**Exercise 02\_08\_01 – Step 1**

In this Exercise, we will learn how to handle some of the tasks that PHP was particularly designed to do, access and manipulate databases.



1. Create a folder named Exercise 02\_08\_01 and open it with your IDE. Create a new file called ***MySQLInfo.php***. Scaffold a basic HTML code layout into it. Complete our standard opening documentation in the ***<head>*** element. Make sure to have the ***modernizr*** <script> linked in. Set the <title> content to ***MySQLInfo***:  
   ***<!DOCTYPE html>  
   <html lang="en">  
   <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width">  
    <title>MySQLInfo</title>  
    <script src="modernizr.custom.65897.js"></script>  
   </head>  
   <body>  
   </body>  
   </html>***
2. In the <body>, create an ***<h2>*** element with content ***MySQL Database Server Information***. Create a set of PHP standard script delimiters.  
    ***<h2>MySQL Database Server Information</h2>  
    <?php  
     
    ?>***Copy the project folder into the appropriate spot on your Web Server and test it.
3. Insert the following code into top of the script section to retrieve the info of the MySQL ***client***. This will be the server PHP version and ID:  
    <?php  
    ***echo "<p>MySQL client version: " . mysqli\_get\_client\_info()   
   . "</p>\n";*** ?>  
   Give this a browser/server test.
4. Insert the following code into the script to make the database ***connection***, print out the contents of the connection ***variable***, and test for success or failure, and ***close*** the connection if it was successful:  
    ***$hostname = "localhost";  
    $username = "root";  
    $password = "";  
    $DBConnect = mysqli\_connect($hostname, $username,   
    $password);  
    if (!$DBConnect) {  
    echo "<p>Connection failed.</p>\n";  
    }  
    else {  
    echo "<p>Connection successful.</p>\n";  
    mysqli\_close($DBConnect);  
    }***  
   Give this a browser/server test.
5. Insert code into the connection success ***else*** clause to obtain more database server ***information***:  
    else {  
    echo "<p>Connection successful.</p>\n";  
    ***echo "<p>MySQL connection: " .   
    mysqli\_get\_host\_info($DBConnect) . "</p>\n";  
    echo "<p>MySQL protocol version: " .   
    mysqli\_get\_proto\_info($DBConnect) . "</p>\n";  
    echo "<p>MySQL server version: " .   
    mysqli\_get\_server\_info($DBConnect) . "</p>\n";*** mysqli\_close($DBConnect);  
    }  
   Give this a browser/server test.

**Exercise 02\_08\_01 – Step 2**



1. Copy the file ***MySQLInfo.php*** to a file named ***CreateNewsletterDB.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Create Newsletter DB.*** In the <body>, create an ***<h2>*** element with content ***Create Newsletter DB***. Create a set of PHP standard script delimiters:  
    ***<h2>Create Newsletter DB</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Modify the code in the script as follows to add a variable to name the new DB and to add error reporting. Purposely ***misname*** the host to test the error reporting:  
    <?php  
    ***$hostname = "localhost";  
    $username = "root";  
    $password = "";  
    $DBName = "newsletter";  
    $DBConnect = mysqli\_connect($hostname, $username,   
    $password);  
    if (!$DBConnect) {  
    echo "<p>Connection error: " . mysqli\_connect\_error() .   
    "</p>\n";  
    }  
    else {  
    mysqli\_close($DBConnect);  
    }*** ?>  
   Give this a server/browser test. Fix the host name and retest.
3. Insert code into the success ***else*** clause to create a new database. Notice the different form of the ***mysqli\_error()*** call. We pass the connection variable to distinguish which DB might have the error we are interested, as we could have multiple DBs open:  
    else {  
    ***$sql = "CREATE DATABASE $DBName";  
    if (mysqli\_query($DBConnect, $sql)) {  
    echo "<p>Successfully created the \"$DBName\"   
    database.</p>\n";  
    }  
    else {  
    echo "<p>Could not create the \"$DBName\"   
    database: " .   
    mysqli\_error($DBConnect) . "</p>\n";  
    }***Give this a browser/server test. Misname the ***$DBConnect*** variable to test the error case. Then fix it for another test. Check for the existence of the database in ***MySQL*** ***Monitor***.

