



第 7 章

實用的 PHP 程式 以購物網站為例

7-1 建立存放商品與顧客資料的資料庫



- 利用 SQL 程式建立資料庫

- 在 phpMyAdmin 上執行 SQL 程式，建立購物網站的資料庫
- 利用文字編輯器開啟檔案後，將檔案內容全選並複製後，貼到
- phpMyAdmin 的指令輸入欄

shop.sql

在 phpMyAdmin 建立資料庫

在 SQL 程式內容貼到指令輸入欄

按下[執行]按鈕，建立資料庫



● 建立資料庫與使用者

—進行資料庫建立與使用者新增，各行程式所代表的功能如下

● drop database 指令

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

● create database 指令

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

● grant 指令

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

● use 指令

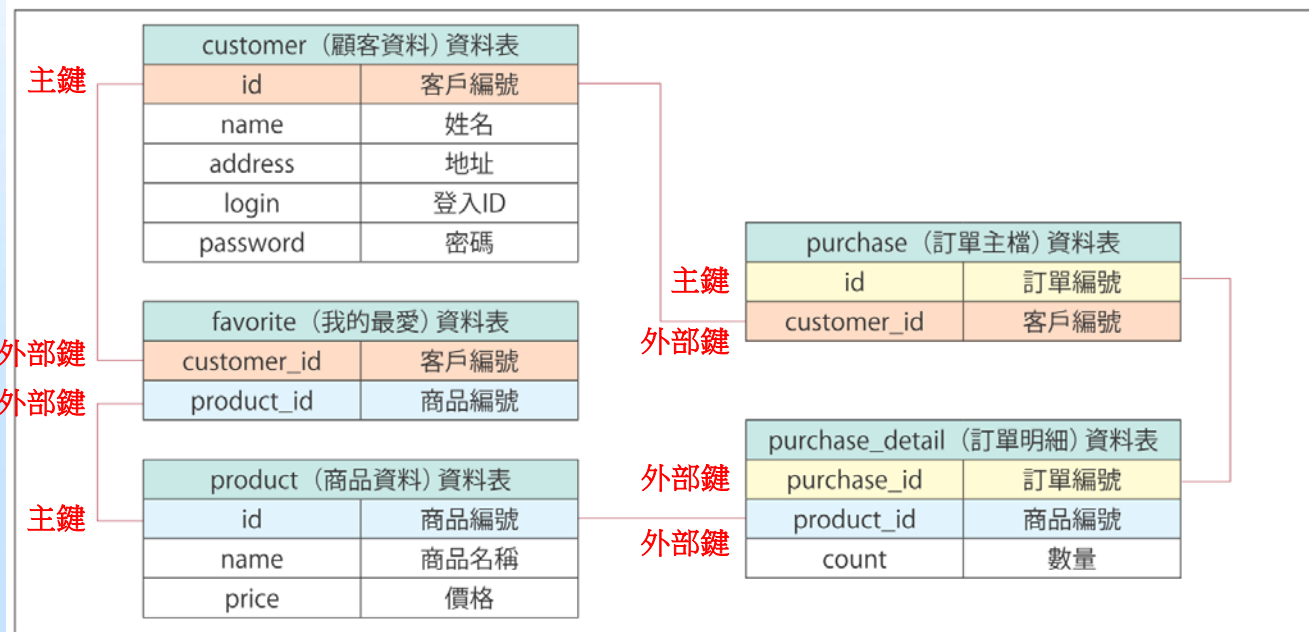
```
use shop;
```

● 建立資料表

-在 shop 資料庫中，利用 create table 指令建立下列資料表

資料表名稱	概述	包含的資料欄
product	商品資料	商品編號、商品名稱、價格
customer	顧客資料	客戶編號、姓名、地址、登入ID、密碼
purchase	訂單主檔	訂單編號、客戶編號
purchase_detail	訂單明細	訂單編號、商品編號、數量
favorite	我的最愛	客戶編號、商品編號

一資料表之間的關係如下圖所示



● 新增資料

-本例的程式只將資料新增到 product（商品）資料表與 customer（顧客）資料表

● product 資料表

```
insert into product values(null, '松果', 700);
```

● product 資料表內新增資料

資料欄	內容值
id	null（自動產生編號）
name	松果
price	700

●customer 資料表:要在 customer 資料表中新增顧客資料，以下行敘述為例

```
insert into customer values(null, '熊木 和夫',  
                             '東京都新宿區西新宿2-8-1', 'kuaka', 'BearTree1');
```

●customer 資料表內新增資料

資料欄	內容值
id	null (自動產生編號)
name	熊木 和夫
address	東京都新宿區西新宿2-8-1
login	kuaka
password	BearTree1

7-2 網站登入與登出



● 製作登入畫面

—首先，製作輸入登入 ID 與密碼的畫面。多數購物網站，都會提供登入功能，使用者只要登入，若要能自動帶出過去使用網站時留存的資料，必須利用 **Session** 機制。

A screenshot of a web page's login section. On the left, there is a vertical brown bar containing two circular PHP logos. To the right, a navigation menu includes links for '商品', '我的最愛', '購買記錄', '購物車', '結帳', '登入', '登出', and '會員登錄'. Below the menu, there are two input fields: '登入ID' and '密碼', followed by a '登入' button.

```
login-input.php
```

```
<?php require '../header.php';?>
<?php require 'menu.php';?>
<form action="login-output.php" method="post">
登入ID<input type="text" name="login"><br>
密碼<input type="password" name="password"><br>
<input type="submit" value="登入">
</form>
<?php require '../footer.php';?>
```



● 功能選單

–利用 `require` 敘述即可將外部 PHP 程式，載入目前所在程式中

```
<?php require 'menu.php'; ?>
```

–`menu.php` 中則是指定了每個功能的對應 PHP 程式

```
<a href="product.php">商品</a>
```

menu.php:

```
<a href="product.php">商品</a>
```

```
<a href="favorite-show.php">我的最愛</a>
```

```
<a href="history.php">購買記錄</a>
```

```
<a href="cart-show.php">購物車</a>
```

```
<a href="purchase-input.php">結帳</a>
```

```
<a href="login-input.php">登入</a>
```

```
<a href="logout-input.php">登出</a>
```

```
<a href="customer-input.php">會員登錄</a>
```

```
<hr>
```

● 製作登入機制

— 參照下列程式，撰寫以登入 ID 與密碼登入網站的程式

```

login-output.php
<?php require '../header.php';?>
<?php require 'menu.php';?>
<?php
session_start(); 使用 session 前，必須先執行 session_start()
unset($_SESSION['customer']);
$db=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
    'staff', 'password');
$sql=$db->prepare('select * from customer where login=? and
password=?');
$sql->execute([$_REQUEST['login'], $_REQUEST['password']]);
foreach ($sql->fetchAll() as $row) {
    $_SESSION['customer']=[ 將使用者個資存入 session。
        'id'=>$row['id'], 'name'=>$row['name'],
        'address'=>$row['address'], 'login'=>$row['login'],
        'password'=>$row['password']];
}
if (isset($_SESSION['customer'])) {
    echo '親愛的', $_SESSION['customer']['name'], '、歡迎光臨。';
} else {
    echo '登入ID或密碼有誤。';
}
?>
<?php require '../footer.php';?>
    
```

