School of Information and Computer Technology

Sirindhorn International Institute of Technology

Thammasat University

ITS351 Database Programming Laboratory

Laboratory #9: Data Manipulation II

Objective: - To learn how to retrieve data from PHP

- To learn how to delete data from PHP

1   Selecting data from MySQL to PHP

A flow chart illustrating the use of PHP together with a database to make a client/server web application is shown as followed:
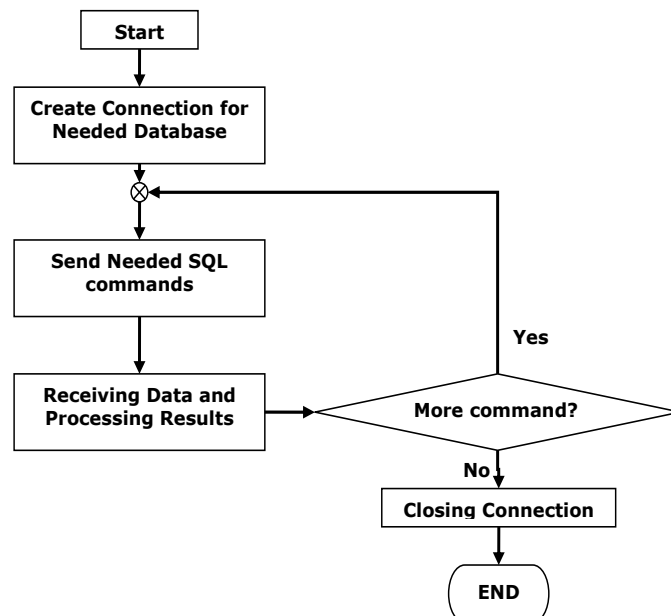
From the figure, we firstly have to establish a connection to the MySQL Server. After that, through the established connection, you can issue as many MySQL statements as needed to the MySQL server. In the end, the database connection is closed to free the used resource. The detail of each step in this work flow is described next.

1.1   Return ID from insertion

After the insertion code is successfully executed, the method `$insert_id` of class mysqli can return the ID generated by a query on a table with a column having the AUTO_INCREMENT attribute. For example,

```php
<?php
      $r=array("Idiot's Guide Book",1200);
      $q="INSERT INTO product(p_name, p_price)
VALUES('".$mysqli->real_escape_string($r[0])."', $r[1])";
      if(!$mysqli->query($q)){
             echo "INSERT failed. Error: ".$mysqli->error ;
      }
      $id = $mysqli->insert_id;
      echo "Record $id is inserted";
?>
```

We can use the method `$mysqli->real_escape_string()` which will properly escape MySQL special characters from the value "Idiot's Guide Book", which contains a single-quote.

$q will be

  "INSERT INTO product(p_name, p_price) VALUES('Idiot\'s Guide Book', 1200)"

(note the backslash in front of the single-quote). The added backslash signals the MySQL that the following character is an actual value, not part of the syntax. With this code, the insertion is successful. Then it returns id of last query of insertion to the screen.

1.2   Display Result Sets in a Table

Continuing from the example of "product" table, let us try to retrieve some data and display it in an HTML table. The following code displays products and their prices which are greater than 100.

```php
<?php
      $q="select p_name, p_price from product where p_price> 100; ";
      if($result=$mysqli->query($q)){
             echo '<table border="1">';
             echo '<tr><th>Name</th><th>Price</th></tr>';
             while($row=$result->fetch_array()){
                    echo "<tr>";
                    echo "<td>".$row['p_name']."</td>";
                    echo "<td>".$row['p_price']."</td>";
                    echo "</tr>";
             }
             echo '</table>';
             $result->free();
      }else{
             echo "Retrieval failed: ".$mysqli->error ;
      }
?>
```

The structure of the code is almost identical to the previous lab except that we now print the result in an HTML table. In the previous lab, `$row` is accessed by a numeric index. In fact, the method `fetch_array()` also allows the values in each row to be accessed by their column names as the keys in the returned associative array. In this particular example, `$row['p_name']` would give the same value as `$row[0]`. Notice that `$mysqli->error` also works for a SELECT query, and will give an error message on a failure.

After the code is executed, the following table is obtained.

