

此投影片為博碩圖書有限公司提供，僅允許學生私下學習用，請勿複製轉載，以免侵犯智財權

---



# 章節大綱

8-1 PHP網頁存取MySQL流程觀念

8-2 PHP網頁取得資料表查詢結果

8-3 PHP取得SQL指令的執行狀態

備註：可依進度點選小節

# 8-1 PHP網頁存取MySQL流程觀念

---



PHP網頁存取MySQL

開啟與關閉MySQL資料庫連結

開啟指定資料庫

操作資料表

# PHP網頁存取MySQL(1/5)

---



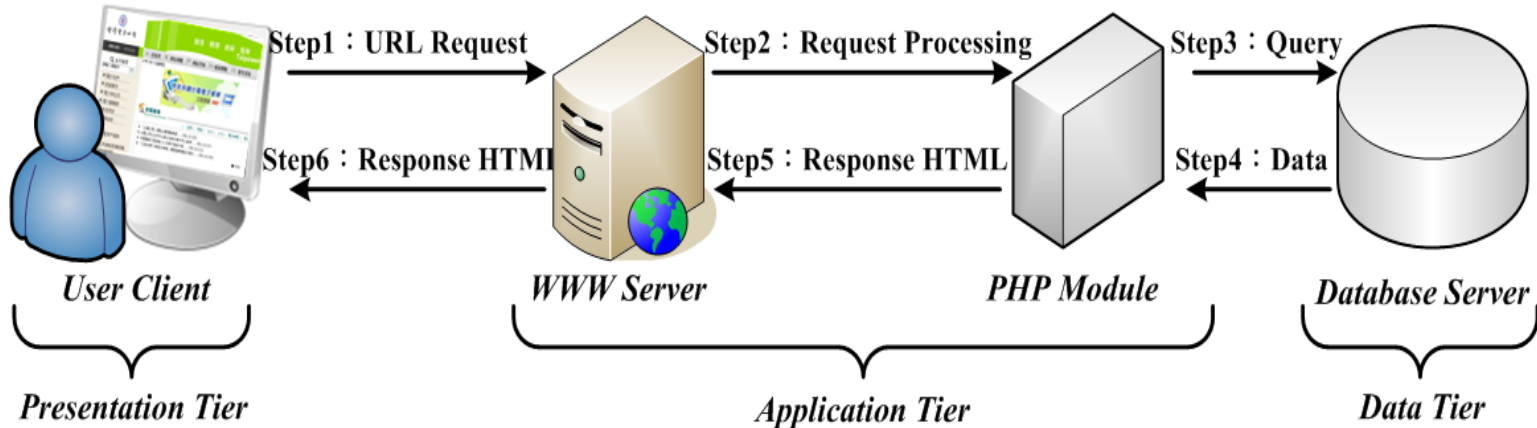
前面章節敘述如何建立資料庫、資料表，以及使用SQL 指令（新增、刪除、修改、查詢）來操作資料表的內容

接下來，本章節將要說明如何使用PHP 程式來存取MySQL 資料庫裡的內容

# PHP網頁存取MySQL(2/5)



下圖是PHP 網路存取MySQL 資料庫資料的運作流程



# PHP網頁存取MySQL(3/5)



在過程中，PHP網頁要存取MySQL資料庫的資料，它必須先連結資料庫管理系統，再來指定資料庫，接著送出SQL指令再取得執行結果，最後關閉資料庫連線

在編輯存取MySQL資料庫資料的PHP網頁程式時可以分成5個步驟，並且每一個步驟都有它專屬的函式，如下圖

# PHP網頁存取MySQL(4/5)



# PHP網頁存取MySQL(5/5)



若將這五大函式寫入PHP網頁程式，其程式如下

PHP原始檔

```
<html>
<head>
<title>PHP & SQL Server</title>
</head>
<body>
<?php
$servername='localhost,1443';
$username='sqlserver';
$password='sqlserver';
$connection = mysql_connect($servername,$username,$password);
mysql_select_db('tempdb', $connection);
$sql="SELECT * FROM power";
$result=mysql_query($sql);
for($i = 0; $i < mysql_num_rows( $result); $i++)
{
    $row = mysql_fetch_row($result);
    $WScript->Echo ("Row ".$i."\n");
    $WScript->Echo ("Id: ".$row[0]."\n");
    $WScript->Echo ("title: ".$row[1]."\n");
    $WScript->Echo ("date: ".$row[2]."\n");
    $WScript->Echo ("time: ".$row[3]."\n");
    $WScript->Echo ("watt: ".$row[4]."\n");
    $WScript->Echo ("type: ".$row[5]."\n");
}
mysql_close($connection);
?>
</body>
</html>
```