**Exercise 02\_08\_01 – Step 3**



1. Copy the file ***CreateNewsletterDB.php*** to a file named ***SelectTest.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Select Test.*** In the <body>, create an ***<h2>*** element with content ***Select Test***. Create a set of PHP standard script delimiters:  
    ***<h2> Select Test </h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Modify the code in the success ***else*** clause to select the new database:  
    else {  
    ***if (mysqli\_select\_db($DBConnect, $DBName)) {  
    echo "<p>Successfully selected the \"$DBName\"   
    database.</p>\n";  
    }  
    else {  
    echo "<p>Could not select the \"$DBName\" database: " .   
    mysqli\_error($DBConnect) . "</p>\n";  
    }*** mysqli\_close($DBConnect);  
    }Give this a browser/server test. Misname the ***$DBName*** variable to test the error case. Then fix it for another test.

**Exercise 02\_08\_01 – Step 4**



1. Copy the file ***SelectTest.php*** to a file named ***CreateSubscribersTable.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Create Subscribers Table.*** In the <body>, create an ***<h2>*** element with content ***Create Subscribers Table***:  
    ***<h2>Create Subscribers Table <h2>***
2. Add a new variable to hold the name of the new table:  
    $DBName = "newsletter";  
    **$tablename = "subscribers";**
3. Modify the code in the success ***mysqli\_select\_db()*** clause to test whether the new table exists, and to create a success/failure framework to act on the results:  
    echo "<p>Successfully selected the \"$DBName\"   
    database.</p>\n";  
    ***$sql = "SHOW TABLES LIKE '$tablename'";  
    $result = mysqli\_query($DBConnect, $sql);  
    if (mysqli\_num\_rows($result) === 0) {  
    echo "The <strong>$tablename</strong> table does   
    not exist, creating table.<br>\n";  
    }  
    else {  
    echo "The <strong>$tablename</strong> table   
    already exists.<br>\n";  
    }***Give this a browser/server test.
4. Add the following code into the success ***mysqli\_num\_rows ()*** clause to add the new table, handling the success and failure scenarios:  
    echo "<p>Successfully selected the \"$DBName\"   
    database.</p>\n";  
    ***$sql = "CREATE TABLE subscribers (subscriberID   
    SMALLINT NOT NULL AUTO\_INCREMENT PRIMARY   
    KEY, name VARCHAR(80), email VARCHAR(100),   
    subscribe\_date DATE, confirmed\_date DATE)";  
    $result = mysqli\_query($DBConnect, $sql);  
    if ($result === false) {  
    echo "<p>Unable to create the   
    <strong>$tablename</strong> table.</p>";  
    echo "<p>Error code: " . mysqli\_error($DBConnect)   
    . "</p>";  
    }  
    else {  
    echo "<p>Successfully created the   
    <strong>$tablename</strong> table.</p>";  
    }***Give this a browser/server test. If successful, run it again and make sure it gives the correct error message. Go to MySQL Monitor and use a ***DESCRIBE subscribers;*** command to check the structure.

