## 深入淺出 PHP+MYSQL 複習重點整理

### 第一章:充滿生機-靜態頁面賦予生命

1. PHP:服務器端腳本語言，允許你在頁面發布到客戶瀏覽器之前處理服務器上的WEB頁面內容。
2. PHP腳本:包含PHP代碼的一個文本文件，在WEB服務器上完成任務。
3. MySQL:允許將數據存儲在數據庫和數據庫表中，使用SQL語言插入和獲取信息。
4. SQL:一種查詢語言，用于與MySQL之類的數據庫應用交互。
5. <?php ?>:PHP腳本中的所有PHP代碼必須包在這組標記之間。
6. Mail():mail($to, $subject, $msg, ’From: ’. $email); 目標地址，EMAIL主題，EMAIL內文，來源地址。
7. 變量:PHP中，所有的變量必須以$符號開頭，如$name。
8. $\_POST:一個存儲表單數據的特殊變量。其結構是數組，例：$email = $\_POST[‘email’];
9. Echo:字串輸出，類似c#的print()。例：echo ’happy day.’;
10. 轉義字符：\n(換行);

### 第二章:如何連接在一起-連接MYSQL

1. CREATE DATABASE 資料庫名。
2. USE 資料庫名。
3. CREATE TABLE 資料表名(欄位名 資料類型)。例：CREATE TABLE first\_table( first\_name varchar(30),last\_name varchar(30));
4. INSERT INTO 資料表名 (欄位1,欄位2,……) VALUES (欄位1的值,欄位2的值,……);
5. SELECT 欄位名 FROM 資料表名 (WHERE 條件子句)
6. $dbc = mysqli\_connect(‘localhost’,’root’,’abcd1234’,’aliendatabase’) or die(‘error querying database.’);域名，帳號，密碼，資料庫名。Die終止程式並輸出錯誤訊息。
7. $result = mysqli\_query($dbc,$query) or die(error querying database.’); $query為SQL查詢語句。
8. Mysqli\_close($dbc);
9. Mysqli\_select\_db($dbc,’指定資料庫名’);

### 第三章:創建與填充數據庫

1. SQL數據類型參考表：

|  |  |
| --- | --- |
| int | 整數 |
| Char(1) | Y OR N |
| DATE | 11/17/1984 |
| TIME | 用餐時間 |
| DEC(4,2) | 48.99 |
| VARCHAR(60) | 字串 |
| CHAR(2) | HR、SA、SD之類的 |
| DATETIME | 日期加時間 |

1. DESCRIBE 資料表名。展示資料表的結構
2. DROP TABLE 資料表名。刪除整個資料表
3. $row = mysqli\_fetch\_array($result);每執行一次取得查詢結果的一行。所以當要循環處理時可以這樣做。While($row = mysqli\_fetch\_array($result)){echo $row[‘first\_name’]……}
4. DELETE FROM 資料表名 (WHERE 條件子句)。刪除資料表資料。

### 第四章:現實的實際應用

1. Isset(變量名)。測試一個變量是否存在(是否已經賦值)。
2. Empty(變量名)。確定一個變量是否包含一個空值。(含0，’’,false,null)。
3. 利用取反!時，將變量用()包起來似乎是個好習慣。
4. 如果遇到重復代碼，去思考如何用一個標示來判斷。如：$output\_form=false;
5. $\_SERVER[‘PHP\_SELF’],存儲著當前腳本的名。所以可以這樣用：<form action=”<?php echo $\_SERVER[‘PHP\_SELF’]; ?>” method=”post”> 直接引用自身，這樣腳本更好維護。
6. ALTER TABLE 表單名 ADD 欄位名 INT NOT NULL AUTO\_INCREMENT FIRST, ADD PRIMARY KEY(欄位名)。在表單中新增欄位，此欄位不能有空值，每新增一行，數值會自動增加，將此欄位放在第一列，設定此欄位為主鍵。
7. 主鍵的5大原則：
   1. 主鍵中的數據不能重復。
   2. 主鍵必須有一個值。
   3. 插入新行時必須設置主鍵。
   4. 主鍵必須盡可能高效。
   5. 主鍵值不能改變。
8. Foreach (母集合 AS 子元素){}; 例：foreach($\_POST[‘todelete’] as $delete\_id){};
9. <input type=”checkbox” value=”值” name=”todelete[]” /> 可直接將值存在陣列中，Submit時會傳出陣列。