選擇資料庫

送出 SQL 指令

連接資料庫管理系統

取出結果

關閉資料庫連線



# 開啟與關閉MySQL資料庫連結(1/5)



從上節得知，若要使用MySQL資料庫的資源，首先要使用MySQL中建置好的帳號和密碼（請參考1-2節的step8）登入MySQL伺服器，建立連線。這在PHP程式中可以呼叫mysql\_connect()函數來建立與MySQL資料庫之間的連線

當從資料庫中拿到所需要的資料後，也就是，不再需要存取MySQL資料庫時，我們可以呼叫mysql\_close()函數來釋放連結資料庫資源所佔用的記憶體

其使用函數格式如下：

# 開啟與關閉MySQL資料庫連結(2/5)



- 建立MySQL資料庫的連結

```
mysql_connect(MySQL伺服器位址, 帳號, 密碼);
```

- 關閉MySQL資料庫的連結

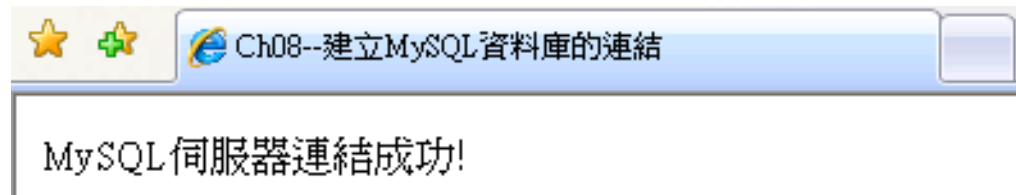
```
mysql_close(mysql_connect(MySQL伺服器位址, 帳號, 密碼));
```

# 開啟與關閉MySQL資料庫連結(3/5)



## 範例

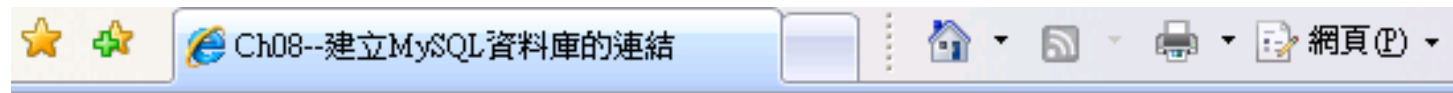
```
<?php
    $db_link = mysql_connect("localhost", "root", "1234"); //開啟MySQL資料庫
    if(!$db_link){
        echo "MySQL伺服器連結失敗!";
    }
    else{
        echo "MySQL伺服器連結成功!";
        mysql_close($db_link); //關閉MySQL資料庫
    }
?>
```



# 開啟與關閉MySQL資料庫連結(4/5)



若我們提供的帳號、密碼有錯誤時，那麼網頁上便會出現PHP 的錯誤訊息，如下圖所示：



```
Warning: mysql_connect() [function.mysql-connect]: Access denied for user  
'root'@'163.26.231.216' (using password: YES) in C:\AppServ\www\ch08  
\mysql_connect.php on line 2  
MySQL伺服器連結失敗!  
Warning: mysql_close(): supplied argument is not a valid MySQL-Link resource in  
C:\AppServ\www\ch08\mysql_connect.php on line 7
```