▼ 登入成功

PHP 商品 我的最愛 購買記錄 購物車 結帳 登入 登出 會員登錄

PHP 親愛的熊木 和夫、歡迎光臨。

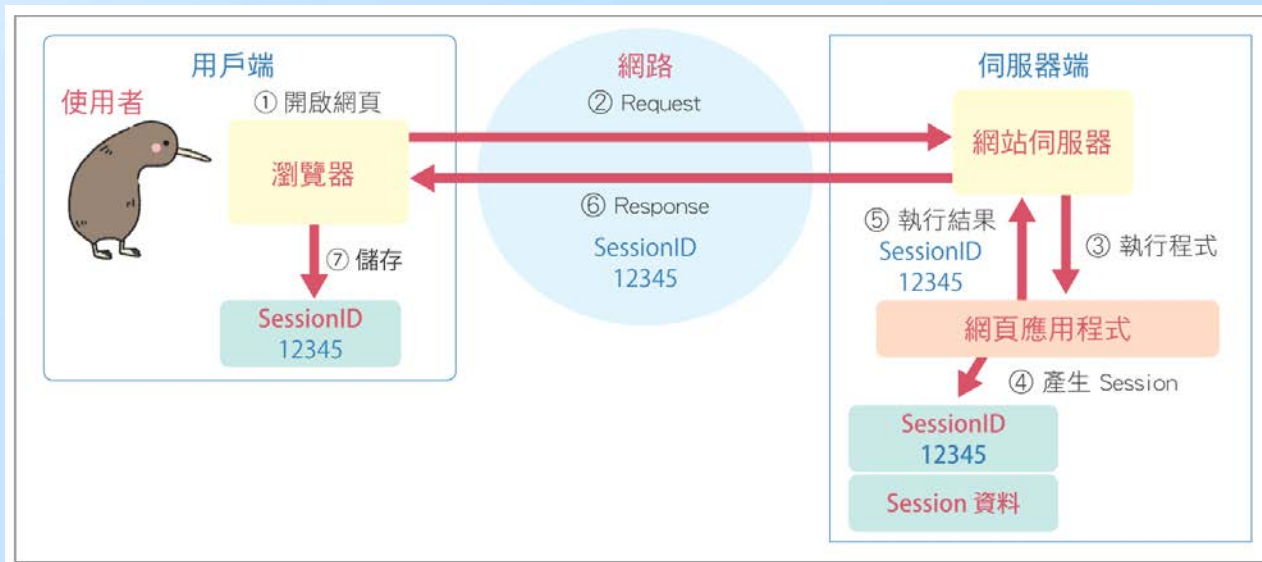
▼ 登入失敗

PHP 商品 我的最愛 購買記錄 購物車 結帳 登入 登出 會員登錄

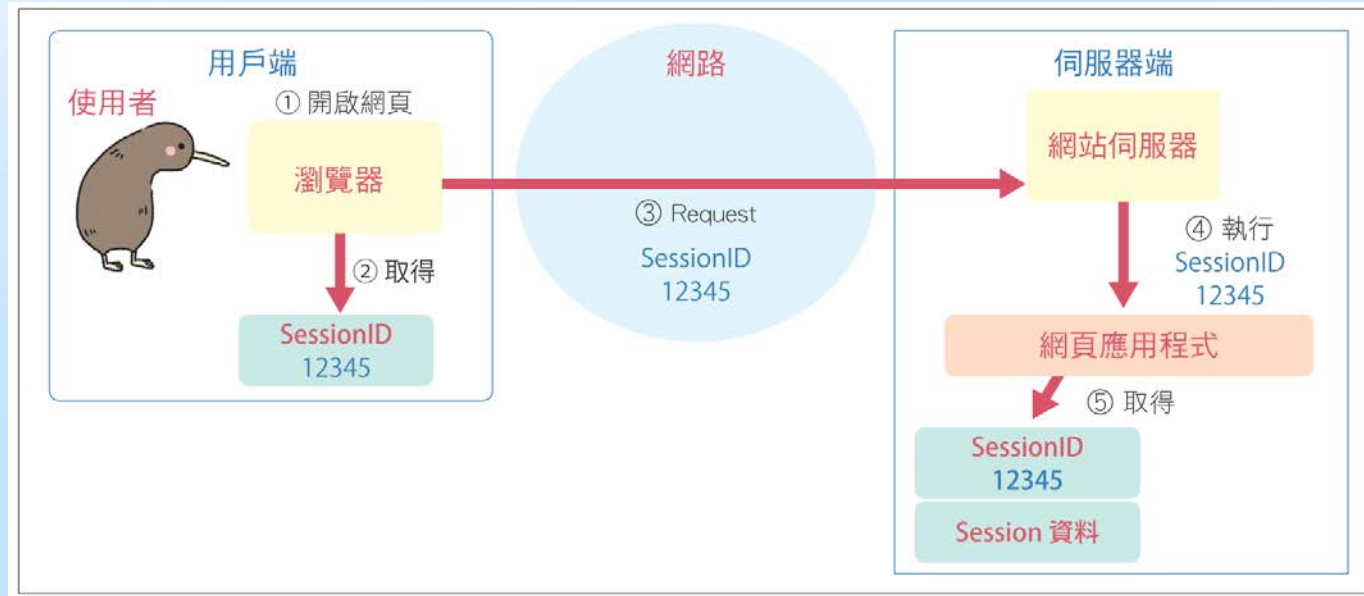
PHP 登入ID或密碼有誤。

● Session

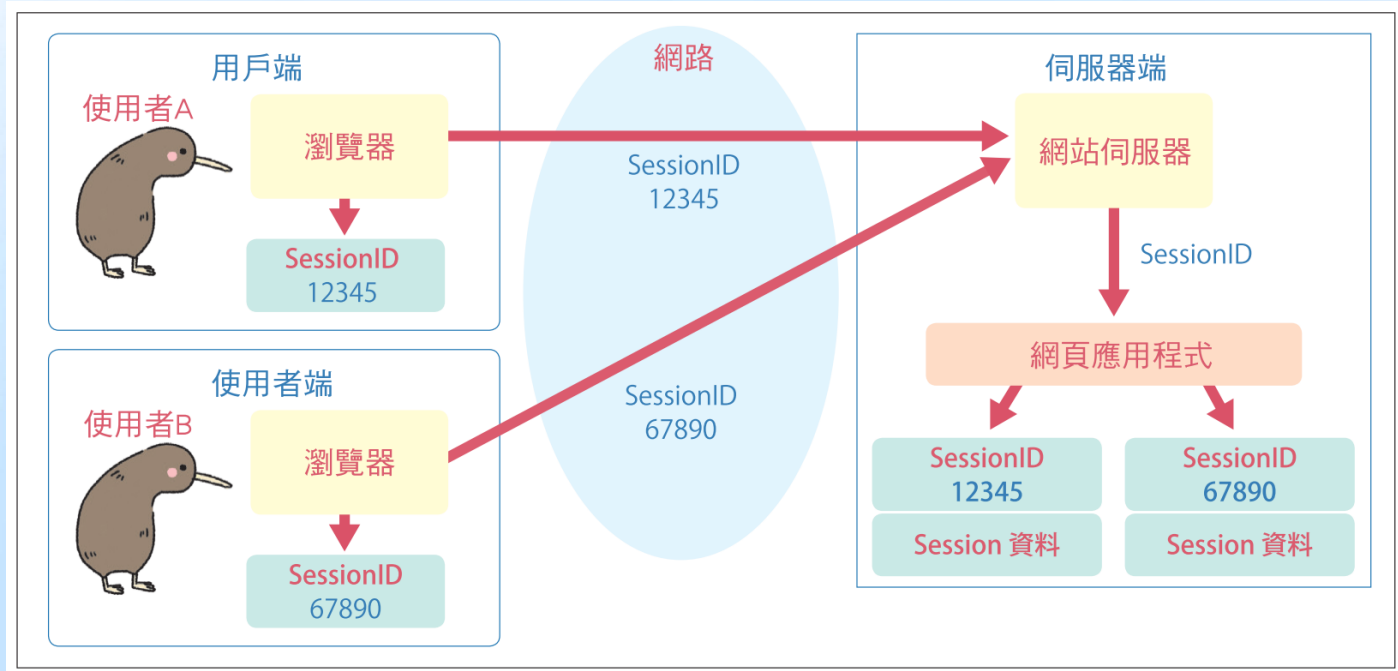
- 「Session」是在網頁應用程式中用來放置各使用者資料的機制
- 產生 SessionID 與 Session 資料



- 在 PHP 中，可透過 `$_SESSION` 陣列存取 Session 資料
- 將 SessionID 傳送到網站伺服器

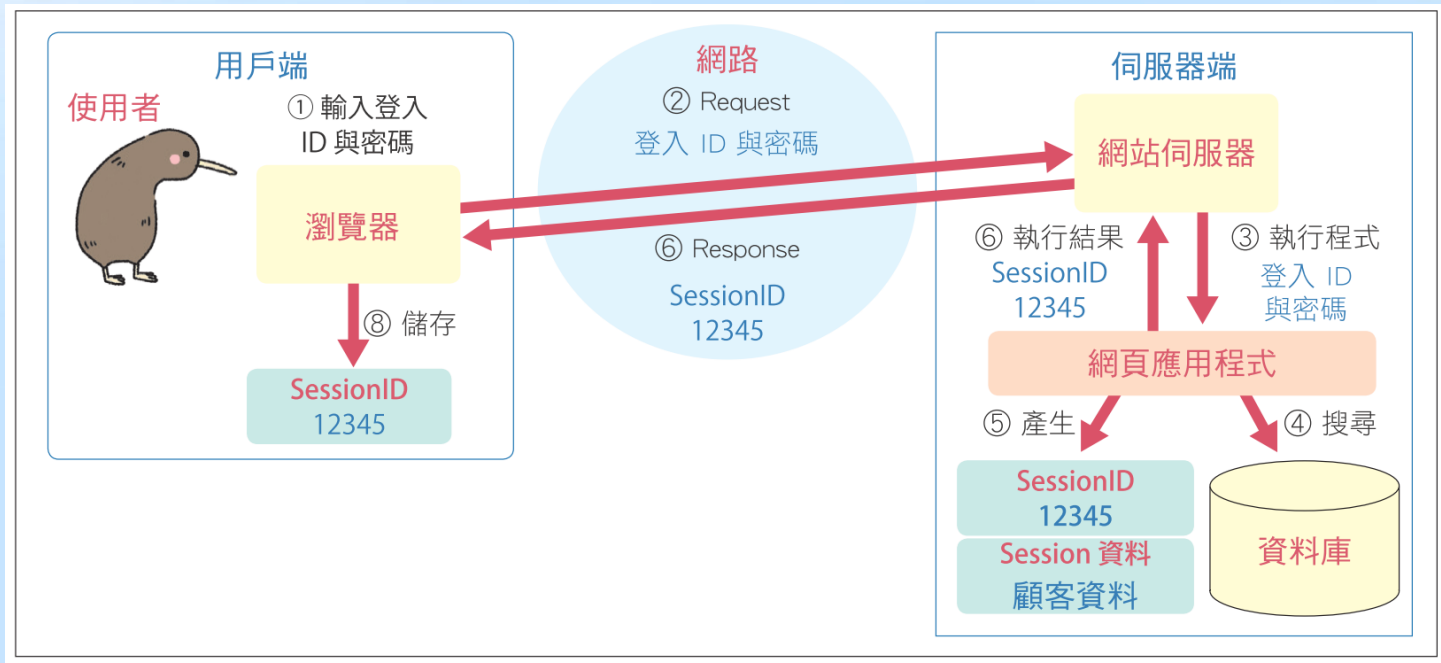


—每個使用者會有自己的 Session 資料



● 利用 Session 製作登入功能

–以 Session 製作的登入功能流程如下所示





● Session 啟用

- PHP** 中要存取 **Session** 之前，須先呼叫 **session_start** 函式啟用 **Session**

```
session_start();
```

- 利用 **\$_SESSION** 陣列存取 **Session** 資料

```
$_SESSION['customer']
```

- 將變數刪除，必須使用 **unset** 函式，下列指令只會刪除所引為 **'customer'** 的元素。

```
unset($_SESSION['customer']);
```



● 檢查登入 ID 與密碼

– 接下來將登入 ID 及密碼合併成一組條件，搜尋資料庫

```
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',  
            'staff', 'password');
```

– 撰寫用來搜尋登入 ID 與密碼的 SQL 指令

```
'select * from customer where login=輸入的登入ID and password=輸入的密碼'
```

– 利用 prepare 方法 (6-4 節 Step3) 預處理 SQL 的敘述

```
$sql=$pdo->prepare(  
    'select * from customer where login=? and password=?');
```

– 利用 execute 方法 (6-4 節 Step3) 執行 SQL 指令

```
[$_REQUEST['login'], $_REQUEST['password']]
```

– 將陣列傳入 execute 方法

```
$sql->execute($_REQUEST['login'], $_REQUEST['password']);
```



