

UNIT – V

Interacting with MySQL using PHP

1. Write the difference between MySQL versus MySQLi functions in PHP?

MySQL is the old database driver, and MySQLi is improved driver. The "i" stands for improved so it is MySQL improved database.

MySQL and MySQLi are PHP database extensions implemented by using the PHP extension framework. PHP database extensions are used to write PHP code for accessing the database. They expose database API to provide interfaces to use database functions.

MySQL extension is deprecated and will not be available in future PHP versions. It is recommended to use the MySQLi extension with PHP 5.5 and above. MySQLi allows `mysqli_query()` function connect to the database. MySQLi API allows executing multiple queries with a single expression using the `multi_query()` function.

MySQL	MySQLi
MySQL extension added in PHP version 2.0	MySQLi extension added in php 5.5 and will work on MySQL 4.1.3 or above.
MySQL provides the procedural interface.	MySQLi provides both procedural and Object-Oriented interface.
MySQL does not support for prepared statements.	MySQLi supports for prepared statements.
It does not support for multiple statements.	It support for multiple statements.
Transactions are handled by SQL queries only.	It supports transactions through API.
It does not support stored procedures.	It supports store procedure.
Extension directory: ext/mysql	Extension directory: ext/mysqli

2. Explain about connecting to MySQL with PHP? Or Making a Connection

In PHP, we can easily connecting to MySQL with using the `mysqli_connect()` function. All the communication between PHP with the MySQL database server takes place through this connection.

Syntax:

```
$conn=mysqli_connect("hostname","username","password","database");
```

OR

```
$conn=new mysqli("hostname","username","password","database");
```

The hostname specifies the hostname (**localhost**), whereas the **username** and **password** parameter specifies the credentials to access MySQL server, and the **database** parameter is the name of the database.

To make a connection between PHP and MySQL

```
<! mysqlconnect.php>
<?php
$conn = mysqli_connect("localhost", "college", "", "testdb");
if (mysqli_connect_errno())
{
    printf("Connect failed: %s\n", mysqli_connect_error());
    exit();
}
```

```

else
printf("Host information: %s\n", mysqli_get_host_info($conn));
mysqli_close($conn);
?>

```

Save the above script as **mysqlconnect.php** and place it in the web server. If the connection was successful, then `mysqli_get_host_info()` function display a message **Host information: localhost via TCP/IP**.

If the connection fails, an error occurred, `mysqli_connect_error()` function prints an error message is **Access denied for user 'college@'localhost'**.

3. Write about executing queries? And retrieving an error messages?

In PHP, the `mysqli_query()` function is used to send our SQL query to MySQL. In this script, first make the connection and then execute a query. The script in creates a simple table called **student**. The `mysqli_query()` function returns a value of true or false, and this value is checked in the if else statement. If the value of `$res` is true, a success message is printed on the screen. If the value of `$res` is not true, and the table was not created, an error message will appear, generated by the `mysqli_error()` function.

```

<! mysqlcreatetable.php>
<?php
$conn=new mysqli("localhost", "college", "", "testdb");
if (mysqli_connect_errno())
{
printf("Connect failed: %s\n", mysqli_connect_error());
exit();
}
else
{
$sql = "CREATE TABLE student (sno INT PRIMARY KEY
AUTO_INCREMENT, sname VARCHAR (25))";
$res = mysqli_query($conn, $sql);
if ($res === TRUE)
echo "Table student successfully created.";
else
printf("Could not create table: %s\n", mysqli_error($conn));
mysqli_close($conn);
}
?>

```

Save the above script **mysqlcreatetable.php** and place it in the web server.

Table student successfully created.

Retrieving an error messages

The `mysqli_error()` function will return a helpful error message when we make a mistake. For example, we have created a table **student**. If we execute the script again we will get the following error message in the browser.

Could not create table: Table 'student' already exists

Working with MySQL Data

4. Write about inserting data with PHP?

The `mysqli_query()` function to execute the basic SQL queries. The basic SQL commands are like INSERT, UPDATE and DELETE queries; no additional scripting is required after the query has been executed.