# 開啟與關閉MySQL資料庫連結(5/5)



上述的錯誤訊息會提供錯誤提示。在程式開發的階段，這些訊息可以幫助開發者修改程式錯誤

若要避免錯誤訊息的出現，我們可以使用「@」來抑制錯誤訊息的產生，如下

```
@mysql_connect(MySQL伺服器位址, 帳號, 密碼);
```

# 開啟指定資料庫(1/2)



當建立MySQL伺服器的連線後，我們必須選擇要開啟的資料庫。

下列為開啟資料庫的使用函數：

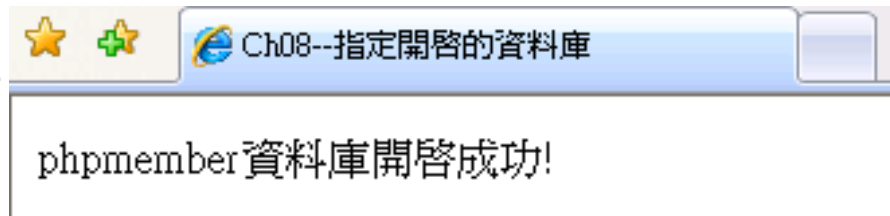
```
mysql_select_db(“資料庫名稱” );
```

# 開啟指定資料庫(2/2)



## 範例

```
<?php
    $db_link = @mysql_connect("localhost", "root", "wdc") or die("MySQL伺服器
    連結失敗!<br>");
    //如果連結失敗，則終止程式執行，並顯示連結失敗的訊息
    $select_db = @mysql_select_db("phpmember");
    if(!$select_db)
        die("無法開啟資料庫!<br>");
    else
        echo "phpmember資料庫開啟成功!";
    mysql_close($db_link);
?>
```



# 操作資料表(1/2)



PHP 程式成功連結MySQL 伺服器並開啟資料庫後，便可進行新增、刪除、修改、查詢資料表的內容（請參考第七章SQL指令教學）。設定好SQL 指令後，我們只要利用mysql\_query() 函數來執行SQL 指令即可在連結的資料庫內執行這些操作



# 操作資料表(2/2)



## (1) 新增—insert

```
$sql_query="INSERT INTO '資料表名稱' ('欄位名1', ...) VALUES ('值1', ...)";  
mysql_query($sql_query);
```

## (2) 刪除—delete

```
$sql_query="DELETE FROM '資料表名稱' WHERE 條件式";  
mysql_query($sql_query);
```

## (3) 修改—update

```
$sql_query="UPDATE '資料表名稱' SET '欄位名1'='值1', ... where 條件式";  
mysql_query($sql_query);
```

## (4) 查詢—select

```
$sql_query="SELECT * FROM '資料表名稱' ";  
mysql_query($sql_query);
```

## 8-2 PHP網頁取得資料表查詢結果

---



讀取查詢結果函數

`mysql_result()` 函式－取得查詢結果

`mysql_db_name()` 函式－取得查詢結果

`mysql_fetch_row()` 函式－取得查詢結果

`mysql_fetch_array()` 函式－取得查詢結果

`mysql_fetch_assoc()` 函式－取得查詢結果

# 讀取查詢結果函數



當用 `mysql_query()` 函式執行 `select` 指令後，可以使用 `mysql_fetch_row()` 函式來取得查詢結果。然而，取得查詢結果函式並非只有上述一種

下表為常用的讀取查詢結果函式：

函式	特徵
<code>mysql_result()</code>	一次只能取得一個欄位的內容。
<code>mysql_db_name()</code>	一次只能取得一個欄位的內容。
<code>mysql_fetch_row()</code>	一次可以取得一筆(列)資料，使用欄位序號來取得欄位內容。
<code>mysql_fetch_array()</code>	一次可以取得一筆(列)資料，可以使用欄位序號或欄位名稱來取得欄位內容。
<code>mysql_fetch_assoc()</code>	一次可以取得一筆(列)資料，使用欄位名稱來取得欄位內容。

# mysql\_result() 函式－取得查詢結果(1/2)

`mysql_result()` 函式用來取得查詢結果，但是它一次只能取得一個欄位的內容。下列為 `mysql_result()` 函式的語法：




```
mysql_result(resource result, int row[, mixed field]);
```

- **resource result**：資源識別字。
- **int row**：記錄的序號；0表示第一筆(列)資料，1表示第二筆(列)記錄。
- **mixed field**：欄位的名稱或欄位的序號；0表示第一個欄位，1表示第二個欄位，以此類推。另外，指定欄位名稱亦可。

# mysql\_result()函式－取得查詢結果(2/2)

## 範例

```
<?php
$db_link = @mysql_connect("localhost", "root", "1234")
    or die("MySQL伺服器連結失敗!<br>");
//如果連結失敗，則終止程式執行，並顯示連結失敗的訊息
$select_db = @mysql_select_db("member");
if(!$select_db)
{
    die("無法開啟資料庫!<br>");
} else {
    echo "member資料庫開啟成功!<br>";
    $sql_query="SELECT * FROM member";
    $result = mysql_query($sql_query);
    echo "第五筆資料的第3個欄位內容".mysql_result($result,4,2)."<br>";
    echo "第五筆資料的".mysql_field_name($result,2)."欄位內容為".
        mysql_result($result,4,account);
}
mysql_close($db_link);
?>
```

   用mysql\_result()函式取得某一欄位的內容

member資料庫開啟成功!

