



第 6 章

與資料庫的 結合運用

6-1 資料庫的基本知識



- 所謂資料庫是將資料匯整在一起機制

▼ 資料庫

未整理過的資料
(資料形式不統一)

熊木和夫
東京都新宿區西新宿2-8-1

地址：神奈川縣橫濱市中區日本大通1
姓名：鳥居健二

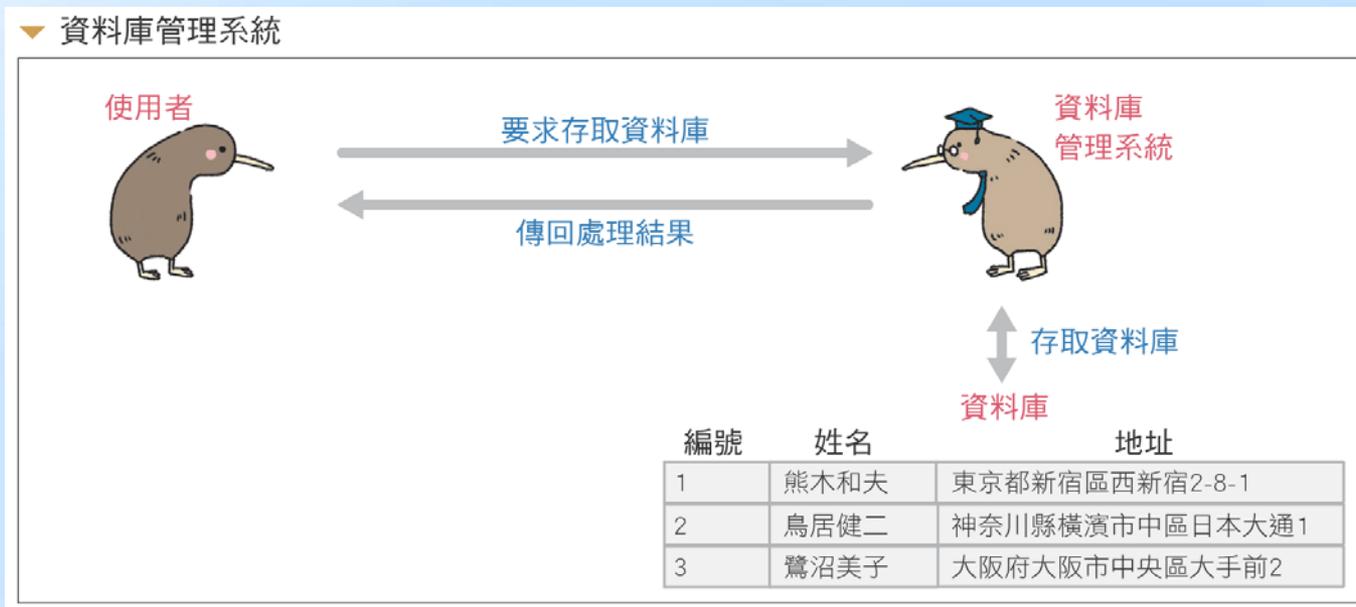
大阪府大阪市中央區大手前2
鷺沼美子小姐



資料庫
(以便於搜尋、更新的形式儲存起來)

編號	姓名	地址
1	熊木和夫	東京都新宿區西新宿2-8-1
2	鳥居健二	神奈川縣橫濱市中區日本大通1
3	鷺沼美子	大阪府大阪市中央區大手前2

- 電腦上用來建立及存取資料庫的軟體，稱為「資料庫管理系統 (DBMS)」



● 資料表與列(橫列)、欄(直欄)

-現在最被廣泛使用的類型為「關聯式資料庫」

-在資料庫中，資料以表格的形式管理，這個表格被稱「資料表」

▼ 資料庫中的資料表

編號	姓名	地址
1	熊木和夫	東京都新宿區西新宿2-8-1
2	鳥居健二	神奈川縣橫濱市中區日本大通1
3	鷺沼美子	大阪府大阪市中央區大手前2

一橫向的格子稱為「列」，縱向的格式稱為「欄」

▼ 列與欄

	欄1	欄2	欄3
列1	1	熊木和夫	東京都新宿區西新宿2-8-1
列2	2	鳥居健二	神奈川縣橫濱市中區日本大通1
列3	3	鷺沼美子	大阪府大阪市中央區大手前2

要點！

資料庫中的資料表是以「列」與「欄」組成。



- 建立表格時，應先決定欄數以及每一欄要存放什麼樣的資料
- 只要在記憶體或磁碟等的儲存容量範圍內，都可追加所需的列數

▼ 列可輕易追加

	欄(編號)	欄(姓名)	欄(地址)
列1	1	熊木和夫	東京都新宿區西新宿2-8-1
列2	2	鳥居健二	神奈川縣橫濱市中區日本大通1
列3	3	鷺沼美子	大阪府大阪市中央區大手前2
列4	4	鷺尾史郎	愛知縣名古屋市中區三之丸3-1-2

要點！

列可輕易地增刪。



一建立資料表時，必須要在一開始就明確定出資料表的用途，**盡量避免**之後要修改資料欄架構的情況發生

	欄(編號)	欄(姓名)	欄(郵遞區號)	欄(地址)
列1	1	熊木和夫	160-0023	東京都新宿區西新宿2-8-1
列2	2	鳥居健二	231-0021	神奈川縣橫濱市中區日本大通1
列3	3	鷺沼美子	540-0008	大阪府大阪市中央區大手前2
列4	4	鷺尾史郎	460-0001	愛知縣名古屋市中區三之丸3-1-2

要點！

設計資料欄時，應考量之後不需再做修改。



每一橫列就是一筆資料，表格除了可以新增一筆資料，也可以根據編號進行修改每一筆資料中某一個欄位的內容。



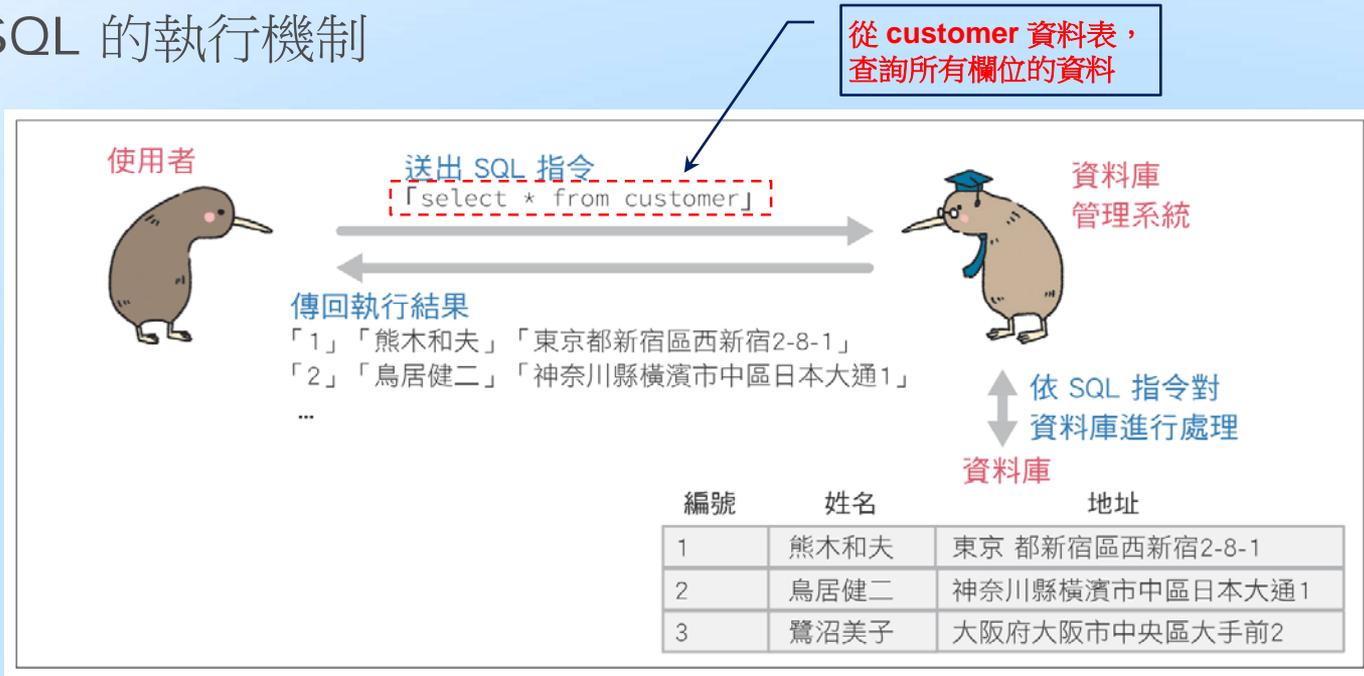
● MySQL (MariaDB)