Inserting data with PHP:

The easiest method for inserting data at this stage simply hardcode the INSERT command.

```
<!mysqliinsert.php>
<?php
$conn = new mysqli("localhost", "college", "", "testdb");
if (mysqli_connect_errno())
{
    printf("Connect failed: %s\n", mysqli_connect_error());
    exit();
}
else
{
    $sql = "INSERT INTO student (sname) VALUES ('Rajesh')";
    $res = mysqli_query($conn, $sql);
    if ($res === TRUE)
        echo "A record has been inserted.";
    else
        printf("Could not insert record: %s\n", mysqli_error($conn));
    mysqli_close($conn);
}
?>
```

A record has been inserted.

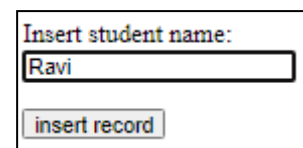
Save the above script **mysqliinsert.php** and place it in the web server.

Running this script will result in the addition of a row to the **student** table. To enter more than one in the script, we can either make a long list of hard coded SQL statements and use `mysqli_query()` multiple times to execute these statements.

Form-based interface to the record addition script:

To create the form for this script, we need only one field because **sno** field can automatically increment. The **action** of the form is the name of the record-addition script.

```
<!insertform.html>
<html>
<body>
<form action="insert.php" method="POST">
<p>Insert student name:<br>
<input type="text" name="sname" size="20">
<p><input type="submit" name="submit" value="insert record"></p>
</form>
</body>
</html>
```



Save the above script **insertform.html** and place it in the web server.

The value entered in the form will replace the hard coded values in the SQL query with a variable called `$_POST["sname"]`.

```
<!insert.php>
```

```
<?php
```

```
$conn= mysqli_connect("localhost", "college", "", "testdb");
```

```
if (mysqli_connect_errno())
```

```
{
```

```
printf("Connect failed: %s\n", mysqli_connect_error());
```

```
exit();
```

```
}
```

```
else
```

```
{
```

```
$sql = "INSERT INTO student (sname) VALUES ('".$_POST["sname"]."')";
```

```
$res = mysqli_query($conn, $sql);
```

```
if ($res === TRUE)
```

```
echo "A record has been inserted.";
```

```
else
```

```
printf("Could not insert record: %s\n", mysqli_error($conn));
```

```
mysqli_close($conn);
```

```
}
```

```
?>
```

A record has been inserted.

Save the above script **insert.php** and place it in the web server.

5. Explain about retrieving data with PHP?

We have a few rows in our (**student**) table, we can write a PHP script to retrieve that data. The **mysqli_query()** function is used to retrieve data from a table. The **mysqli_num_rows()** function to retrieve the number of rows in the result set (\$res) and it places the value in a variable called **\$number_of_rows**. The number should be equal to the number of records in (**student**) table.

There are some records in the table, we can get fancy and fetch the actual contents of those records to retrieve each row as an array. We use while statement to each record in the result set, placing the values of each fields into a specific variable, and then displaying the results on the screen.

```
$newArray=mysqli_fetch_array($res);
```

The use of **mysqli_free_result()** function before closing the connection with **mysqli_close()** ensures that all memory associated with the query.

Example:

```
<!select.php>
```

```
<?php
```

```
$conn = mysqli_connect("localhost", "college", "", "testdb");
```

```
if (mysqli_connect_errno())
```

```
{
```

```
printf("Connect failed: %s\n", mysqli_connect_error());
```

```
exit();
```

```
}
```

```
else
```

```
{
```

```
$sql = "SELECT * FROM student";
```

```

$res = mysqli_query($conn, $sql);
if ($res)
{
$number_of_rows = mysqli_num_rows($res);
printf("Result set has %d rows.\n", $number_of_rows, "\n");
while($newArray = mysqli_fetch_array($res))
{
$sno=$newArray['sno'];
$name=$newArray['sname'];
echo "<br>";
echo "Student number is ".$sno." & student name is ".$name;
}
}
else
printf("Could not retrieve records: %s\n", mysqli_error($conn));
mysqli_free_result($res);
mysqli_close($conn);
}
?>