### 第五章:使用儲存在文件中的數據

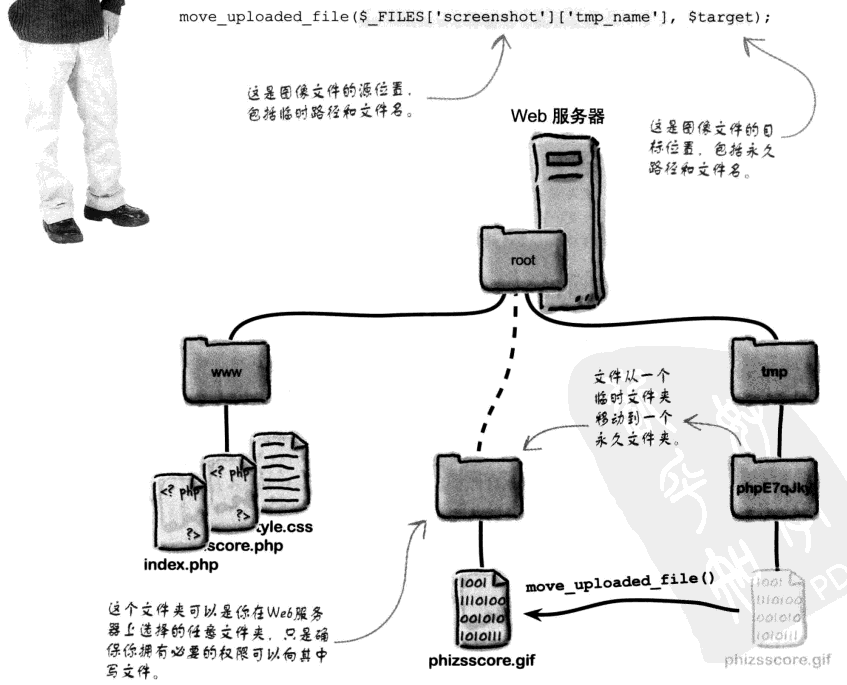
1. 修改資料庫表格欄位資料的使用方式：ALTER TABLE

* ADD COLUMN：為數據庫表增加一個新列，只需在ADD COLUMN後面指定列名及其類型。
  + 例：ALTER TABLE guitarwars ADD COLUMN age TINYINT
* DROP COLUMN：從一個數據庫表刪除一列(以及其中存儲的所有數據)，只需在DROP COLUMN後面指定列名。
  + 例：ALTER TABLE guitarwars DROP COLUMN age
* CHANGE COLUMN：修改一列的列名和數據類型，只需在CHANGE COLUMN後面指定原列名、新列名以及新列的數據類型。
  + 例：ALTER TABLE guitarwas CHANGE COLUMN score high\_score INT
* MODIFY COLUMN：修改一個數據庫表中某一列的數據瞺型或位置。要修改一列的位置，需要指定列名及其具體位置，可指定FIRST或是相對位置 AFTER 和另一個原有的列。
  + 例：ALTER TABLE guitarwars MODIFY COLUMN date DATETIME AFTER age

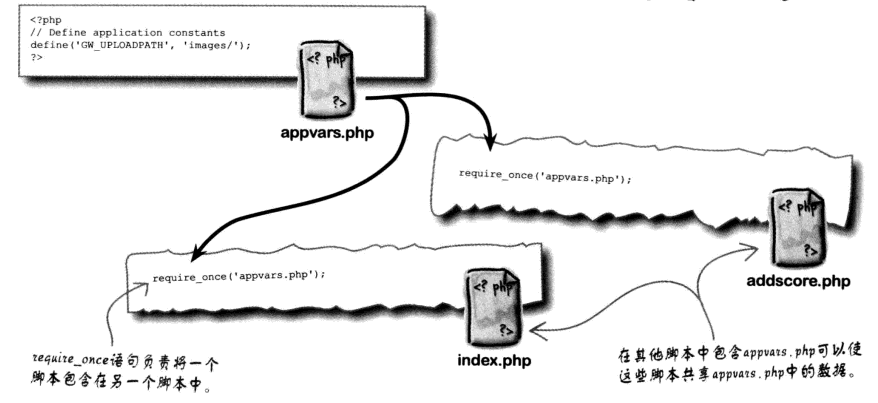
1. 要使HTML表單支持上傳功能，在表單必須注意兩個部份：
   * <form enctype=”multipart/form-data”……
   * <input type=”file”……
2. 可在表單中，增加一個隱藏的欄位來限制上傳檔案的大小：
   * <input type=”hidden”name=”MAX\_FILE\_SIZE”value=”32768”/>

這邊代表的是限制32KB(32768字節)

1. NOW(),MYSQL NOW()函數用來插入當前日期/時間。
   * 例：INSERT INTO guitarwars VALUES (0,NOW(),’$name’,’$score’)
2. PHP用來做檔案上傳的超級變數$\_FILES用法：
   * $\_FILES[‘screenshot’][‘name’]：得到上傳文件的文件名。
   * $\_FILES[‘screenshot’][‘size’]：得到上傳文件的大小(字節數)。
   * $\_FILES[‘screenshot’][‘type’]：得到上傳文件的MIME類型。
   * $\_FILES[‘screenshot’][‘tmp\_name’]：得到上傳文件的在服務器上的暫存位置。
   * $\_FILES[‘screenshot’][‘error’]：得到上傳的錯誤碼。0表示成功，其它值則是失敗。
3. 在WEB面面上放置一個圖像只需要該圖像文件的引用。
   * <img src=”phizsscore.jpg”alt=”score image”/>
4. is\_file(檔案位置/名稱)用來檢查服務器中，該檔案位置是否有該名稱的檔案。
5. filesize(檔案位置/名稱)用來取得該檔案的檔案大小，可用來檢查該檔案是不是一個空文件。
6. move\_uploaded\_file(來源位置,目標位置)，用來將檔案從暫存資料夾移到我們使定的資料夾。
   * 例：move\_uploaded\_file($\_FILES[‘screenshot’][‘tmp\_name’],’imadges/’);