**Exercise 02\_08\_01 – Step 5**



1. Copy the file ***CreateSubscribersTable.php*** to a file named ***NewsletterSubscribe.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Newsletter Subscribe.*** In the <body>, create an ***<h2>*** element with content ***Newsletter Subscribe***. Comment out all of the code in the PHP script delimiters, except the variable declarations at the top:  
    ***<h2> Newsletter Subscribe<h2>***
2. Add some new variables after the existing ones to help out with the code:  
    $tablename = "subscribers";  
    ***$subscriberName = "";  
    $subscriberEmail = "";  
    $showForm = false;***
3. This will be an all-in-one form, so add the following HTML code into the ***<body>*** below the PHP script delimiters:  
    ***<form action="NewsletterSubscribe.php" method="post">  
    <p><strong>Your Name: </strong><br>  
    <input type="text" name="subName" value="<?php echo   
    $subscriberName; ?>"></p>  
    <p><strong>Your Email Address: </strong><br>  
    <input type="email" name="subEmail" value="<?php echo   
    $subscriberEmail; ?>"></p>  
    <p><input type="submit" name="submit"   
    value="Submit"></p>  
    </form>***  
   Give this a browser/server test.
4. Remove all of the code from the first ***$sql*** assignment statement, through and including the entire ***else*** clause containing the statement:  
   ***echo "The $tablename table already exists.<br>\n"***Give this a browser/server test.
5. Uncomment the rest of the PHP code so the remaining code is as follows:  
    $showForm = false;  
    ***$DBConnect = mysqli\_connect($hostname, $username,   
    $password);  
    if (!$DBConnect) {  
    echo "<p>Connection error: " . mysqli\_connect\_error() .   
    "</p>\n";  
    }  
    else {  
    if (mysqli\_select\_db($DBConnect, $DBName)) {  
    echo "<p>Successfully selected the \"$DBName\"   
    database.</p>\n";  
    }  
    else {  
    echo "<p>Could not select the \"$DBName\" database:   
    " . mysqli\_error($DBConnect) . "</p>\n";  
    }  
    mysqli\_close($DBConnect);  
    }***Give this a browser/server test. The database should be successfully selected.
6. Just before the closing PHP script delimiter, add the following if statement::  
    ***if ($showForm) {***  
    ?>
7. Just before the closing ***</html>*** tag, add the following PHP script:  
   </html>  
   ***<?php  
   }  
   ?>***Give this a browser/server test. The form should not show, but the database should be successfully selected.
8. Directly under the variable declarations, add the following ***if…else*** clause to test for form ***submit***:  
    $showForm = false;  
    ***if (isset($\_POST['submit'])) {  
    $formErrorCount = 0;  
    }  
    else {  
    $showForm = true;  
    }***  
   Give this a browser/server test. The form should show and the database should be successfully selected and disappear on a submit.
9. Now move all of the database code into the ***if*** part of the clause, from the ***$DBConnect*** assignment, down to, but not including, the ***if ($showform)*** statement. The code should look like this:  
    ***if (isset($\_POST['submit'])) {  
    $formErrorCount = 0;  
    $DBConnect = mysqli\_connect($hostname, $username,   
    $password);  
    if (!$DBConnect) {  
    echo "<p>Connection error: " . mysqli\_connect\_error()   
    . "</p>\n";  
    }  
    else {  
    if (mysqli\_select\_db($DBConnect, $DBName)) {  
    echo "<p>Successfully selected the \"$DBName\"   
    database.</p>\n";  
    }  
    else {  
    echo "<p>Could not select the \"$DBName\"   
    database: " .   
    mysqli\_error($DBConnect) . "</p>\n";  
    }  
    mysqli\_close($DBConnect);  
    }  
    }***Give this a browser/server test. The form should show and the database should ***not*** be selected. The form should disappear on a submit, and the database should be selected.
10. Change the DB connection failure to the following ***mysqli*** error code:  
     if (!$DBConnect) {  
     ***echo "<p>Connection error: " . mysqli\_connect\_error()   
     . "</p>\n";*** }
11. Enclose all of the ***$DBConnect*** code in the following ***if...else*** statement:  
     $formErrorCount = 0;  
     ***if ($formErrorCount === 0) {  
     $showForm = false;*** $DBConnect = mysqli\_connect($hostname, $username,   
     $password);  
     if (!$DBConnect) {  
     echo "<p>Connection error: " . mysqli\_connect\_error() .   
     "</p>\n";  
     }  
     else {  
     if (mysqli\_select\_db($DBConnect, $DBName)) {  
     echo "<p>Successfully selected the \"$DBName\"   
     database.</p>\n";  
     }  
     }  
     else {  
     echo "<p>Could not select the \"$DBName\" database: "   
     . mysqli\_error($DBConnect) . "</p>\n";  
     }  
     mysqli\_close($DBConnect);  
     }  
     ***}  
     else {  
     $showForm = true;  
     }***Give this a browser/server test.
12. Let’s start building out the code to validate the form. Directly under the variable ***$formErrorCount*** initialization, add the following ***if…else*** clause to test for form errors on the subscriber ***name***:  
     $formErrorCount = 0;  
     ***if (!empty($\_POST['subName'])) {  
     $subscriberName =   
     stripslashes($\_POST['subName']);  
     $subscriberName = trim($subscriberName);  
     if (strlen($subscriberName) === 0) {  
     echo "<p>You must include your   
     <strong>name</strong></p>\n";  
     ++$formErrorCount;  
     }  
     }  
     else {  
     echo "<p>Form submittal error, no   
     <strong>Name</strong> field!</p>\n";  
     ++$formErrorCount;  
     }***  
    Give this a browser/server test. Be thorough about entering nothing, then just spaces, then a real name.
13. Directly below the previous code, let’s test for form errors on the subscriber ***email***:   
     ***if (!empty($\_POST['subEmail'])) {  
     $subscriberEmail = stripslashes($\_POST['subEmail']);  
     $subscriberEmail = trim($subscriberEmail);  
     if (strlen($subscriberEmail) === 0) {  
     echo "<p>You must include your <strong>email   
     address</strong></p>\n";  
     ++$formErrorCount;  
     }  
     }  
     else {  
     echo "<p>Form submittal error, no   
     <strong>email</strong> field!</p>\n";  
     ++$formErrorCount;  
     }***  
    Give this a browser/server test. Be thorough about entering nothing, then just spaces, then a real email.
14. Now we can actually add the data to the database. Add the following code to the ***mysqli\_select\_db()*** in the successful ***if*** clause below the ***echo***. Notice the use of mysqli\_insert\_id() used to retrieve the ***AUTO\_INCREMENT PRIME KEY*** data that was generated   
     if (mysqli\_select\_db($DBConnect, $DBName)) {  
     echo "<p>Successfully selected the \"$DBName\"   
     database.</p>\n";  
     ***$subscriberDate = date("Y-m-d");  
     $sql = "INSERT INTO $tablename (name, email,   
     subscribe\_date) VALUES('$subscriberName',   
     '$subscriberEmail', '$subscriberDate')";  
     $result = mysqli\_query($DBConnect, $sql);  
     if ($result === false) {  
     echo "<p>Unable to insert the values into the   
     <strong>$tablename</strong>   
     table.</p>";  
     echo "Error code <strong>" .   
     msqli\_errorno($DBConnect) . ": " .   
     mysqli\_error($DBConnect) .   
     "</strong></p>";  
     }  
     else {  
     $subscriberID =   
     mysqli\_insert\_id($DBConnect);  
     echo "<p><strong>" .   
     htmlentities($subscriberName) .   
     "</strong>, you are now subscribed to our   
     newsletter.<br>";  
     echo "Your subscriber ID is   
     <strong>$subscriberID</strong>.<br>";  
     echo "Your email address is <strong>" .   
     htmlentities($subscriberEmail) .   
     "</strong>.</p>";  
     }***  
    Give this a browser/server test. Add at least 5 subscribers to test