● 產生 Session 資料

-execute 方法執行 SQL 指令後的結果

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

-以各資料欄的名稱做為索引，建立陣列如下

```
[  
    'id'=>$row['id'],  
    'name'=>$row['name'],  
    'address'=>$row['address'],  
    'login'=>$row['login'],  
    'password'=>$row['password']  
]
```



一將陣列指定為 \$_SESSION['customer']

```
$_SESSION['customer']=[  
    'id'=>$row['id'], 'name'=>$row['name'],  
    'address'=>$row['address'], 'login'=>$row['login'],  
    'password'=>$row['password']];
```

一取得顧客資料

資料項目	寫法
客戶編號	<code>\$_SESSION('customer')['id']</code>
客戶姓名	<code>\$_SESSION('customer')['name']</code>
地址	<code>\$_SESSION('customer')['address']</code>
登入ID	<code>\$_SESSION('customer')['login']</code>
密碼	<code>\$_SESSION('customer')['password']</code>



● 顯示登入結果

-利用可檢查變數是否已宣告的 `isset` 函式

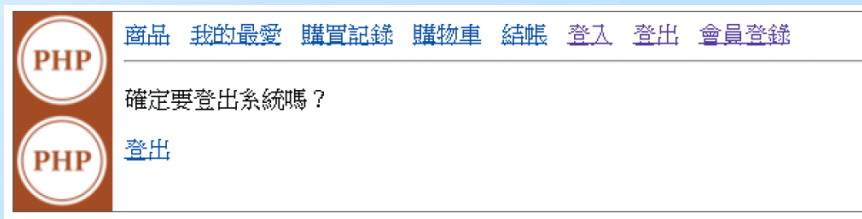
```
if (isset($_SESSION['customer'])) {
```

-當登入成功時，顯示歡迎訊息


```
echo '親愛的', $_SESSION['customer']['name'], '、歡迎光臨。';
```

● 製作登出功能

- 登出系統時，必須清除登入時所建立的 Session 資料
- 確認登出畫面



```

List  logout-input.php PHP
<?php require '../header.php';?>
<?php require 'menu.php';?>
<p>確定要登出系統嗎? </p>
<a href="logout-output.php">登出</a>
<?php require '../footer.php';?>
    
```

一登出處理的程式內容如下所示

```

List  logout-output.php  PHP
<?php require '../header.php';?>
<?php require 'menu.php';?>
<?php
session_start();
if (isset($_SESSION['customer'])) {
    unset($_SESSION['customer']);
    echo '登出成功。';
} else {
    echo '您原本就已登出。';
}
?>
<?php require '../footer.php';?>
    
```

▼ 登出成功

	商品 我的最愛 購買記錄 購物車 結帳 登入 登出 會員登錄
	登出成功。

▼ 原本就已登出時

	商品 我的最愛 購買記錄 購物車 結帳 登入 登出 會員登錄
	您原本就已登出。

● 清除 Session 資料

–登出系統的程式也需存取 Session，因此一開始必須先呼叫 `session_start` 函式

```
session_start();
```

–接著檢查使用者是否已登入系統

```
if (isset($_SESSION['customer'])) {
```

–若使用者已登入系統，則進行登出處理

```
unset($_SESSION['customer']);
```


7-3 登錄會員資料



● 顯示會員資料輸入畫面

— 首先，製作會員資料的輸入畫面，可以新增、修改使用者的基本資料。

A screenshot of a web form for member registration. The form has a dark red sidebar on the left with three circular 'PHP' logos. The main content area is white and contains a navigation menu at the top with links: 商品, 我的最愛, 購買記錄, 購物車, 結帳, 登入, 登出, 會員登錄. Below the menu are four input fields: 姓名, 地址, 登入ID, and 密碼. At the bottom left of the form is a 確定 (Confirm) button.

PHP	商品 我的最愛 購買記錄 購物車 結帳 登入 登出 會員登錄
PHP	姓名 <input type="text"/>
PHP	地址 <input type="text"/>
PHP	登入ID <input type="text"/>
	密碼 <input type="text"/>
	<input type="button" value="確定"/>



```

customer-input.php
<?php require '../header.php';?>
<?php require 'menu.php';?>
<?php
session_start();
$name=$address=$login=$password=''; 清空變數內的資料。
if (isset($_SESSION['customer'])) {
    $name=$_SESSION['customer']['name'];
    $address=$_SESSION['customer']['address'];
    $login=$_SESSION['customer']['login'];
    $password=$_SESSION['customer']['password'];
}
echo '<form action="customer-output.php" method="post">';
echo '<table>';
echo '<tr><td>姓名</td><td>';
echo '<input type="text" name="name" value="', $name, '">';
echo '</td></tr>';
echo '<tr><td>地址</td><td>';
echo '<input type="text" name="address" value="', $address, '">';
echo '</td></tr>';
echo '<tr><td>登入ID</td><td>';
echo '<input type="text" name="login" value="', $login, '">';
echo '</td></tr>';
echo '<tr><td>密碼</td><td>';
echo '<input type="password" name="password" value="', $password, '">';
echo '</td></tr>';
echo '</table>';
echo '<input type="submit" value="確定">';
echo '</form>';
?>
<?php require '../footer.php';?>
    
```

如果已經登入，從 **session** 取出資料，放入變數中。

顯示個人資料。

● 顯示登錄的資料

-因要對 **Session** 進行存取，因此須先呼叫 `session_start` 函式

```
session_start();
```

-宣告用來暫存姓名、地址、登入 ID、密碼的變數，並先代入空字串

```
$name=$address=$login=$password='';
```

-檢查 **Session** 內是否已存放了顧客資料

```
if (isset($_SESSION['customer'])) {
```

-若顧客資料已存在，則從 **Session** 中取出顧客資料

```
$name=$_SESSION['customer']['name'];
```

-變數會用來產生 `<input>` 標籤

```
echo '<input type="text" name="name" value="", $name, "">';
```

-實際產生出來的 `<input>` 標籤如下所示

```
<input type="text" name="name" value="熊木 和夫">
```

顧客資料的新增與修改處理

—參照下列程式，撰寫處理顧客資料新增與修改的程式

```

customer-output.php
<?php require '../header.php';?>
<?php require 'menu.php';?>
<?php
session_start();
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
'staff','password');
if (isset($_SESSION['customer'])) {
    $id=$_SESSION['customer']['id'];
    $sql=$pdo->prepare('select * from customer where id!=? and
login=?');
    $sql->execute([$id, $_REQUEST['login']]);
} else {
    $sql=$pdo->prepare('select * from customer where login=?');
    $sql->execute([$_REQUEST['login']]);
}
if (empty($sql->fetchAll())) {
    if (isset($_SESSION['customer'])) {
        $sql=$pdo->prepare('update customer set name=?, address=?,
'login=?, password=? where id=?');
        $sql->execute([
            $_REQUEST['name'], $_REQUEST['address'],
            $_REQUEST['login'], $_REQUEST['password'], $id]);
        $_SESSION['customer']=[
            'id'=>$id, 'name'=>$_REQUEST['name'],
            'address'=>$_REQUEST['address'], 'login'=>$_REQUEST['login'],
            'password'=>$_REQUEST['password']];
        echo '客戶資料修改完成。';
    } else {
        $sql=$pdo->prepare('insert into customer values(null,?,?,?,?)');
        $sql->execute([
            $_REQUEST['name'], $_REQUEST['address'],
            $_REQUEST['login'], $_REQUEST['password']]);
        echo '客戶資料新增完成。';
    }
} else {
    echo '登入 ID 已被使用，請重新設定。';
}
?>
<?php require '../footer.php';?>
    
```

