

# 資料庫管理系統

資應菁英班四甲

## 第 6 週 更新資料庫的資料

許雯綾

chiaohsu@nutc.edu.tw

參考資料：最新 PHP+MySQL+AJAX 網頁程式設計，第 2 版，施威銘研究室，旗標  
PHP7+MySQL+Ajax 網頁設計範例教本，第 5 版，陳會安，旗標

### 大 綱



6-1 新增資料



6-2 刪除資料



6-3 更新資料



6-4 課堂練習

## 6-1 新增資料

- 若要新增一筆記錄到資料庫，使用 `mysqli_query`(連線物件, INSERT 敘述) 函數來執行 INSERT 敘述。

(此範例用 POST 傳遞資訊)

```
<?php
include("mysql.inc.php");

//如果網頁表單的 name 與 qty 欄位都不是空字串
if (!empty( $_POST['name'] ) && !empty( $_POST['qty'] ) ){
    //將 name 與 qty 欄位值新增至 【inventory】 資料表
    $sql="INSERT inventory (書籍名稱, 數量) VALUES (' {$_POST['name']} ', '{$_POST['qty']} ' )";
    mysqli_query($conn, $sql);
}
?>
```

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>書籍存貨管理系統</title>
</head>
<body>
```

*<!--建立一個新增書籍的表單 -->*

```
<form method="post" action="<?php $_SERVER['PHP_SELF'] ?>">
    書名: <input name="name">
    數量: <input name="qty" style="width: 73px">
    <input name="submit" type="submit" value="新增">
</form>
```

```
<?php
```

*//使用【書籍名稱】排序, 查詢【inventory】資料表的所有資料*

```
$sql= "SELECT * FROM inventory ORDER BY 書籍名稱 ASC";
```

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

*//如果查到的記錄筆數大於 0, 便使用迴圈顯示所有資料*

```
if ( mysqli_num_rows($result) >0 ){
    echo "<hr><table border='1'>
        <tr><th>書籍名稱</th><th>數量</th></tr>";

    while ( $row = mysqli_fetch_array($result) ) {
        echo "<tr><td>{$row['書籍名稱']}</td>
            <td>{$row['數量']}</td></tr>";
    }
    echo '</table>';
}
?>
```

```
</body>
</html>
```

成功連結資料庫伺服器

書名:  數量:

書籍名稱	數量
HTML5參考手冊	135
Linux 架站實務	300
PHP 程式語言	100
Windows 使用手冊	500

取得目前的網址

成功連結資料庫伺服器

書名:  數量:

書籍名稱	數量
HTML5參考手冊	135
Linux 架站實務	300
PHP 程式語言	100
PHP+MySQL+Ajax	580
Windows 使用手冊	500

- inventory 資料表結構及內容如下：

型別 int, Primary Key,  
自動編號 (auto\_increment)

編號	書籍名稱	數量
1	PHP 程式語言	100
2	Linux 架站實務	300

型別 varchar(20)

型別 int

PS: 編號為的資料型別為“自動編號”，因此新增時不需要給值

● auto\_increment 自動編號 (流水編號)

一筆記錄新增時，就從「1」開始(或 001, 0001 000001, 視其編碼而定)編號，每筆新增記錄編號都累加 1，中間不間斷、不重複，就像 1，2，3，4，5，6.....一直延伸下去的自然數，其性質就像流水一樣(不間斷、不重複)，連續而不斷，故稱為流水編號。號碼一旦使用了，就不能再使用。

使用流水編號做為主鍵(PK)的好處有幾個：

1. 自動產生，auto increment 的序號不用程式另外定義。
2. 保持資料的單一性。每筆資料有唯一值，就算刪除以後也不會出現相同 id 的資料。
3. 與商業邏輯無關，所以在商業邏輯變動時比較不會影響 PK 的設計。
4. SQL 語法變得比較短
5. 數字當作比較條件會比字串或其他資料型態來得快。

流水編號為主鍵的缺點有：

1. 流水號 PK 對商業邏輯來說沒有意義，對系統使用者來，編號 0009527 的客戶是誰不是那麼直覺 (故通常流水編號 PK 只給資料庫和程式看的，不是給使用者看的。因此這個欄位大都不會在畫面上呈現給使用者看。跟使用者溝通還是都用商業邏輯的 key。)
2. 如果其他資料也有被設定為 PK 的需要(像是使用者編號為流水號，但是使用者帳號也是唯一)，就變成必須有兩個欄位要維護 PK 的特性。
3. 系統維護變複雜，例如若遇到資料合併，兩邊的流水編碼有重複時，必定有部分記錄要重編碼，衍生其他關聯表格要同步更新的問題。

- 如果需要取得剛才新增記錄的編號，可以使用 mysqli\_insert\_id()

例如：  
`mysqli_query($conn, $sql);` //執行 Insert 之後，取得新增的編號  
`$orderId = mysqli_insert_id ($conn);`  
`echo " 新增書籍編號：";`  
`echo $orderId;`

## 6-2 刪除資料

- 若要從資料庫中刪除一筆記錄，使用 `mysqli_query`(連線物件, `DELETE` 敘述) 函數來執行 `DELETE` 敘述。

### 6-2.php

