

第十五章

使用mysql與pdo進行連線處理

閱讀本章之前



- 自從PHP3搭配MySQL進行資料存取，mysql元件提供的各式函數一直是兩者溝通的方法。
- 從PHP 5時代開始，mysql元件提供的各式函數開始出現功能不足且速度緩慢等問題。
- PHP5於mysql元件擴增新的功能，稱之為mysqli，提供了參數查詢與多個SQL語法支援。
- PHP7不再支援mysql元件，所以如若進行PHP版本升級，請留意資料庫連結存取語法的重大變更。
- mysqli提供一般與物件導向兩種方式操作，基於未來與物件導向相關技術結合，所以本章節只針對物件導向的操作進行介紹。

與Server連線與選擇資料庫



- 我們必須先與資料庫伺服器進行連線再來選資料庫，這一段過程須經過兩段驗證：
 - 第一階段是判斷帳號、密碼與host主機。
 - 第二階段再判斷是否有權限操作指定的資料庫。

與Server連線

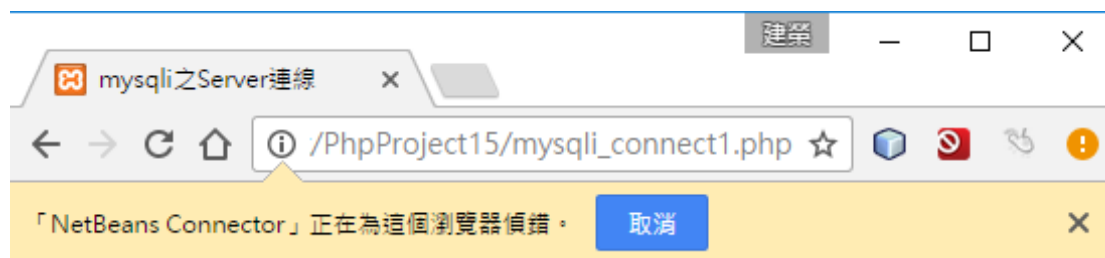


- 透過建立mysqli物件的方式與資料庫主機進行連線與選取資料庫，建立mysqli連線物件的方式如下：
 - `mysqli連線物件=new mysqli('主機','帳號','密碼','資料庫');`
- 如果mysqli連線物件操作時產生錯誤，將會造成該物件的connect_error成立，藉由該物件的connect_error是否成立來判斷資料庫連線狀況。

與Server連線



- 請嘗試使用mysql_connect()函數與資料庫伺服器連線及選取pcschool資料庫（檔案名稱：「PhpProject15」資料夾內「mysql_connect1.php」）。

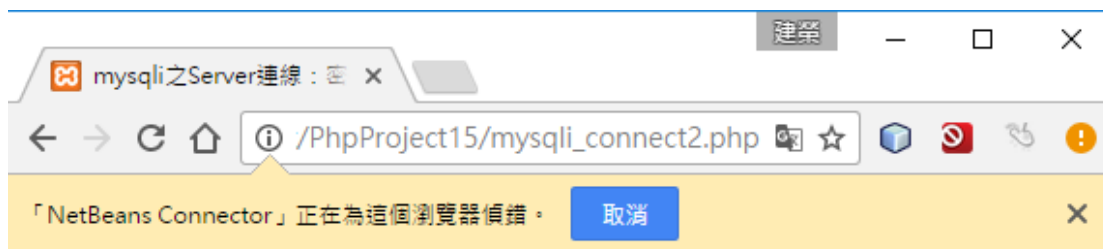


資料庫伺服器連結成功及資料庫開啟成功

與Server連線



- 如果連線失敗，會有什麼訊息呢？請嘗試連線到主機的密碼請設定為錯誤密碼，讓連線產生錯誤（檔案名稱：「PhpProject15」資料夾內「mysqli_connect2.php」）：

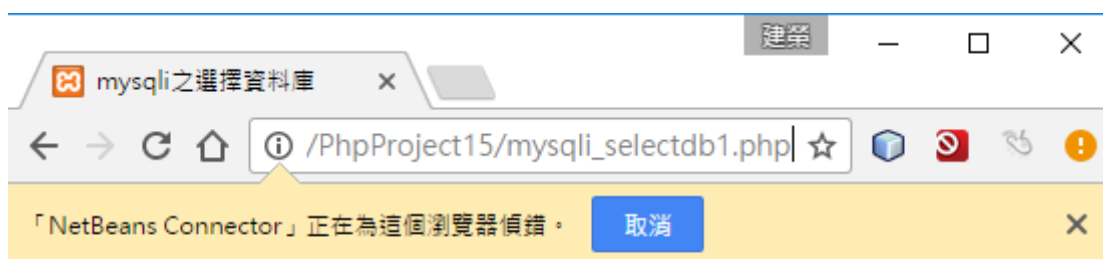


Warning: mysqli::__construct(): (HY000/1045): Access denied for user 'root'@'localhost' (using password: YES) in C:\xampp\htdocs\PhpProject15\mysqli_connect2.php on line 7
連結錯誤訊息: Access denied for user 'root'@'localhost' (using password: YES)

變更資料庫



- 您可透過mysqli物件的select_db()方法替換選擇的資料庫，請嘗試於連線成功後進行變更資料庫為test1（「PhpProject15」資料夾內「mysqli_selectdb1.php」）。



資料庫伺服器連結成功及資料庫開啟成功
切換至test資料庫，開啟成功!

直接執行SQL語法

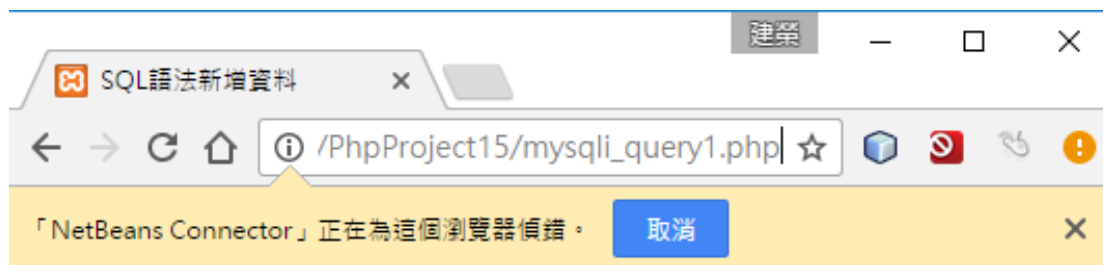


- 順利連結到資料庫之後，接著我們就可以針對資料庫內的資料表與眾多資料直接執行SQL語法進行新增、刪除、更新與查詢的作業。
- 直接執行SQL與法會有SQL injection風險，建議針對資料存取動作進行過濾動作較為保險。