- 在電腦的世界裡，已有多種廣受歡迎的 RDBMS 產品
- 在使用方式上與 MySQL 完全相同

RDBMS：關聯式資料庫管理系統，**Relational Database Management System**

● SQL

- SQL 是用來操作 RDBMS 的程式語言
- SQL 的執行機制



6-2 建立商品資料庫



- 啟動 MySQL

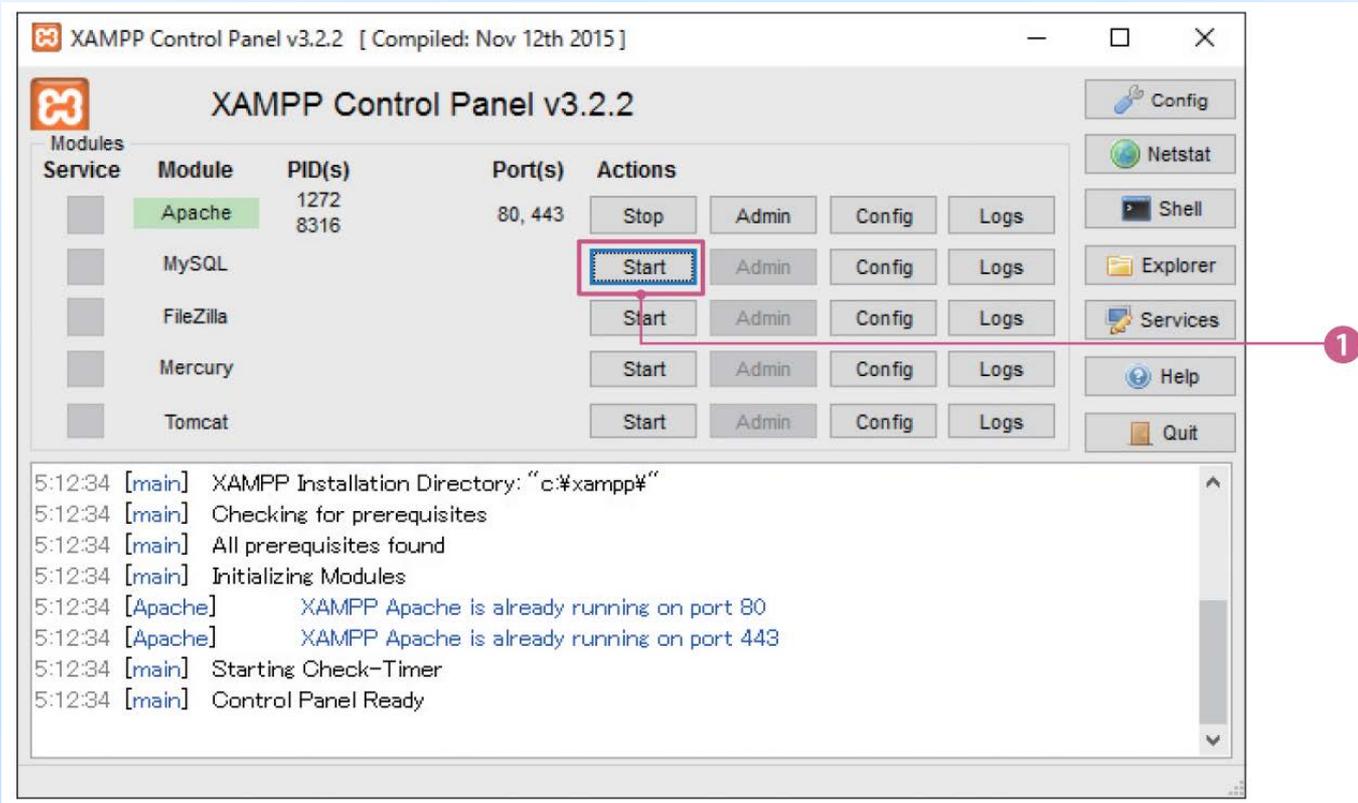
- 要執行 **SQL** 指令則必須先啟動 **RDBMS**

- 以下操作都是在啟動 XAMPP 控制面板，並執行 Apache 的環境下進行

▼ XAMPP 圖示



– 利用 XAMPP 控制面板啟動 MySQL



XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]

XAMPP Control Panel v3.2.2

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	1272 8316	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL			Start Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

5:12:34 [main] XAMPP Installation Directory: "c:\xampp#\n5:12:34 [main] Checking for prerequisites\n5:12:34 [main] All prerequisites found\n5:12:34 [main] Initializing Modules\n5:12:34 [Apache] XAMPP Apache is already running on port 80\n5:12:34 [Apache] XAMPP Apache is already running on port 443\n5:12:34 [main] Starting Check-Timer\n5:12:34 [main] Control Panel Ready

● 執行建立資料庫的 SQL 程式

- 以 SQL 撰寫的程式，稱為 SQL Script。每個 SQL Script 中，會包含 1 個或多個 SQL 指令
- 本章範例程式中，包含了建立 **shop 資料庫** 的 SQL 程式

▼ Table 資料庫定義

項目	名稱
資料庫名稱	shop
資料表名稱	product
使用者名稱	staff
密碼	password

要先建立資料庫 (**shop**)，才可以在此資料庫中建立資料表 (**product**)。

—這裡要建立的資料庫、資料表、使用者等各項的關係如下

▼ 商品資料庫

shop (商店) 資料庫

product (商品) 資料表

id (商品編號)	name (商品名稱)	price (價格)
1	松果	700
2	核桃	270
3	葵花子	210
4	杏仁	220

...



使用者

使用者名稱 : staff
密碼 : password



```
product.sql
drop database if exists shop;
create database shop default character set utf8 collate utf8_
general_ci;
grant all on shop.* to 'staff'@'localhost' identified by
'password';
use shop;

create table product (
    id int auto_increment primary key,
    name varchar(200) not null,
    price int not null
);

insert into product values(null, '松果', 700);
insert into product values(null, '核桃', 270);
insert into product values(null, '葵花子', 210);
insert into product values(null, '杏仁', 220);
insert into product values(null, '腰果', 250);
insert into product values(null, '巨人玉米', 180);
insert into product values(null, '開心果', 310);
insert into product values(null, '夏威夷豆', 600);
insert into product values(null, '南瓜子', 180);
insert into product values(null, '花生', 150);
insert into product values(null, '枸杞', 400);
```