已登入的情況：
檢查使用者名稱是否重複

未登入的情況：
檢查使用者名稱是否重複

已登入的情況：
修改使用者資料

未登入的情況：
新增使用者資料

找不到資料！

一輸入的顧客資料

資料欄	內容值
姓名	貓田 重藏
地址	靜岡縣靜岡市葵區追手町9-6
登入ID	nekota
密碼	CatField10

一顧客資料輸入

PHP

[商品](#)
[我的最愛](#)
[購買記錄](#)
[購物車](#)
[結帳](#)
[登入](#)
[登出](#)
[會員登錄](#)

姓名

地址

登入ID

密碼

一顧客資料登錄

PHP

[商品](#)
[我的最愛](#)
[購買記錄](#)
[購物車](#)
[結帳](#)
[登入](#)
[登出](#)
[會員登錄](#)

客戶資料新增完成。

一顧客資料更新

PHP

[商品](#)
[我的最愛](#)
[購買記錄](#)
[購物車](#)
[結帳](#)
[登入](#)
[登出](#)
[會員登錄](#)

客戶資料修改完成。



● 檢查登入 ID 是否重複

–指令中的 `id` 要指定為目前使用者的客戶編號，`login` 則指定為輸入的登入 ID

```
select * from customer where id!=? and login=?
```

–指令中的 `login` 即指定為輸入的登入 ID (帳號)

```
select * from customer where login=?
```

–若被指定為傳入參數的變數或運算式為空值，則 `empty` 函式會傳回 `TRUE` 值

```
if (empty($sql->fetchAll())) {
```



● 寫入顧客資料

-利用 `isset` 函式檢查 `Session` 資料是否存在，就可藉此判斷使用者的登入狀態

```
if (isset($_SESSION['customer'])) {
```

-若使用者已登入，則利用下行所示的 `SQL update` 敘述修改資料庫的資料

```
update customer set name=?, address=?, login=?, password=? where id=?
```

-利用 `execute` 方法執行 `SQL` 指令，別代入輸入的姓名、地址、登入 `ID`、密碼、客戶編號等資料

```
$sql->execute([  
    $_REQUEST['name'], $_REQUEST['address'],  
    $_REQUEST['login'], $_REQUEST['password'], $id]);
```




● 登錄顧客資料

–若使用者未登入系統，則利用下行所示的 SQL insert 敘述

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

–「？」部份同樣要分別代入姓名、地址、登入 ID、密碼等資料

```
$sql->execute([
    $_REQUEST['name'], $_REQUEST['address'],
    $_REQUEST['login'], $_REQUEST['password']]);
```

7-4 購物車功能



● 顯示商品清單

—顯示商品清單頁，並可從這裡進入個別商品的明細資料頁

執行 **product.php** 要

1. 建立會員資料
2. 登入

```
product.php
<?php require '../header.php';?>
<?php require 'menu.php';?>
<form action="product.php" method="post">
商品搜尋
<input type="text" name="keyword">
<input type="submit" value="搜尋">
</form>
<hr>
<?php
echo '<table>';
echo '<th>商品編號</th><th>商品名稱</th><th>價格</th>';
$pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
'staff','password');
if (isset($_REQUEST['keyword'])) {
    $sql=$pdo->prepare('select * from product where name like ?');
    $sql->execute(['%' . $_REQUEST['keyword'] . '%']);
} else {
    $sql=$pdo->prepare('select * from product');
    $sql->execute([]);
}
foreach ($sql->fetchAll() as $row) {
    $id=$row['id'];
    echo '<tr>';
    echo '<td>', $id, '</td>';
    echo '<td>';
    echo '<a href="detail.php?id=', $id, '">', $row['name'], '</a>';
    echo '</td>';
    echo '<td>', $row['price'], '</td>';
    echo '</tr>';
}
echo '</table>';
?>
<?php require '../footer.php';?>
```

商品一覽

商品 我的最愛 購買記錄 購物車 結帳 登入 登出 會員登錄

商品搜尋 搜尋

商品編號	商品名稱	價格
1	松果	700
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

有商品搜尋的資料

無商品搜尋的資料

點在商品名稱時，呼叫detail.php

● 顯示搜尋結果

-在進行資料顯示的處理時，若 REQUEST 參數中包含了搜尋關鍵字，利用 if 判斷式

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

● 顯示商品明細頁

—在商品清單中點選商品，就可進入商品明細頁

```

List detail.php PHP
<?php require '../header.php';?>
<?php require 'menu.php';?>
<?php
$db=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
    'staff', 'password');
$sql=$db->prepare('select * from product where id=?');
$sql->execute($_REQUEST['id']);
foreach ($sql->fetchAll() as $row) {
    echo '<p></p>';
    echo '<form action="cart-insert.php" method="post">';
    echo '<p>商品編號：', $row['id'], '</p>';
    echo '<p>商品名稱：', $row['name'], '</p>';
    echo '<p>價格：', $row['price'], '</p>';
    echo '<p>數量：<select name="count">';
    for ($i=1; $i<=10; $i++) {
        echo '<option value=', $i, '>', $i, '</option>';
    }
    echo '</select></p>';
    echo '<input type="hidden" name="id" value=', $row['id'], '>';
    echo '<input type="hidden" name="name" value=', $row['name'], '>';
    echo '<input type="hidden" name="price" value=', $row['price'], '>';
    echo '<p><input type="submit" value="放入購物車"></p>';
    echo '</form>';
    echo '<p><a href="favorite-insert.php?id=', $row['id'],
        '>">加入我的最愛</a></p>';
}
?>
<?php require '../footer.php';?>
    
```

form 範圍



● 取得商品資料

–範例是以下列 **select** 敘述取得指定商品編號所對應的商品資料

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

–利用 **execute** 方法（6-4 節 Step3）執行 SQL 指令

```
$sql->execute([$REQUEST['id']]);
```

–以 **fetchAll** 方法（6-4 節 Step3）取得 SQL 的執行結果，並利用 **foreach** 迴圈顯示商品資料

```
echo '<p>商品名稱：', $row['name'], '</p>';
```

