10/4/2017

Course Work 2: Online Food Ordering Database And Site.

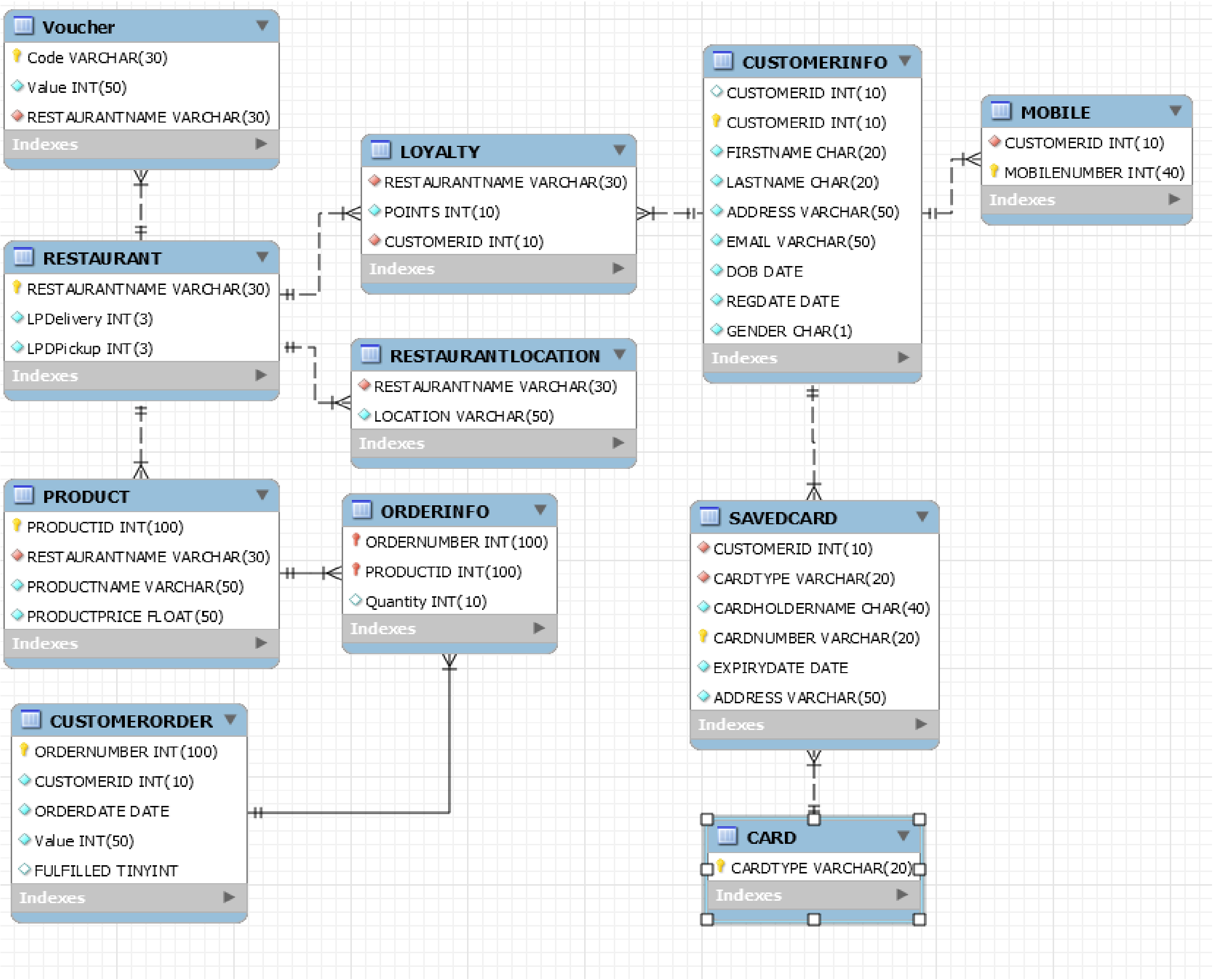
By: Savannah Alphonso (ska7), Ismail Marashi(im31), Ishaq Marashy(im30)

