
Term:	Fall 2019
Project:	Application