–商品照片則是利用 **** 標籤顯示

```
echo '<p></p>';
```



● 加入購物車

–來解說用來將商品加入購物車的表單

```
echo '<form action="cart-insert.php" method="post">';
```

–表單中應有可用來選填訂購數量的下拉式選單

```
echo '<p>數量:<select name="count">';  
...  
echo '</select></p>';
```

–加入代表 1 到 10 的 `<option>` 標籤，讓使用者可指定訂購數量為 1 到 10

```
for ($i=1; $i<=10; $i++) {  
    echo '<option value="' . $i . '>' . $i . '</option>';  
}
```



—設定 for 迴圈開始時的變數 i 為 1，重複執行迴圈內的處理

```
<option value="1">1</option>
<option value="2">2</option>
...
```

—針對商品編號，使用像 `<input type="hidden" name="id" value="5">` 這樣的 `<input>` 標籤，傳送商品編號的值為 5

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



● 「加到我的最愛」連結

–設定 REQUEST 參數名 id 對應的值為商品編號

```
<a href="favorite-insert.php?id=5">加到我的最愛</a>
```

–連結可以用下行程式產生，其中 `<a>` 標籤用來產生連結，`<p>` 標籤則用來產生段落

```
echo '<p><a href="favorite-insert.php?id=', $row['id'],  
'">加到我的最愛</a></p>';
```


● 在購物車中加入商品

—在商品明細頁中，點按〔加入購物車〕按鈕後要執行的程式如下

```

List cart-insert.php

<?php require '../header.php';?>
<?php require 'menu.php';?>
<?php
session_start();
$id=$_REQUEST['id'];
if (!isset($_SESSION['product'])) {
    $_SESSION['product']=[];
}
$count=0;
if (isset($_SESSION['product'][$id])) {
    $count=$_SESSION['product'][$id]['count'];
}
$_SESSION['product'][$id]=[
    'name'=>$_REQUEST['name'],
    'price'=>$_REQUEST['price'],
    'count'=>$count+$_REQUEST['count']
];
echo '<p>商品放入購物車成功。</p>';
echo '<hr>';
require 'cart.php';
?>
<?php require '../footer.php';?>
    
```

▼ 加入購物車

商品 [我的最愛](#) [購買記錄](#) [購物車](#) [結帳](#) [登入](#) [登出](#) [會員登錄](#)

商品放入購物車成功。

商品編號	商品名稱	價格	數量	小計	
5	腰果	250	1	250	刪除
合計				250	

▼ 相同商品再次加入購物車

商品 [我的最愛](#) [購買記錄](#) [購物車](#) [結帳](#) [登入](#) [登出](#) [會員登錄](#)

商品放入購物車成功。

商品編號	商品名稱	價格	數量	小計	
5	腰果	250	3	750	刪除
合計				750	

▼ 不同商品加入購物車

商品 [我的最愛](#) [購買記錄](#) [購物車](#) [結帳](#) [登入](#) [登出](#) [會員登錄](#)

商品放入購物車成功。

商品編號	商品名稱	價格	數量	小計	
5	腰果	250	3	750	刪除
4	杏仁	220	2	440	刪除
合計				1190	

● Session 內保存的資料

–在變數 `$_SESSION` 所代表的 Session 中，以下列變數存放購物車內的商品資料

▼ 購物車內的商品資料

變數	保存的資料種類
<code>\$_SESSION['product']['商品編號']['name']</code>	商品名稱
<code>\$_SESSION['product']['商品編號']['price']</code>	價格
<code>\$_SESSION['product']['商品編號']['count']</code>	數量

▼ 購物車內的商品資料例①

變數	保存的資料值
<code>\$_SESSION['product']['5']['name']</code>	'腰果'
<code>\$_SESSION['product']['5']['price']</code>	250
<code>\$_SESSION['product']['5']['count']</code>	3

▼ 購物車內的商品資料例②

變數	保存的資料值
<code>\$_SESSION['product']['4']['name']</code>	'杏仁'
<code>\$_SESSION['product']['4']['price']</code>	220
<code>\$_SESSION['product']['4']['count']</code>	2



● 購物車初始化

—要檢查購物車變數是否已被定義，可利用 `isset` 函式

```
if (!isset($_SESSION['product'])) {
```

—當購物車變數未定義時，執行下列程式

```
$_SESSION['product']=[];
```

空陣列



● 取得購物車內已保存的數量

–若是要放入購物車的商品，在購物車內已經存在，則進行將數量合併計算的處理

```
$count=0;
```

–利用 `isset` 函式檢查要放入購物車的商品，在**購物車內是否已經存在**

```
if (isset($_SESSION['product'][$id])) {
```

–若購物車內已存在相同商品，則取得購物車內此商品的數量

```
$count=$_SESSION['product'][$id]['count'];
```



● 在購物車內新增商品

—依照前述的購物車結構，將商品名稱、價格、數量等資料放入購物車

```
$_SESSION['product'][$id]=[  
    'name'=>$_REQUEST['name'],  
    'price'=>$_REQUEST['price'],  
    'count'=>$count+$_REQUEST['count']  
];
```

在 **product** 的 **session** 存放某一商品 **id** 的
商品名稱、商品價格、購買數量

—最後顯示購物車內的商品清單

```
require 'cart.php';
```



● 顯示購物車內的商品清單

—參照下列程式撰寫，並將檔案儲存為 chapter7\cart.php

```
cart.php
<?php
if (!empty($_SESSION['product'])) {
    echo '<table>';
    echo '<th>商品編號</th><th>商品名稱</th>';
    echo '<th>價格</th><th>數量</th><th>小計</th>';
    $total=0;
    foreach ($_SESSION['product'] as $id=>$product) {
        echo '<tr>';
        echo '<td>', $id, '</td>';
        echo '<td><a href="detail.php?id=', $id, '">',
            $product['name'], '</a></td>';
        echo '<td>', $product['price'], '</td>';
        echo '<td>', $product['count'], '</td>';
        $subtotal=$product['price']*$product['count'];
        $total+=$subtotal;
        echo '<td>', $subtotal, '</td>';
        echo '<td><a href="cart-delete.php?id=', $id, '">刪除</a></td>';
        echo '</tr>';
    }
    echo '<tr><td>合計</td><td></td><td></td><td></td><td>', $total,
        '</td><td></td></tr>';
    echo '</table>';
} else {
    echo '購物車內無商品。';
}
?>
```

在 product 的 session
存放在 id 的資料指定給
product 變數。

購物車內要刪除商
品的檔案連結。

- 檢查購物車是否為空

–利用可檢查變數是否為空值的 `empty` 函式（6-5 節），以下列程式判斷

```
if (!empty($_SESSION['product'])) {
```

