CPSC 2221 - Database Systems

Group Project - Implementation of a Relational Database

Project Title:	Online Sport Shoe Store Database
Project Milestone:	4

#	Student Name	Student ID	Email Address
1	Hok Nam Ko	100399595	hko07@mylangara.ca
2	Bryan Tompkins	100384247	btompkins00@mylangara.ca
3	Isabelle Wang	100396670	iwang11@mylangara.ca
4	Jerry Au	100390444	wau05@mylangara.ca

By keying our names and student IDs in the above table, we certify that the work submitted with this cover page was performed solely by those whose names and student IDs are included above.

Also, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Langara College.

Project SetUp Guide

Setup guide (We'll be assuming the marker has PHPMyAdmin + Xampp Setup already):

Unzip Files:

- Unzip the files that our group has uploaded.

Database Setup:

- Run the "Create Table Script" to create the necessary tables in the database.
- Run the "Insert Table Script" to populate the tables with initial data.

Web Application Setup:

- Drag and drop your HTML and PHP files to the htdocs directory of the XAMPP folder.
- Change the values of connect.php $dbpass, $dbuser, and $db to your local values. (refer to image)

Start XAMPP:

- Ensure that XAMPP is installed and running on your local machine.

Access Localhost:

- Open a web browser and go to http://localhost to access the locally hosted web application.

Project Accomplishment

We built an online shoe store database that stores information about product, stock, order, shopping cart, customer, staff, and shipping.
Users of this database will be able to enjoy the following features:

1. **Insert Feature:**
   Insert a product to 2 tables
   insert product stock to 3 tables
   insert a customer/a staff
   insert a shipping company

2. **Display Feature:**
   Display all product;
   Display all orders;
   Display customers who made all purchases

3. **Search Feature:**
   Search for a ProductID based on product name
   Search for StockID and ProductID with price greater than 150
   search for orders bought by a certain customer
   Search for ProductID with a certain size
   search for shipping time based on order_id

4. **Update Feature**
   Update a stock price
   Update stock count
   Update customer info
   Update staff info

5. **Delete Feature Done**
   Delete a product by specifying productID
   Delete an order, customer, staff

6. **Calculation Feature**
   Show count of total product sold
   Show the average price of total product
   Show the total number of shoes bought by each customer

Scripts/Queries used in all php files

Queries for inserting feature

Insert a product to 2 tables

```
$query1 = "INSERT INTO product VALUES ('$Product_ID', '$Product_Name')";
$query2 = "INSERT INTO product_sub VALUES ('$Product_Name', '$Description',
'$Category')";
```

insert product stock to 3 tables

```
$query1 = "INSERT INTO stock_count
VALUES('$Stock_ID','$Stock_Count','$Product_ID')";
$query2 = "INSERT INTO stock_color_size
VALUES('$Stock_ID','$Pro_Color','$Pro_Size')";
$query3 = "INSERT INTO stock_price VALUES('$Stock_ID', '$Product_ID',
'$Pro_Price')";
```

insert a customer/a staff

```
$query = "INSERT INTO customer
VALUES('$Customer_ID','$First_Name','$Last_Name','$Address','$Phone_Number','$Shipping_
Address')";
```

insert a shipping company

```
$query = "INSERT INTO shipping_company
VALUES('$Company_ID','$Comp_Name','$Comp_Address','$Phone_Number')";
```

Queries for display feature

Display all product;

```
SELECT * FROM Product
```

Display all orders;

```
SELECT * FROM Pro_Order
```

Display customers who made certain purchases

```
SELECT DISTINCT c.Customer_ID, c.First_Name, c.Last_Name
FROM Customer c
JOIN Pro_Order po ON c.Customer_ID = po.Customer_ID
JOIN Order_Items oi ON po.Order_ID = oi.Order_ID
```

```
WHERE oi.Product_ID IN (
        SELECT Product_ID
        FROM Product
        WHERE Product_Name = '$PName'
)
```

Display customers who made all purchases

```
SELECT c.Customer_ID, c.First_Name, c.Last_Name
    FROM Customer c
    WHERE NOT EXISTS (
      SELECT p.Product_ID
      FROM Product p
      WHERE NOT EXISTS (
        SELECT oi.Product_ID
        FROM Order_Items oi
        JOIN Pro_Order po ON oi.Order_ID = po.Order_ID
        WHERE po.Customer_ID = c.Customer_ID AND oi.Product_ID =
p.Product_ID
      )
        )
```