第五筆資料的第3個欄位內容為：ivy

第五筆資料的account欄位內容為：ivy

# mysql\_db\_name() 函式－取得查詢結果



`mysql_db_name( )` 函式的用法與 `mysql_result( )` 函式相同，一次只能取得一個欄位的內容。下列為 `mysql_db_name( )` 函式的語法：

```
mysql_db_name(resource result, int row[, mixed field]);
```

- **resource result**：資源識別字。
- **int row**：記錄的序號；0表示第一筆(列)資料，1表示第二筆(列)記錄。
- **mixed field**：欄位的名稱或欄位的序號；0表示第一個欄位，1表示第二個欄位，以此類推。另外，指定欄位名稱亦可。

## mysql\_fetch\_row()函式－取得查詢結果(1/3)

`mysql_fetch_row()` 函式也是用來取得查詢結果的函式，與 `mysql_result()` 不同之處在於它一次可以取得一筆資料，並將記錄指標移到下一筆資料

`mysql_fetch_row()` 函式將查詢結果之單列拆到陣列變數中，所以它是以欄位序號取得，0 表示第一個欄位，1 表示第二個欄位，以此類推

下列為 `mysql_fetch_row()` 函式的語法：

```
mysql_fetch_row(resource result);
```

## mysql\_fetch\_row() 函式－取得查詢結果(2/3)

一般而言，mysql\_fetch\_row( ) 函式通常是搭配陣列使用

假設總共有 3 筆查詢資料，我們分別用陣列 row0、row1、row2 將查詢結果存放在陣列裡。若要顯示第3 筆資料的第5 個欄位，其語法如下：

```
$row0 = mysql_fetch_row(resource result);  
$row1 = mysql_fetch_row(resource result);  
$row2 = mysql_fetch_row(resource result);  
  
echo $row3[4];
```



# mysql\_fetch\_row() 函式 — 取得本給結果 (2/2)

## 範例

id	name	account	password	sex	birthday	email	url	telephone	address
5	賴勝恩	shane	7790ffb26cf6409e707	男	1984-06-20	shane@superstar.com	NULL	0946820035	台南市南台街13巷200號

```
<?php
```

```
$db_link = @mysql_connect("localhost", "root", "1234")
    or die("MySQL 伺服器連結失敗!<br>");
//如果連結失敗，則終止程式執行，並顯示連結失敗的訊息
$select_db = @mysql_select_db("member");
$sql_query="SELECT * FROM member WHERE account = 'shane'";
$result = mysql_query($sql_query);

echo "<table border = '1'><tr align='center'>";
for($i=0; $i<mysql_num_fields($result); $i++)
{
    echo "<td>".mysql_fetch_field($result, $i)->name."</td>";
}
echo "</tr>";

while($row = mysql_fetch_row($result))
{
    echo "<tr>";
    for($j=0; $j<mysql_num_fields($result); $j++)
    {
        echo "<td>$row[$j]</td>";
    }
    echo "<tr/>";
}

mysql_close($db_link);
```

```
?>
```

## mysql\_fetch\_array()函式－取得查詢結果(1/3)



`mysql_fetch_array()` 函式與 `mysql_fetch_row()` 函式相似。不同之處在於 `mysql_fetch_array()` 函式可以使用欄位序號或欄位名稱取得欄位內容。下列為 `mysql_fetch_row()` 函式的語法：

```
mysql_fetch_array(resource result [, int result_type]);
```

■ `resource result`：資源識別字。

■ `int result_type`：指定取得欄位內容的方式，但可以不指定。分別的參數值有 `MYSQL_NUM`（使用欄位序號取得欄位內容）、`MYSQL_ASSOC`（使用欄位名稱取得欄位內容）、`MYSQL_BOTH`（使用欄位序號及欄位名稱取得欄位內容）。