Course Code: F27WD

Course: Web Design And Databases

Instructor: Shaikh Tilal

ER Diagram for Our Database:



In our database, we implemented the queries that allow you to:

1. Update a customer. (Made by Savannah)

2. Add a customer. (Made by Savannah)
3. Add an order to a customer. (Made by Ismail)
4. Add products to a customer’s order. (Made by Ismail)
5. Add credit cards to a customer. (Made by Ismail)
6. Get a customer’s voucher. (Made by Ismail)
7. Apply a voucher to an order. (Made by Ismail)
8. Search for a customer by ID. (Made by Ishaq)
9. Get a customer’s orders by ID. (Made by Ishaq)
10. Get a customer’s account summary by name. (Made by Savannah)
11. Get a customer’s credit cards by ID. (Made by Ismail)
12. Get a customer’s loyalty by name. (Made by Ishaq)
13. Add a restaurant. (Made by Savannah)
14. Add a location to a restaurant. (Made by Ishaq)
15. Add products. (Made by Ismail)
16. Get a restaurants products. (Made by Ishaq)
17. Get find restaurants by their location. (Made by Ishaq)
18. Get the total amount of reward points awarded by a restaurant name. (Made by Ishaq)

To explain how the database works we decided to Clarify these 9 major queries:

1. Get a customer’s account summary by name.
2. Update a customer.
3. Add a restaurant.
4. Get a customer’s orders.
5. Get a customer’s voucher.
6. Apply a voucher to an order.
7. Search for a customer by ID.
8. Add a location to a restaurant.
9. Get the total amount of reward points awarded by a restaurant name.

Get a customer’s account summary by name (Find the address of John Smith):

To do this you would input the information i.e. First and Last name in “CustomerAccount.html,” the action performed, on the click of ‘Submit’, by the form is to redirect the inputs to “DisplayCustomerSummary.php,” where First and Last name where it will be stored as $first and $last., respectfully. It uses the database and selects from the table CUSTOMERINFO, where the FIRSTNAME and LASTNAME match the $first and $last. The system should show all the rows/details of any/all the customers with that name. If there is no such name ($numrows== 0) a print command is executed by the system displaying a message, “No customer with name $first $last.”

Add a new restaurant:

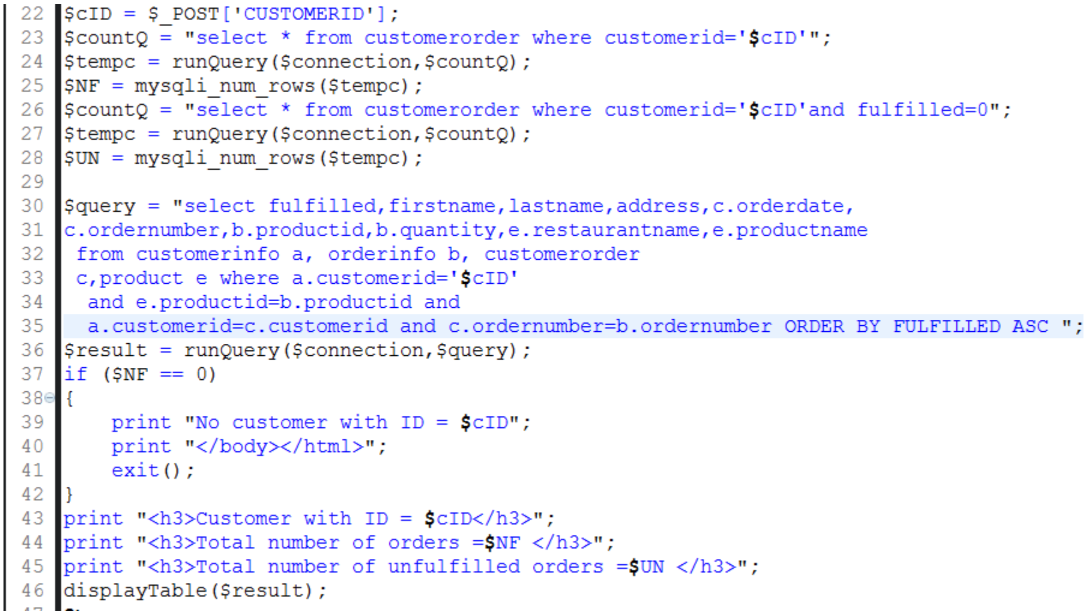
To add a new restaurant, you need to enter the name, loyalty points for pickup and delivery into their relevant text boxes in “AddRestaurant.html.” On click of ‘Submit’ the action performed by the form is to redirect the inputs to “AddRestaurant.php” and store the values as $name, $lpickup and $ldelivery respectfully. Using $query and $insResult the values are inserted into the table RESTAURANT of the database. If this is true, the system executes a print command where "Details for " .$name ." have been inserted” will be displayed on the screen. If this is not the case then it will simply exit.

