F27WD Web Design and Databases
Coursework 3

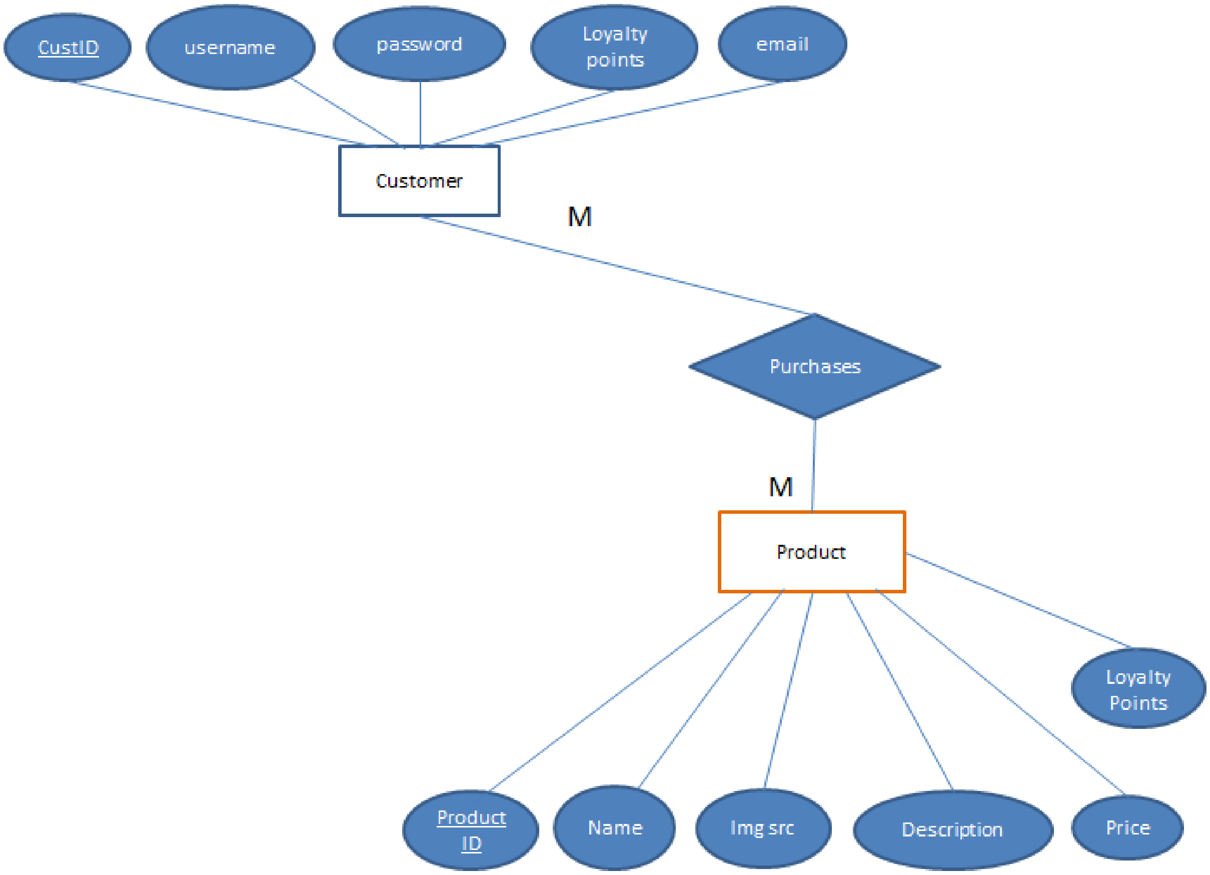
Team name: teeshop

Team members: Naqeeb Mohammed Naseer
Christian Chester Corpuz

TEESHOP v2.0

The second iteration of Teeshop implements the same simplistic design elements and layout as the first one but comes with a major update. Teeshop now employs dynamic pages by the use of PHP and MySQL. This allows for the elimination of duplicate pages and also a register/ log-in facility.

