

CS430/630 – Homework 4

Released Nov 23, Due Dec 04

Instructions: The homework is due on Fri, Dec 4th, 23:59:59. Submission is done in electronic format, create a folder HW4 in your course directory and place source code and compilation scripts there. Recall to set the proper permissions for files!

Question 1 (50 points) - JDBC

You must create a JDBC application for managing the orders of customers from a shop. The schema for this is as follows:

```
Products(pid: integer, name: string, price: integer)
Sales(pid: integer, cid: integer, quantity: integer)
Customers(cid: integer, name: string, budget: integer)
```

`Products` will store the information of a given product which the shop sells including its ID, name, and price. The `Customers` table will store information on a customer including their ID and name. This table will also store the budget that the customer has available to spend at this shop. The budget CAN be a negative number! Think of it as a line of credit taken from the shop. The `Sales` table will store information on sales made at the shop including the product id of the product sold, the customer id to whom it was sold, and the quantity sold to that customer. The quantity must be non-negative but it can be 0!

You must create the above schema definition in your submission in a file called `schema.sql`. You are allowed flexibility on the exact attribute types you use for your schema, given they reasonably match the specification above (e.g. in terms of number types, string types). Also, you must create a Java JDBC-based application run by customers, with name `Shop.java`. The application must have a command-line interface menu that allows the user to select one option as described below. Once that menu function is completed, the program must return results if applicable and then return to the main menu. For each menu option, you are allowed (and even recommended, if needed) to have multiple steps (or “screens”) to complete the tasks. You will use the DBS3 Oracle instance as DBMS.

Customer Menu

The application starts by requesting the customer ID. Verify that this is a valid customer id, otherwise display an error and allow the user to try again. The remaining session assumes that customer is active. If the value “-1” is given, the user should be prompted with all the necessary information to create a new customer. The main menu of the program is as follows:

P – Products: Lists all products in products table

O – Order: Order a product given a product id and quantity. Each customer can only order each product once so make sure that there is no duplication. *Subtract the total cost of this order from the budget of the active customer.*

R – Return: Given a product id, return that product to the shop with a quantity of 1. Ensure that the customer has ordered this product before and that the quantity ordered in the sales table will not drop below 0 due to this return. *Must add the cost of this product back to the customer’s budget.*

S – Product Search: Searches the products table given a substring of the product name

E – Expenditure: Lists all orders the current customer has made. It should display the product ID, product name, and the total cost of the order (quantity * price).

C – Current Budget: List the budget for the current customer.

X – Exit: Exit application