Update a customer’s information:

To update a customer’s information, you would need to go to “CustomerUpdate.html,” where you will enter the First and Last name of the customer you wish to update. Once you click ‘Submit,’ the system redirects to “CustomerUpdate.php’” where it will store the inputed first and last name as $first and $last, respectfully. This then selects from the table CUSTOMERINFO where $first and $last match the FIRSTNAME and LASTNAME. Once the match(s) are found, the system will fetch and display all the details from the table. You can now amend the details expect for the CUSTOMERID, since it is auto increment and a unique identifier. The new data will be inserted into the table using “UpdateCustomer.php.” Upon successful update, the system will display a message using a print command, "Details for ".$first . "".$last . " have been inserted.” If there is no match for FRISTNAME and LASTNAME the system will display message, "No customer with name = $first $last.”

Get a Customers Orders and Count the number of orders they have (DisplayCustomersOrders.php):

To retrieve a Customers orders and count them, we select all the orders that belong to the customer id($cID). Counting the rows using mysqli_num_rows returns how many rows/orders a customer has. Then when run a second query to count how many of those queries are fulfilled. Finally, we run a third query that retrieves all the items in all the orders and we sort them according to whether they have been fulfilled or not. Finally, we display the results using print.

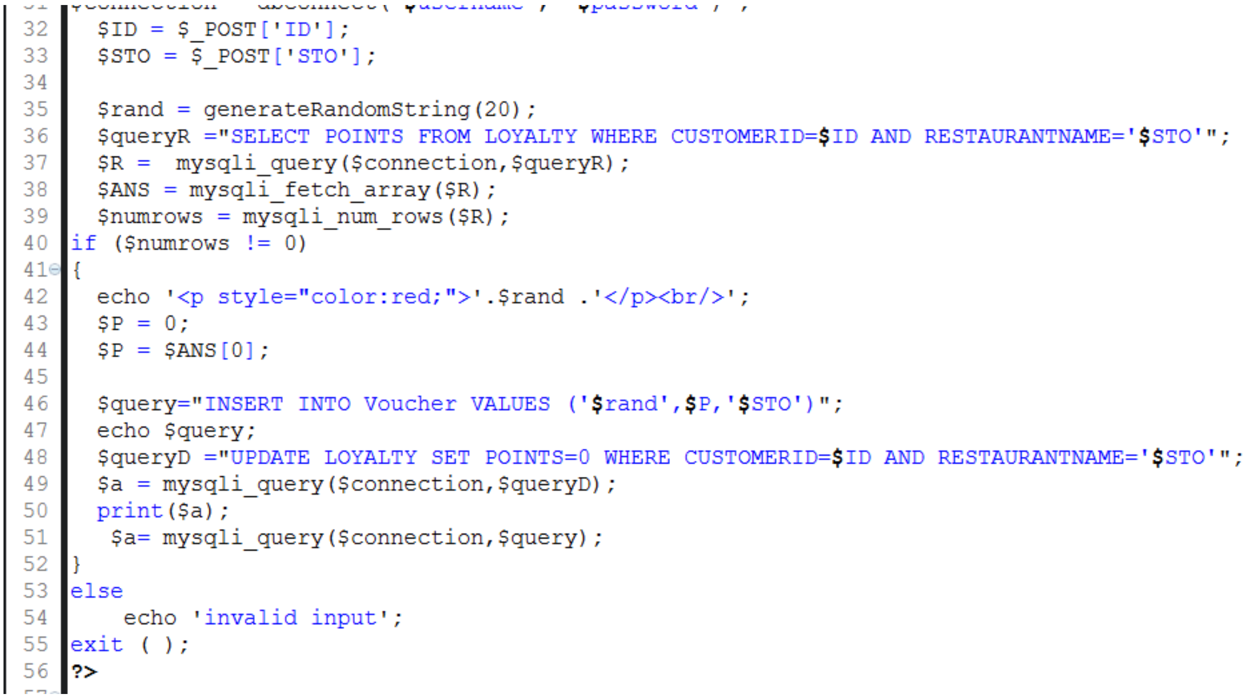


```
$cID = $_POST['CUSTOMERID'];
$countQ = "select * from customerorder where customerid='$cID'";
$tempc = runQuery($connection,$countQ);
$NF = mysqli_num_rows($tempc);
$countQ = "select * from customerorder where customerid='$cID'and fulfilled=0";
$tempc = runQuery($connection,$countQ);
$UN = mysqli_num_rows($tempc);