```
<?php
include("mysql.inc.php");
//如果網頁表單的 name 與 qty 欄位都不是空字串
if (!empty($_POST['name']) && !empty($_POST['qty'])){
    //將 name 與 qty 欄位值新增至 【inventory】 資料表
    $sql="INSERT inventory (書籍名稱, 數量) VALUES ('".$_POST['name']."','".$_POST['qty']."')";
    mysqli_query($conn, $sql);
}
?>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>書籍存貨管理系統</title>
</head>
<body>
<body>
    <form method="post" action="<?php $_SERVER["PHP_SELF"] ?>">
        書名: <input name="name">
        數量: <input name="qty" style="width: 73px">
        <input name="submit" type="submit" value="新增">
    </form>

    <?php
    //使用【書籍名稱】排序, 查詢【inventory】資料表的所有資料
    $sql="SELECT * FROM inventory ORDER BY 書籍名稱 ASC";
    $result=mysqli_query($conn, $sql);

    //如果查到的記錄筆數大於 0, 便使用迴圈顯示所有資料
    if (mysqli_num_rows($result) > 0){
        echo "<hr><table border='1'>
            <tr><td>書籍名稱</td><td>數量</td></tr>";

        while ($row = mysqli_fetch_array($result)) {
            echo "<tr><td>{$row['書籍名稱']}</td>
                <td>{$row['數量']}</td>
                <td><a href=' 6-2-1.php?del={$row['編號']}'>刪除</a></td> //這裏不是透過 form 去傳遞
            </tr>";
        }
        echo '</table>';
    }
    ?>
</body>
</html>
```

成功連結資料庫伺服器

書名:  數量:

書籍名稱	數量	
HTML5參考手冊	135	<a href="#">刪除</a>
Linux 架站實務	300	<a href="#">刪除</a>
PHP 程式語言	100	<a href="#">刪除</a>
PHP+MySQL+Ajax	580	<a href="#">刪除</a>
PHP+MySQL+Ajax	580	<a href="#">刪除</a>
Windows 使用手冊	500	<a href="#">刪除</a>

## 6-2-1.php

```
<?php
header( 'Content-Type: text/html; charset=utf-8' );
include("mysql.inc.php");

//如果以 GET 方式傳遞過來的 del 參數不是空字串
if ( !empty($_GET['del']) ) {
    //將 del 參數所指定的編號的記錄刪除
    $sql="DELETE FROM inventory WHERE 編號 = '{$_GET['del']}' ";
    mysqli_query($conn, $sql);

    //取得被刪除的記錄筆數
    $rowDeleted = mysqli_affected_rows($conn);

    //如果刪除的筆數大於 0，則顯示成功，若否，便顯示失敗
    if ( $rowDeleted > 0 ) {
        echo "刪除成功";
    }
    else {
        echo "刪除失敗";
    }
}
?>
<p><a href="6-2.php">回系統首頁</a></p> //沒有帶參數
```

## 6-2.php

成功連結資料庫伺服器

書名:  數量:

書籍名稱	數量	
HTML5參考手冊	135	<a href="#">刪除</a>
Linux 架站實務	300	<a href="#">刪除</a>
PHP 程式語言	100	<a href="#">刪除</a>
PHP+MySQL+Ajax	580	<a href="#">刪除</a>
PHP+MySQL+Ajax	580	<a href="#">刪除</a>
Windows 使用手冊	500	<a href="#">刪除</a>

## 6-2.php

成功連結資料庫伺服器

書名:  數量:

書籍名稱	數量	
HTML5參考手冊	135	<a href="#">刪除</a>
Linux 架站實務	300	<a href="#">刪除</a>
PHP 程式語言	100	<a href="#">刪除</a>
PHP+MySQL+Ajax	580	<a href="#">刪除</a>
Windows 使用手冊	500	<a href="#">刪除</a>

← → ↻ 🏠 ⓘ localhost/bookstore/6-2-1.php?del=7

成功連結資料庫伺服器刪除成功

[回系統首頁](#)

## 6-3 更新資料

- 若要更新／修改資料庫中的記錄，使用 `mysqli_query`(連線物件, UPDATE 敘述) 函數來執行 UPDATE 敘述。

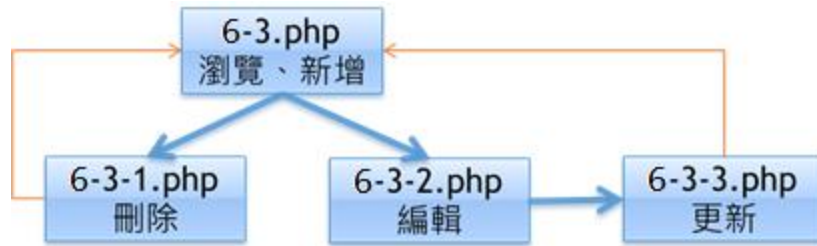
### 6-3.php

