CS4092 Project Description
Online Shopping Application

March 17, 2024

1 Project Overview

In this project, each group builds its own online shopping application using a PostgreSQL database backend to store information about products, their availability, orders of customers, and staff information for the store.

1.1 Project outputs

The project has three deliverables.

1.1.1 ER-model

Each group should develop an ER model for the application. This should be submitted in jpeg, png, or pdf.

1.1.2 Relational schema

The second deliverable is a translation of the ER model into a relational schema implemented as an SQL script. The script should use Postgres SQL dialect and should execute without errors. Besides from defining tables and constraints, this script should create indices where appropriate. Please submit the script as a simple sql file.

1.1.3 Application

The last deliverable is a online shopping application that uses the relational schema defined in the first two deliverables. This application can be either a web or desktop application.

1.2 Submission

All deliverables, i.e., the codes, ER model, and SQL scripts, must be submitted to the **group repository**. The repository should have **three separate directories** for each submission. In addition to this, each group should submit a **5-minute recorded demo video** demonstrating their application by the end of the semester. Some of the requirements are optional for bonus points. **Every member of the group has to contribute**

in each phase of the project and the project will be graded based on individual contribution and on the overall project result.

1.3 Timeline

As shown in the syllabus in Canvas, the deadlines are as follows:

- ER model due: April 4 at 11:59 pm
- Database design (SQL scripts) due: April 4 at 11:59 pm
- Implementation due: April 18 at 11:59 pm

2 Database Requirements

2.1 Users

The application should support two types of users.

- **Customers**: Customers of the store can search for products and look up information about products, set up an account and change their preferences and account details, order products, and make payments. For each customer, the application should record the `name` of the customer, one or more `addresses` (address for delivery and/or for payment), and `credit card` information. A customer can have multiple credit cards and each credit card should be associated with a payment address. For each customer, the database should maintain the current `balance` of the customer's account, i.e., the dollar amount of outstanding payments for the customer.
- **Staff Members**: Staff of the store can modify and create products, update the availability of products in the stock, query customer information, and process orders. The database stores `name`, `address`, `salary`, and `job title`.

2.2 Online store information

The database should record information about customers, orders, items, and stock.

- **Product**: A product is an item that the store sells, e.g., shoes, an apple, etc. Products are of a certain category, e.g., apparel, food, etc. Additional information should be stored for each product. For example, all products have types, their brands, sizes, and short description of the product.
- **Warehouse**: The store has multiple warehouses. Each warehouse is located at a certain address.
- **Stock**: The database should record how many items of each products are currently stored in which warehouse.
- **Product price**: The database should maintain the prices for products.

- **Orders**: A customer can order items from the online store. An order consists of products, each with an associated quantity. For example, a customer may order one pair of shoes and six apples. Each order should be recorded when it is issued with its status (issued, sent, and received) and which credit card was used to pay for the order.
- **Delivery plan**: Each order has its own delivery plan. The database should maintain the delivery type (express or standard), delivery price, delivery date, and ship date.

2.3 Bonus points

2.3.1 Suppliers

The database stores information about suppliers. Each supplier has an address and a name. A supplier sells some items (not necessarily all items that exist) for a supplier-specific price.

2.3.2 Warehouse capacity

Each warehouse has its own size that is used when a new stock of products is added.

3 Application Requirements

The application should support the following actions. Staff and/or customors should be able to execute each corresponding action.

- Customer
  - Search and browse products, add products into a shopping cart, and place an order
  - Add, delete, and modify a credit card and address
- Staff
  - Add, delete, and modify a product and its price
  - Add stock to a warehouse

3.1 Searching for products and placing orders

Customers can search for available products and browse through the catalog of such products grouped by product type. The application should maintain a shopping cart for a customer that stores the items the customer has selected so far and their quantity. Customers can add and delete items from the shopping cart and change an items quantity. Once a customer is satisfied with the shopping cart content, they can submit an order to order all the items that are currently in their shopping cart. A default delivery plan should be standard. The customer can make a change it to express if needed. For each order, the customer can select one of the existing payment methods (credit cards). Once

an order has been placed, the total cost of the order is added to the customer's account balance. Furthermore, the available quantity of products in the warehouse should be reduced accordingly.

3.2 Add/Delete/Modify a credit card and addresses

Customers can add, delete, and modify credit cards associated with their account. Each credit card has an associated payment address. Customers can also add, modify, and delete addresses.

3.3 Add/Delete/Modify a products and its price

Staff members can add, delete, and modify products and set product prices.

3.4 Manage stock

Staff members can add products to individual warehouses.

3.5 Bonus points

3.5.1 Check availability

When an order is submitted, check whether enough products are available in the warehouse to fullfil the order.

3.5.2 Product images

The application provides images for products.

3.5.3 Check storage limits

When new stock is added to a warehouse, check that the total size of all products stored in this warehouse does not exceed the size of the warehouse.