$query = "select fulfilled,firstname,lastname,address,c.orderdate,
c.ordernumber,b.productid,b.quantity,e.restaurantname,e.productname
 from customerinfo a, orderinfo b, customerorder
 c,product e where a.customerid='$cID'
  and e.productid=b.productid and
  a.customerid=c.customerid and c.ordernumber=b.ordernumber ORDER BY FULFILLED ASC ";
$result = runQuery($connection,$query);
if ($NF == 0)
{
    print "No customer with ID = $cID";
    print "</body></html>";
    exit();
}
print "<h3>Customer with ID = $cID</h3>";
print "<h3>Total number of orders =$NF </h3>";
print "<h3>Total number of unfulfilled orders =$UN </h3>";
displayTable($result);
```

Get A Customers Voucher (GetVoucher.php):

Getting a voucher has 3 steps, getting the points a voucher will represent, creating the voucher, and zeroing out the points the customer has after claiming it. First get the amount of points that the customer($ID) has with a certain restaurant($STO). Then we insert and print the string generated by generateRandomString () into the voucher table with its value in points and the restaurant that it belongs to. Finally, we set the amount of points the customer has for that restaurant to zero.



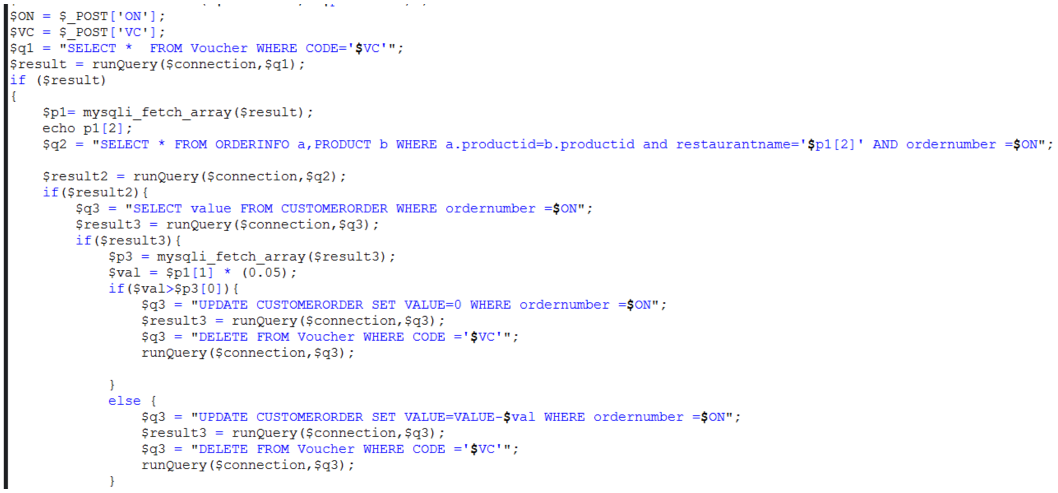
```
  $ID = $_POST['ID'];
  $STO = $_POST['STO'];

  $rand = generateRandomString(20);
  $queryR ="SELECT POINTS FROM LOYALTY WHERE CUSTOMERID=$ID AND RESTAURANTNAME='$STO'";
  $R =  mysqli_query($connection,$queryR);
  $ANS = mysqli_fetch_array($R);
  $numrows = mysqli_num_rows($R);
if ($numrows != 0)
{
  echo '<p style="color:red;">'.$rand .'</p><br/>';
  $P = 0;
  $P = $ANS[0];

  $query="INSERT INTO Voucher VALUES ('$rand',$P,'$STO')";
  echo $query;
  $queryD ="UPDATE LOYALTY SET POINTS=0 WHERE CUSTOMERID=$ID AND RESTAURANTNAME='$STO'";
  $a = mysqli_query($connection,$queryD);
  print($a);
   $a= mysqli_query($connection,$query);
}
else
    echo 'invalid input';
exit ( );
?>
```