## mysql\_fetch\_array()函式－取得查詢結果(2/3)



### 範例(1/2)

```
<?php
$db_link = @mysql_connect("localhost", "root", "1234")
           or die("MySQL伺服器連結失敗!<br>");
//如果連結失敗，則終止程式執行，並顯示連結失敗的訊息
$select_db = @mysql_select_db("member");
$sql_query="SELECT * FROM member WHERE account = 'shane'";
$result = mysql_query($sql_query);
echo "<table border = '1'><tr align='center'>";
for($i=0; $i<mysql_num_fields($result); $i++)
{
    echo "<td>".mysql_fetch_field($result, $i)->name."</td>";
}
echo "</tr>";
```

## mysql\_fetch\_array()函式－取得查詢結果(3/3)



### 範例(2/2)

```
while($row = mysql_fetch_array($result,MYSQL_ASSOC))
{
    echo "<tr>";
    echo "<td>".$row["id"]."</td>";
    echo "<td>".$row["name"]."</td>";
    echo "<td>".$row["account"]."</td>";
    echo "<td>".$row["password"]."</td>";
    echo "<td>".$row["sex"]."</td>";
    echo "<td>".$row["birthday"]."</td>";
    echo "<td>".$row["email"]."</td>";
    echo "<td>".$row["url"]."</td>";
    echo "<td>".$row["telephone"]."</td>";
    echo "<td>".$row["address"]."</td>";
    echo "<tr/>";
}
mysql_close($db_link);
```

?>



用mysql\_fetch\_array()函式取得某一筆資料的內容



id	name	account	password	sex	birthday	email	url	telephone	address
5	賴勝恩	shane	7790fbb26cf6409e707	男	1984-06-20	shane@superstar.com	NULL	0946820035	台南市南台街13巷200號

## mysql\_fetch\_assoc() 函式－取得查詢結果(1/3)



`mysql_fetch_assoc()` 函式與 `mysql_fetch_row()` 函式相似，但是它是  
以欄位名稱來取得欄位內容。下列為 `mysql_fetch_assoc()` 函式的語  
法：

```
mysql_fetch_assoc(resource result);
```

■ **resource result**：資源識別字。

■ **int result \_type**：指定取得欄位內容的方式，但可以不指定。分別的參數值有 **MYSQL\_NUM**（使用欄位序號取得欄位內容）、**MYSQL\_ASSOC**（使用欄位名稱取得欄位內容）、**MYSQL\_BOTH**（使用欄位序號及欄位名稱取得欄位內容）。

## mysql\_fetch\_assoc()函式－取得查詢結果(2/3)



### 範例(1/2)

```
<?php
$db_link = @mysql_connect("localhost", "root", "1234")
           or die("MySQL伺服器連結失敗!<br>");
//如果連結失敗，則終止程式執行，並顯示連結失敗的訊息
$select_db = @mysql_select_db("member");
$sql_query="SELECT * FROM member WHERE account = 'shane'";
$result = mysql_query($sql_query);
echo "<table border = '1'><tr align='center'>";
for($i=0; $i<mysql_num_fields($result); $i++)
{
    echo "<td>".mysql_fetch_field($result, $i)->name."</td>";
}
echo "</tr>";
```

## mysql\_fetch\_assoc()函式－取得查詢結果(3/3)



### 範例(2/2)

```
while($row = mysql_fetch_assoc($result))
{
    echo "<tr>";
    echo "<td>".$row["id"]."</td>";
    echo "<td>".$row["name"]."</td>";
    echo "<td>".$row["account"]."</td>";
    echo "<td>".$row["password"]."</td>";
    echo "<td>".$row["sex"]."</td>";
    echo "<td>".$row["birthday"]."</td>";
    echo "<td>".$row["email"]."</td>";
    echo "<td>".$row["url"]."</td>";
    echo "<td>".$row["telephone"]."</td>";
    echo "<td>".$row["address"]."</td>";
    echo "<tr/>";
}
mysql_close($db_link);
```

?>



用mysql\_fetch\_assoc()函式取得某一筆資料的內容

id	name	account	password	sex	birthday	email	url	telephone	address
5	賴勝恩	shane	7790ffb26cf6409e707	男	1984-06-20	shane@superstar.com	NULL	0946820035	台南市南台街13巷200號