執行變更資料表動作



- 首先我們執行SQL語法變更資料表內容，我們將嘗試於資料表新增資料（檔案名稱：「PhpProject15」資料夾內「mysqli_query1.php」）。



資料庫伺服器連結成功及資料庫開啟成功
新增成功

執行SQL語法時的錯誤操作



- 執行SQL語法時可能會出現若干錯誤，首先是編碼問題，請您將mysqli_query1.php另存新檔為mysqli_query2.php，接著請您修改以下部分(其他相同)，主要是第14行改變數內容為中文，而第17行加上註解。

The screenshot shows a web browser window with the address bar displaying `/PhpProject15/mysqli_query2.php`. A yellow notification bar at the top states: "「NetBeans Connector」正在為這個瀏覽器偵錯。" (NetBeans Connector is debugging this browser). Below the notification, a message reads: "資料庫伺服器連結成功及資料庫開啟成功 新增成功" (Database server connection successful and database opened successfully. New record added successfully).










Below the message is a table with the following data:

				id	username	password
<input type="checkbox"/>		編輯		複製		刪除
				1	admin	php
<input type="checkbox"/>		編輯		複製		刪除
				2	admin2	mysql
<input type="checkbox"/>		編輯		複製		刪除
				3	try	again
<input type="checkbox"/>		編輯		複製		刪除
				4	ā,æ-ı	again

執行SQL語法時的錯誤操作



- 第17行的mysqli物件執行query動作，動作內容為「set names utf8」，代表設定執行過程中的編碼為utf8，這部分請不要遺漏，免得造成資料為亂碼。
- 若將第17行註解取消，重新執行一次，中文就不再是亂碼。

<div>← T →</div>					id	username	password
<input type="checkbox"/>		編輯	 複製	 刪除	1	admin	php
<input type="checkbox"/>		編輯	 複製	 刪除	2	admin2	mysql
<input type="checkbox"/>		編輯	 複製	 刪除	3	try	again
<input type="checkbox"/>		編輯	 複製	 刪除	4	ä,æ-†	again
<input type="checkbox"/>		編輯	 複製	 刪除	5	中文	again

執行SQL語法時的錯誤操作

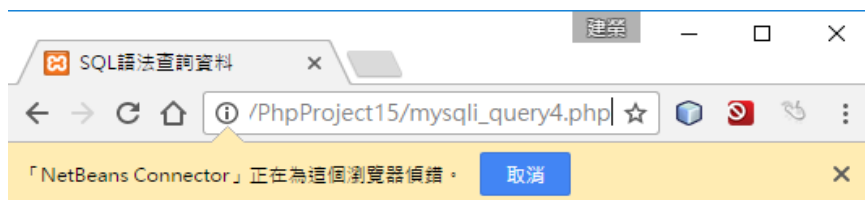


- SQL語法執行過程中若有狀況，錯誤資訊將儲存於mysqli物件的error成員，若執行產生錯誤時可顯示error內容，不過假如是對外服務網頁，建議預設不要顯示error內容，避免被有心人士利用。
- 請您將mysqli_query2.php另存新檔為mysqli_query3.php，接著請您修改以下部分(其他相同)，主要是第16行SQL語法內沒有加上單引號，SQL語法內字串資料需要前後加上單引號，否則會產生錯誤訊息。第22行顯示SQL語法操作錯誤訊息。

執行查詢語法



- 請嘗試以mysqli物件的query()動作查詢「ch13」資料庫的「customers」資料表內前十筆資料的「CustomerID」與「CompanyName」欄位內容（檔案名稱：「PhpProject15」資料夾內「mysqli_query4.php」）。



資料庫伺服器連結成功及資料庫開啟成功
有 10 筆資料喔！

ALFKI--Alfreds Futterkiste
ANATR--Ana Trujillo Emparedados y helados
ANTON--Antonio Moreno Taqueria
AROUT--Around the Horn
BERGS--Berglunds snabbkop
BLAUS--Blauer See Delikatessen
BLONP--Blondel pere et fils
BOLID--Bolido Comidas preparadas
BONAP--Bon app'
BOTTM--

執行查詢語法



- 資料庫伺服器與挑選資料庫的語法為第07行到第13行，請參考15-1-1節介紹說明。
- 第14行將SQL語法儲存至\$sql變數內；第15及第16行代表\$mysqli物件使用query動作以執行SQL語法。
- 第15行執行的SQL語法是設定編碼為UTF-8，而第16行則執行第14行的SQL語法，執行完SQL語法後再將結果儲存於查詢結果物件，此範例內物件名稱為\$sql2。

執行查詢語法



- \$sql2物件已儲存了查詢結果，所以接著很多動作都以這個物件為主。
- 第17行代表\$sql2物件執行「num_rows」這個動作來計算筆數，並將計算後的結果儲存於\$rows變數內。
- 第21行看到\$sql2物件執行fetch_object()方法，讀取出每一筆記錄的各欄位資訊後儲存於\$list3物件內。
- 在迴圈內再將\$list3物件內的欄位資料取出，例如本例中取出CustomerID與CompanyName兩個欄位資料。
- 當查詢動作完成後，於第25行執行close()方法以關閉查詢結果物件。當資料庫連線結束，於第26行執行close()方法以關閉連結資料庫物件。

執行查詢語法



- 在此整理mysqli連結資料庫流程為：

Step 1、請以 `new mysqli()` 方式產生一個連結資料庫物件，該物件可連結伺服器與指定資料庫。

Step 2、使用連結資料庫物件的 `query()` 動作設定編碼。

Step 3、使用 `query()` 動作執行 SQL 語法，SQL 語法可分為兩種：

1. SQL 語法若是 `insert into`、`delete`、`update` 三個語法則會變更資料表內容。
2. SQL 語法若是 `select` 語法，使用連結資料庫物件的 `query()` 動作後再將結果儲存於查詢結果物件，接著就可以使用該查詢結果物件的 `num_rows()` 動作取得查詢後筆數或者利用 `fetch_object()` 動作取出每一筆記錄的內容。