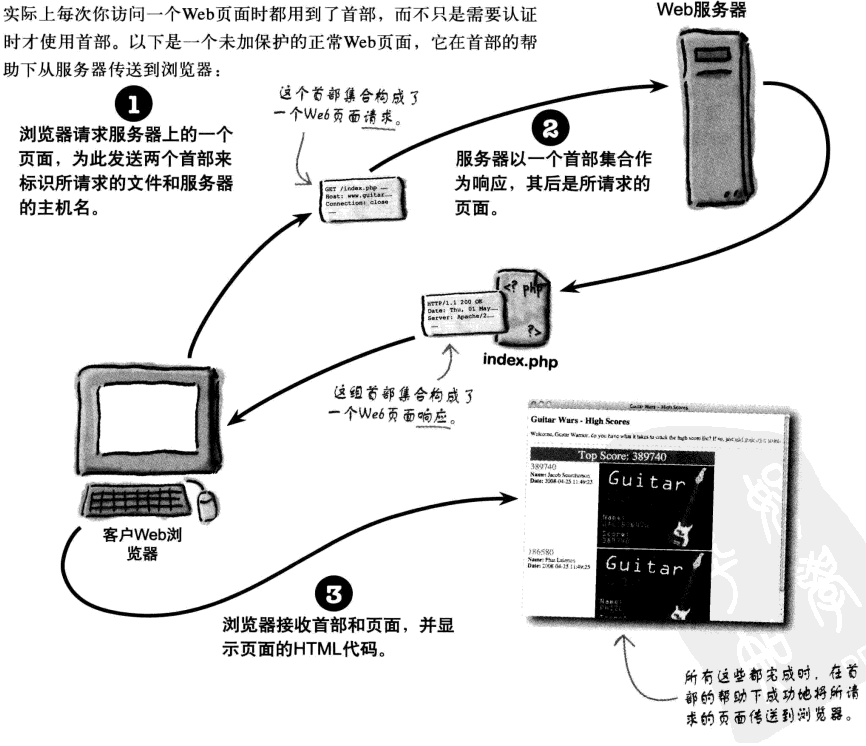
1. Php 可以透過修改php.ini來改變上傳的暫存位置：upload\_tmp\_dir。
2. 如果發現上傳功能不正常，很有可能是上傳的文件夾權限設定沒有開放，可以將權限設為chmod 777 images。
3. 為方便做程式的維護，可以將上傳文件夾的位置設定成常量來與上傳檔案名做組合。例：$target=GW\_UPLOADPATH.$screenshot;
4. PHP中定義常量的語法：define(‘GW\_UPLOADPATH’,’images/’);
5. 為了避免上傳的檔案名和其它使用者上傳的重覆，我們可以在檔名後加上服務器的時間來避免檔名重覆。
   * 例：$target=GW\_UPLOADPATH.time().$screenshot;
   * Time()將取得當前伺服器的時間。
6. 為了更好的維護程式碼，我們可以將常常重用的常量定義在一個固定的腳本中，讓其它腳本包含。
   * 例：require\_once(‘appvars.php’);



