CS 425 Database Organization - Summer 2022
Course Project: An Online Grocery Store*

---

1 Project Timeline

The project has three deliverables. Deadlines are announced on the course Blackboard and the class. Each group will demo their application at the end of the semester. The deliverables are:

- ER-model: Each group should develop an ER-model for the application. This can be uploaded as any type of image file (please do not use esoteric formats).
- Relational schema: The second deliverable is a translation of the ER-model into a relational schema implemented as an SQL script. Besides from defining tables and constraints, this script should create indexes where appropriate. Please upload the script as a simple text file.
- Application: The last deliverable is an online grocery management application that uses the relational schema defined in the first two deliverables. This application can be either a web or desktop application.

~~Some of the requirements are marked as optional **bonus** requirements. You are free to not realize these requirements, but you can get extra credits by implementing them.~~

Every member of the group has to contribute in each phase of the project and you will be graded based on your individual contribution and on the overall project result.

2 Overview

The goal is to build an US online grocery store application using a database backend to store information about products, availability of products in the stock, and customers of the store. The application should support two types of users. Customers of the store can search for products and look up information about products, setup an account and change their preferences and account details, order products, and make payments. Staff of the store

*Credit: Boris Glavic, IIT

can modify and create products, update the availability of products in the stock, query customer information, and process orders

3 Data Requirements

3.1 Users

We will consider two types of users:

- **Customers**: Customers of the store. For each customer we should record the **name** of the customer, one or more **addresses** (addresses can be delivery and/or payment addresses), and **credit card** information. A customer can have multiple credit cards and for each credit card we associate it with a payment address. For each customer we also maintain the current **balance** of the customers account, i.e., the dollar amount of outstanding payments for the customer.
- **Staff Member**: Staff of the store. We store **name**, **address**, **salary**, and **job title**.

3.2 Store information

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

- **Product**: A product is an item that can be brought by customers of the store, e.g., a banana or a beef steak. Products are of a certain category, e.g., food, drinks,.... Additional information should be stored for each product based on its type. For example, all products have types, but nutrition information is only stored for food and alcohol content is only stored for alcoholic drinks. All products have a size in cubic feet.
- **Warehouse**: The grocery store has multiple warehouses. Each warehouse is located at a certain address and has a storage capacity (in cubic feet).
- **Stock**: The database should record how many items of each products are currently stored in which warehouse.
- **Product pricing**: The grocery store maintains prices for products that are set per state. The price a customer is paying for a product is set based on the state of their delivery address.

- **Orders**: A customer can order items from the store. An order consists of products, each with an associated quantity. For example, customer Peter may order 3 bananas, and 5 six-packs of beer. For an order we should record when it was issued, what its status is (issued, send, received), and which credit card was used to pay for the order.

3.2.1 Suppliers

- Also store information about suppliers in the database. Each supplier has an address and a name. A supplier sells some items (not necessary all items that exist) for a supplier specific price.

4 Application Requirements

The application should support the following actions. We indicate for each action whether it can be executed by staff and/or customer.

- Search and browse products, put products into a shopping cart, and place an order (`customer`)
- Add/Delete/Modify a credit cards and addresses (`customer`)
- Add/Delete/Modify a product and product pricing (`staff`)
- Add stock to a warehouse (`staff`)

4.1 Searching for products and placing orders

Customers can search for products available in their home state (delivery address) 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. 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 in the customers state should be reduced accordingly.

4.1.1 Check availability

When an order is submitted check that enough products are available in the warehouse in the customers home state to fulfill the order.

4.1.2 Save shopping carts between sessions

Enables the user to continue with their shopping cart when the log out and in again.

4.2 Add/Delete/Modify a credit card and addresses

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

4.3 Add/Delete/Modify a products and product pricing

Staff members can add/delete/modify products and set product prices per state, e.g., price of bananas in Illinois.

4.3.1 Product images

Also show images for products.

4.4 Manage stock

Staff members can add products to individual warehouses.

4.4.1 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.