Name	Price
Mouse	600
Printer	4000

1.3   Get the Number of Rows

There are many circumstances where, besides the actual result set, the number of rows in the result set is needed. The class `mysqli_result` has a property `num_rows` for this purpose. The following code demonstrates how to use it. We assume `$mysqli` has already been constructed.

```php
<?php
    $q="select p_id from product where p_name like 'P%'; ";
    if($result=$mysqli->query($q)){
        $count=$result->num_rows;
        echo "There are $count products starting with P.";
        $result->free();
    }else{
        echo "Query failed: ".$mysqli->error ;
    }
?>
```

In this example, we try to find the number of product names which start with 'P'. The number can be obtained by referring to `$result->num_rows`.

There is another way to get only the number of rows. That is to query "`select count(*) from Product where p_name like 'P%' `", and use `fetch_array()` to get the count value. If only the count is needed, then one may issue an SQL COUNT statement. However, if the actual result set is also needed, we recommend the first way which is to use `$result->num_rows` to get the count. In this way, both the result set and the count can be obtained.

1.4   Get the Number of Columns

Often, the number of columns is known in advance when the query is constructed. However, in the case that the query is dynamically constructed (i.e., columns to query depend on a user input), or the query has "*" for all columns, the number of columns may be unknown. The class `mysqli_result` has a property `field_count` for this purpose.

```php
<?php
    $q="select * from Product limit 1;";
    if($result=$mysqli->query($q)){
        $count=$result->field_count;
        echo "There are $count columns.";
        $result->free();
    }else{
        echo "Query failed: ".$mysqli->error ;
    }
?>
```

In this example, we try to find the number of columns (fields) in the "Product" table. On a success, "There are 3 columns" will be printed out.

1.5   Seek a Row in the Result Set

The object `mysqli_result` containing the result set works by maintaining an internal pointer which points to the current row. Rows in the set are retrieved by moving this pointer (by calling `$result->fetch_array()`) sequentially from the beginning to the end. However, in

Last Updated: 2/9/12

some cases, we may be interested in only a particular row in the result set. This is when the method `data_seek()` of class `mysqli_result` comes in handy. For example, we want to find the product which has the third lowest price.

```php
<?php
    $q='select p_name, p_price from product order by p_price limit 3;';
    if($result=$mysqli->query($q)){
        // Seek to the third row (row index starts from 0)
        $result->data_seek(2);
        $row=$result->fetch_array();
        echo $row['p_name']." has the third lowest price which is
".$row['p_price'];
        $result->free();
    }else{
        echo "Query failed: ".$mysqli->error;
    }
?>
```

In this example, we query the products and order them by their prices in ascending order. To get the product having the third lowest price, we move the internal pointer of `$result` to index 2 by using `$result->data_seek(2)`. So, the next fetch by `$result->fetch_array()` will give the result of the third row. After executed, the output of this code is "Mouse has the third lowest price which is 600".

2 Deleting data from MySQL via PHP

The DELETE statement is used to delete existing records in a table.

Syntax

DELETE FROM table_name

WHERE some_column=some_value

Note: Notice the WHERE clause in the DELETE syntax. The WHERE clause specifies which record or records that should be deleted. If you omit the WHERE clause, all records will be deleted!.

2.1 Delete the existing record

The DELETE query is very similar to the INSERT Query in the previous example. We need to choose a table, tell MySQL to perform the deletion, and provide the requirements that a record must have for it to be deleted.

```php
<?php
    $q="DELETE FROM product where p_id=5";
    if(!$mysqli->query($q)){
        echo "DELETE failed. Error: ".$mysqli->error ;
    }
?>
```

2.2   Delete via form in PHP

Now we are going to delete a row in the database. In this example, we use a product table which contains: name and price from previous example. We will be retrieving the data, making delete, and then delete the row in the database.

We are going to need to create two files in order to delete the data.

- viewinfo.php – We will retrieve all product data from the Database.

- delinfo.php – We will send id from viewinfo.php to this form and delete record in the database.