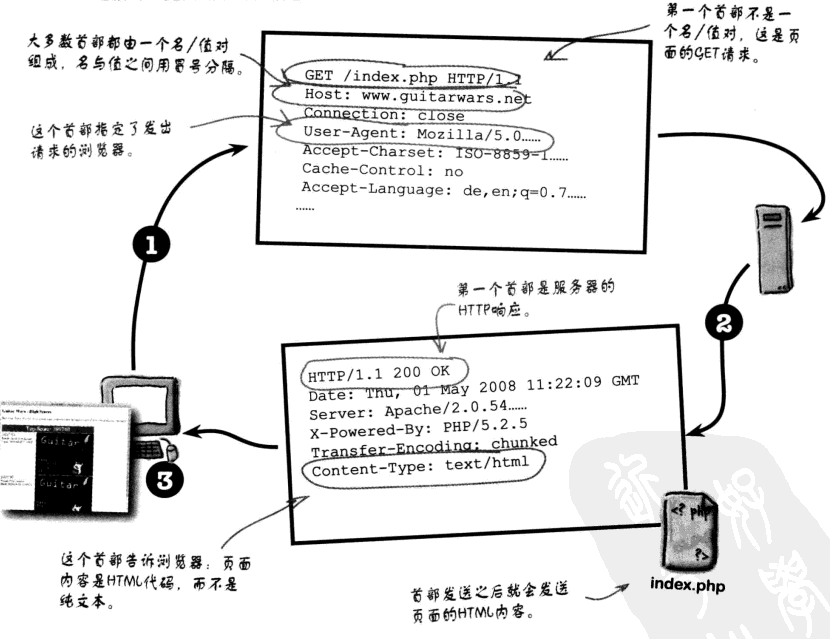
1. include()函數和require\_once()功能類似，二者的區別在於，如果未找到包含文件，則require\_once()會產生一個錯誤，而include不會顯示任何錯誤。
2. ORDER BY 用來做sql查詢結果的排序，其中asc代表升序，desc代表降序。
3. 用於表示web圖像的MIME類型：
   * GIF：image/gif
   * JPEG：image/jpeg or image/pjpeg
   * PNG：image/png
4. unlink可從服務器中刪除一個文件。例：unlink($\_FILES[‘screenshot’][‘tmp\_name’])
5. @可以防止顯示錯誤報告。例：@unlink();
6. $\_GET：透過GET的方式取得的資料都存在$\_GET的陣列中，其使用方式與$\_POST類似。
7. GET和POST的差別：
   * POST：用於向服務器發送數據，從而以某種方式導致服務器上狀態的改變，如在數據庫中插入數據。數據還可以在響應中返回。不同於GET，POST請求只能通過WEB表單的動作完成。另外與GET不同，POST請求中發送的數據是隱藏不可見的。
   * GET：一般用於數據獲取，而不會使服務器有任何改變。對於少量的數據，GET非常有用，可以直接在URL中向服務器發送數據。與POST不同，GET主要適用於發送少量的數據。(註：代表通過URL接收數據的腳本不能完成任何INSERT、DELETE、FROM或其它更改數據庫的操作。)
8. LIMIT：在SQL查詢中可用於行數的限制。
   * 例：DELETE FROM guitarwars WHERE ID=’1’LIMIT 1這樣一次只會刪除一行資料。
   * 例：DELETE FROM guitarwars LIMIT 10 ,5 這樣會刪除11到15行。這邊的第一個參數是跳過多少行，第二個參數是返回多少行。
9. $\_FILES[‘’][‘error’]錯誤訊息一覽：
   * UPLOAD\_ERR\_OK：其值為0，沒有錯誤發生，文件上傳成功。
   * UPLOAD\_ERR\_INI\_SIZE：其值為1，上傳的文件超過了php.ini中upload\_max\_filesize選項限制的值。
   * UPLOAD\_ERR\_FORM\_SIZE：其值為2，上傳文件的大小超過了html表單中MAX\_FILE\_SIZE選項指定的值。
   * UPLOAD\_ERR\_PARTIAL：其值為3，文件只有部份被上傳。
   * UPLOAD\_ERR\_NO\_FILE：其值為4，沒有文件被上傳。
   * UPLOAD\_ERR\_NO\_TMP\_DIR：其值為6，找不到臨時文件夾。
   * UPLOAD\_ERR\_CANT\_WRITE：其值為7，文件寫入失敗。

### 第六章:保證應用安全

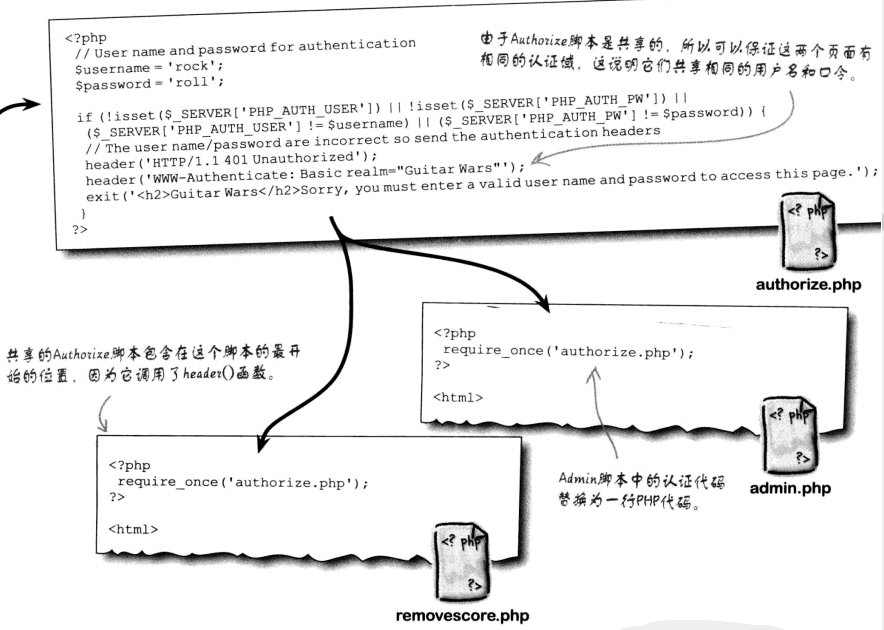
1. http認證的工作原理：用戶嘗試訪問一個通過認證保護的頁面時，如我們的admin頁面，會為用戶提供一個窗口，要求輸入一個用戶名和口令。