Step 4、執行 `close()` 方法以關閉查詢結果物件。

Step 5、執行 `close()` 方法以關閉連結資料庫物件。

Query動作的風險



- 依循前面各節的動作，目前ch13資料庫的login資料表內共有五筆資料，如果您的資料筆數不一致就依照您目前看到的筆數開始進行練習。
- 請於phpMyAdmin的SQL視窗內輸入「**select * from login where username='jiannrong' and password=' ';**」，請留意上述語法使用的都是單引號。執行SQL語法後，因欄位沒有符合條件而不會有資料顯示。

✓ MySQL 回傳空的查詢結果 (即零資料列)。 (查詢花費 0.0005 秒。)

```
select * from login where username='jiannrong' and password=' '
.....
```

Query動作的風險



- SQL語法的where條件內請加入or進行條件分析，修改後輸入「**select * from login where username='jiannrong' and password=' 'or ' '= ' '**」，請留意上述語法使用的都是單引號。執行後將會出現所有資料，但沒有一筆資料的「username」欄位內容為「jiannrong」。

顯示第 0 - 4 列 (總計 5 筆, 查詢花費 0.0006 秒。)

```
select * from login where username='jiannrong' and password=' 'or ' '= ' '
```

[行內編輯] [編輯] [SQL 語句分析] [產生 PHP 程式碼]

☐ 全部顯示 | 資料列數: 25 | 搜尋資料列: 搜尋此資料表

依主鍵排序: 無

+ 選項

		id	username	password
<input type="checkbox"/>	編輯 複製 刪除	1	admin	php
<input type="checkbox"/>	編輯 複製 刪除	2	admin2	mysql
<input type="checkbox"/>	編輯 複製 刪除	3	try	again
<input type="checkbox"/>	編輯 複製 刪除	4	a_æ-ı	again
<input type="checkbox"/>	編輯 複製 刪除	5	中文	again

Query動作的風險



- 因為「' ' ='」這一個條件代表「空字串=空字串」，而這個條件永遠成立；or代表只要左右兩邊有一邊條件成立就算成立。
- 若username或password空白或錯誤，但因為or右邊的條件成立，所以where敘述成立，可把所有資料（select * from login）列出。

Query動作的風險



- 請設計以下的表單網頁來輸入帳號與密碼（檔案名稱：「PhpProject15」資料夾內的「mysql-input-object1.html」）。
- 再請您設計以下的表單網頁（檔案名稱：「PhpProject15」資料夾內的「mysql-query-object1.php」）。

Query動作的風險



- 表單網頁內帳號請輸入「yeh」，密碼請輸入「'or"='」（均為單引號），表單網頁內其實沒有這筆資料，但是這樣的語法造成所有資料都顯示出來。

請輸入帳號: yeh
請輸入密碼: 'or"='
提交 重設

SQL Injection物件方式

select * from login where username='yeh' and password="' or "='"
有 5 筆資料喔！

username	password
admin	php
admin2	mysql
try	again
ä,æ-†	again
中文	again

SQL injection的解決方式

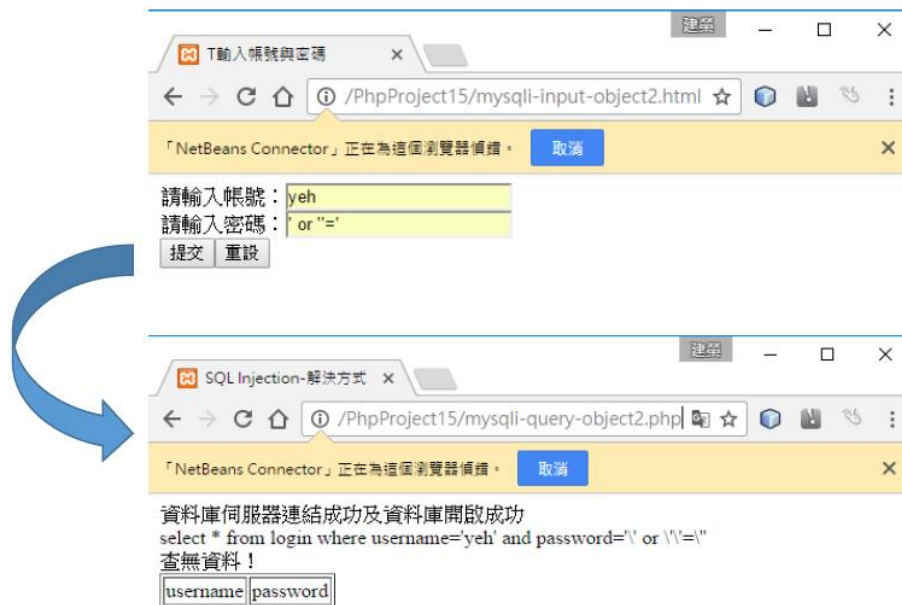


- mysqli提供了real_escape_string()方法可將SQL語法內特殊符號做轉換，考慮到PHP5另有magic_quotes_gpc設定進行過濾，所以我們在執行real_escape_string()方法前先進行過濾。
- 由於real_escape_string()方法可能造成數值資料誤加上過濾符號，所以我們得先進行資料分類，若設定為數值資料就不需要進行加上過濾符號的動作。

SQL injection的解決方式



- 請將「mysql-input-object1.html」另存新檔為「mysql-input-object2.html」，請修改表單的form action為mysql-query-object2.php。再請修改如下（檔案名稱：「PhpProject15」資料夾內的「mysql-query-object2.php」）。



SQL injection的解決方式



- 第14行建立一個名為mysql_prepare_for_request的函數名稱，接收三個參數，第一個參數是資料庫物件\$link，第二個參數是傳進資料表的值\$value，第三個是類型\$type，如果不給值預設為string。
- 第16行將進行\$type的判斷：
 - 如果是string執行第18行到第22行之間的語法，然後第23行跳出。
 - 如果是數值則會執行number區塊的判斷，執行第25行判斷\$returnvalue資料是否不是數值，若不是數值則將\$returnvalue資料設定為0。

SQL injection的解決方式



- 第18行判斷是否有magic_quotes_gpc設定進行過濾，這設定會自動加上反斜線，所以於第20行先刪除這反斜線。
- 請留意magic_quotes_gpc設定於PHP7不再提供，目的是資料是否要加上反斜線應該由程式設計師決定，而不是自動加入，加上PHP7也有其他的資料存取語法，不一定要使用query方法，所以這個設定於PHP7取消。
- 接著於第22行透過real_escape_string()方法加入反斜線進行過濾