First create viewinfo.php (Extend from previous example)

```php
<?php
     $q="select * from product";
     if($result=$mysqli->query($q)){
          echo '<table border="1">';
          echo '<tr><th>Name</th><th>Price</th><th>Delete</th></tr>';
          while($row=$result->fetch_array()){
               echo "<tr>";
               echo "<td>".$row['p_name']."</td>";
               echo "<td>".$row['p_price']."</td>";
               echo "<td><a href='delinfo.php?id="
                    .$row['p_id']."'> Delete</a></td>";
               echo "</tr>";
          }
          echo '</table>';
          $result->free();
     }else{
          echo "Retrieval failed: ".$mysqli->error ;
     }
?>
```

Output result will be

Name	Price	Delete
Pencil	10	Delete
Eraser	5	Delete
Mouse	600	Delete
Printer	4000	Delete
Idiot's Guide Book	1200	Delete

`<a href='delinfo.php?id=".`
`$row['p_id']."'> Delete</a>`

When mouse over this
link ../delinfo.php?id=5

Delete menu will bring the program into delinfo.php file where it is used to delete the data which have been selected in the above form. delinfo.php file as follow:

Last Updated: 2/9/12

Second create delinfo.php

```php
<?php
    $p_id = $_GET['id'];

    $mysqli = new mysqli('localhost','root','root','staff');
    if($mysqli->connect_errno){
        echo $mysqli->connect_errno.": ".$mysqli->connect_error;
    }

    $q="DELETE FROM product where p_id=$p_id";

    if(!$mysqli->query($q)){
            echo "DELETE failed. Error: ".$mysqli->error ;
    }
    $mysqli->close();
    //redirect
    header("Location: viewinfo.php");
?>
```

Note that we must use "$_GET" to receive ID. That is all you need to allow a user to delete the information.

Worksheet

1. Import database named "STAFF" from given resource file in database folder. The following figure shows the structure of STAFF database. Note that all fields must set to allow NULL value except primary key.

USERGROUP Table

Field	Type	Length Values	Extra	Primary Key
USERGROUP_ID	INT		Auto_increment	Yes
USERGROUP_CODE	VARCHAR	50		
USERGROUP_NAME	VARCHAR	50		
USERGROUP_REMARK	VARCHAR	255		
USERGROUP_URL	VARCHAR	50		

USER Table

Field	Type	Length Values	Extra	Primary Key
USER_ID	INT		Auto_increment	Yes
USER_TITLE	INT			
USER_FNAME	VARCHAR	50		
USER_LNAME	VARCHAR	50		
USER_GENDER	INT			
USER_EMAIL	VARCHAR	50		
USER_NAME	VARCHAR	25		
USER_PASSWD	VARCHAR	25		
USER_GROUPID	INT			
DISABLE	INT			

Last Updated: 2/9/12

2. Write PHP code in "group.php" to check whether data is coming from "add_group.php" or not?

- If input data found, please insert it to table "USERGROUP" and retrieve data from "USERGROUP" table to display on the screen.
- If user direct to this page using menu, please retrieve data from "USERGROUP" table and display on the screen **(no insertion).**
- Note that you need to display total number of rows at the end of the table.

3. Write PHP code to delete the selected record that user want to delete in the "USERGROUP" table.

group.php (worksheet 1)

group.php (worksheet 2)

Last Updated: 2/9/12

group.php (worksheet 3)

Last Updated: 2/9/12

Exercise

1. Modify PHP code in "add_user.php" to select master data from "TITLE", "GENDER", and "USERGROUP" table in database "STAFF". For "TITLE" and "USERGROUP" Table, please display retrieving data in the form of combo box while "GENDER" display in the form of radio button. Note that value of these three form fields must be id from the database.

2. Modify PHP code in "user.php" to check whether data is coming from "add_user.php" or not?

- If input data found, please insert it to table "USER" and retrieve data from "USER" table to display on the screen instead of data coming from add_user.php.
- If user direct to this page using menu, please retrieve data from "USER" table and display on the screen **(no insertion).**
- Data that display in Title and User Group column must be name (retrieve name from TITLE and USERGROUP Table).
- Note that you need to display total number of rows at the end of the table.

3. Write PHP code to delete the selected record that user want to delete in the "USER" table.

user.php (Worksheet 1)

Last Updated: 2/9/12

user.php (Worksheet 2)

user.php (Worksheet 3)

Last Updated: 2/9/12