1. php中透過$\_SERVER全局變量可以取得從認證窗口得到的用戶名和口令。
   * $\_SERVER[‘PHP\_AUTH\_PW’];
   * $\_SERVER[‘PHP\_AUTH\_USER’];
2. Http首部剖析：首部准確地控制著如何在WEB瀏覽器和WEB服務器之間來回傳遞信息，以及傳遞何種信息。每個首部通常包含標識著一段信息的一個名/值對，如一個WEB頁面(HTML)的內容類型。首先一組首部會作為WEB頁面請求的一部份發送到服務器，然後返回另一組首部作為響應的一部份發回給瀏覽器。下面將更仔細地查看各組首部，從而明確客戶與服務器相互通信時究竟發送了哪些信息。



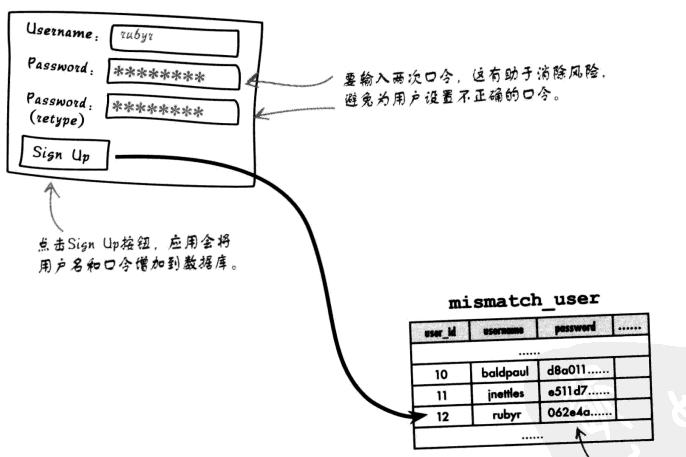
1. Header()函數可以從服務器向瀏覽器發送首部。
   * 例：header(‘Content-Type:text/html’);
2. 發起HTTP認證需要兩個首部：
   * HTTP/1.1 401 Unauthorized：這個首部讓瀏覽器知道用戶沒有得到查看頁面的授權。
   * WWW\_Authenticate:Basic realm=”Guitar Wars”：”Basic realm”(基本域)是用于唯一標識這個特定認證的一個短語，它會出現在認證窗口。
3. 定義用戶退出認證的顯示訊息，使用exit()函數。
   * 例：exit(‘Sorry,you must enter a valid user name and password to access this page.’);
4. 其它補充的首部用法：
   * Header(‘Location:http://www.quitarwars.net/about.php’);重定向
   * Header(‘Refresh: 5;url= <http://www.quitarwars.net/about.php>’);五秒後重定向
   * Header(‘Content-Type:text/plain’);內容作為純文本傳送到瀏覽器。
5. 基本域的實際作用：它定義了一個特定用戶名和口令所保護的安全區。一旦成功的輸入了對應一個給定域的用戶名和口令，瀏覽器會記住這個用戶名和口令，而對同一個域中後續的認證首部不會再顯示認證窗口。換句話說，基本域使得瀏覽器可以記住你已滿足一組給定頁面的安全需求。
6. 當有多個頁面需要用同一個HTTP認證時，應使用包含腳本的作法會更有效率。



1. MYSQL數據類型TINYINT(1)其功能等同BOOL。
2. UPDATE 資料表名 SET 修改欄位資料用來修改已存在的資料表資料。
   * 例：UPDATE guitarwars SET approved=1 WHERE id=’$id’;
3. MYSQL支援枚舉類型，所以也可以宣告成以下的方式：
   * ALTER TABLE guitarwars ADD COLUMN approved ENUM(‘yes’,’no’)
4. SQL注入攻擊：使用表單數據作為一種途徑來修改查詢的基本操作。
   * 例：在表單輸入1000000’,’ETHELSSCORE2.GIF’,1) –
   * 透過這種方式，在本行兩個橫線後面的sql語句會被註釋掉。
5. 加強表單的安全性：
   * Trim()函數可以很容易地去除前導或末尾的空格。例：$name=trim($\_POST[‘name’]);
   * Mysqli\_real\_escape\_string()函數會將有危險的字符進行轉義，使它不能有意地影響查詢的執行。例：$name=mysqli\_real\_escape\_string($dbc,trim($\_POST[‘name’]));
6. 為SQL欄位設定預設資料，只要加上DEFAULT即可。
   * 例：ALTER TABLE guitarwars MODIFY COLUMN approved TINYINT(1) DEFAULT 0;
7. is\_numeric(變數名)可以用來檢查該變數是否為數字。