```

Save the above script **select.php** and place it in the web server.

```

Result set has 5 rows.
Student number is 1 & student name is Rajesh
Student number is 2 & student name is Ravi
Student number is 3 & student name is Raju
Student number is 4 & student name is Raji
Student number is 5 & student name is Radha

```

Creating an Online Address Book

6. Discuss about Planning and creating the database tables?

When we think of an address book, the obvious fields come to mind: name, address, telephone number, email address. However, if we look at our own paper based address book, we have several entries for one person. Maybe that person has three telephone numbers, or two email addresses and so forth. In our online address book, a set of related tables will help alleviate the redundancy and repetition of information.

We will use actual SQL statements to create the tables, but first we should look at this information and try to see the relationship appear which fields should be primary or unique keys.

Table Name	Field Names
master_name	id, date_added, date_modified, f_name, l_name
address	id, master_id, date_added, date_modified, address, city, state, zipcode, type
telephone	id, master_id, date_added, date_modified, tel_number, type
Fax	id, master_id, date_added, date_modified, fax_number, type
Email	id, master_id, date_added, date_modified, email, type
personal_notes	id, master_id, date_added, date_modified, note

Each table has date_added and date_modified field. The fields will help maintain our data.

The master_name table has 5 fields the id, date_added, date_modified, f_name, l_name. The id field is the primary key. No other keys need to be primary or unique we want to limit our address book.

Table: master_name

```
CREATE TABLE master_name (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    date_added datetime,  
    date_modified datetime,  
    f_name varchar(75),  
    l_name varchar(75)  
);
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

We will create the supplementary tables, which all relate back to the master_name table. For, instance, the address table has the basic primary key id field, the date_added field, and date_modified field plus the field through which the relationship will be made the master_id field.

The master_id will be equal to the id field in the master_name table, matching the person the address. The master_id field is not a unique key because it is a perfectly valid assumption that one person may have several address entries. The type field as an enumerated list containing three options: **home**, **work** or **other**. A person may have one or more of all three types, so no other keys are present in this table besides the primary key id.

Table: address

```
CREATE TABLE address (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    master_id INT NOT NULL,  
    date_added DATETIME,  
    date_modified DATETIME,  
    address VARCHAR(255),  
    city varchar(30),  
    state CHAR(2),  
    zipcode varchar(10),  
    type ENUM('home','work','other')  
);
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

The telephone, fax, and email tables are all variations on the same theme.

Table: telephone

```
CREATE TABLE telephone(  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    master_id INT NOT NULL,  
    date_added DATETIME,  
    date_modified DATETIME,  
    tel_number VARCHAR(255),  
    type ENUM('home','work','other')  
);
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

Table: email

```
CREATE TABLE email(  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    master_id INT NOT NULL,
```

```

date_added DATETIME,
date_modified DATETIME,
email VARCHAR(150),
type ENUM('home','work','other')
);

```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

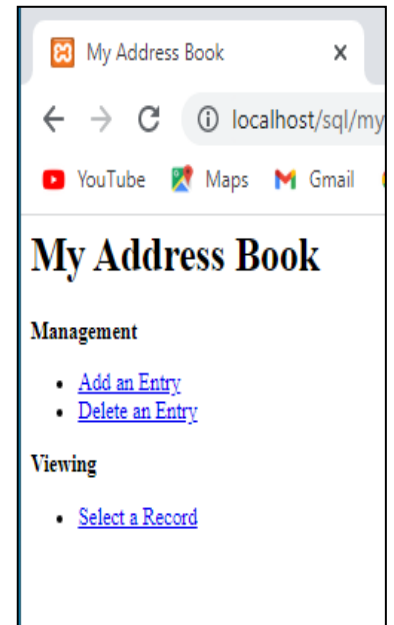
7. Write about creating a Menu in an online address book?

Online address book will contain several actions, so it creates a menu for our links. Creates a simple menu for all the scripts we will create the script **mymenu.html**.