表單對SQL Injection的處理建議



- SQL Injection是一個網頁上的填空遊戲，可輕鬆地在沒有帳號權限控制下新增、刪除、更新與查詢資料，會產生這個問題有以下幾點原因：
 1. 表單輸入資料的欄位沒有針對特殊符號進行過濾
 2. 欄位沒有限制長度及內容
 3. 資料庫連結無密碼連結
 4. 攻擊者猜測到資料表欄位的部分資訊

表單對SQL Injection的處理建議



- 所以建議可用以下的方式避免遭到攻擊：
- 1. 過濾特殊符號
- 2. 欄位限制長度及內容
- 3. 資料庫連結權限控管
- 4. 加上session或cookie限制，避免攻擊者反覆測試

資源物件的參數語法與PDO操作方式

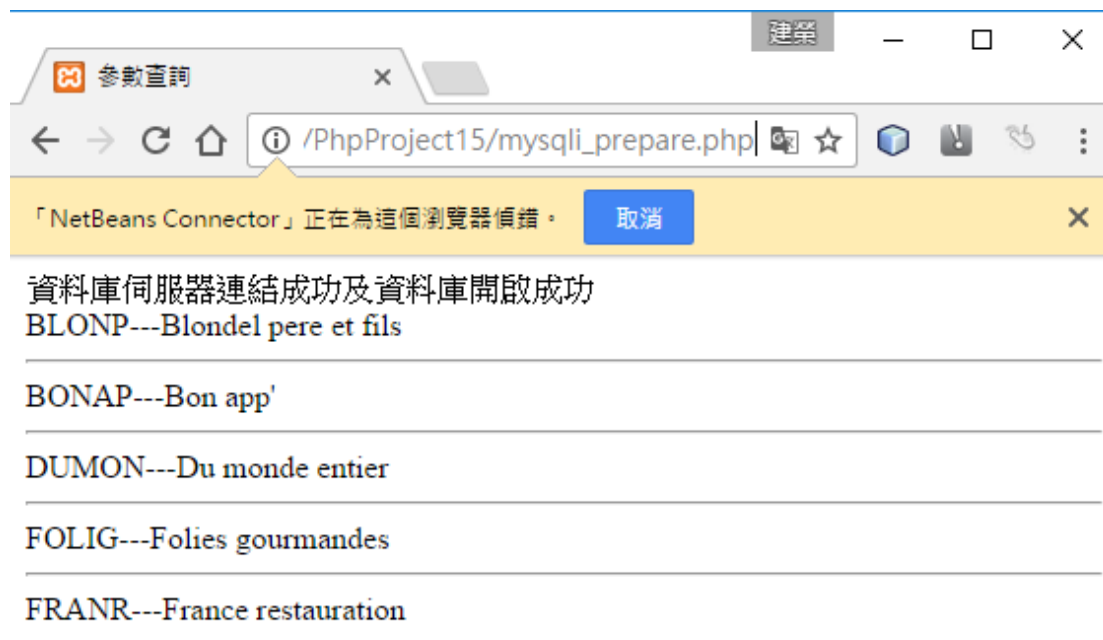


- 我們於15-2-4節可以看到SQL injection問題，15-2-5節針對直接輸入SQL語法的query動作進行SQL injection的處置，15-2-6節介紹表單與伺服器的處理建議。
- PHP與資料庫並非只有直接輸入SQL語法這種方式溝通，另外有資源物件的參數語法以及PDO操作方式均可以避免SQL injection攻擊。

資源物件語法



- 以資源物件語法方式可以避開SQL injection風險，我們來看以下的範例介紹（檔案名稱：「PhpProject15」資料夾內「mysqli_prepare.php」）。



資源物件語法



- 不論是動態變更資料或著查詢資料，資料欄位於資源物件語法內是以「?」來代替，而資源型態共有四種：
 - 「s」代表字串及「i」代表整數
 - 「b」代表Blob及「d」代表浮點數
- 資料庫伺服器與挑選資料庫的語法為第07行到第13行，請參考15-1-1節介紹說明。
- 第14行設定傳輸編碼為utf8，第15行預設\$country變數內容為france，接著於第16行進行select查詢，查詢語法交給資料庫物件的prepare()方法進行準備，建立名為\$stmt資源物件。

資源物件語法



- 資源物件建立後再使用資源物件的bind_param()方法將變數與資源內的「？」做指定型態的結合。
- 資源物件的bind_param()函數內共有兩個參數，依序為「資料型態、要結合的變數」。
- 資源型態所以第20行將\$country變數以字串型態與\$stmt資源物件內的「？」進行結合。

資源物件語法



- 第21行執行資源物件的execute()函數執行SQL語法。這是一個執行查詢資料的SQL語法，所以執行execute()函數後會有結果。
- 第22行執行資源物件的bind_result()動作將資源查詢的結果，對應到指定的變數，就可於第23行執行資源物件的fetch()動作將結果列出。

資源物件語法



- 請留意第22行`bind_result()`方法內有多個參數，代表您指定要輸出的變數名稱，本例中由第16與第17行語法得知我們要查詢`customerid`,`companyname`兩個欄位內容。
- 資源物件的`bind_result()`函數內自訂了`$col1`與`$col2`兩個變數，分別對應`customerid`與`companyname`兩個欄位，再使用資源物件的`fetch()`方法將`$col1`與`$col2`兩個變數內容逐一取出。
- 資源用到最後建議要做關閉的動作，第27行執行資源物件的`close()`動作關閉資源及第29行執行連結資料庫物件的`close()`動作關閉資料庫連線。