如果 **shop** 資料庫存在，則刪除

建立 **shop** 資料庫

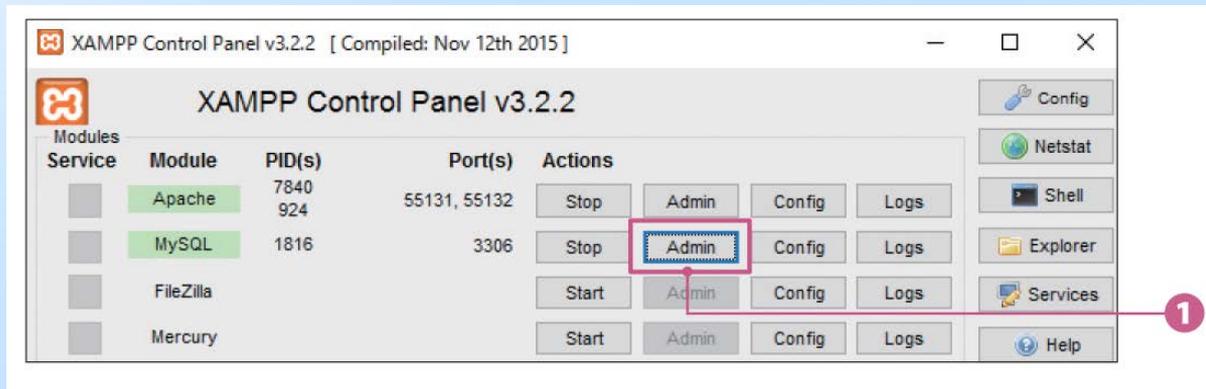
授權資料庫所有
權限給使用者

使用 **shop** 資料庫

建立 **product** 資料表

將資料新增至
product 資料表中

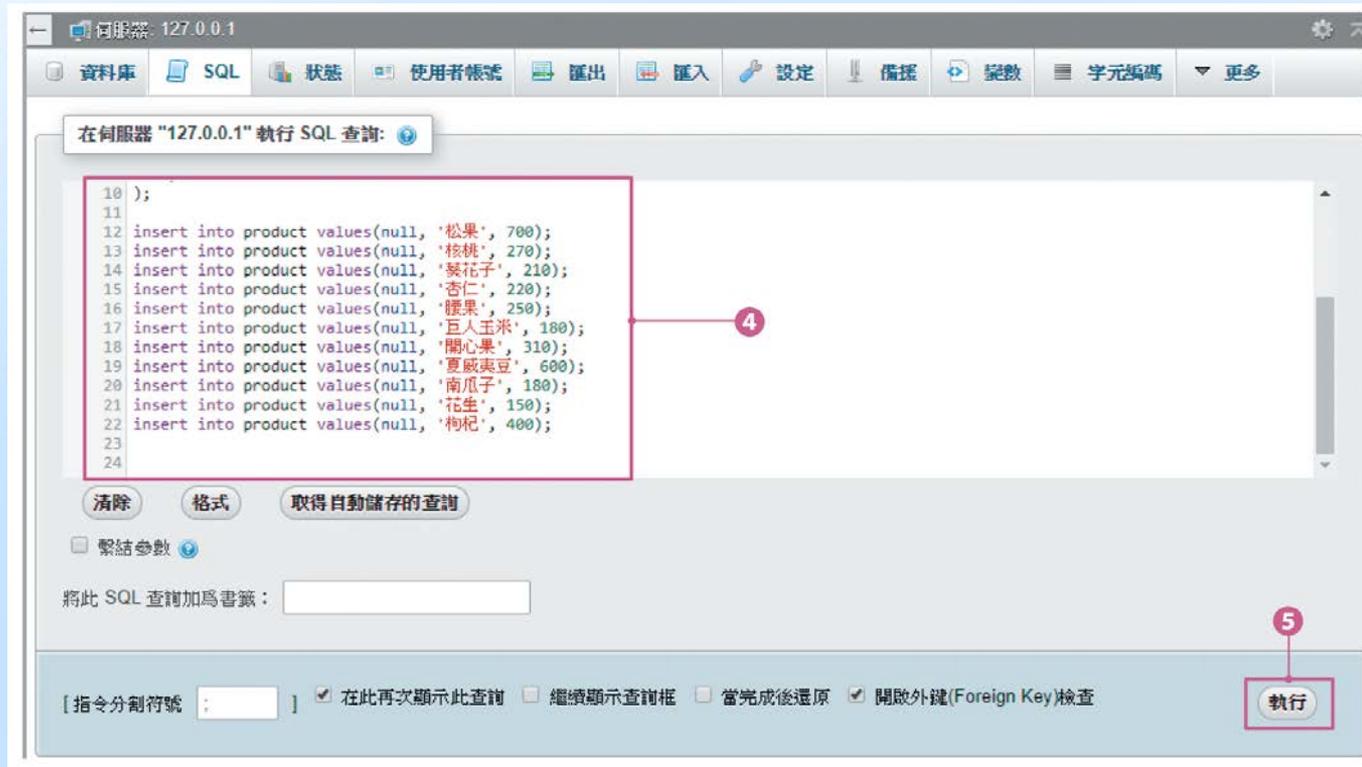
- 輸入與執行 SQL 程式的方法
 - 從 XAMPP 控制面板啟動 phpMyAdmin



一可設定要使用的語系

The screenshot shows the phpMyAdmin interface in a browser window. The address bar displays 'localhost/phpmyadmin/'. The main content area is divided into two sections: '一般設定' (General Settings) and '外觀設定' (Appearance Settings). In the '外觀設定' section, the '語系 - Language' dropdown menu is highlighted with a red box and labeled with a red circle containing the number '2'. The selected language is '中文 - Chinese traditional'. Below it, the '主題' (Theme) is set to 'pmahomme'.

– 在 SQL 的輸入區塊中貼上 SQL 程式內



註：SQL 程式建立資料庫 shop 要完成後(需要等候片刻)，才可 use shop。

一指令被正常執行完成後，畫面上會顯示綠色的打勾符號



←在畫面左側，列出 MySQL 所管理的所有資料庫與資料表清單





● 建立資料庫與使用者

– 接下來將說明 SQL 指令檔 (product.sql) 的內容

● 刪除資料庫(如果 shop 資料庫存在，則先刪除)

```
drop database if exists shop;
```

● 建立資料庫

```
create database shop default character set utf8 collate utf8_general_ci;
```

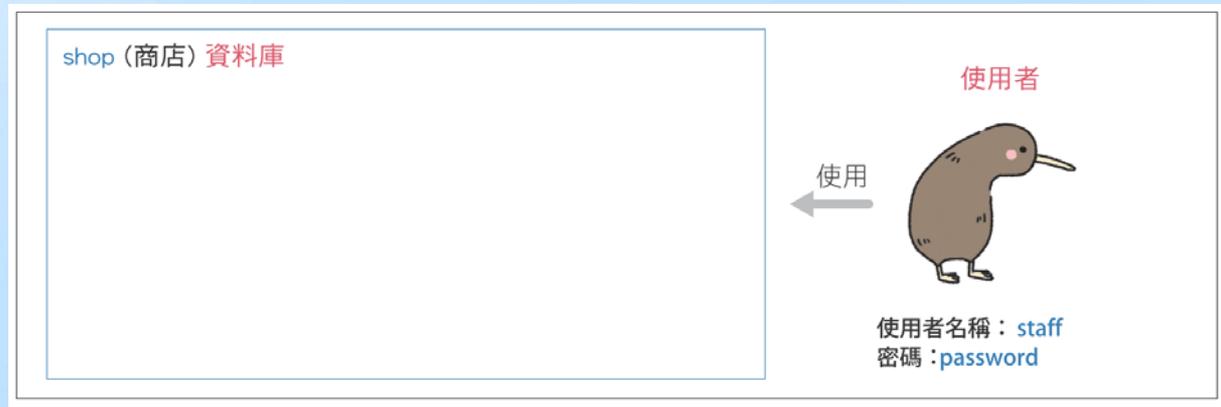
● 建立使用者 (如果使用者不存在)

```
grant all on shop.* to 'staff'@'localhost' identified by 'password';
```

● 連線資料庫

```
use shop;
```

-use 是用來連線到資料庫的指令





● 建立資料表

-底下指令代表在 **shop** 資料庫中，建立 **product** 資料表

```
create table product (  
    id int auto_increment primary key,  
    name varchar(200) not null,  
    price int not null  
);
```

-商品編號欄(自動編號且為主鍵)

```
id int auto_increment primary key,
```

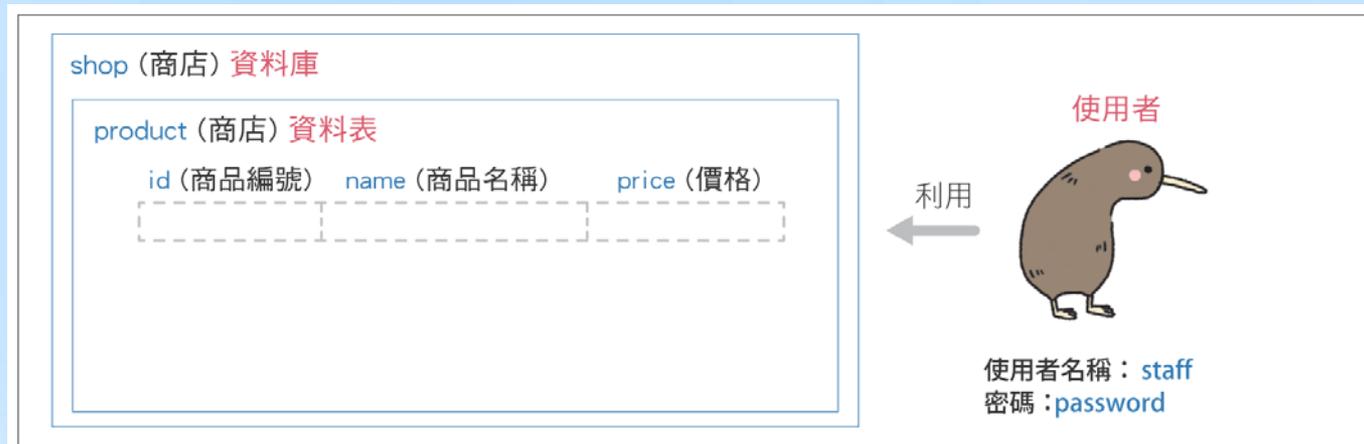
—商品名稱欄(變動字串，長度200，不允許空值)

```
name varchar(200) not null,
```

—價格資料欄(整數資料，不允許空值)

```
price int not null
```

—藉由上述處理，就可在 shop 資料庫中建立 product 資料表





● 新增資料

—最後，將商品資料放入 **product** 資料表

```
insert into product values(null, '松果', 700);  
insert into product values(null, '核桃', 270);  
insert into product values(null, '葵花子', 210);  
...
```

—例如，若商品名稱為「松果」，價格為「700」，應寫成

```
null, '松果', 700
```

編號 (id)：設為自動編號，不需輸入任何資料，因此設為 **null**。
商品名稱 (name)：字串資料型別，要將資料加上"，也可以用""。
價格 (price)：整數資料型別，以數字形式輸入。

—下列資料也可利用 insert 指令新增到資料表

▼ 新增資料

shop (商店) 資料庫

product (商店) 資料表

id (商品編號)	name (商品名稱)	price (價格)
1	松果	700
2	核桃	270
3	葵花子	210
4	杏仁	220

...