```

<! mymenu.html>
<html>
<head>
<title>My Address Book</title>
</head>
<body>
<h1>My Address Book</h1>
<p><b>Management</b></p>
<ul>
<li><a href="addentry.php">Add an Entry</a></li>
<li><a href="delentry.php">Delete an Entry</a></li>
</ul>
<p><b>Viewing</b></p>
<ul>
<li><a href="selentry.php">Select a Record</a></li>
</ul>
</body>
</html>

```



8. Write about creating the record addition mechanism?

The basic record addition script has two parts: which the form should be displayed and what actions to take if the form is being submitted.

```

<!addentry.php>
<?php
if (!$_POST)
{
$display_block = "
<form method=\"post\" action=\"".$_SERVER["PHP_SELF"].">
<p><b>First/Last Names:</b><br/>
<input type=\"text\" name=\"f_name\" size=\"30\" maxlength=\"75\">
<input type=\"text\" name=\"l_name\" size=\"30\" maxlength=\"75\"></p>

<p><b>Address:</b><br/>
<input type=\"text\" name=\"address\" size=\"30\"></p>
<p><b>City/State/Zip:</b><br/>
<input type=\"text\" name=\"city\" size=\"30\" maxlength=\"50\">
<input type=\"text\" name=\"state\" size=\"5\" maxlength=\"2\">
<input type=\"text\" name=\"zipcode\" size=\"10\" maxlength=\"10\"></p>

```

```

<p><b>Address Type:</b><br/>
<input type=\"radio\" name=\"add_type\" value=\"home\" checked> home
<input type=\"radio\" name=\"add_type\" value=\"work\"> work
<input type=\"radio\" name=\"add_type\" value=\"other\"> other</p>

```

```

<p><b>Telephone Number:</b><br/>
<input type=\"text\" name=\"tel_number\" size=\"30\" maxlength=\"25\">
<input type=\"radio\" name=\"tel_type\" value=\"home\" checked> home
<input type=\"radio\" name=\"tel_type\" value=\"work\"> work
<input type=\"radio\" name=\"tel_type\" value=\"other\"> other</p>

```

```

<p><b>Email Address:</b><br/>
<input type=\"text\" name=\"email\" size=\"30\" maxlength=\"150\">
<input type=\"radio\" name=\"email_type\" value=\"home\" checked> home
<input type=\"radio\" name=\"email_type\" value=\"work\"> work
<input type=\"radio\" name=\"email_type\" value=\"other\"> other</p>

```

```

<p><input type=\"submit\" name=\"submit\" value=\"Add Entry\"></p>
</form>";

```

```

}
else if ($_POST)
{
if (($_POST["f_name"] == "") || ($_POST["l_name"] == ""))
{
header("Location: addentry.php");
exit;
}
//connect to database
$mysqli = mysqli_connect("localhost", "college", "", "testDB");
if (mysqli_connect_errno())
{
printf("Connect failed: %s\n", mysqli_connect_error());
exit();
}
//add to master_name table
$add_master_sql = "INSERT INTO master_name (date_added, date_modified,
f_name, l_name)
VALUES (now(), now(), '".$_POST["f_name"]."', '".$_POST["l_name"]."')";
$add_master_res = mysqli_query($mysqli, $add_master_sql) or
die(mysqli_error($mysqli));

//get master_id for use with other tables
$master_id = mysqli_insert_id($mysqli);
if (($_POST["address"]) || ($_POST["city"]) || ($_POST["state"]) ||
($_POST["zipcode"]))
{
//something relevant, so add to address table

```