資源物件語法



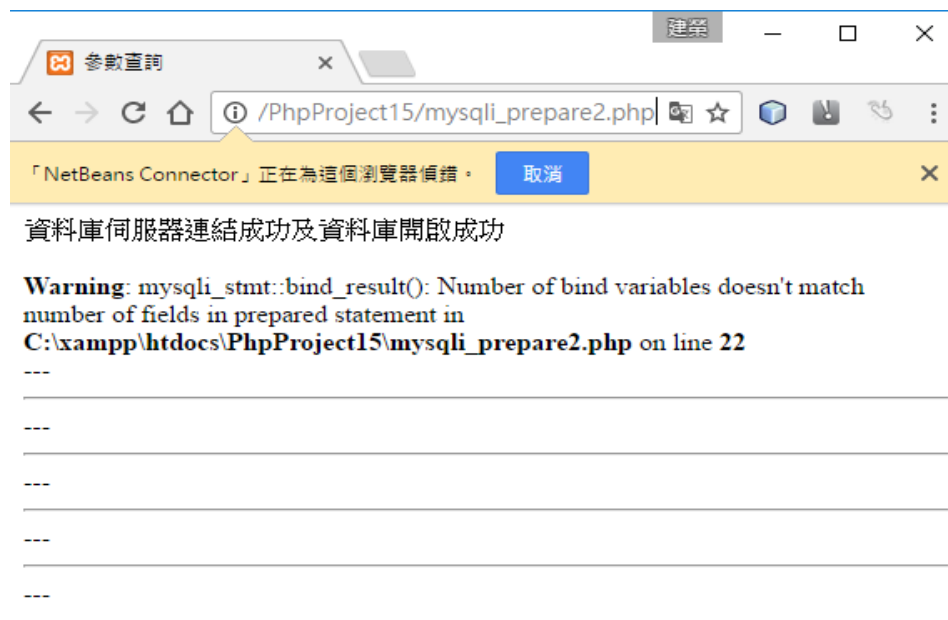
- 使用資源物件語法有關的動作有以下幾種：

物件	方法動作名稱	方法動作說明說明
連結資料庫物件	prepare ()	將 SQL 語法建立一個資源物件。函數內參數為 SQL 語法。
	close()	關閉資料庫連線。
資源物件	bind_param()	將變數與資源內的「？」做指定型態的結合。
	execute()	執行結合後的 SQL 語法。
	bind_result()	bind_result()函數將資源查詢的結果，對應到指定的變數
	fetch()	每一筆記錄的欄位內容逐筆輸出。
	close()	關閉資源。

資源物件語法



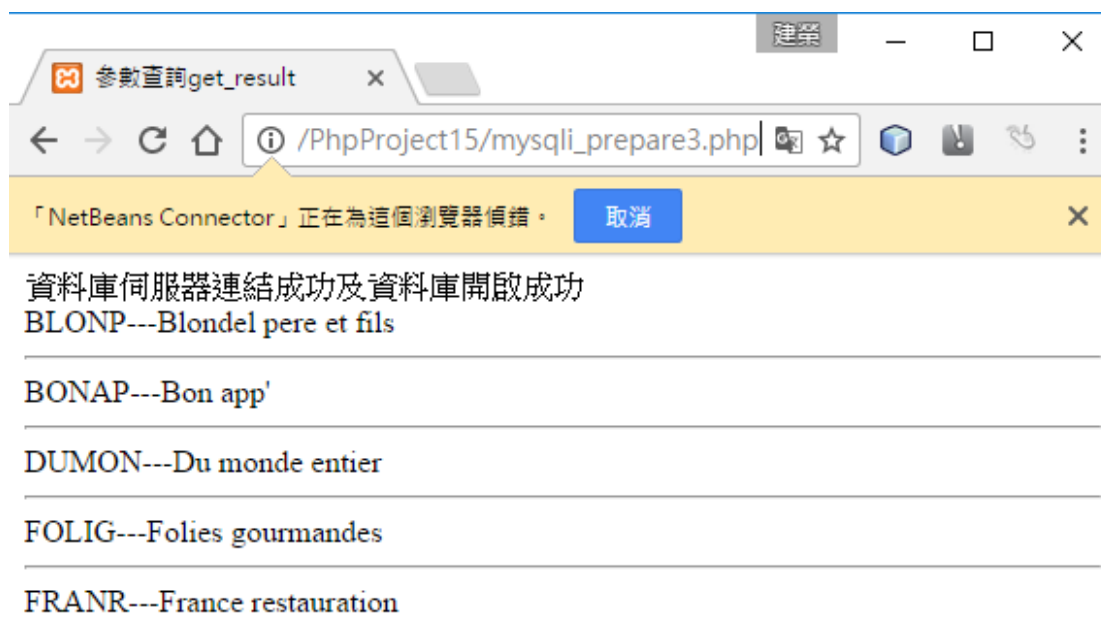
- 請將「mysql_prepare.php」改為「mysql_prepare2.php」，再請您修改第16行到第18行的SQL語法，改為所有欄位方式查詢，這個程式執行時將會產生錯誤。



資源物件語法



- 如果您要進行所有欄位的檢索方式，請您改為 `get_result()` 動作，請查看以下的練習（檔案名稱：「PhpProject15」資料夾內「`mysqli_prepare3.php`」）。



資源物件語法



- 請留意第21行執行資源物件的get_result動作後再儲存於\$result查詢結果物件，第22行再將\$result查詢結果物件的內容依照欄位順序儲存至\$row陣列內，再讀取\$row陣列內各欄位內容，請留意欄位名稱與檢索時的資料表欄位名稱大小寫必須一致。
- bind_result () 方法的優點：
 - 更為簡單
 - 不需要牢記欄位名稱
 - 使用fetch()方式帶出資料
- bind_result () 方法的缺點：
 - 不能查詢全部欄位內容

資源物件語法



- `get_result ()` 方法的優點：
 - 可以查詢全部欄位內容
 - 使用 `fetch_array()` 方式帶出資料
- `get_result ()` 方法的缺點：
 - 必須再另外建立陣列儲存資料
 - 必須牢記欄位的大小寫名稱