6-3 顯示商品清單



- 利用 phpMyAdmin 顯示商品清單

- 在資料庫一覽中選擇 shop (此動作就是 **use shop**)

The screenshot shows the phpMyAdmin interface. The left sidebar lists databases: information_schema, mysql, performance_schema, phpmyadmin, shop, and test. The main area shows the '資料庫' (Databases) tab selected, with a table listing databases and their collations. The 'shop' database is highlighted with a red box and a red circle labeled '2'. The '資料庫' tab is also highlighted with a red box and a red circle labeled '1'.

資料庫	編碼與排序	動作
<input type="checkbox"/> information_schema	utf8_general_ci	<input type="checkbox"/> 檢查權限
<input type="checkbox"/> mysql	latin1_swedish_ci	<input type="checkbox"/> 檢查權限
<input type="checkbox"/> performance_schema	utf8_general_ci	<input type="checkbox"/> 檢查權限
<input type="checkbox"/> phpmyadmin	utf8_bin	<input type="checkbox"/> 檢查權限
<input checked="" type="checkbox"/> shop	utf8_general_ci	<input type="checkbox"/> 檢查權限
<input type="checkbox"/> test	latin1_swedish_ci	<input type="checkbox"/> 檢查權限
總計: 6	latin1_swedish_ci	

一執行結果將會顯示包含 id、name、price 這三個資料欄的商品清單

+ 選項				id	name	price			
<input type="checkbox"/>		編輯		複製		刪除	1	松果	700
<input type="checkbox"/>		編輯		複製		刪除	2	核桃	270
<input type="checkbox"/>		編輯		複製		刪除	3	葵花子	210
<input type="checkbox"/>		編輯		複製		刪除	4	杏仁	220
<input type="checkbox"/>		編輯		複製		刪除	5	腰果	250
<input type="checkbox"/>		編輯		複製		刪除	6	巨人玉米	180
<input type="checkbox"/>		編輯		複製		刪除	7	開心果	310
<input type="checkbox"/>		編輯		複製		刪除	8	夏威夷豆	600
<input type="checkbox"/>		編輯		複製		刪除	9	南瓜子	180
<input type="checkbox"/>		編輯		複製		刪除	10	花生	150
<input type="checkbox"/>		編輯		複製		刪除	11	枸杞	400

● SQL 的 select 敘述

-利用 select 敘述，可取得指定資料表中，**指定的資料欄的資料**

語法 select

```
select 資料欄名稱 from 資料表名稱;
```

如果使用 * 表示所有欄位



● 在 PHP 中連結資料庫

—開啟文字編輯器，撰寫 PHP 程式如下

```
List all.php PHP
<?php require '../header.php';?>
<?php
    $pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
                'staff', 'password');
?>
<?php require '../footer.php';?>
```

—程式若正確執行，則會顯示出完全空白的畫面。若有錯誤訊息，請檢查發生錯誤的那一行

● 利用 PDO 連接資料庫

- PDO 即是一種類別 (**class**)，裡面包含了操作資料庫時會用到的變數與函式
- 類別中的變數稱為「屬性 (property)」，類別內的函式稱為「方法 (Method)」





—登入的使用者名稱

- 使用者名稱必須如下所示，以單引號「'」框住

```
'staff'
```

—登入密碼

- 與使用者名稱一樣，必須以單引號「'」框住

```
'password'
```



● 用 PHP 顯示商品清單

—從連線的資料庫中取得商品資料，並顯示在瀏覽器畫面上

```
List all2.php PHP
<?php require '../header.php';?>
<?php
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
              'staff', 'password');
foreach ($pdo->query('select * from product') as $row) {
    echo '<p>';
    echo $row['id'], ':';
    echo $row['name'], ':';
    echo $row['price'];
    echo '</p>';
}
?>
<?php require '../footer.php':?>
```



一程式若正確執行，會顯示出 product 資料表中的所有商品資料

▼ 利用 PHP 顯示商品清單

PHP	1:松果:700
PHP	2:核桃:270
PHP	3:葵花子:210
PHP	4:杏仁:220
PHP	5:腰果:250
PHP	6:巨人玉米:180
PHP	7:開心果:310
PHP	8:夏威夷豆:600
PHP	9:南瓜子:180
PHP	10:花生:150
PHP	11:枸杞:400

● 在 PHP 中執行 select 敘述

–要在 PHP 程式中執行 SQL 的 select 敘述（p.6-20），程式應撰寫如下

```
$pdo->query('select * from product')
```

–執行 SQL 指令

語法 執行 SQL 指令

```
PDO ->query('SQL指令')
```



● 逐筆處理取得的資料

—將 `query` 方法與 `foreach` 迴圈合在一起，很容易地就能處理多筆資料。範例程式寫法如下

```
foreach ($pdo->query('select * from product') as $row) {  
    ...  
}
```



- 取出單筆資料中的指定資料項

—代入資料的變數 `$row` 是一種陣列，因此程式撰寫方式如下

```
$row['id']
```

—將資料欄名稱指定為陣列索引

語法 取得資料欄內的資料

```
陣列名稱['資料欄名稱']
```


● 簡化程式

—程式可依下列紅字部份修改

```

List all3.php PHP
<?php require '../header.php';?>
<?php
    $pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
                  'staff', 'password');
    foreach ($pdo->query('select * from product') as $row) {
        echo "<p>$row[id]:$row[name]:$row[price]</p>";
    }
?>
<?php require '../footer.php';?>

```

- 在字串內插入變數值

- 以雙引號框住的字串，具有可在字串之中插入變數值的功能

```
echo "$row[name]";
```

- 以字串做為陣列索引時，程式應如下以單引號將索引框住

```
$row['name'];
```

- 在被雙引號框住的字串中，不需再用單引號框住陣列索引

```
"$row[name]"
```


● 利用 **HTML 製作表格** 讓結果更易閱讀

—將程式依下列紅字部份修改

```

List  all4.php PHP
<?php require '../header.php';?>
<table>
<tr><th>商品編號</th><th>商品名稱</th><th>商品價格</th></tr>
<?php
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
              'staff', 'password');
foreach ($pdo->query('select * from product') as $row) {
    echo '<tr>';
    echo '<td>', $row['id'], '</td>';
    echo '<td>', $row['name'], '</td>';
    echo '<td>', $row['price'], '</td>';
    echo '</tr>';
    echo "\n";
}
?>
</table>
<?php require '../footer.php';?>

```


● HTML 中製作表格的標籤

-本例程式中使用了下列 HTML 標籤產生表格

標籤	功能
<table>	定義表格
<tr>	在表格中，定義1個橫列
<th>	在橫列中，定義做為標題欄的內容
<td>	在橫列中，定義資料欄的內容



● 輸出換行符號

—的程式中，定義表格內每一列內容的敘述，會分別顯示成一行

```
<tr><td>1</td><td>松果</td><td>700</td></tr>
<tr><td>2</td><td>核桃</td><td>270</td></tr>
...
```

—若是輸出時沒有換行，則多行敘述會併成一行(不易閱讀)

```
<tr><td>1</td><td>松果</td><td>700</td></tr><tr><td>2</td><td>核
桃</td><td>270</td></tr>...
```

—要讓輸出換行的程式如下

```
echo "\n";
```



● 確保資料正確顯示

—將程式依下列紅字部份修改

```
List all5.php PHP
<?php require '../header.php';?>
<table>
<tr><th>商品編號</th><th>商品名稱</th><th>商品價格</th></tr>
<?php
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
              'staff', 'password');
foreach ($pdo->query('select * from product') as $row) {
    echo '<tr>';
    echo '<td>', htmlspecialchars($row['id']), '</td>';
    echo '<td>', htmlspecialchars($row['name']), '</td>';
    echo '<td>', htmlspecialchars($row['price']), '</td>';
    echo '</tr>';
    echo "\n";
}
?>
</table>
<?php require '../footer.php';?>
```

- 避免特殊字造成版面錯置

-在以下行的程式，將由資料庫取出的資料原原本本的顯示出來

```
$row['name']
```

-利用 PHP 的 `htmlspecialchars` 函式，將程式改寫如下就可先將資料經適當處理後再顯示

```
htmlspecialchars($row['name'])
```

PHP `htmlspecialchars` 函數的功能是用來轉換 **HTML** 特殊符號為僅能顯示用的編碼，例如有人利用網站表單輸入一些清除資料庫的語法或塞入後門程式，通常都會用到一些特殊符號，為了安全起見，所有表單傳遞的資料都應該利用 **PHP `htmlspecialchars` 函數**做第一層的把關，這是網頁設計的安全基礎。

6-4 搜尋商品資料



- 利用 SQL 以商品名稱搜尋

- 首先說明用來搜尋商品資料的 SQL 指令

```
select * from product where name='腰果';
```

▼ 顯示搜尋結果

+ 選項			
← T →			
	id	name	price
<input type="checkbox"/> 編輯	5	腰果	250

- select 敘述的 where 子句

- 若要搜尋 name 資料欄中，值為「腰果」的資料列

```
where name='腰果'
```

- 「=」是 SQL 中用來進行比對的算符

語法 where

```
where 資料欄名稱 = '搜尋的鍵值'
```

● 以商品名稱搜尋商品（輸入畫面）

—請參照下列程式製作輸入商品名稱用的表單

```

List search-input.php PHP
<?php require '../header.php';?>
請輸入商品名稱：
<form action="search-output.php" method="post">
<input type="text" name="keyword">
<input type="submit" value="搜尋">
</form>
<?php require '../footer.php';?>

```

▼ 商品名稱輸入畫面



The screenshot shows a search form with a PHP logo on the left, the text '請輸入商品名稱：' followed by a text input field, and a '搜尋' button.