```

$add_address_sql = "INSERT INTO address (master_id, date_added,
date_modified, address, city, state, zipcode, type) VALUES ('".$master_id."',
now(), now(), '".$_POST["address"]."', '".$_POST["city"]."',
".$_POST["state"]."', '".$_POST["zipcode"]."', '".$_POST["add_type"]."');
$add_address_res = mysqli_query($mysqli, $add_address_sql) or
die(mysqli_error($mysqli));
}
if ($_POST["tel_number"])
{
//something relevant, so add to telephone table
$add_tel_sql = "INSERT INTO telephone (master_id, date_added,
date_modified,
tel_number, type) VALUES ('".$master_id."', now(), now(),
".$_POST["tel_number"]."', '".$_POST["tel_type"]."');
$add_tel_res = mysqli_query($mysqli, $add_tel_sql) or
die(mysqli_error($mysqli));
}
if ($_POST["email"])
{
//something relevant, so add to email table
$add_email_sql = "INSERT INTO email (master_id, date_added,
date_modified,
email, type) VALUES ('".$master_id."', now(), now(),
".$_POST["email"]."', '".$_POST["email_type"]."');
$add_email_res = mysqli_query($mysqli, $add_email_sql) or
die(mysqli_error($mysqli));
}
mysqli_close($mysqli);
$display_block = "<p>Your entry has been added. Would you like to <a
href=\"addentry.php\">add
another</a>?</p>";
}
?>
<html>
<head>
<title>Add an Entry</title>
</head>
<body>
<h1>Add an Entry</h1>
<?php echo $display_block; ?>
</body>
</html>

```

Add an Entry

Your entry has been added. Would you like to [add another?](#)

Add an Entry

First/Last Names:

Address:

City/State/Zip:

Address Type:

☒ home ☐ work ☐ other

Telephone Number:

Email Address:

9. Write about viewing Records in online address book?

The select and view script class selentry.php, which has two parts: the record selection form and to display the record contents.

```
<?php
$mysqli = mysqli_connect("localhost", "college", " ", "testDB");
if (mysqli_connect_errno())
{
    printf("Connect failed: %s\n", mysqli_connect_error());
    exit();
}
if (!$_POST)
{
    $display_block = "<h1>Select an Entry</h1>";
    //get parts of records
    $get_list_sql = "SELECT id, CONCAT_WS(' ', l_name, f_name) AS display_name
    FROM master_name ORDER BY l_name, f_name";
    $get_list_res = mysqli_query($mysqli, $get_list_sql) or die(mysqli_error($mysqli));
    if (mysqli_num_rows($get_list_res) < 1)
    {
        $display_block .= "<p><em>Sorry, no records to select!</em></p>";
    }
    else
    {
        //has records, so get results and print in a form
        $display_block .= "
        <form method=\"post\" action=\"\".$_SERVER[\"PHP_SELF\"]\">
        <p><b>Select a Record to View:</b><br>
        <select name=\"sel_id\">
        <option value=\"\">-- Select One --</option>";
        while ($recs = mysqli_fetch_array($get_list_res))
        {
            $id = $recs['id'];
            $display_name = stripslashes($recs['display_name']);
            $display_block .= "<option value=\"\".$id.\">\".$display_name.\"</option>";
        }
        $display_block .= "
        </select>
        <p><input type=\"submit\" name=\"submit\" value=\"View Selected Entry\"></p>
        </form>";
    }
    mysqli_free_result($get_list_res);
}
else if ($_POST)
{
    //check for required fields
    if ($_POST["sel_id"] == "")
    {
        header("Location: selentry.php");
        exit;
    }
}
```