使用PDO與連結資料庫主機

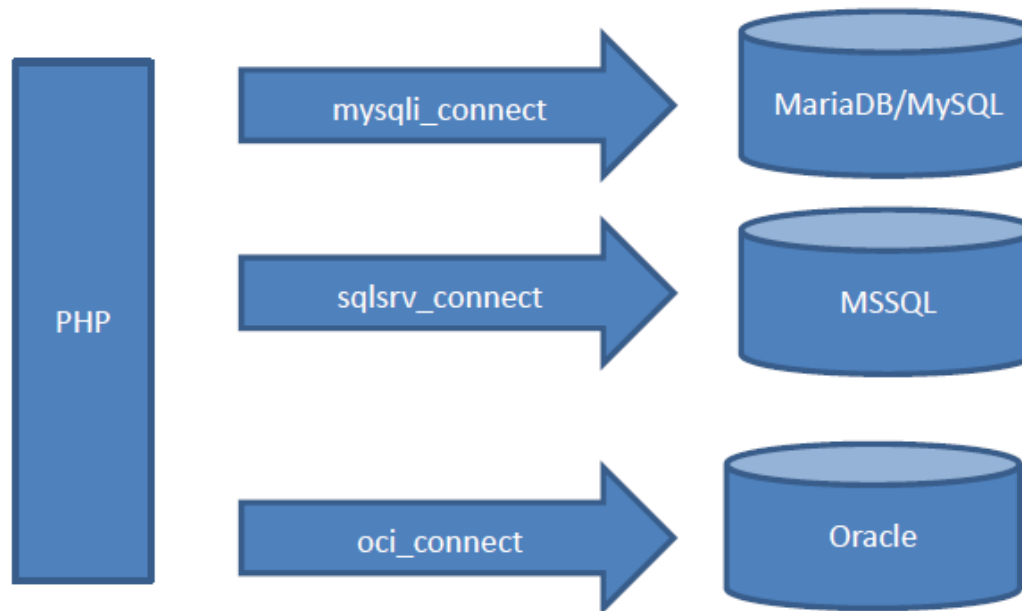


- PDO(PHP Data Objects) 是一個相當好用的擴充元件，主要有以下的理由：
- 1.採用物件導向的概念設計。
- 2.採用一致的連線方式。如果您的後端資料庫可能是MariaDB/MySQL、PostgreSQL或著是MS SQL Server，您得使用不同的連結方式與執行SQL的語法，一旦更換資料庫您就得更換不同的語法，但PDO可以將存取資料庫的動作抽象化，呼叫不同資料庫時會載入不同的擴充元件，但使用的語法是相同，參數略有不同。讓使用者可以使用一致的方式存取和處理資料的作業。

使用PDO與連結資料庫主機



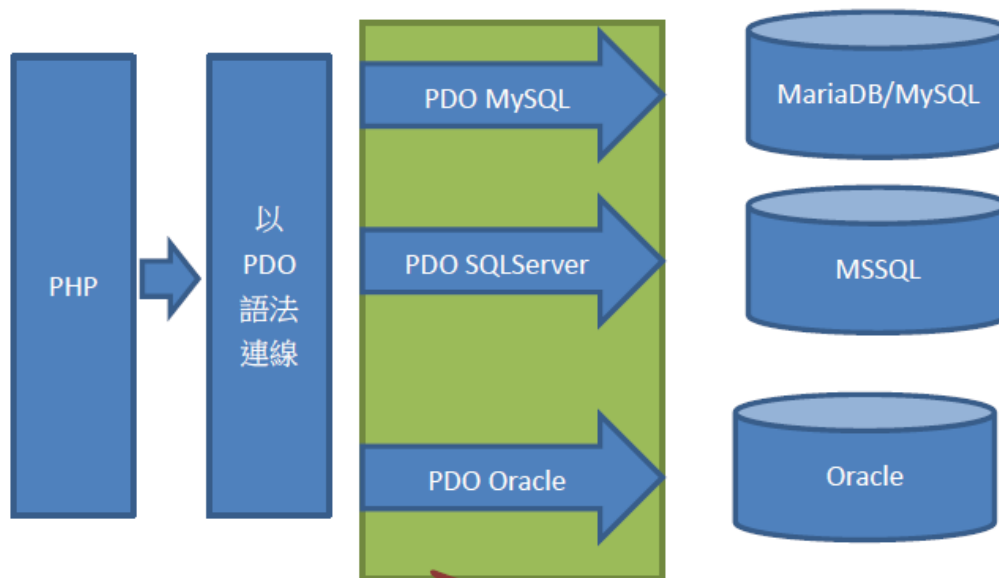
- 3.避免SQL injection發生。
- 不同資料庫有不同的連線與執行SQL的語法



使用PDO與連結資料庫主機



- 使用PDO方式連線不同資料庫



呼叫不同的擴充元件，但使用的語法是相同，參數略有不同。

使用PDO與連結資料庫主機



- XAMPP內已經加入PDO的支援，您可以輸入 <http://localhost/dashboard/phpinfo.php> 進行檢驗，如果您沒有這個檔案，建議您建立一個PHP檔案，這個檔案只要有以下的資訊；

```
<?php  
phpinfo();  
?>
```

使用PDO與連結資料庫主機



- 接著請您執行這個網頁，您將會看到您的PHP支援的項目，請搜尋PDO，如果找到PDO相關資訊那就代表已經啟用。
- 以XAMPP為例，他已經啟用了MariaDB/MySQL與SQLite這兩種資料庫的PDO支援功能。

PDO

PDO support	enabled
PDO drivers	mysql, sqlite

pdo_mysql

PDO Driver for MySQL	enabled
Client API version	mysqlnd 5.0.12-dev - 20150407 - \$Id: fd3bfdab644600c96b25beaf69d021447b7b4ec5 \$