### 第七之一章:構建個性化web應用-還記得我嗎？(COOKIE &SESSION)

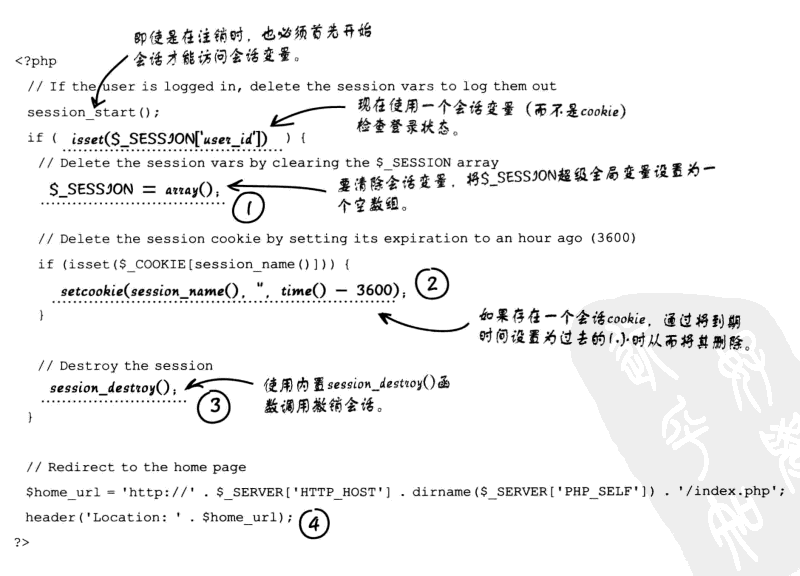
1. SHA()加密口令：MYSQL提供了一個名為SHA()的函數，可以對文本串進行加密算法，其結果是一個加密串，長度固定為40個16進制字符。
   * 例：INSERT INTO mismatch\_user(username,password,join\_date) values(‘jnettles’,SHA(‘tatlover’),NOW());
2. SHA()只提供單向加密，無法還原。
3. 針對SHA()加密來判斷登入密碼時，可以這樣做：
   * SELECT \* FROM mismatch user WHERE password = SHA(‘tatlover’)
4. 由於SHA()函數最後產生的字符串有40字符長，所以在宣告資料欄位時必須注意將欄位宣告為VARCHAR(40)。
5. 其它的加密算法與PHP相應用法：
   * MYSQL：SHA() PHP：sha1() ..一般認為SHA算法比MD5算法更安全一些。
   * MYSQL：MD5() PHP：md5()
6. 由於用戶註冊時，通常我們會將密碼欄位用星號屏蔽，所以多做一個密碼確認欄位是比較好的方式。



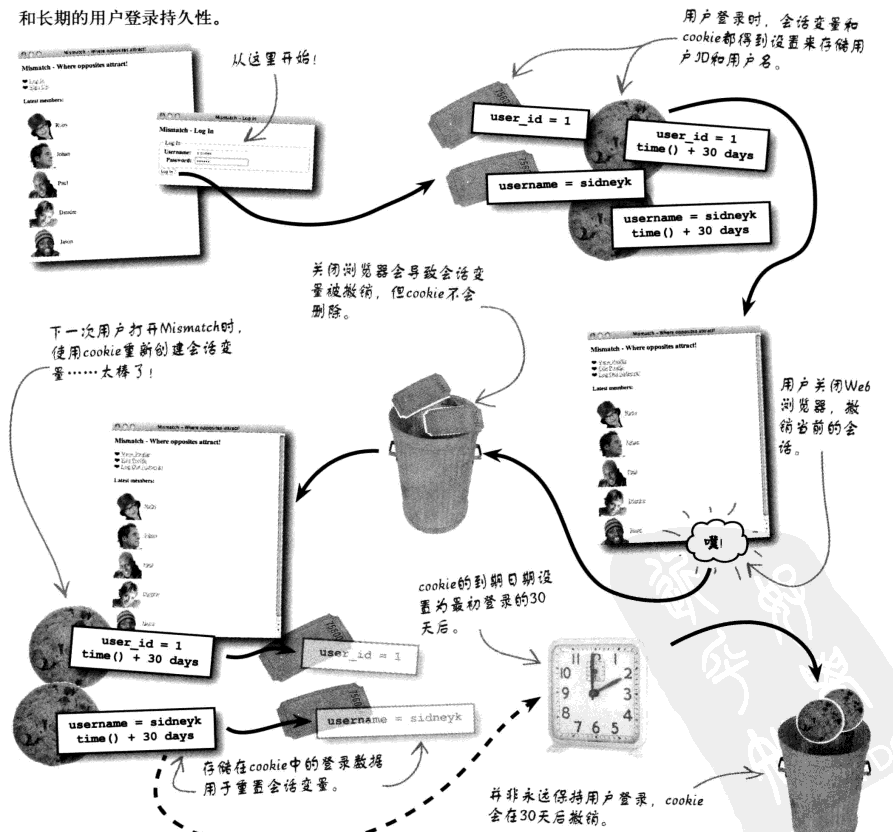
1. 如果在一個單引號字符串中要使用’符號要記得用\’進行轉義。
2. Mysqli\_num\_rows(變數)，用來取得SQL查詢結果的資料筆數。
   * 例：
   * $query=”SELECT \* FROM mismatch\_user WHERE username=’$username’”;
   * $data=mysqli\_query($dbc,$query);
   * If(mysqli\_num\_rows($data==0){…}
3. Exit()：在PHP中用來直接退出腳本。
4. Cookie與http認證的差異：