Apply a Voucher(ApplyVoucher.php):

Applying a voucher requires two inputs, voucher number(VC), and order number(ON). First, we make a query that checks if the voucher exists and the restaurant the voucher belongs to exists. Second, we check if the order contains products that belongs to that restaurant. Third, we retrieve the total price of the order and check if the value of the voucher is higher than the price of the order. If it is, the order price is set to zero. Otherwise, we subtract the price of the order with the value of the voucher. Finally, we remove the voucher from our voucher table.



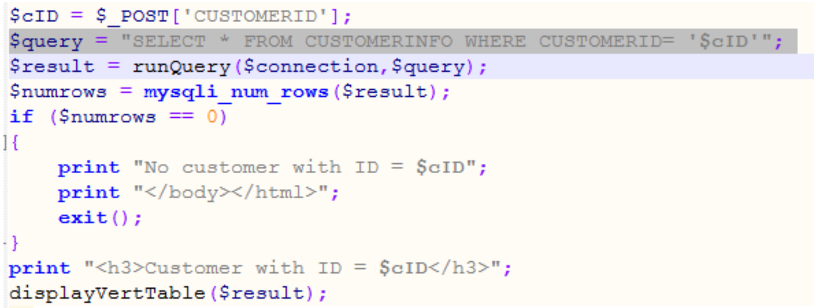
```
$ON = $_POST['ON'];
$VC = $_POST['VC'];
$q1 = "SELECT *  FROM Voucher WHERE CODE='$VC'";
$result = runQuery($connection,$q1);
if ($result)
{
    $p1= mysqli_fetch_array($result);
    echo p1[2];
    $q2 = "SELECT * FROM ORDERINFO a,PRODUCT b WHERE a.productid=b.productid and restaurantname='$p1[2]' AND ordernumber =$ON";

    $result2 = runQuery($connection,$q2);
    if($result2){
        $q3 = "SELECT value FROM CUSTOMERORDER WHERE ordernumber =$ON";
        $result3 = runQuery($connection,$q3);
        if($result3){
            $p3 = mysqli_fetch_array($result3);
            $val = $p1[1] * (0.05);
            if($val>$p3[0]){
                $q3 = "UPDATE CUSTOMERORDER SET VALUE=0 WHERE ordernumber =$ON";
                $result3 = runQuery($connection,$q3);
                $q3 = "DELETE FROM Voucher WHERE CODE ='$VC'";
                runQuery($connection,$q3);

            }
            else {
                $q3 = "UPDATE CUSTOMERORDER SET VALUE=VALUE-$val WHERE ordernumber =$ON";
                $result3 = runQuery($connection,$q3);
                $q3 = "DELETE FROM Voucher WHERE CODE ='$VC'";
                runQuery($connection,$q3);
            }
```

Search for a customer by ID. (DisplayOneCustomer.php):

