

CSC365 Final Project Description

Introduction

Our project emulates a Bicycle E-commerce store that utilizes a MySQL database to manage multiple tables that hold information that is vital for the store to operate. Our tables manage Customers, Bicycles, Manufacturers, Order Information, and different Item Orders. There are also auxiliary tables that support the main functionality of the previously listed tables (Address, State, Order Status). In total there are 9 tables being managed through our frontend implementation, which supports all of the necessary functionality to insert, update, and remove tuples from each table.

Connection to our Database

Our project relies heavily on the java.sql.jdbc package to connect with our database. First a connection is made in our DBConnection file, which outputs a Connection Object to the rest of our program. The DBCreating file searches our existing database for the tables necessary for the operation of our program, and if it does not detect the tables, it executes the appropriate create table statements to ensure that the tables exist.

Main Menu

The main menu consists of a image resource with a toolbar on the left side. This toolbar allows the user to navigate through the different functionality of the application. We used JButtons in the JavaSwing library to redirect users based on a mouse click action.

Manufacturer

The Manufacturer menu consists of a table page that is populated with the contents of the Manufacturer table, as well as two text fields that allow the user to Add, Update, and Delete from the Manufacturer table. Since the Manufacturer table does not contain any foreign keys, inserts to this table can be done independently from other tables. The text in these text fields are used as inputs for the user-specified action. Each of the buttons triggered a function that performed their respective action in SQL.

Category

The Category table refers to the different Categories of Bicycles supported by our store. This may range from bikes like Mountain Bikes, Road Bikes, etc. The page itself consists of a table that displays all of the tuples in the Category table, as well as Add, Update, and Delete buttons that perform their respective actions using the two text fields for Category ID and Category Name. The text in these text fields are used as inputs for the user-specified action. Each of the buttons triggered a function that performed their respective action in SQL.

Bicycle

The Bicycle table page consists of a table that lists all of the different kinds of Bicycles sold by our store. These tuples are attained using a SELECT query. Like the other previous pages, there are buttons for Add, Update, and Delete. These buttons trigger functions in the background that will execute the appropriate DDL statements to meet the user's desired functionality.

Customer

The Customer page has a table that is populated with a select query to our Customer table. Below that table is a menu of text fields to be filled by the user. In the menu, there is a drop down menu of different States, which is filled by our States table. The user's specific choice of State is used in the DDL statements for each of the Add, Update, or Delete buttons. The object that we used for this dropdown menu is a JComboBox.

Order

The Order page is different than the other previous pages, as it contains a table that displays all of the past orders, sorted by most recent Date, as well as another table that displays all of the Item orders made that pertains to a particular Order_info tuple. We utilized multiple joins in order to achieve this bottom table. There are also text fields that are made for the user to Update or Delete orders. There are certain restrictions for the Delete option, where a user is restricted from deleting a tuple if the Order Status is set to "Done".

Add Order

The Add Order page is used to add Orders to the Order_info table and the Item_order table. There are two tables presented to the user, the Customer table and the Bicycle table. The user is prompted to select a tuple from both tables, and specify the quantity of Bicycle to order. Then a parallel insert is executed such that integrity constraints are maintained. The function that is called by the Add button pulls the necessary information from the tuples that the user selected in the Customer and Bicycle tables, and uses that information for the DDL statements. When a order is made, a trigger is used to update/decrement the stock of the bike that was just ordered.

Sales Report

Finally there is the Sales Report page, where multiple complex queries are being used to fill tables based on which query is specified. An example of these queries are Top 10 Buyers with the Highest Total Purchase and Customers who bought bicycles in all Categories and Manufacturers. These complex queries utilize nested queries, views, and division. It's purpose is to display what we have learned for this course, as well as to highlight the different types of queries that could be made for our database.