|  |  |  |
| --- | --- | --- |
|  | HTTP認證 | cookie |
| 資料存在時間 | 在瀏覽器未被關閉前，或手動控制瀏覽器將其清除時會永久存在。 | 可以用腳本控制其存在時間。 |
| 可否用腳本控制刪除 | 不可 | 可以 |
| 在PHP中的設置方式 | 使用HTTP認證時，會自動儲存$\_SERVER[‘PHP\_AUTH\_USER’]  $\_SERVER[‘PHP\_AUTH\_PW’] | Setcookie(‘username’,’sidneyk’,time()+(60\*60\*8)’)  參數1：KEY  參數2：VALUE  參數3：消毀日期，單位為秒數(可選) |
| 在PHP中的取得方式 | $\_SERVER[‘PHP\_AUTH\_USER’]  $\_SERVER[‘PHP\_AUTH\_PW’] | $\_COOKIE[‘username’] |
| PHP中消毀方式 | 只能將瀏覽器關閉 | Setcookie(‘username’,’sidneyk’,time()-3600’)  要刪除一個cookie，只需要將到期日期設置為過去的一個時間。 |

1. 在php腳本裡取得網頁路徑的方式：
   * $home\_url=’http://.$\_SERVER[‘HTTP\_HOST’].dirname($\_SERVER[‘PHP\_SELF’]).’/index.php’;
   * $\_SERVER[‘HTTP\_HOST’]取得網域路徑
   * dirname($\_SERVER[‘PHP\_SELF’])取得目錄路徑。
   * \_\_FILE\_\_可取得文件的絕對路徑本含文件名。
   * dirname(\_\_FILE\_\_\_) 函数返回的是[脚本](http://www.jbxue.com/jb/)所在路径
2. 有些用戶可能會設定瀏覽器限制cookie的使用，這時我們就要使用會話SESSION來將資料儲存在服務器上。
3. SESSION的生命和時間，會話之所以稱為會話有一個原因，它們有非常明確的開始和結束。與一個會話關聯的數據會隨該會話的生命期生存和毀滅，對此可以通過php代碼控制。只有一種情況下無法控制會話的生命期，就是用戶關閉瀏覽器時，這會導致會話結束，並使會話變量被銷毀。
   * Session\_start();：開始一個會話，此函數不會設置任何數據，它的工作只是建立會話並開始運行，此時會自動建立一個SESSION ID, WEB瀏覽器會使用這個ID來將會話與多個頁面關聯。
   * $\_SESSION[‘username’]=’sidneyk’：儲存數據。
   * Echo $\_SESSION[‘username’]：使用數據。
   * Session\_destroy();：銷毀一個會話。注意，此時只是關閉會話，會話變量實際上並沒有銷毀。
4. 註銷會話的流程：
   * Session\_start();：開啟一個會話。
   * If(isset($\_SESSION[‘user\_id’]){…}：檢查是否有使用者登入。
   * $\_SESSION=array();：清空會話變量。
   * If(isset($\_COOKIE[session\_name()])){}：檢查是否存在會話COOKIE。
   * Setcookie(session\_name(),’’,time()-3600)：刪除會話COOKIE。
   * Session\_destroy()：撤銷會話。
   * Header(‘Location：’.$home\_url)：重定向回首頁。