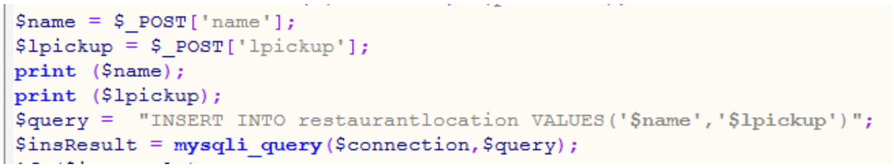
The user enters the customers I.D they wish to get the details of in “RequestCustomerDetails.html“ and click the submit button. Then “DisplayOneCustomer.php” receives the value in “$cID” using $_POST[]. The I.D that was received is then used to display all info that belongs to that customer. “SELECT * CUSTOMERINFO” Indicates that we want to display all values in the database table CUSTOMERINFO, and “WHERE CUSTOMERID=’$cID’” sets the condition for what values are related to what we are looking for. In this case we are looking for the information related to the ID in “$cID” that was entered in the previous page. runQuery() actually runs the sql script in “$query” and stored the result in “$result”. “$numrows” checks the result for the number of rows that were returned. In the case, there are 0 rows that means there are no customers in customerinfo table that contain that ID. displayVertTable() is used to display the customers with an ID similar to the one the user is looking for.



```
$cID = $_POST['CUSTOMERID'];
$query = "SELECT * FROM CUSTOMERINFO WHERE CUSTOMERID= '$cID'";
$result = runQuery($connection,$query);
$numrows = mysqli_num_rows($result);
if ($numrows == 0)
{
    print "No customer with ID = $cID";
    print "</body></html>";
    exit();
}
print "<h3>Customer with ID = $cID</h3>";
displayVertTable($result);
```

Add a location to a restaurant. (AddRestaurantLocation.php):

The user enters the restaurant name they wish to add the location of in “AddRestaurantLocation.html“and click the submit button. Then “AddRestaurantLocation.php” receives the value in “$name” and in “$lpickup” using $_POST[]. The location and name that were received is then used to add the new data into the restaurantlocation table. In this case, it is a new location for a given restaurant. runQuery() then runs the sql script in “$query” and stored the result in “$insResult”.

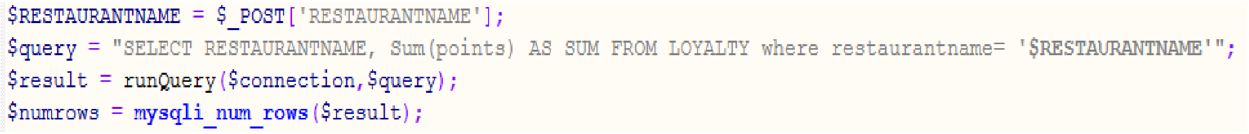


```
$name = $_POST['name'];
$lpickup = $_POST['lpickup'];
print ($name);
print ($lpickup);
$query =  "INSERT INTO restaurantlocation VALUES('$name','$lpickup')";
$insResult = mysqli_query($connection,$query);
```

Get the total amount of reward points awarded by a restaurant name. (RestaurantPoints.php):

The user enters the restaurant name they wish to get the total given loyalty points of in “RestaurantPoints.html“and click the submit button. Then “RestaurantPoints.php” receives the name in “$RESTAURANTNAME” using $_POST []. The restaurant name that was received is then used to sum all loyalty points in the Loyalty table using Sum () and displayed as the table named “SUM”. “WHERE RESTAURANTNAME=’$RESTAURANTNAME’” sets the condition for what is included in the sum, which in this case the condition is the name of restaurant the user entered.

runQuery () then runs the sql script in “$query” and stored the result in “$insResult”.



```
$RESTAURANTNAME = $_POST['RESTAURANTNAME'];
$query = "SELECT RESTAURANTNAME, Sum(points) AS SUM FROM LOYALTY where restaurantname= '$RESTAURANTNAME'";
$result = runQuery($connection,$query);
$numrows = mysqli_num_rows($result);
```