pdo_sqlite

PDO Driver for SQLite 3.x	enabled
SQLite Library	3.8.10.2

使用PDO與連結資料庫主機

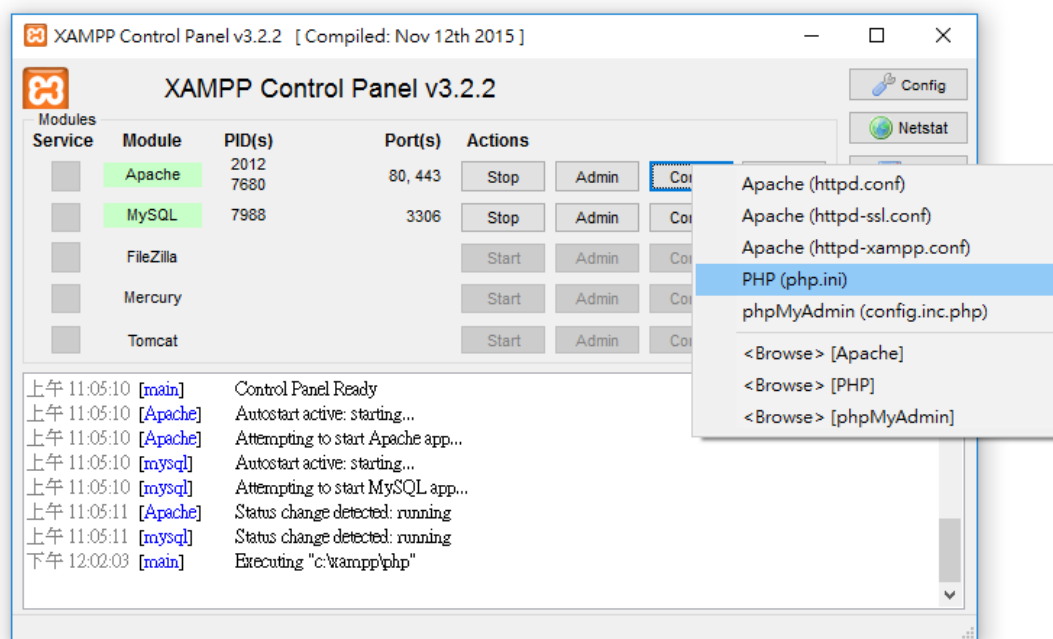


- 如果沒有啟用，接著我們將修改設定。若您使用XAMPP，您可開啟「XAMPP Control Panel」後點選「Apache」旁的「Config」，再請點選「PHP(phi.ini)」，就可以開啟phi.ini。
- 開啟後請搜尋「pdo」，您會發現到一行「;extension=phi_pdo_mysql.dll」的設定。

使用PDO與連結資料庫主機



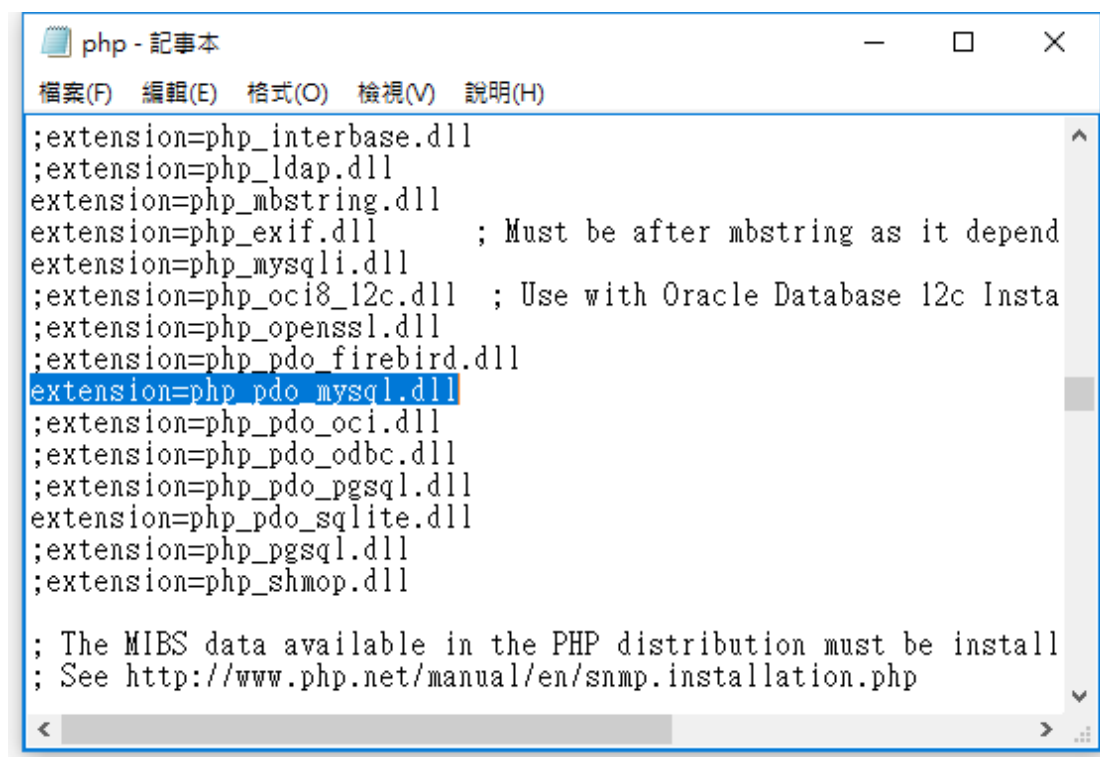
- 「;」代表這一行為註解，所以請您將「;extension=php_pdo_mysql.dll」的「;」刪除，修改為「extension=php_pdo_mysql.dll」，再請您儲存php.ini。



使用PDO與連結資料庫主機



- php.ini內關於PDO的設定

A screenshot of a Notepad window titled "php - 記事本". The window displays the contents of a php.ini file, specifically the section for extensions. The line ";extension=php_pdo_mysql.dll" is highlighted in blue. Other visible lines include ";extension=php_interbase.dll", ";extension=php_ldap.dll", "extension=php_mbstring.dll", "extension=php_exif.dll ; Must be after mbstring as it depend", "extension=php_mysqli.dll", ";extension=php_oci8_12c.dll ; Use with Oracle Database 12c Insta", ";extension=php_openssl.dll", ";extension=php_pdo_firebird.dll", ";extension=php_pdo_oci.dll", ";extension=php_pdo_odbc.dll", ";extension=php_pdo_pgsql.dll", "extension=php_pdo_sqlite.dll", ";extension=php_pgsql.dll", and ";extension=php_shmop.dll". At the bottom, there is a comment about MIBS data and a URL: "; The MIBS data available in the PHP distribution must be install", "; See http://www.php.net/manual/en/snmp.installation.php".

```
php - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

;extension=php_interbase.dll
;extension=php_ldap.dll
extension=php_mbstring.dll
extension=php_exif.dll      ; Must be after mbstring as it depend
extension=php_mysqli.dll
;extension=php_oci8_12c.dll ; Use with Oracle Database 12c Insta
;extension=php_openssl.dll
;extension=php_pdo_firebird.dll
extension=php_pdo_mysql.dll
;extension=php_pdo_oci.dll
;extension=php_pdo_odbc.dll
;extension=php_pdo_pgsql.dll
extension=php_pdo_sqlite.dll
;extension=php_pgsql.dll
;extension=php_shmop.dll

; The MIBS data available in the PHP distribution must be install
; See http://www.php.net/manual/en/snmp.installation.php
```

使用PDO與連結資料庫主機



- 儲存php.ini後請記得重新啟動Apache。
- 若您使用XAMPP，您可開啟「XAMPP Control Panel」後點選「Apache」旁的「Stop」鈕，使系統關閉，該鈕就會變更為「Start」鈕。
- 請您接著於「XAMPP Control Panel」後點選「Apache」旁的「Start」鈕，使系統開啟，該鈕就會變更為「Stop」鈕。
- 請您開啟phpinfo網頁查詢PDO是否有開啟，若您搜尋到這項目，代表可以執行後續的各項範例。