```

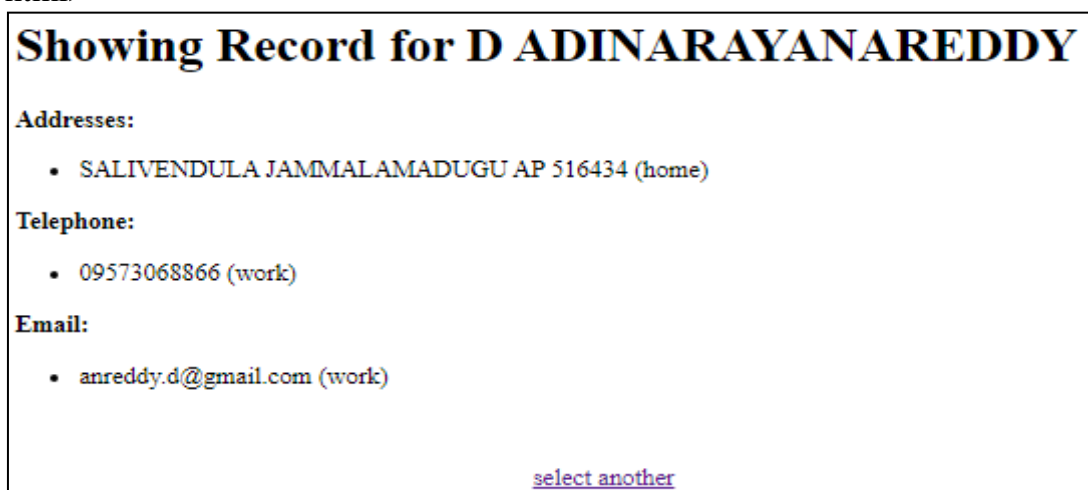
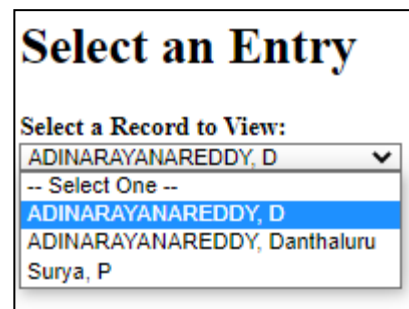
//get master_info
$get_master_sql = "SELECT concat_ws(' ', f_name, l_name) as display_name
FROM master_name WHERE id = '".$_POST["sel_id"].'";
$get_master_res = mysqli_query($mysqli, $get_master_sql) or
die(mysqli_error($mysqli));
while ($name_info = mysqli_fetch_array($get_master_res))
{
$display_name = stripslashes($name_info['display_name']);
}
$display_block = "<h1>Showing Record for ".$display_name."</h1>";
//free result
mysqli_free_result($get_master_res);
//get all addresses
$get_addresses_sql = "SELECT address, city, state, zipcode, type
FROM address WHERE master_id = '".$_POST["sel_id"].'";
$get_addresses_res = mysqli_query($mysqli, $get_addresses_sql) or
die(mysqli_error($mysqli));
if (mysqli_num_rows($get_addresses_res) > 0)
{
$display_block .= "<p><b>Addresses:</b><br/>
<ul>";
while ($add_info = mysqli_fetch_array($get_addresses_res))
{
$address = stripslashes($add_info['address']);
$city = stripslashes($add_info['city']);
$state = stripslashes($add_info['state']);
$zipcode = stripslashes($add_info['zipcode']);
$address_type = $add_info['type'];
$display_block .= "<li>$address $city $state $zipcode ($address_type)</li>";
}
$display_block .= "</ul>";
}
//free result
mysqli_free_result($get_addresses_res);
//get all tel
$get_tel_sql = "SELECT tel_number, type FROM telephone
WHERE master_id = '".$_POST["sel_id"].'";
$get_tel_res = mysqli_query($mysqli, $get_tel_sql) or die(mysqli_error($mysqli));
if (mysqli_num_rows($get_tel_res) > 0)
{
$display_block .= "<p><b>Telephone:</b><br/><ul>";
while ($tel_info = mysqli_fetch_array($get_tel_res))
{
$tel_number = stripslashes($tel_info['tel_number']);
$tel_type = $tel_info['type'];
$display_block .= "<li>$tel_number ($tel_type)</li>";
}
$display_block .= "</ul>";
}
//free result
mysqli_free_result($get_tel_res);

```