● 顯示商品清單

– 購物車內的商品，是以下列名稱保存在變數 `$_SESSION['product']` 裡

```
$_SESSION['product'][商品編號A]
$_SESSION['product'][商品編號B]
...
```

– 要顯示購物車內的商品清單，需使用 `foreach` 迴圈

```
foreach ($_SESSION['product'] as $id=>$product) {
```

– 購物車內的內容

變數	值
<code>\$_SESSION('product')(5)('name')</code>	'腰果'
<code>\$_SESSION('product')(5)('price')</code>	250
<code>\$_SESSION('product')(5)('count')</code>	3

–以 \$product 表示時

變數	值
\$product['name']	'腰果'
\$product['price']	250
\$product['count']	3

–要顯示商品編號時，可利用變數 \$id 撰寫程式如下

```
echo '<td>', $id, '</td>';
```

–程式執行後實際產生的內容如下

```
<td>5</td>
```

–要顯示價格時，則可如下所示利用變數 \$product 撰寫程式

```
echo '<td>', $product['price'], '</td>';
```

–程式執行後實際產生的內容如下

```
<td>250</td>
```



● 刪除購物車內的商品

– 在每項商品後面顯示下列連結，用來刪除購物車內的商品

```
<a href="cart-delete.php?id=商品編號">刪除</a>
```

– 舉例來說，當商品編號為 5 時，產生的連結如下

```
<a href="cart-delete.php?id=5">刪除</a>
```

– 這個連結要放在 HTML 表格的資料格中，所以會一併產生 `<td>` 標籤

```
echo '<td><a href="cart-delete.php?id=', $id, '">刪除</a></td>';
```



● 計算小計金額與合計金額

—將各項商品以「小計 = 價格 * 數量」逐項計算小計金額。「*」是乘法的算符，實際程式應撰寫如下

```
$subtotal=$product['price']*$product['count'];
```

—一開始先將合計金額設定為 0

```
$total=0;
```

—購物車內所有商品都會執行這行程式，就可藉此求出全部的合計金額。實際程式如下

```
$total+=$subtotal;
```

● 刪除購物車內的商品

— 參照下列程式，撰寫進行購物車內商品刪除處理的程式

```

List cart-delete.php PHP
<?php require '../header.php';?>
<?php require 'menu.php';?>
<?php
session_start();
unset($_SESSION['product'][$_REQUEST['id']]);
echo '所選商品已移出購物車。';
echo '<hr>';
require 'cart.php';
?>
<?php require '../footer.php';?>

```

刪除購物車內商品。

▼ 刪除購物車內的商品

PHP	商品 我的最愛 購買記錄 購物車 結帳 登入 登出 會員登錄																		
PHP	所選商品已移出購物車。																		
PHP	<table border="1"> <thead> <tr> <th>商品編號</th> <th>商品名稱</th> <th>價格</th> <th>數量</th> <th>小計</th> <th></th> </tr> </thead> <tbody> <tr> <td>5</td> <td>蘋果</td> <td>250</td> <td>3</td> <td>750</td> <td>刪除</td> </tr> <tr> <td colspan="4">合計</td> <td>750</td> <td></td> </tr> </tbody> </table>	商品編號	商品名稱	價格	數量	小計		5	蘋果	250	3	750	刪除	合計				750	
商品編號	商品名稱	價格	數量	小計															
5	蘋果	250	3	750	刪除														
合計				750															

▼ 刪除購物車內所有商品

PHP	商品 我的最愛 購買記錄 購物車 結帳 登入 登出 會員登錄
PHP	所選商品已移出購物車。
PHP	購物車內無商品。



● 刪除商品資料

—要刪除變數，必須使用 `unset` 函式

```
unset($_SESSION['product'][$_REQUEST['id']]);
```

7-5 將商品加入我的最愛



● 加入我的最愛

—其程式撰寫如下，檔案儲存為 chapter7\favorite-insert.php

```
favorite-insert.php
<?php require '../header.php';?>
<?php require 'menu.php';?>
<?php
session_start();
if (isset($_SESSION['customer'])) {
    $pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
        'staff', 'password');
    $sql=$pdo->prepare('insert into favorite values(?,?)');
    $sql->execute([$_SESSION['customer']['id'], $_
REQUEST['id']]);
    echo '商品加入我的最愛成功。';
    echo '<hr>';
    require 'favorite.php';
} else {
    echo '請先登入，才能將商品加入我的最愛。';
}
?>
<?php require '../footer.php';?>
```

將商品加入我的最愛

商品 我的最愛 購買記錄 購物車 結帳 登入 登出 會員登錄

商品加入我的最愛成功。

商品編號	商品名稱	價格	
8	夏威夷豆	600	刪除

未登入時

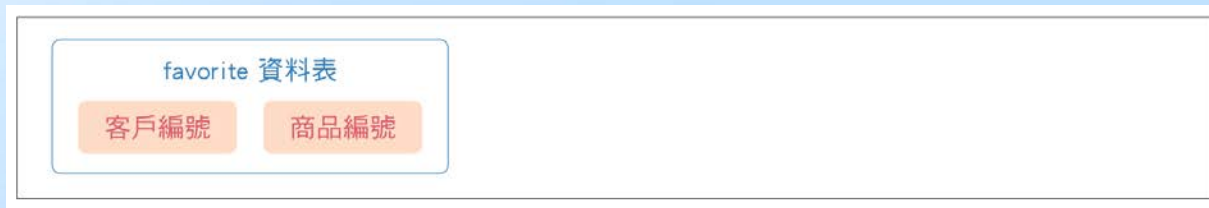
商品 我的最愛 購買記錄 購物車 結帳 登入 登出 會員登錄

請先登入，才能將商品加入我的最愛。

\$_SESSION['customer']['id']: 客戶編號
\$_REQUEST['id']: 商品編號
在 favorite 記錄客戶喜愛的商品編號

- 資料庫內的結構

-我的最愛中的商品，會以下列結構儲存在資料庫的 favorite 資料表中



-favorite 資料表的內容



一使用者追加商品時

favorite 資料表	
客戶編號	商品編號
1	8
1	2

一多個使用者同時登錄資料時

favorite 資料表	
客戶編號	商品編號
1	8
1	2
2	5
2	3
3	11



● 將商品加入我的最愛

-可使用 `isset` 函式檢查 `Session` 物件中 `customer` 是否已存在

```
if (isset($_SESSION['customer'])) {
```

-使用者已登入，則執行下列 `SQL` 指令，將商品登錄到我的最愛

```
insert into favorite values(?,?)
```

-其中「？」的部份應代入客戶編號與商品編號。實際的程式如下

```
$sql=$pdo->prepare('insert into favorite values(?,?)');  
$sql->execute([$_SESSION['customer']['id'], $_REQUEST['id']]);
```