-在這裡輸入的搜尋關鍵字將會轉輸出用程式

```
<form action="search-output.php" method="post">
```

-輸入的商品名稱用來做為搜尋關鍵字

```
<input type="text" name="keyword">
```

● 以商品名稱搜尋商品（輸出用的程式）

–以 PHP 撰寫程式如下，製作以表單所輸入的關鍵字搜尋商品的功能

search-output.php
PHP

```

<?php require '../header.php';?>
<table>
<tr><th>商品編號</th><th>商品名稱</th><th>商品價格</th></tr>
<?php
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
              'staff', 'password');
$sql=$pdo->prepare('select * from product where name=?');
$sql->execute($_REQUEST['keyword']);
foreach ($sql->fetchAll() as $row) {
    echo '<tr>';
    echo '<td>', $row['id'], '</td>';
    echo '<td>', $row['name'], '</td>';
    echo '<td>', $row['price'], '</td>';
    echo '</tr>';
    echo "\n";
}
?>
</table>
<?php require '../footer.php';?>

```

▼ 輸入「腰果」

請輸入商品名稱：

▼ 「腰果」的搜尋結果

商品編號	商品名稱	商品價格
5	腰果	250

- 預處理 SQL 指令（prepare 方法）

- 以本例來說，SQL 指令應寫為

```
select * from product where name=?
```

- 變數 \$pdo 即為 PDO 類別的 instance(6-3 節)

```
$pdo->prepare('select * from product where name=?')
```

語法 `prepare`

```
PDO的變數->prepare('SQL指令')
```

–prepare 方法在執行後，會回傳已設定好 SQL 指令的 PDOStatement 實例

```
$sql=$pdo->prepare('select * from product where name=?');
```



execute() 方法準備要執行的 SQL 語句，SQL 語句可以包含零個或多個問號 (?) 參數標記，參數在 SQL 執行時會被替換。



● 執行 SQL 指令（execute 方法）

—要執行以傳入參數傳入 prepare 方法的 SQL 指令，必須利用 PHP 內建的 PDOStatement 類別的 execute 方法

```
$sql->execute($_REQUEST['keyword']);
```

語法 execute

```
變數->execute(值)
```

PHP PDO



● 取得 SQL 指令執行結果（fetchAll 方法）

–變數 \$sql 是 PDOStatement 類別的實例

```
foreach ($sql->fetchAll() as $row) {
```

語法 fetchAll

```
foreach (PDO的變數->fetchAll() as 要將取得結果代入的變數)
```

–先將取得的資料逐筆代入變數 \$row

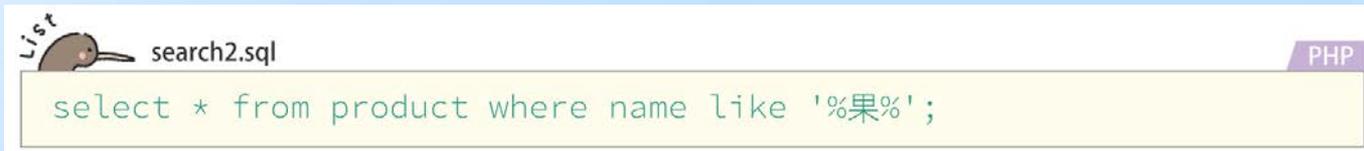
```
$row['name']
```

–再用下行程式將商品名稱輸出到畫面上

```
echo '<td>', $row['name'], '</td>';
```

● 搜尋部份符合的商品

—指定了 shop 資料庫後，點選 [SQL] 頁籤，然後在輸入欄中貼上 chapter6\search2.sql 的SQL 指令如下



```
select * from product where name like '%果%';
```

▼ 顯示搜尋結果

+ 選項		id	name	price
<input type="checkbox"/>	編輯 複製 刪除	1	松果	700
<input type="checkbox"/>	編輯 複製 刪除	5	腰果	250
<input type="checkbox"/>	編輯 複製 刪除	7	開心果	310

● 利用 like 算符做模糊搜尋

–利用 like 算符就可在設定條件時使用「%」符號

```
%果%
```

- ▶ 果 → 單獨一個「果」字。
- ▶ 腰果 → 在「果」字前面還有字串
- ▶ 果物專賣店 → 在「果」字後面還有字串
- ▶ 堅果奶油 → 在「果」字前後都有字串

–利用 like 算符將 where 子句改寫如下

```
where name like '%果%';
```

● 以商品名稱模糊搜尋商品

—參照下列 **PHP** 程式撰寫模糊搜尋功能，找出商品名稱中包含了表單輸入字串的所有商品

```

List  search-output2.php PHP
<?php require '../header.php';?>
<table>
<tr><th>商品編號</th><th>商品名稱</th><th>商品價格</th></tr>
<?php
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
            'staff', 'password');

$sql=$pdo->prepare('select * from product where name like ?');
$sql->execute(['%'.$_REQUEST['keyword'].'%']);
foreach ($sql->fetchAll() as $row) {
    echo '<tr>';
    echo '<td>', $row['id'], '</td>';
    echo '<td>', $row['name'], '</td>';
    echo '<td>', $row['price'], '</td>';
    echo '</tr>';
    echo "\n";
}
?>
</table>
<?php require '../footer.php';?>

```

–所撰寫的 PHP 程式，才能透過輸入表單執行「search-output2.php」

```

List search-input2.php PHP
<?php require '../header.php';?>
請輸入商品名稱：
<form action="search-output2.php" method="post">
<input type="text" name="keyword">
<input type="submit" value="搜尋">
</form>
<?php require '../footer.php';?>
    
```

▼ 輸入「果」



請輸入商品名稱：

▼ 「果」的搜尋結果



商品編號	商品名稱	商品價格
1	松果	700
5	腰果	250
7	開心果	310





● like 算符與萬用字元的使用

– 在要傳入 `prepare` 方法的 SQL 指令中，若要使用 `like` 算符，則程式寫法如下

```
$sql=$pdo->prepare('select * from product where name like ?');
```

– 要代入「？」位置的字串，應像「% 果 %」這樣前後兩側都以 % 框住

```
'%'.$_REQUEST['keyword'].'%'
```

– 將加上「%」之後的字串，以陣列型式傳入 `execute` 方法的傳入參數，就可執行 SQL 指令

```
$sql->execute(['%'.$_REQUEST['keyword'].'%']);
```

6-5 在資料表內新增資料



- 新增商品資料用的 SQL 指令

-在 SQL 的輸入欄中輸入範例檔案 chapter6\insert.sql 內的 SQL 指令如下

```
insert into product values(null, '奶油花生', 200);
```

資料新增完成

✔ 新增了 1 列。
新增的資料列行 id: 12 (查詢花費 0.0684 秒。)

```
insert into product values(null, '奶油花生', 200)
```

新增一筆資料的資料表內容

+ 選項						
← T →						
			id	name	price	
<input type="checkbox"/>	編輯	複製	刪除	1	松果	700
<input type="checkbox"/>	編輯	複製	刪除	2	核桃	270
<input type="checkbox"/>	編輯	複製	刪除	3	葵花子	210
<input type="checkbox"/>	編輯	複製	刪除	4	杏仁	220
<input type="checkbox"/>	編輯	複製	刪除	5	腰果	250
<input type="checkbox"/>	編輯	複製	刪除	6	巨人玉米	180
<input type="checkbox"/>	編輯	複製	刪除	7	開心果	310
<input type="checkbox"/>	編輯	複製	刪除	8	夏威夷豆	600
<input type="checkbox"/>	編輯	複製	刪除	9	南瓜子	180
<input type="checkbox"/>	編輯	複製	刪除	10	花生	150
<input type="checkbox"/>	編輯	複製	刪除	11	枸杞	400
<input type="checkbox"/>	編輯	複製	刪除	12	奶油花生	200