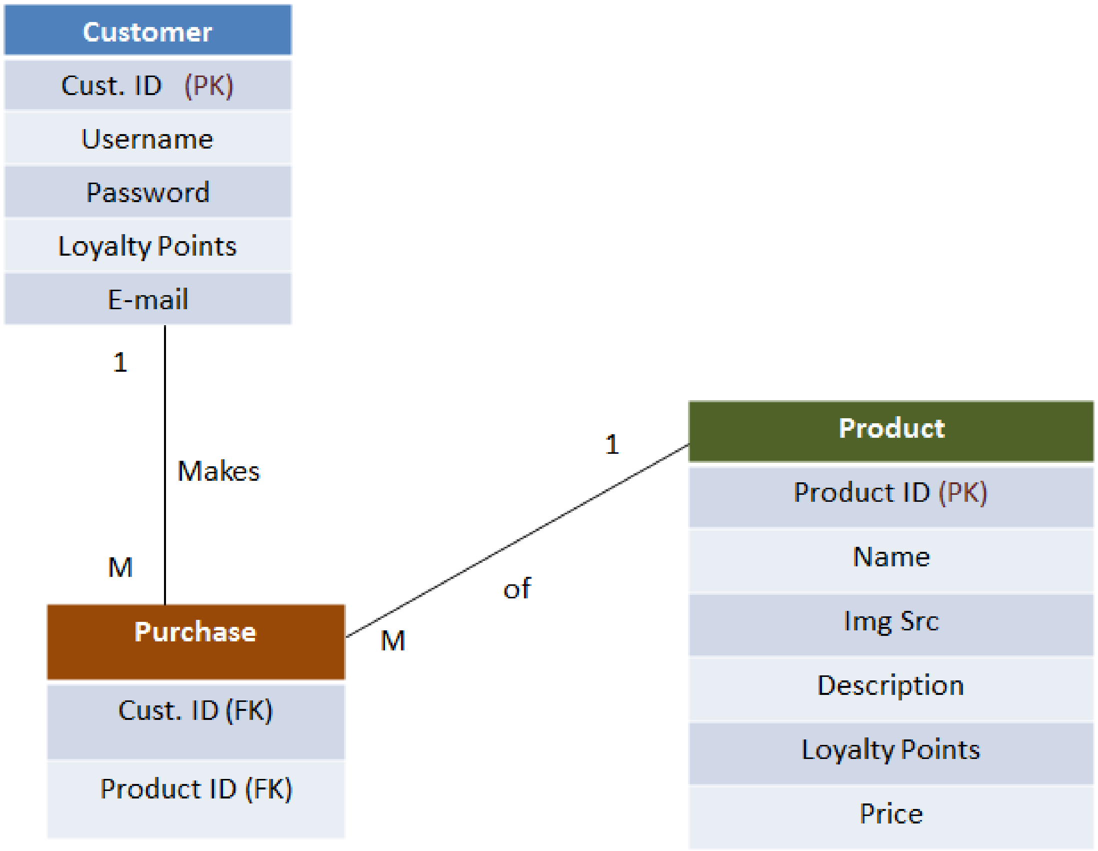
ENTITY RELATIONSHIP DIAGRAM



The ER diagram for teeshop consists of two tables who share a Many-Many relationship with each other. The Customer table, whose primary key is CustID, holds information of the registered customers. This table is used to validate a login as well as to discount the price if a customer chooses to use his/her loyalty points.

The Product table, whose primary key is ProductID, holds information regarding products for sale. This table allows for the products in the product page of the website to be dynamically generated rather than being hard coded as list items. Moreover, this table also helps in dynamically generating the product description page thereby eliminating the need to create duplicate description pages for each product.

DATABASE DESIGN

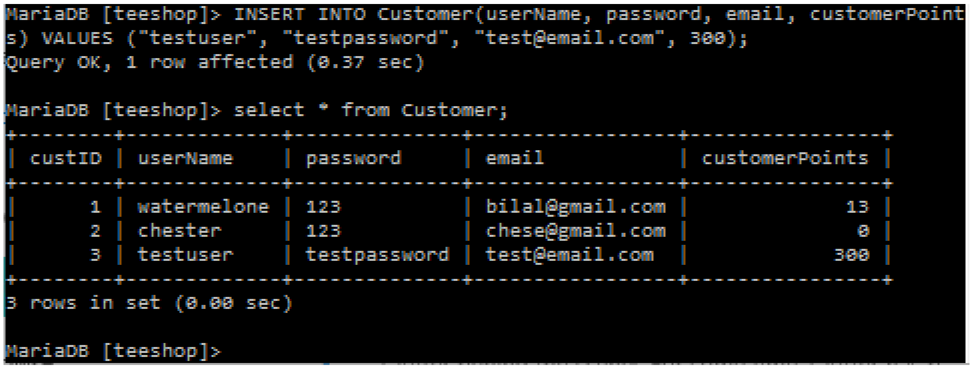


The database design implements the same tables seen in the ER diagram but changes the Many-Many relationship to a One-Many / Many-One relationship by introducing a third table ‘Purchase’ which uses CustID from the Customer table and ProductID from the Product table as foreign keys. The Purchase table is used to display the recent purchases on the user’s profile page.