Queries for Search feature

Search for a ProductID based on product name

```
$query = "SELECT Product_ID, Product_Name FROM product
WHERE Product_Name = '$input'";
```

Search for StockID and ProductID with price greater than 150

```
$query = "SELECT Stock_ID, Product_ID, Pro_Price FROM stock_price
WHERE Pro_Price > $input";
```

search for orders bought by a certain customer

```
$query = "SELECT PO.*, C.First_Name FROM pro_order PO, customer C
WHERE PO.Customer_ID = C.Customer_ID
AND C.First_Name = '$input'";
```

Search for ProductID with a certain size

```
$query = "SELECT SC.Product_ID, SCZ.Pro_Size FROM stock_color_size SCZ,
stock_count SC
WHERE SCZ.Stock_ID = SC.Stock_ID
AND SCZ.Pro_Size = $input";
```

search for shipping time based on order_id

```
$query = "SELECT Estimated_Shipping_Time FROM shipping_order
WHERE Order_ID = $input";
```

Queries for update feature

Update a stock price

```
$query = "UPDATE Stock_Price SET Stock_Price = $Stock_Price WHERE
Stock_ID = $Stock_ID";
```

Update stock count

```
$query = "UPDATE Stock_Count SET Stock_Count = $Stock_Count WHERE
Stock_ID = $Stock_ID";
```

Update customer info

```
$query = "UPDATE customer SET First_Name = $First_Name WHERE
Customer_ID = $Customer_ID";
$query1 = "UPDATE customer SET Last_Name = $Last_Name WHERE
Customer_ID = $Customer_ID";
$query2 = "UPDATE customer SET Address = $Customer_Address WHERE
Customer_ID = $Customer_ID";
$query3 = "UPDATE customer SET Phone_Number = $Phone_Number WHERE
Customer_ID = $Customer_ID";
$query4 = "UPDATE customer SET Shipping_Address = $Shipping_Address
WHERE Customer_ID = $Customer_ID";
```

Update staff info

```
$query = "UPDATE staff SET First_Name = $First_Name WHERE Staff_ID =
$Staff_ID";
$query1 = "UPDATE staff SET Last_Name = $Last_Name WHERE Staff_ID =
$Staff_ID";
$query2 = "UPDATE staff SET Address = $Staff_Address WHERE Staff_ID =
$Staff_ID";
$query3 = "UPDATE staff SET Phone_Number = $Phone_Number WHERE
Staff_ID = $Staff_ID";
$query4 = "UPDATE staff SET Role = $Role WHERE Staff_ID = $Staff_ID";
```

Queries for delete feature

Delete a product by specifying productID

```
$query = "DELETE FROM product WHERE Product_ID = $Product_ID";
```

Delete an order

```
$query1 = "DELETE FROM order_items WHERE Order_ID = $Order_ID";
$query2 = "DELETE FROM pro_order WHERE Order_ID = $Order_ID";
```

Delete customer

```
$query = "DELETE FROM customer WHERE Customer_ID = $Customer_ID";
```

Delete staff

```
$query = "DELETE FROM customer WHERE Staff_ID = $Staff_ID";
```

Queries for Calculation feature

Show count of total product sold

```
SELECT po.Order_ID, po.Order_Total_Price, COUNT(oi.Product_ID) AS ItemCount
FROM Pro_Order po
LEFT JOIN Order_Items oi ON po.Order_ID = oi.Order_ID
WHERE po.Order_Status = 'delivered'
GROUP BY po.Order_ID, po.Order_Total_Price
```

Show the average price of total product

```
SELECT AVG(Pro_Price) AS AveragePrice FROM Stock_Price
```

Show the total number of shoes bought by each customer

```
SELECT c.Customer_ID, c.First_Name, c.Last_Name,
COUNT(oi.Product_ID) AS TotalShoesPurchased
FROM Customer c
JOIN Pro_Order po ON c.Customer_ID = po.Customer_ID
JOIN Order_Items oi ON po.Order_ID = oi.Order_ID
GROUP BY c.Customer_ID, c.First_Name, c.Last_Name
```