```
<?php
include("mysql.inc.php");
//如果網頁表單的 name 與 qty 欄位都不是空字串
if (!empty($_POST['name']) && !empty($_POST['qty'])){
    //將 name 與 qty 欄位值新增至 【inventory】 資料表
    $sql="INSERT inventory (書籍名稱, 數量) VALUES ('".$_POST['name']."','".$_POST['qty']."')";
    mysqli_query($conn, $sql);
}
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>書籍存貨管理系統</title>
</head>
<body>
    <form method="post" action="<?php $_SERVER["PHP_SELF"] ?>">
        書名: <input name="name">
        數量: <input name="qty" style="width: 73px">
        <input name="submit" type="submit" value="新增">
    </form>
    <?php
    //使用【書籍名稱】排序, 查詢 【inventory】 資料表的所有資料
    $sql="SELECT * FROM inventory ORDER BY 書籍名稱 ASC";
    $result=mysqli_query($conn, $sql);

    //如果查到的記錄筆數大於 0, 便使用迴圈顯示所有資料
    if (mysqli_num_rows($result) > 0){
        echo "<hr /><table border='1'>
            <tr><td>書籍名稱</td><td>數量</td></tr>";

        while ($row = mysqli_fetch_array($result)) {
            echo "<tr><td>{$row['書籍名稱']}</td>
                <td>{$row['數量']}</td>
                <td><a href='6-3-1.php?del={$row['編號']}'>
                    刪除</a></td>
                <td><a href='6-3-01.php?edit={$row['編號']}'>
                    編輯</a></td></tr>";
        }
        echo '</table>';
    }
    ?>
</body>
</html>
```





localhost/bookstore/6-3.php

成功連結資料庫伺服器

書名:  數量:

書籍名稱	數量		
HTML5參考手冊	135	<a href="#">刪除</a>	<a href="#">編輯</a>
Linux 架站實務	300	<a href="#">刪除</a>	<a href="#">編輯</a>
PHP 程式語言	100	<a href="#">刪除</a>	<a href="#">編輯</a>
PHP+MySQL+Ajax	580	<a href="#">刪除</a>	<a href="#">編輯</a>
Windows 使用手冊	500	<a href="#">刪除</a>	<a href="#">編輯</a>

localhost/bookstore/6-3.php

成功連結資料庫伺服器

書名:  數量:

書籍名稱	數量		
HTML5參考手冊	120	<a href="#">刪除</a>	<a href="#">編輯</a>
Linux 架站實務	300	<a href="#">刪除</a>	<a href="#">編輯</a>
PHP 程式語言	100	<a href="#">刪除</a>	<a href="#">編輯</a>
PHP+MySQL+Ajax	580	<a href="#">刪除</a>	<a href="#">編輯</a>
Windows 使用手冊	500	<a href="#">刪除</a>	<a href="#">編輯</a>

localhost/bookstore/6-3-2.php?edit=4

成功連結資料庫伺服器

書名: HTML5參考手冊 數量:

[回系統首頁](#)

localhost/bookstore/6-3-3.php

成功連結資料庫伺服器資料更新成功

[回系統首頁](#)

6-3-1.php (同 6-2-1.php, 修改返回系統首頁 6-3.php)

```

<?php
header('Content-Type: text/html; charset=utf-8');
include("mysql.inc.php");

//如果以 GET 方式傳遞過來的 del 參數不是空字串
if ( !empty($_GET['del']) ) {
    //將 del 參數所指定的編號的記錄刪除
    $sql="DELETE FROM inventory WHERE 編號 = '{$_GET['del']}' ";
    mysqli_query($conn, $sql);