SAMPLE QUERIES AND OUTPUTS

The following MySQL queries are core to the functioning of Teeshop and its dynamic elements (with sample outputs using test data):

1. INSERT INTO Customer (userName, password, email, customerPoints) VALUES ('$_POST[username]','$_POST[password]','$_POST[email]', '0');
   This query is made when the user clicks the submit button after filling the form on the Register page. This is done to make records of new users into the Customer table.



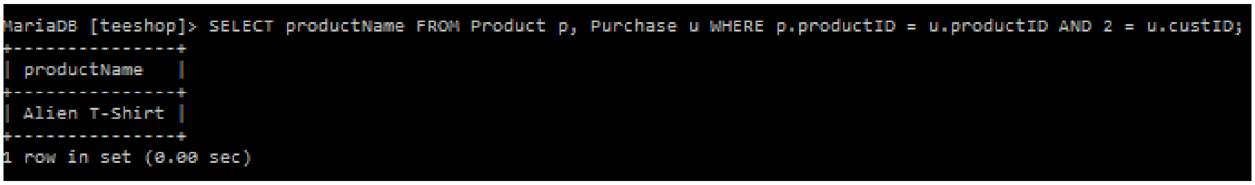
```
MariaDB [teeshop]> INSERT INTO Customer(userName, password, email, customerPoint
s) VALUES ("testuser", "testpassword", "test@email.com", 300);
Query OK, 1 row affected (0.37 sec)

MariaDB [teeshop]> select * from Customer;
+--------+-------------+--------------+-----------------+----------------+
| custID | userName    | password     | email           | customerPoints |
+--------+-------------+--------------+-----------------+----------------+
|      1 | watermelone | 123          | bilal@gmail.com |             13 |
|      2 | chester     | 123          | chese@gmail.com |              0 |
|      3 | testuser    | testpassword | test@email.com  |            300 |
+--------+-------------+--------------+-----------------+----------------+
3 rows in set (0.00 sec)

MariaDB [teeshop]>
```

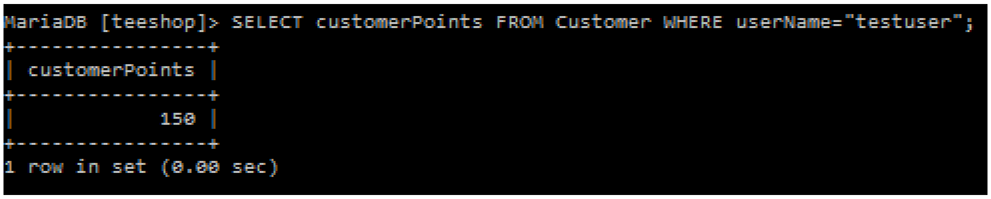
2. SELECT * FROM Product WHERE productName = \"$productID\";
   This query is made to produce a dynamically generated Product Description page. Depending on the shirt the user clicks the Description pages changes details accordingly. This shirt's id is stored in $productID.
   N.B. Sample output not provided due to the lack of readability of the Product table when displayed on the MySQL shell.

3. SELECT productName FROM Product p, Purchase u WHERE p.productID = u.productID AND $id = u.custID;
   This query is made in order to return the purchase history of a given user, $id being the session variable which holds the value of the current user's id.



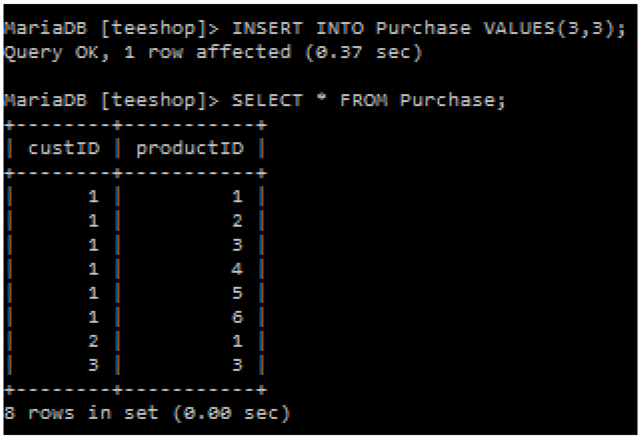
```
MariaDB [teeshop]> SELECT productName FROM Product p, Purchase u WHERE p.productID = u.productID AND 2 = u.custID;
+---------------+
| productName   |
+---------------+
| Alien T-Shirt |
+---------------+
1 row in set (0.00 sec)
```