利用 PDO 連結



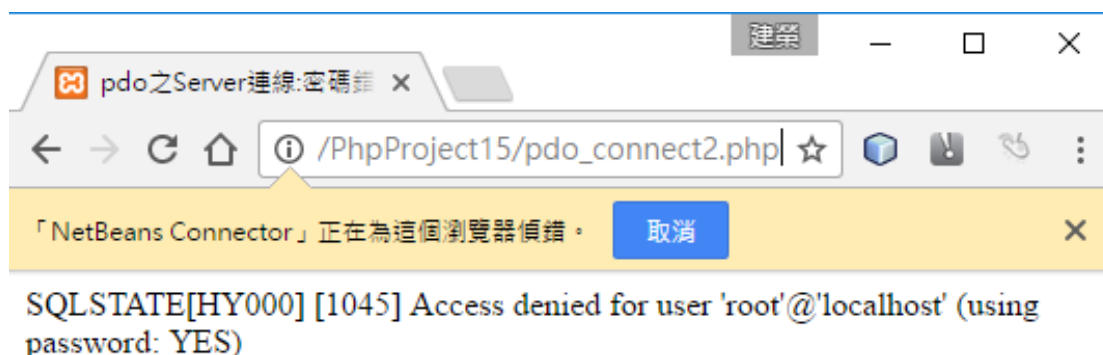
- 我們要進行資料庫主機連結與資料庫開啟，MariaDB/MySQL的連結方式為：

```
PDO 物件=new PDO("mysql:host=主機;dbname=資料庫","帳號","密碼")
```


利用PDO連結



- 請查看操作練習（檔案名稱：「PhpProject15」資料夾內「pdo_connect1.php」）。
- PDO可搭配try catch語法進行連結與資料庫操作，過程中若產生例外則會拋出，例如以下的練習為密碼錯誤（檔案名稱：「PhpProject15」資料夾內「pdo_connect2.php」）。



利用PDO進行資料操作

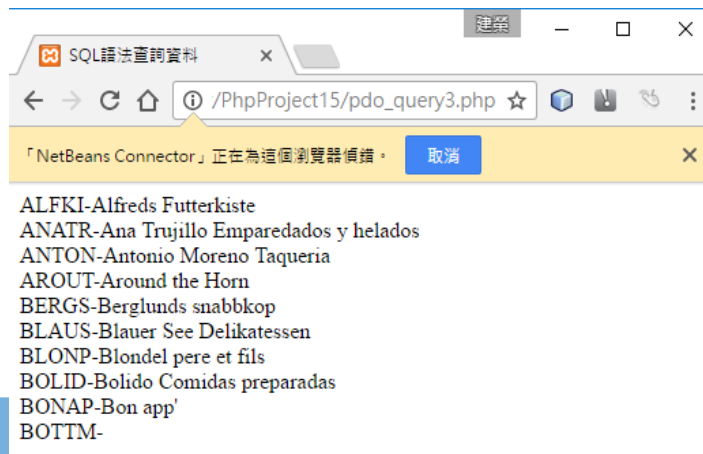


- 我們先來了解如何新增資料到資料表，請您查看以下的練習（檔案名稱：「PhpProject15」資料夾內「pdo_query1.php」）。
- 第15行代表設定編碼為utf8，第16與第17行代表要存入資料表的\$user與\$pass兩個變數內容，第18行與第19行為insert into語法，我們設定時可綁定欄位參數，這裡綁定的欄位參數分別為:user與:pass，透過prepare方法進行準備，並將資料儲存於\$sql物件。
- 第20行到23行執行以\$sql物件為參數的execute動作，將\$user與\$pass兩個變數分別存入:user與:pass兩個欄位參數。

利用PDO進行資料操作



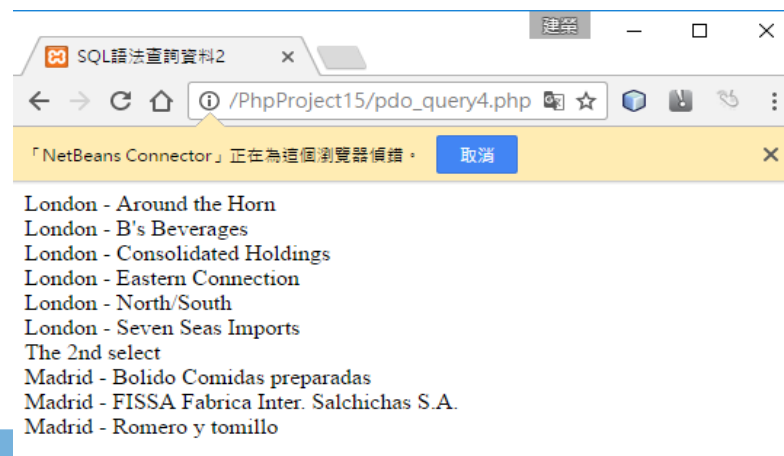
- 新增刪除更新除了可用陣列方式進行給值，也可用?方式進行欄位設定，請您將pdo_query1.php另存為pdo_query2.php，在此僅列出兩個檔案不同之處（檔案名稱：「PhpProject15」資料夾內「pdo_query2.php」）。
- 接著我們來了解如何查詢資料表資料，請您查看以下的練習（檔案名稱：「PhpProject15」資料夾內「pdo_query3.php」）。



利用PDO進行資料操作



- 第15行到第16行為SQL語法，第17行以query()方法執行SQL語法後將動作儲存於\$result物件內，第18行透過fetchAll()方法取出所有資料後儲存為陣列，再搭配foreach()語法逐筆由取出的資料撈出所需要的欄位。
- 檢索資料時也可以用指定欄位參數方式操作，請您查看以下的練習（檔案名稱：「PhpProject15」資料夾內「pdo_query4.php」）。



問題與討論



- 1.請說明mysqli連結伺服器與挑選資料庫的流程。
- 2.請使用資源物件語法方式，分別利用bind_result（）與get_result（）方法查詢products資料表內容。
- 3.請使用PDO語法查詢orders資料表內容。
- 4.請使用mysqli基本語法、資源物件語法與PDO語法查詢employees資料表內三個欄位內容。