**Exercise 02\_08\_01 – Step 6**



1. Copy the file ***CreateSubscribersTable.php*** to a file named ***NewsletterSubscribers.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Newsletter Subscribers.*** In the <body>, create an ***<h2>*** element with content ***Newsletter Subscribers***. Comment out all of the code in the PHP script delimiters, except the variable declarations at the top:  
    ***<h2> Newsletter Subscribers<h2>***
2. Remove all of the code that comprises the entire ***if (mysqli\_num\_rows()*** statement with its ***else*** clause. Modify the ***$sql*** statement and add a debug line of code beneath the ***mysqli\_query()*** call as follows:  
    echo "<p>Successfully selected the \"$DBName\"   
    database.</p>\n";  
    ***$sql = "SELECT \* FROM $tablename";  
    $result = mysqli\_query($DBConnect, $sql);  
    echo "<p>Number of rows in   
    <strong>$tablename</strong>: " .   
    mysqli\_num\_rows($result) . ".</p>\n";  
    mysqli\_free\_result($result);***Give this a browser/server test.
3. Now let’s build out an HTML <table> into which we will load the retrieved data:  
    echo "<p>Number of rows in <strong>$tablename</strong>:   
    " . mysqli\_num\_rows($result) . ".</p>\n";  
    ***echo "<table width='100%' border='1'>\n";  
    echo "<tr><th>Subscriber ID</th>" .   
    "<th>Name</th>" . "<th>Email</th>" .   
    "<th>Subscriber Date</th>" .   
    "<th>Name</th></tr>\n";  
    echo "</table>\n";*** mysqli\_free\_result($result);  
   Give this a browser/server test.
4. Now we can retrieve the data into a ***result*** ***set*** using the ***mysqli\_fetch\_row()*** function. Enter code just above the closing ***</table>*** tag as follows:  
    ***while ($row = mysqli\_fetch\_row($result)) {  
    echo "<tr>";  
    echo "<td>{$row[0]}</td>";  
    echo "<td>{$row[1]}</td>";  
    echo "<td>{$row[2]}</td>";  
    echo "<td>{$row[3]}</td>";  
    echo "<td>{$row[4]}</td>";  
    echo "<tr>\n";  
    }*** echo "</table>\n";  
   Give this a browser/server test.

**Exercise 02\_08\_01 – Step 7**



1. Copy the file ***NewsletterSubscribers.php*** to a file named ***NewsletterSubscribers2.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Newsletter Subscribers 2.*** In the <body>, create an ***<h2>*** element with content ***Newsletter Subscribers 2***:  
    ***<h2> Newsletter Subscribers 2<h2>***
2. Now we can retrieve the data into a ***result*** ***set*** using the ***mysqli\_fetch\_assoc()*** function. Change the while loop and its associated code as follows. Notice that we can now use a foreach loop to go through the fields:  
    ***while ($row = mysqli\_fetch\_assoc($result)) {  
    echo "<tr>";  
    foreach ($row as $field) {  
    echo "<td>{$field}</td>";  
    }  
    echo "<tr>\n";  
    }*** echo "</table>\n";  
   Give this a browser/server test.