● SQL 的 insert 敘述

–在 SQL 中利用 insert 敘述，即可在指定資料表中加入一筆新的資料

```
insert into product
```

–新資料列的各欄位值，每個值之間以逗號「,」分隔

```
values(null, '奶油花生', 200);
```

–**id** 資料欄的值是由系統自動產生，因此在指令中只需將值指定為 **null** 即可

語法 insert

```
insert into 資料表名稱 values(資料欄 1 的值, 資料欄 2 的值, ...)
```

● 新增商品資料（輸入畫面）

— 參照下列程式，撰寫輸入商品名稱與價格的表單

```

List  insert-input.php PHP
<?php require '../header.php';?>
<p>資料新增：</p>
<form action="insert-output.php" method="post">
  商品名稱<input type="text" name="name">
  價格<input type="text" name="price">
  <input type="submit" value="確定新增">
</form>
<?php require '../footer.php';?>
  
```

▼ 商品資料輸入畫面



資料新增：

商品名稱 價格

● 新增商品資料（輸出處理程式）

— 參照下列程式，撰寫將表單中所輸入的商品名稱與價格等商品資料寫入資料庫的功能

```

List insert-output.php PHP
<?php require '../header.php';?>
<?php
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
              'staff', 'password');
$sql=$pdo->prepare('insert into product values(null, ?, ?)');
if ($sql->execute([$REQUEST['name'], $REQUEST['price']])) {
    echo '新增成功。';
} else {
    echo '新增失敗。';
}
?>
<?php require '../footer.php';?>

```

▼ 輸入商品名稱與價格

資料新增：

按下 [確定新增] 按鈕後，應會顯示「新增成功。」的訊息



在資料表的最後，應會多了一筆「炒花生」的資料

PHP	商品編號	商品名稱	商品價格
	1	松果	700
	2	核桃	270
	3	葵花子	210
PHP	4	杏仁	220
	5	腰果	250
	6	巨人玉米	180
PHP	7	開心果	310
	8	夏威夷豆	600
	9	南瓜子	180
PHP	10	花生	150
	11	枸杞	400
	12	奶油花生	200
PHP	13	炒花生	220

- 在 PHP 程式執行 insert 敘述

- 在 PHP 程式執行的 insert 敘述必須撰寫如下

```
insert into product values(null, ?, ?)
```

- 要執行 insert 敘述時，必須先利用 prepare 方法

```
$pdo->prepare('insert into product values(null, ?, ?)')
```

- prepare 方法會傳回 PDOStatement 的實例

```
$sql=$pdo->prepare('insert into product values(null, ?, ?)');
```




● 先檢查輸入值後再新增

- 在之前程式中，並沒有檢查使用者輸入的值有沒有問題，就直指將它新增到資料庫
- 如果輸入畫面的商品名稱輸入字串「`<script>alert("hello"); </script>`」，價格可輸入任意數值

A screenshot of a web form titled "資料新增:" (Add Data:). The form contains two input fields: "商品名稱" (Item Name) and "價格" (Price). The "商品名稱" field contains the malicious payload `<script>alert("hello"); </script>`, and the "價格" field contains the value "0". A "確定新增" (Confirm Add) button is located to the right of the input fields. A PHP logo is visible in the top left corner of the form's container.

資料新增:		
商品名稱 <input type="text" value='<script>alert("hello"); </script>'/>	價格 <input type="text" value="0"/>	確定新增

一應先確認商品名稱與價格欄的輸入值是否適當，並加入讓特殊意義的標籤失去作用的處理

```

List insert-output2.php PHP
<?php require '../header.php';?>
<?php
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
              'staff', 'password');
$sql=$pdo->prepare('insert into product values(null, ?, ?)');
if (empty($_REQUEST['name'])) {
    echo '請輸入商品名稱。';
} else
if (!preg_match('/[0-9]+/', $_REQUEST['price'])) {
    echo '請以整數輸入商品價格。';
} else
if ($sql->execute(
    htmlspecialchars($_REQUEST['name']), $_REQUEST['price'])
)) {
    echo '新增成功。';
} else {
    echo '新增失敗。';
}
?>
<?php require '../footer.php';?>

```


—輸入商品名稱與價格後，按下 [確定新增] 按鈕

▼ 輸入商品名稱與價格。

	資料新增：	
	商品名稱 <input type="text" value="蜂蜜花生"/>	價格 <input type="text" value="240"/>
		<input type="button" value="確定新增"/>

—按下 [確定新增] 按鈕後，將會顯示「新增成功。」訊息

	新增成功。
---	-------

—價格的值不是整數時，也同樣不能新增

	資料新增：	
	商品名稱 <input type="text" value="蜂蜜花生"/>	價格 <input type="text" value="abc"/>
		<input type="button" value="確定新增"/>

● 檢查輸入值

```
if (empty($_REQUEST['name'])) {
```

–Empty

語法 empty

```
empty(值)
```

–接下來，檢查價格是否為整數。

```
if (!preg_match('/[0-9]+/', $_REQUEST['price'])) {
```


6-6 修改資料庫內的商品資料



● 修改商品資料的 SQL 指令

-在 SQL 指令輸入欄中輸入檔案 chapter6\update.sql 中的指令如下後，按下右下的 [執行] 按鈕，執行 SQL 指令

```
update product set name='高級松果', price=900 where id=1;
```

修改前的資料表內容

PHP	商品編號	商品名稱	商品價格
1	1	松果	700
2	2	核桃	270
3	3	葵花子	210
4	4	杏仁	220
5	5	腰果	250
6	6	巨人玉米	180
7	7	開心果	310
8	8	夏威夷豆	600
9	9	南瓜子	180
10	10	花生	150
11	11	枸杞	400

修改後的資料表內容

PHP	商品編號	商品名稱	商品價格
1	1	高級松果	900
2	2	核桃	270
3	3	葵花子	210
4	4	杏仁	220
5	5	腰果	250
6	6	巨人玉米	180
7	7	開心果	310
8	8	夏威夷豆	600
9	9	南瓜子	180
10	10	花生	150
11	11	枸杞	400



● 修改商品資料（輸入畫面）

—接著改以 PHP 製作修改資料庫中商品資料的程式

```
update-input.php
<?php require '../header.php';?>
<table>
<tr><th>商品編號</th><th>商品名稱</th><th>商品價格</th></tr>
<?php
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
    'staff', 'password');
foreach ($pdo->query('select * from product') as $row) {
    echo '<tr><form action="update-output.php" method="post">';
    echo '<input type="hidden" name="id" value="' . $row['id'],
    '">';
    echo '<td>', $row['id'], '</td>';
    echo '<td>';
    echo '<input type="text" name="name" value="' . $row['name'],
    '">';
    echo '</td>';
    echo '<td>';
    echo '<input type="text" name="price" value="' .
    $row['price'], '">';
    echo '</td>';
    echo '<td><input type="submit" value="確定修改"></td>';
    echo '</form></tr>';
    echo "\n";
}
?>
</table>
<?php require '../footer.php';?>
```

一程式若正確執行，則會顯示出商品一覽表

▼ 修改資料的輸入畫面

PHP	商品編號	商品名稱	商品價格	
PHP	1	<input type="text" value="松果"/>	<input type="text" value="700"/>	<input type="button" value="確定修改"/>
PHP	2	<input type="text" value="核桃"/>	<input type="text" value="270"/>	<input type="button" value="確定修改"/>
PHP	3	<input type="text" value="葵花子"/>	<input type="text" value="210"/>	<input type="button" value="確定修改"/>
PHP	4	<input type="text" value="杏仁"/>	<input type="text" value="220"/>	<input type="button" value="確定修改"/>
PHP	5	<input type="text" value="腰果"/>	<input type="text" value="250"/>	<input type="button" value="確定修改"/>
PHP	6	<input type="text" value="巨人玉米"/>	<input type="text" value="180"/>	<input type="button" value="確定修改"/>
PHP	7	<input type="text" value="開心果"/>	<input type="text" value="310"/>	<input type="button" value="確定修改"/>
PHP	8	<input type="text" value="夏威夷豆"/>	<input type="text" value="600"/>	<input type="button" value="確定修改"/>
PHP	9	<input type="text" value="南瓜子"/>	<input type="text" value="180"/>	<input type="button" value="確定修改"/>
PHP	10	<input type="text" value="花生"/>	<input type="text" value="150"/>	<input type="button" value="確定修改"/>
PHP	11	<input type="text" value="枸杞"/>	<input type="text" value="400"/>	<input type="button" value="確定修改"/>



● 將 type 屬性設為 hidden

–商品名稱則利用 `<input>` 標籤撰寫如下，此時的 type 屬性指定為 `hidden`

```
<input type="hidden" name="id" value=" 1" >
```

type 屬性設定為 hidden 使用者不可見，其目的是收集信息

–要產生這樣的 HTML 標記，程式如下

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

● 更新商品資料（顯示結果）

一參照下列程式，撰寫將表單輸入的**商品名稱**與**價格**更新到資料庫的 PHP 程式

```

update-output.php
<?php require '../header.php';?>
<?php
$db=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
            'staff', 'password');
$sql=$db->prepare('update product set name=?, price=? where
id=?');
if (empty($_REQUEST['name'])) {
    echo '請輸入商品名稱。';
} else 可用 elseif 取代
if (!preg_match('/[0-9]+/', $_REQUEST['price'])) {
    echo '請以整數輸入商品價格。';
} else
if ($sql->execute(
    [htmlspecialchars($_REQUEST['name']), $_REQUEST['price'],
    $_REQUEST['id']]
)) {
    echo '修改成功。';
} else {
    echo '修改失敗。';
}
?>
<?php require '../footer.php';?>