4. SELECT customerPoints FROM Customer WHERE userName=\"$username\";
   This query is made to return the loyalty points of the current user, $username being the session variable which holds the current user's username.



```
MariaDB [teeshop]> SELECT customerPoints FROM Customer WHERE userName="testuser";
+----------------+
| customerPoints |
+----------------+
|            150 |
+----------------+
1 row in set (0.00 sec)
```

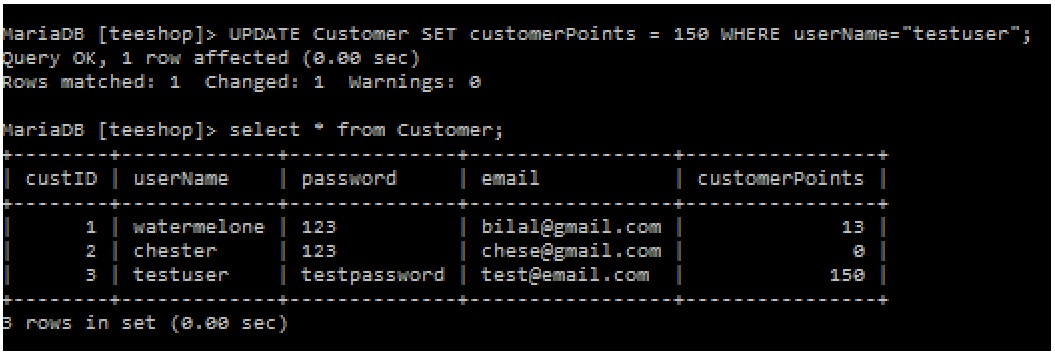
5. INSERT INTO Purchase VALUES ($id, $realID);
   This query is made whenever a user makes a purchase. The product id of the product the user bought is assigned to his/her customer id.



```
MariaDB [teeshop]> INSERT INTO Purchase VALUES(3,3);
Query OK, 1 row affected (0.37 sec)

MariaDB [teeshop]> SELECT * FROM Purchase;
+--------+-----------+
| custID | productID |
+--------+-----------+
|      1 |         1 |
|      1 |         2 |
|      1 |         3 |
|      1 |         4 |
|      1 |         5 |
|      1 |         6 |
|      2 |         1 |
|      3 |         3 |
+--------+-----------+
8 rows in set (0.00 sec)
```

6. UPDATE Customer SET customerPoints=$newpoints WHERE userName=\"$username\";
   This query is used to update the loyalty points of a customer whenever he/she makes a purchase, $newpoints being the new value that is to be updated to the current loyalty points.



```
MariaDB [teeshop]> UPDATE Customer SET customerPoints = 150 WHERE userName="testuser";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [teeshop]> select * from Customer;
+--------+-------------+--------------+-----------------+----------------+
| custID | userName    | password     | email           | customerPoints |
+--------+-------------+--------------+-----------------+----------------+
|      1 | watermelone | 123          | bilal@gmail.com |             13 |
|      2 | chester     | 123          | chese@gmail.com |              0 |
|      3 | testuser    | testpassword | test@email.com  |            150 |
+--------+-------------+--------------+-----------------+----------------+
3 rows in set (0.00 sec)
```

DEV NOTES

Teeshop is not a fully functional website. Certain elements of the website have not been implemented which includes:

1. The About Us section.
2. The sorting by category and price in the Products page.
3. The Contact Us page.
4. The size and color options on Product Description page.

Furthermore,

1. Images used on the website are referenced on the Legal page found in the footer.
2. IA is on the site map page found in the footer.
3. A new button has been implemented on the navbar which either acts as a login button or as a dropdown menu (for the profile page or logout) depending on if a session is running or not.
4. In the recent purchases section of the Profile page, duplicate purchases are not displayed. Only the most recent purchase of the same shirt is displayed.

RESOURCES USED

1. Sublime Text
2. Cssjquery.com coverflow generator
3. XAMPP

REFERENCES

Help for some of the coding elements implemented in Teeshop was received from the following links-

1. Displaying data from a database table into a website using PHP-
   http://forums.htmlhelp.com/index.php?showtopic=13686

2. Sessions and Login tutorial-
   http://www.formget.com/login-form-in-php/

3. Page Redirection-
   http://www.sitepoint.com/forums/showthread.php?586625-Redirect-using-_SERVER-http_referer

4. Displaying form buttons in line with each other-
   http://stackoverflow.com/questions/1266734/display-multiple-form-submit-buttons-inline-same-line-in-ie8-in-table