```

//get all email
$get_email_sql = "SELECT email, type FROM email
WHERE master_id = '".$_POST["sel_id"]."''";
$get_email_res = mysqli_query($mysqli, $get_email_sql) or
die(mysqli_error($mysqli));
if (mysqli_num_rows($get_email_res) > 0)
{
$display_block .= "<p><b>Email:</b><br/>
<ul>";
while ($email_info = mysqli_fetch_array($get_email_res))
{
$email = stripslashes($email_info['email']);
$email_type = $email_info['type'];
$display_block .= "<li>$email ($email_type)</li>";
}
$display_block .= "</ul>";
}
//free result
mysqli_free_result($get_email_res);
$display_block .= "<br/>
<p align='\"center\"'><a href='\"".$_SERVER["PHP_SELF"]."\">select
another</a></p>";
}
//close connection to MySQL
mysqli_close($mysqli);
?>
<html>
<head>
<title>My Records</title>
</head>
<body>
<?php echo $display_block; ?>
</body>
</html>

```



10. Write about creating the record addition mechanism?

The record deletion mechanism is virtually identical to the script used to view a record.

```

<!delentry.php?
<?php
$mysqli = mysqli_connect("localhost", "college", "", "testDB");
if (mysqli_connect_errno())
{
printf("Connect failed: %s\n", mysqli_connect_error());
exit();
}
if (!$_POST)
{
//haven't seen the selection form, so show it
$display_block = "<h1>Select an Entry</h1>";
//get parts of records
$get_list_sql = "SELECT id, CONCAT_WS(' ', l_name, f_name) AS display_name
FROM master_name ORDER BY l_name, f_name";
$get_list_res = mysqli_query($mysqli, $get_list_sql) or die(mysqli_error($mysqli));
if (mysqli_num_rows($get_list_res) < 1)
{
//no records
$display_block .= "<p><em>Sorry, no records to select!</em></p>";
}
else
{
//has records, so get results and print in a form
$display_block .= "
<form method=\"post\" action=\"".$_SERVER["PHP_SELF"]."\">
<p><b>Select a Record to Delete : </b><br>
<select name=\"sel_id\">
<option value=\"\">-- Select One --</option>";
while ($recs = mysqli_fetch_array($get_list_res))
{
$id = $recs['id'];
$display_name = stripslashes($recs['display_name']);
$display_block .= "<option value=\"".$id."\">".$display_name."</option>";
}
$display_block .= "
</select>
<p><input type=\"submit\" name=\"submit\" value=\"Delete Selected Entry\"></p>
</form>";
}
//free result
mysqli_free_result($get_list_res);
}
else if ($_POST)
{
//check for required fields
if ($_POST["sel_id"] == "")
{
header("Location: delentry.php");
exit;
}
}

```

```

//issue queries
$del_master_sql = "DELETE FROM master_name WHERE id =
    '$_POST["sel_id"].'";
$del_master_res = mysqli_query($mysqli, $del_master_sql) or
    die(mysqli_error($mysqli));

$del_address_sql = "DELETE FROM address WHERE id = '$_POST["sel_id"].'";
$del_address_res = mysqli_query($mysqli, $del_address_sql) or
    die(mysqli_error($mysqli));

$del_tel_sql = "DELETE FROM telephone WHERE id = '$_POST["sel_id"].'";
$del_tel_res = mysqli_query($mysqli, $del_tel_sql) or die(mysqli_error($mysqli));

$del_email_sql = "DELETE FROM email WHERE id = '$_POST["sel_id"].'";
$del_email_res = mysqli_query($mysqli, $del_email_sql) or
    die(mysqli_error($mysqli));
$display_block = "<h1>Record(s) Deleted</h1>
<p>Would you like to
<a href='\"".$_SERVER["PHP_SELF"]."\">delete another</a>?</p>";
}
?>
<html>
<head>
<title>My Records</title>
</head>
<body>
<?php echo $display_block; ?>
</body>
</html>

```