```


6-7 刪除資料庫內的商品資料



- 刪除商品資料的 SQL 指令

```
List delete.sql SQL  
delete from product where id=1;
```

—刪除資料前的資料表內容

	商品編號	商品名稱	商品價格
PHP	1	松果	700
	2	核桃	270
PHP	3	葵花子	210
	4	杏仁	220
	5	腰果	250
PHP	6	巨人玉米	180
	7	開心果	310
	8	夏威夷豆	600
	9	南瓜子	180
PHP	10	花生	150
	11	枸杞	400

一 刪除資料後的資料表內容

	商品編號	商品名稱	商品價格
PHP	2	核桃	270
	3	葵花子	210
PHP	4	杏仁	220
	5	腰果	250
	6	巨人玉米	180
PHP	7	開心果	310
	8	夏威夷豆	600
	9	南瓜子	180
PHP	10	花生	150
	11	枸杞	400

● SQL 的 delete 敘述

–delete 敘述是用來刪除指定資料庫中的指定資料列

```
delete from product
```

語法 delete (刪除所有資料)

```
delete from 資料表名稱
```

這個語法中沒有 **where** 條件子句，執行後的結果會將該資料表的內容全數刪除。

● 刪除商品資料（輸入畫面）

```

List delete-input.php PHP
<?php require '../header.php';?>
<table>
<tr><th>商品編號</th><th>商品名稱</th><th>商品價格</th></tr>
<?php
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
              'staff', 'password');
foreach ($pdo->query('select * from product') as $row) {
    echo '<tr>';
    echo '<td>'.$row['id'].'</td>';
    echo '<td>'.$row['name'].'</td>';
    echo '<td>'.$row['price'].'</td>';
    echo '<td>';
    echo '<a href="delete-output.php?id=', $row['id'], '">確定刪除</a>';
    echo '</td>';
    echo '</tr>';
    echo "\n";
}
?>
</table>
<?php require '../footer.php';?>

```

▼ 加上 [確定刪除] 連結後的商品一覽

商品編號	商品名稱	商品價格	
1	松果	700	確定刪除
2	核桃	270	確定刪除
3	葵花子	210	確定刪除
4	杏仁	220	確定刪除
5	腰果	250	確定刪除
6	巨人玉米	180	確定刪除
7	開心果	310	確定刪除
8	夏威夷豆	600	確定刪除
9	南瓜子	180	確定刪除
10	花生	150	確定刪除
11	枸杞	400	確定刪除

● 含有 REQUEST 參數的連結

- [確定刪除] 連結必須利用 HTML 的 `<a>` 標籤製作

```
<a href="delete-output.php">確定刪除</a>
```

- 必須利用 REQUEST，讓程式知道現在要刪除的是哪一筆資料

```
<a href="delete-output.php?id=1">確定刪除</a>
```



單一 REQUEST 參數：

要開啟的檔案名稱?REQUEST參數名=值

多個 REQUEST 參數：

要開啟的檔案名稱?REQUEST參數名1=值1 & REQUEST參數名2=值2 ...

● 刪除商品資料（顯示結果）

—接著製作所指定用來刪除商品資料的 PHP 程式

```

delete-output.php
<?php require '../header.php';?>
<?php
$db=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
            'staff', 'password');
$sql=$db->prepare('delete from product where id=?');
if ($sql->execute([$$_REQUEST['id']])) {
    echo '刪除成功。';
} else {
    echo '刪除失敗。';
}
?>
<?php require '../footer.php';?>
    
```

▼ 刪除成功

 刪除成功。

▼ 資料刪除後的商品資料表

商品編號	商品名稱	商品價格
2	核桃	270
3	葵花子	210
4	杏仁	220
5	腰果	250
6	巨人玉米	180
7	開心果	310
8	夏威夷豆	600
9	南瓜子	180
10	花生	150
11	枸杞	400

經由 phpMyAdmin 顯示刪除後結果 →

- 從 PHP 程式中執行 delete 敘述

- 要從 PHP 程式執行的 delete 敘述如下

```
delete from product where id=?
```

- 將 delete 敘述的字串，傳入 prepare 方法中

```
$pdo->prepare('delete from product where id=?');
```

- 與前面的範例一樣，將代入的變數名稱指定為 \$sql

```
$sql=$pdo->prepare('delete from product where id=?');
```

- 利用 if 判斷執行是否成功

```
if ($sql->execute([$_REQUEST['id']])) {
```

6-8 資料庫功能的整合



● 製作**新增功能**的表單

—首先製作用來新增商品資料的表單

```
edit.php
<?php require '../header.php';?>
<table>
<tr><th>商品編號</th><th>商品名稱</th><th>商品價格</th></tr>
<tr>
<td><input type="hidden" name="command" value="insert"> 記錄何種功能
</td></td>
<td><input type="text" name="name"></td>
<td><input type="text" name="price"></td>
<td><input type="submit" value="確定新增"></td>
</form>
</tr>
</table>
<?php require '../footer.php';?>
```

▼ 新增資料用的表單

	商品編號	商品名稱	商品價格	<input type="button" value="確定新增"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>	

- 利用 REQUEST 參數區分功能

- 宣告一個名稱 **command** 的 **REQUEST** 參數，用來傳送表示欲執行功能的字串

```
<input type="hidden" name="command" value="insert">
```

● 製作修改與刪除資料用的表單

—接著製作修改、刪除商品資料的表單。程式內容如下

```

edit2.php
<?php require '../header.php';?>
<table>
<tr><th>商品編號</th><th>商品名稱</th><th>價格</th></tr>
<?php
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
    'staff', 'password');
foreach ($pdo->query('select * from product') as $row) {
    echo '<tr>';
    echo '<form action="edit3.php" method="post">';
    echo '<input type="hidden" name="command" value="update">';
    echo '<input type="hidden" name="id" value="' . $row['id'] . '>';
    echo '<td>', $row['id'], '</td>';
    echo '<td>';
    echo '<input type="text" name="name" value="' . $row['name'] . '>';
    echo '</td>';
    echo '<td>';
    echo '<input type="text" name="price" value="' . $row['price'] . '>';
    echo '</td>';
    echo '<td><input type="submit" value="確定修改"></td>';
    echo '</form>';
    echo '<form action="edit3.php" method="post">';
    echo '<input type="hidden" name="command" value="delete">';
    echo '<input type="hidden" name="id" value="' . $row['id'] . '>';
    echo '<td><input type="submit" value="確定刪除"></td>';
    echo '</form>';
    echo '</tr>';
    echo "\n";
}
?>
<tr>
<form action="edit3.php" method="post">
<input type="hidden" name="command" value="insert">
<td></td>
<td><input type="text" name="name"></td>
<td><input type="text" name="price"></td>
<td><input type="submit" value="確定新增"></td>
</form>
</tr>
</table>
<?php require '../footer.php';?>
    
```

一程式若正確執行

▼ 修改、刪除資料表的表單

PHP	商品編號	商品名稱	價格		
PHP	1	<input type="text" value="松果"/>	<input type="text" value="700"/>	<input type="button" value="確定修改"/>	<input type="button" value="確定刪除"/>
PHP	2	<input type="text" value="核桃"/>	<input type="text" value="270"/>	<input type="button" value="確定修改"/>	<input type="button" value="確定刪除"/>
PHP	3	<input type="text" value="葵花子"/>	<input type="text" value="210"/>	<input type="button" value="確定修改"/>	<input type="button" value="確定刪除"/>
PHP	4	<input type="text" value="杏仁"/>	<input type="text" value="220"/>	<input type="button" value="確定修改"/>	<input type="button" value="確定刪除"/>
PHP	5	<input type="text" value="腰果"/>	<input type="text" value="250"/>	<input type="button" value="確定修改"/>	<input type="button" value="確定刪除"/>
PHP	6	<input type="text" value="巨人玉米"/>	<input type="text" value="180"/>	<input type="button" value="確定修改"/>	<input type="button" value="確定刪除"/>
PHP	7	<input type="text" value="開心果"/>	<input type="text" value="310"/>	<input type="button" value="確定修改"/>	<input type="button" value="確定刪除"/>
PHP	8	<input type="text" value="夏威夷豆"/>	<input type="text" value="600"/>	<input type="button" value="確定修改"/>	<input type="button" value="確定刪除"/>
PHP	9	<input type="text" value="南瓜子"/>	<input type="text" value="180"/>	<input type="button" value="確定修改"/>	<input type="button" value="確定刪除"/>
PHP	10	<input type="text" value="花生"/>	<input type="text" value="150"/>	<input type="button" value="確定修改"/>	<input type="button" value="確定刪除"/>
PHP	11	<input type="text" value="枸杞"/>	<input type="text" value="400"/>	<input type="button" value="確定修改"/>	<input type="button" value="確定刪除"/>
		<input type="text"/>	<input type="text"/>	<input type="button" value="確定新增"/>	