● 製作共用程式顯示我的最愛清單

— 參照下述程式，撰寫用來顯示我的最愛清單的共用程式

```

favorite.php
<?php
if (isset($_SESSION['customer'])) {
    echo '<table>';
    echo '<th>商品編號</th><th>商品名稱</th><th>價格</th>';
    $pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
        'staff', 'password');
    $sql=$pdo->prepare(
        'select * from favorite, product '
        'where customer_id=? and product_id=product.id');
    $sql->execute([$_SESSION['customer']['id']]);
    foreach ($sql->fetchAll() as $row) {
        $id=$row['id'];
        echo '<tr>';
        echo '<td>', $id, '</td>';
        echo '<td><a href="detail.php?id='.$id.'">', $row['name'],
            '</a></td>';
        echo '<td>', $row['price'], '</td>';
        echo '<td><a href="favorite-delete.php?id=', $id,
            '">刪除</a></td>';
        echo '</tr>';
    }
    echo '</table>';
} else {
    echo '請先登入，才能顯示我的最愛。';
}
?>

```

合併查詢：
其目的是要在 **favorite** 資料表中只有 **id** 的資訊，透過 **product** 資料表取得商品名稱、商品價格的資料。

- 取得並顯示我的最愛

- 此時可利用 `isset` 函式，檢查變數 `$_SESSION['customer']` 是否已定義

```
if (isset($_SESSION['customer'])) {
```

- favorite 資料表內的狀況

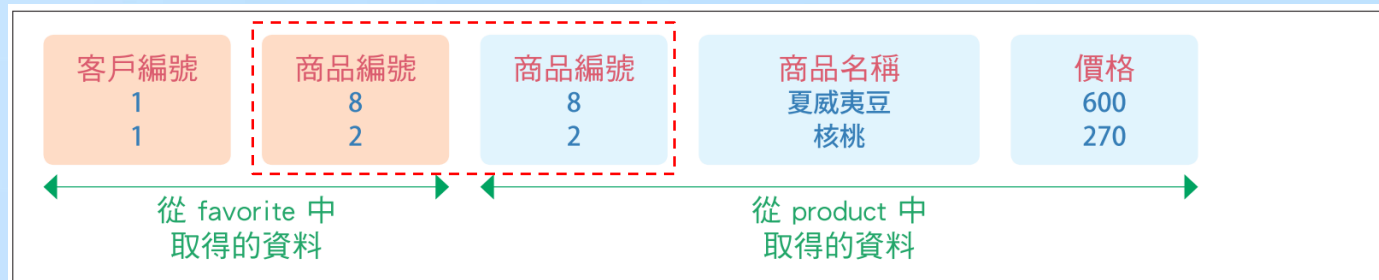
favorite 資料表	
客戶編號	商品編號
1	8
1	2
2	5
2	3
3	11

-在 SQL 指令中，以逗號「,」分隔 favorite 與 product 資料表

```
select * from favorite, product where customer_id=?
```

-因為要找出 favorite 資料表中的商品編號（product_id 資料欄），與 product 資料表中的商品編號（id 資料欄）一致的資料，因此要在 where 子句追加條件如下

where customer_id=? and product_id=id



customer_id, product_id 是 favorite 的欄位名稱。id 是 product 的欄位名稱。



-SQL 指令較長，可利用將字串連結的算符「.」，將指令切割成多行

```
$sql=$pdo->prepare(
    'select * from favorite, product '.
    'where customer_id=? and product_id=id');
$sql->execute([$SESSION['customer']['id']]);
```

-利用 **foreach** 迴圈處理取得的資料列。將資料列逐筆指定給變數 **\$row**

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

合併查詢範例



favorite

customer_id	product_id
1	1
2	2

product

id	name	price
1	松果	700
2	核桃	210
3	杏仁	220

select * from favorite, product

favorite		product		
customer_id	product_id	id	name	price
1	1	1	松果	700
2	2	1	松果	700
1	1	2	核桃	210
2	2	2	核桃	210
1	1	3	杏仁	220
2	2	3	杏仁	220

select * from favorite, product where product_id = id

favorite		product		
customer_id	product_id	id	name	price
1	1	1	松果	700
2	2	1	松果	700
1	1	2	核桃	210
2	2	2	核桃	210
1	1	3	杏仁	220
2	2	3	杏仁	220

Annotations: A green arrow points to the first row (1, 1, 1, 松果, 700). Yellow arrows point to the first and second columns of the first row. Dashed yellow lines connect the product_id column of the first row to the id column of the first row, and the product_id column of the second row to the id column of the second row.

select * from favorite, product where product_id = id and customer_id=1



—利用下行程式取得商品編號，並將它代入變數 \$id

```
$id=$row['id'];
```

—在顯示出商品編號、商品名稱、價格之後，還需顯示可從我的最愛中刪除商品的〔刪除〕連結

```
<a href="favorite-delete.php?id=', $id, '">刪除</a>
```

—〔刪除〕連結是用下列程式產生

```
echo '<td><a href="favorite-delete.php?id=', $id, '">刪除</a></td>';
```


● 刪除我的最愛

— 參照 列程式， 撰寫進行我的最愛刪除處理的程式

```

favorite-delete.php
<?php require '../header.php';?>
<?php require 'menu.php';?>
<?php
session_start();
if (isset($_SESSION['customer'])) {
    $pdo=new PDO('mysql:host=localhost;dbname=shop;charset=utf8',
        'staff', 'password');
    $sql=$pdo->prepare(
        'delete from favorite where customer_id=? and product_
id=?');
    $sql->execute([$_SESSION['customer']['id'], $_
REQUEST['id']]);
    echo '所選商品已從我的最愛移除。';
    echo '<hr>';
} else {
    echo '請先登入，才能從我的最愛移除商品。';
}
require 'favorite.php';
?>
<?php require '../footer.php';?>
    
```

▼ 刪除我的最愛中的商品

[商品](#)
[我的最愛](#)
[購買記錄](#)
[購物車](#)
[結帳](#)
[登入](#)
[登出](#)
[會員登錄](#)

所選商品已從我的最愛移除。

商品編號	商品名稱	價格	
2	核桃	270	刪除

● 資料庫的存取方式

– 要從我的最愛中刪除商品，必須將 favorite 資料表中指定客戶編號與商品編號的資料列刪除

```
delete from favorite where customer_id=? and product_id=?
```

– 以下列程式執行這個 SQL 指令

```
$sql=$pdo->prepare(  
    'delete from favorite where customer_id=? and product_id=?');  
$sql->execute($_SESSION['customer']['id'], $_REQUEST['id']);
```

customer 的 id

product 的 id

小結



- 本章以購物網站為題材，介紹一些實用的程式範例。本章所提供的範例程式既可以完全套用到您要開發的網站，也可以只針對登入或購物車等，選擇您所需的功能套用。希望能成為幫助各位學習與開發的材料。