    //取得被刪除的記錄筆數
    $rowDeleted = mysqli_affected_rows($conn);

    //如果刪除的筆數大於 0, 則顯示成功, 若否, 便顯示失敗
    if ( $rowDeleted > 0 ) {
        echo "刪除成功";
    }
    else {
        echo "刪除失敗";
    }
}
?>
  
```



<p><a href="6-3.php">回系統首頁</a></p>

## 6-3-2.php

<?php

```
include("mysql.inc.php");
```

//如果以 GET 方式傳遞過來的 edit 參數不是空字串

```
if (!empty($_GET['edit'])){
```

//查詢 edit 參數所指定編號的記錄, 從資料庫將原有的資料取出

```
$sql="SELECT * FROM inventory WHERE 編號 = '{$_GET['edit']}' ";
```

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

//將查詢到的資料放在 \$row 陣列

```
$row=mysqli_fetch_array($result);
```

```
}
```

```
else {
```

//如果沒有 edit 參數, 表示此為錯誤執行, 所以轉向回主頁面

```
header("Location:6-3.php");
```

```
}
```

>

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>書籍存貨管理系統</title>
```

```
</head>
```

```
<body>
```

<!--定義一個編輯資料的表單, 並且將編輯好的資料傳遞給 6-3-3.php 進行處理 -->

```
<form method="post" action="6-3-3.php">
```

```
書名: <?php echo $row['書籍名稱'];?>
```

<!--將原本的資料設定於<input>標籤的 value 參數, 如此【數量】欄位內就會自動填上原本的資料 -->

```
數量: <input name="qty" value="<?php echo $row['數量'];?>" style="width: 73px">
```

<!--將編號設定於隱藏的 <input> 標籤, 以便將編號數字傳遞給 6-3-3.php -->

```
<input name="id" type="hidden" value="<?php echo $row['編號'];?>"
```

```
<input name="submit" type="submit" value="送出">
```

```
</form>
```

```
<p><a href="6-3.php">回系統首頁</a></p>
```

```
</body>
```

```
</html>
```



編號用 hidden 的方式傳送出去



### 6-3-3.php

```
<?php
header('Content-Type: text/html; charset=utf-8');
include("mysql.inc.php");

//如果以 POST 方式傳遞過來的 id, qty 參數都不是空字串
if ( !empty($_POST['id']) && !empty($_POST['qty']) ){
    //更新 id 參數所指定編號的記錄
    $sql="UPDATE inventory
        SET 數量 = '{$_POST['qty']}'
        WHERE 編號 = '{$_POST['id']}' ";

    mysqli_query($conn, $sql);
}

//取得被更新的記錄筆數
$rowUpdated=mysqli_affected_rows($conn);

//如果更新的筆數大於 0，則顯示成功，若否，便顯示失敗
if ($rowUpdated > 0){
    echo "資料更新成功";
}
else {
    echo "更新失敗，或者您輸入的資料與原本相同";
}
?>
<p><a href="6-3.php">回系統首頁</a></p>
```

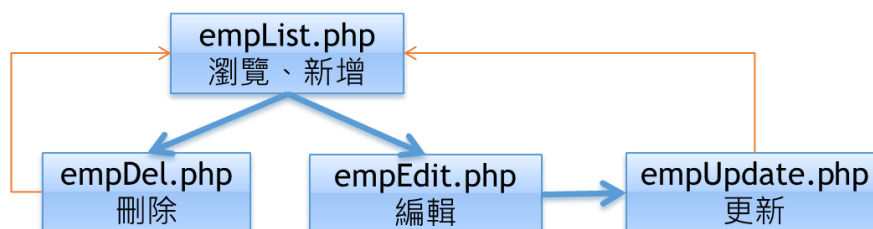
localhost/bookstore/6-3-3.php

成功連結資料庫伺服器資料更新成功

[回系統首頁](#)

## 6-4 課堂練習

- 參考前幾章的講義，完成員工資料管理的相關功能頁面。
- 各頁面網址如下所示。
- 其中 empList.php 必須含分頁、排序、限制查詢筆數，以及新增等功能。



- 完成後，請將測試的執行畫面貼在 Word 檔，連同網站資料夾，一併壓縮後上傳至智慧大師。
- 上傳的資料包括：
  - 網站資料夾：上述四個 php 檔案、mysql.inc.php
  - bookstore.sql
  - 測試畫面 Word 檔