CSC372 Web Development
Term Project Milestone 3: Schema
Due: Tuesday, April 1st, 2025

Introduction

The next step to making our mock-ups dynamic is to develop a database that will power the UI we have designed. Have a solid database that is flexible enough for unexpected new requirements but also quick to respond to queries is critical in web-based systems. In this milestone, you will create a schema diagram for the ecommerce database. You will also write files containing the SQL statements necessary to create and drop the database tables and constraints used by your web application.

Your schema will include details about the following tables:

- Users
- Categories
- Products
- Carts
- CartProducts

In addition to these tables, other tables may need to be created to support the product details page that you mocked up. For example, if your site runs promotions on products, you may need a table to handle them.

Specifications

Users.

The users table holds information about the users in our system. Users can be admins or shoppers. A user can only have one shopping cart. The users table has the following required fields:

- An identity column that contains an id that is automatically assigned.
- The date and time the user record was or modified.
- The user's name, email and password.
- The type of user (admin or shopper).

Categories.

The categories table contains a list of categories that will be shown in the menu and is used to categorize products. The categories table has the following required fields:

- An identity column that contains an id that is automatically assigned.
- A name column containing the name of the category.
- Any other column that would help arrange your site, for example if categories have different priority levels.

Products.

The products table contains all the details about the product. The products table has the following fields, none of which can be null except for some of the product details if they only exist for a subset of your products:

- An identity column that contains an id that is automatically assigned.
- The name, description, image url, price and other details about your product. Each field that is in your product details prototype will need a column to contain that data.
- The category id that the product belongs to.
- Any other column that would help arrange your products, for example if it is a featured/promoted product.

Carts.

The carts table is used to track the details of the user's cart. It should contain the following columns:

- An identity column that contains an id that is automatically assigned.
- A status column to track if the cart is new, abandoned or purchased.
- The date the cart was created.
- The user id for the user that owns the cart.

CartProducts.

The CartProducts table contains a list of all the products that are in the cart. It should contain the following columns:

- An identity column that contains an id that is automatically assigned.
- An id for the cart the item belongs to.
- The id of the product in the cart.
- The quantity of that item.

Technical Requirements.

- Schema Diagram
    - Create a schema diagram of your tables and how they relate to each other. Save it as an image file and include it in your code base.
- SQL: Create the SQL statements necessary to start your application up.
    - create_tables.sql and drop_tables.sql containing the SQL statements necessary to create and drop your tables, respectively. Observe the following guidelines when constructing your SQL statements.
        - Use the plural form of an entity's name for its table (i.e. a users table holds user records).
        - Each table must have a primary key defined on it.
        - Columns whose values are required should have "not null" constraints declared on them.
        - Columns containing unique values (other than a primary key) should have a unique constraint declared on it.
        - Two tables connected by a relationship should have an appropriate foreign key declared between their connecting columns. Generally, the table containing entities that "belong to" entities in the other table has the foreign key defined on it.
        - Note that the order of the create table and drop table statements will be important. For instance, a table on which a foreign key constraint is to be defined needs to be created before the table that declares the constraint. Similarly, a table defining a foreign key constraint must be dropped before the table against which the constraint is declared.
    - insert_categories.sql: Add the SQL statements necessary to create several default product categories.
    - insert_products.sql: Add the SQL statements necessary to create a few products for starting the application. The product categories should refer to categories that have been created by the query mentioned above.

Your SQL files must execute correctly in SQLite and without errors or warnings when run against a database. Add the .sql files to your code base.