# 8-3 PHP取得SQL指令的執行狀態



---

取得SQL指令的執行狀態



# 取得SQL指令的執行狀態(1/6)

若想知道執行SQL 指令後會有多少筆記錄與欄位受到影響，我們可以使用mysql\_num\_rows( )、mysql\_num\_fields( )、mysql\_affected\_rows( ) 等函式

下表為這三個函式的特徵介紹

函式	特徵
mysql_num_rows()	適用於select指令，它會回傳返回列的數目
mysql_num_fields()	適用於select指令，它會回傳被篩選出來的欄位數目
mysql_affected_rows()	適用於insert、delete以及update指令，它會回傳受到該指令影響的記錄數目

# 取得SQL指令的執行狀態(2/6)

`mysql_num_rows()` 函式適用於 `select` 指令，它會回傳返回列的數目（即被篩選出來的記錄的資料筆數）

下列為 `mysql_num_rows()` 函式的語法，其參數 `resource result` 為資源識別字

```
mysql_num_rows(resource result);
```

# 取得SQL指令的執行狀態(3/6)

`mysql_num_fields( )` 函式適用於 `select` 指令，它會回傳被篩選出來的欄位數目

下列為 `mysql_num_fields( )` 函式的語法，其參數 `resource result` 為資源識別字

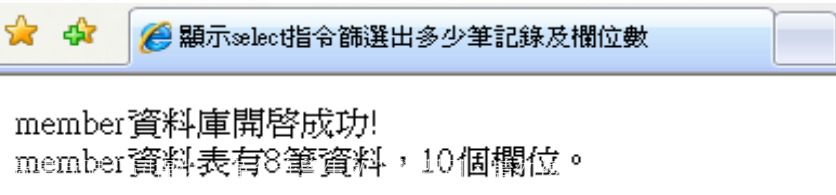
```
mysql_num_fields(resource result);
```

# 取得SQL指令的執行狀態(4/6)

範例：顯示select指令篩選出多少筆記錄及欄位數

```
<?php
$db_link = @mysql_connect("localhost", "root", "l307")
    or die("MySQL伺服器連結失敗!<br>");
//如果連結失敗，則終止程式執行，並顯示連結失敗的訊息
$select_db = @mysql_select_db("member");
if(!$select_db)
{
    die("無法開啟資料庫!<br>");
}
else{
    echo "member資料庫開啟成功!<br>";
    $sql_query="SELECT * FROM member";
    $result = mysql_query($sql_query);
    echo "member資料表有".mysql_num_rows($result)."筆資料";
    echo "，".mysql_num_fields($result)."個欄位。";
}
mysql_close($db_link);
```

?>



# 取得SQL指令的執行狀態(5/6)

`mysql_affected_row( )` 函式適用於insert、delete 以及update 指令，它會回傳受到該指令影響的記錄數目

下列為`mysql_affected_row( )` 函式的語法，其參數resource link\_identifier 為連接識別字

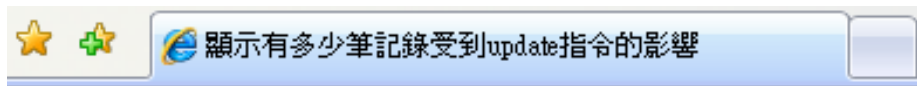
```
mysql_affected_rows([resource link_identifier]);
```

# 取得SQL指令的執行狀態(6/6)

範例：顯示有多少筆記錄受到update指令的影響

```
<?php
$db_link = @mysql_connect("localhost", "root", "l307")
    or die("MySQL伺服器連結失敗!<br>");
//如果連結失敗，則終止程式執行，並顯示連結失敗的訊息
$select_db = @mysql_select_db("member");
if(!$select_db)
{
    die("無法開啟資料庫!<br>");
}else{
    echo "member資料庫開啟成功!<br/>";
    $sql_query="UPDATE member SET sex = '女' WHERE id > 5 AND id < 16";
    mysql_query($sql_query);
    echo "有".mysql_affected_rows()."個資料受到UPDATE指令影響";
}
mysql_close($db_link);
```

?>



member資料庫開啟成功!  
有3個資料受到UPDATE指令影響

# 本章結束

## Q&A討論時間

---