● 製作刪除按鈕

–[確定刪除] 按鈕的標籤寫法如下

```
<input type="submit" value="確定刪除">
```

–必須再加入下行所示的 `<input>` 標籤

```
<input type="hidden" name="id" value="1">
```

–`<input>` 標籤可用下行程式產生。`$row['id']` 是指從資料庫取得的商品編號

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

依據這一行`$row['id']`的內容，達到刪除資料和修改資料的目的。



● 執行新增、修改、刪除

—最後，製作進行新增、修改、刪除處理的程式。程式內容如下

```
edit3.php PHP
<?php require '../header.php';?>
<table>
<tr><th>商品編號</th><th>商品名稱</th><th>價格</th></tr>
<?php
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
              'staff', 'password');
if (isset($_REQUEST['command'])) {
    switch ($_REQUEST['command']) {
        case 'insert':
            if (empty($_REQUEST['name']) ||
                !preg_match('/[0-9]+/', $_REQUEST['price'])) break;
            $sql=$pdo->prepare('insert into product values(null,?,?)');
            $sql->execute(
                [htmlspecialchars($_REQUEST['name']), $_REQUEST['price']]);
            break;
        case 'update':
            if (empty($_REQUEST['name']) ||
                !preg_match('/[0-9]+/', $_REQUEST['price'])) break;
            $sql=$pdo->prepare(
                'update product set name=?, price=? where id=?');
            $sql->execute(
                [htmlspecialchars($_REQUEST['name']), $_REQUEST['price'],
                 $_REQUEST['id']]);
            break;
        case 'delete':
            $sql=$pdo->prepare('delete from product where id=?');
            $sql->execute($_REQUEST['id']);
            break;
    }
}
```

● 重新讀取資料列

```

foreach ($pdo->query('select * from product') as $row) {
    echo '<tr>';
    echo '<form action="edit3.php" method="post">';
    echo '<input type="hidden" name="command" value="update">';
    echo '<input type="hidden" name="id" value="", $row['id'], "">';
    echo '<td>', $row['id'], '</td>';
    echo '<td>';
    echo '<input type="text" name="name" value="", $row['name'], "">';
    echo '</td>';
    echo '<td>';
    echo '<input type="text" name="price" value="", $row['price'], "">';
    echo '</td>';
    echo '<td><input type="submit" value="確定修改"></td>';
    echo '</form>';
    echo '<form action="edit3.php" method="post">';
    echo '<input type="hidden" name="command" value="delete">';
    echo '<input type="hidden" name="id" value="", $row['id'], "">';
    echo '<td><input type="submit" value="確定刪除"></td>';
    echo '</form>';
    echo '</tr>';
    echo "\n";
}
?>
<tr>
<form action="edit3.php" method="post">
<input type="hidden" name="command" value="insert">
<td></td>
<td><input type="text" name="name"></td>
<td><input type="text" name="price"></td>
<td><input type="submit" value="確定新增"></td>
</form>
</tr>
</table>
<?php require '../footer.php';?>

```

● 新增商品資料

一 要新增資料時，要先在畫面下方的新增用欄位輸入商品名稱與價格後，按下 [確定新增] 按鈕

PHP	商品編號	商品名稱	價格		
PHP	1	松果	700	確定修改	確定刪除
PHP	2	核桃	270	確定修改	確定刪除
PHP	3	葵花子	210	確定修改	確定刪除
PHP	4	杏仁	220	確定修改	確定刪除
PHP	5	腰果	250	確定修改	確定刪除
PHP	6	巨人玉米	180	確定修改	確定刪除
PHP	7	開心果	310	確定修改	確定刪除
PHP	8	夏威夷豆	600	確定修改	確定刪除
PHP	9	南瓜子	180	確定修改	確定刪除
PHP	10	花生	150	確定修改	確定刪除
PHP	11	枸杞	400	確定修改	確定刪除
		巧克力花生	260	確定新增	

按下 [確定新增] 按鈕後，商品資料就會被新增到資料庫

商品編號	商品名稱	價格		
1	松果	700	確定修改	確定刪除
2	核桃	270	確定修改	確定刪除
3	葵花子	210	確定修改	確定刪除
4	杏仁	220	確定修改	確定刪除
5	腰果	250	確定修改	確定刪除
6	巨人玉米	180	確定修改	確定刪除
7	開心果	310	確定修改	確定刪除
8	夏威夷豆	600	確定修改	確定刪除
9	南瓜子	180	確定修改	確定刪除
10	花生	150	確定修改	確定刪除
11	枸杞	400	確定修改	確定刪除
12	巧克力花生	260	確定修改	確定刪除
			確定新增	

● 修改商品資料

—只要在商品一覽表對應的資料欄位中，直接修改商品名稱和價格後，按下【確定修改】按鈕即可

▼ 輸入要修改的資料

PHP	商品編號	商品名稱	價格		
	1	松果	700	確定修改	確定刪除
	2	核桃	270	確定修改	確定刪除
PHP	3	葵花子	210	確定修改	確定刪除
	4	杏仁	220	確定修改	確定刪除
	5	腰果	250	確定修改	確定刪除
PHP	6	巨人玉米	180	確定修改	確定刪除
	7	開心果	310	確定修改	確定刪除
	8	夏威夷豆	600	確定修改	確定刪除
PHP	9	南瓜子	180	確定修改	確定刪除
	10	花生	150	確定修改	確定刪除
PHP	11	枸杞	400	確定修改	確定刪除
	12	黑巧克力花生	260	確定修改	確定刪除
PHP				確定新增	

▼ 修改完成後

PHP	商品編號	商品名稱	價格		
	1	松果	700	確定修改	確定刪除
	2	核桃	270	確定修改	確定刪除
PHP	3	葵花子	210	確定修改	確定刪除
	4	杏仁	220	確定修改	確定刪除
	5	腰果	250	確定修改	確定刪除
PHP	6	巨人玉米	180	確定修改	確定刪除
	7	開心果	310	確定修改	確定刪除
	8	夏威夷豆	600	確定修改	確定刪除
PHP	9	南瓜子	180	確定修改	確定刪除
	10	花生	150	確定修改	確定刪除
PHP	11	枸杞	400	確定修改	確定刪除
	12	黑巧克力花生	260	確定修改	確定刪除
PHP				確定新增	

● 刪除商品資料

—以刪除「黑巧克力花生」為例，按下它的 [確定刪除] 按鈕 ①

PHP	商品編號	商品名稱	價格		
	1	松果	700	確定修改	確定刪除
	2	核桃	270	確定修改	確定刪除
PHP	3	葵花子	210	確定修改	確定刪除
	4	杏仁	220	確定修改	確定刪除
	5	腰果	250	確定修改	確定刪除
PHP	6	巨人玉米	180	確定修改	確定刪除
	7	開心果	310	確定修改	確定刪除
	8	夏威夷豆	600	確定修改	確定刪除
PHP	9	南瓜子	180	確定修改	確定刪除
	10	花生	150	確定修改	確定刪除
	11	枸杞	400	確定修改	確定刪除
PHP	12	黑巧克力花生	260	確定修改	確定刪除
				確定新增	

①

▼ 刪除完成後

PHP	商品編號	商品名稱	價格		
	1	松果	700	確定修改	確定刪除
	2	核桃	270	確定修改	確定刪除
PHP	3	葵花子	210	確定修改	確定刪除
	4	杏仁	220	確定修改	確定刪除
	5	腰果	250	確定修改	確定刪除
PHP	6	巨人玉米	180	確定修改	確定刪除
	7	開心果	310	確定修改	確定刪除
	8	夏威夷豆	600	確定修改	確定刪除
PHP	9	南瓜子	180	確定修改	確定刪除
	10	花生	150	確定修改	確定刪除
PHP	11	枸杞	400	確定修改	確定刪除
				確定新增	



● 指定要執行的功能

-在程式開始處理之前，必須先判斷是否有 `command` 這個 `REQUEST` 參數。利用 `isset` 函式（4-1 節）檢查

```
if (isset($_REQUEST['command'])) {
```


小結



- 資料庫的基本操作包含了新增（Create）、讀取（Read）、修改（Update）、刪除（Delete）等功能。可取各功能的首字母，將它們合稱為 CRUD。這些功能分別對應到 SQL 語法中的 insert、select、update、delete。Chapter6 即介紹了在 PHP 程式中執行這些處理的方法。透過 Chapter6，您可以學到所有資料庫的基本操作。
- 在下一章 Chapter7 中，將以建立一個購物網站為例，帶您活用在本章所學的資料庫操作。