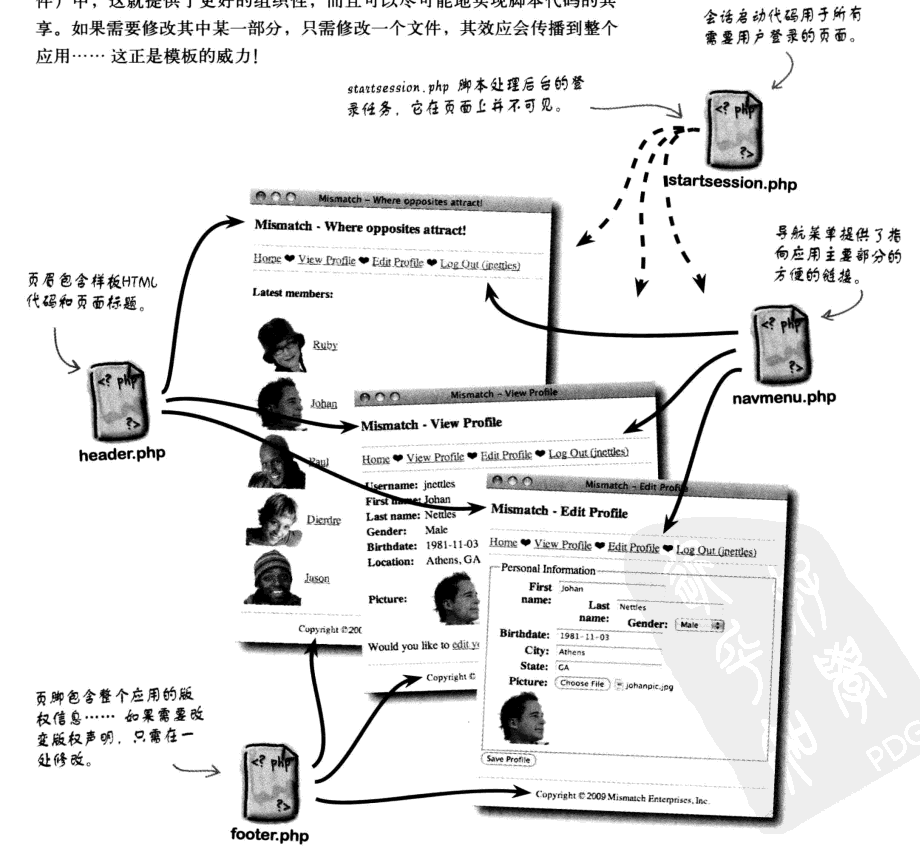
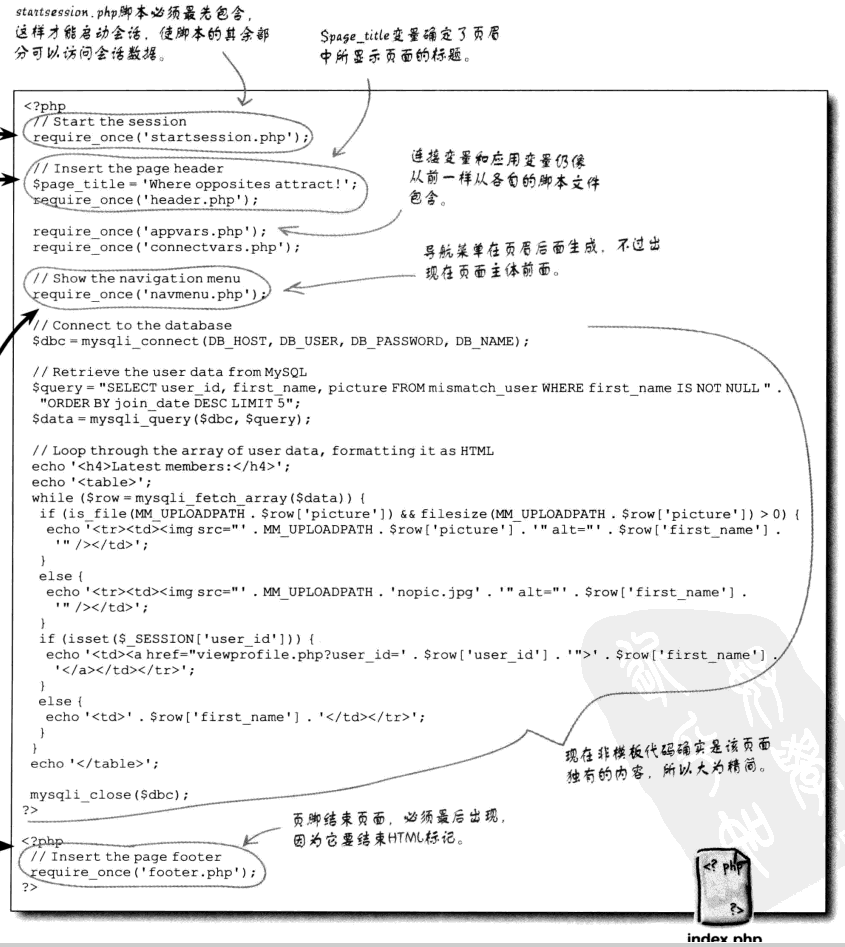


1. COOKIE被禁用時，SESSION將如何響應的具體細節，由php.ini配置文件通過以下三個屬性來設置：
   * Session.use\_cookies
   * Session.use\_only\_cookies
   * Session.use\_trans\_sid 🡨在禁用COOKIE的情況下，要讓會話正常工作，就要將這個屬性設置為true。
   * 如果無法更改伺服器的php.ini設定檔，那就必需手動向會話面頁傳遞會話ID。例：<a href=”viewprofile.php?<?php echo SID; ?>”>view your profile</a>
2. SID：SID超級全局變量包含了會話ID，它可以通過URL傳遞，使頁面能夠知道會話。
3. SESSION+COOKIE=更優秀的登錄持久性。



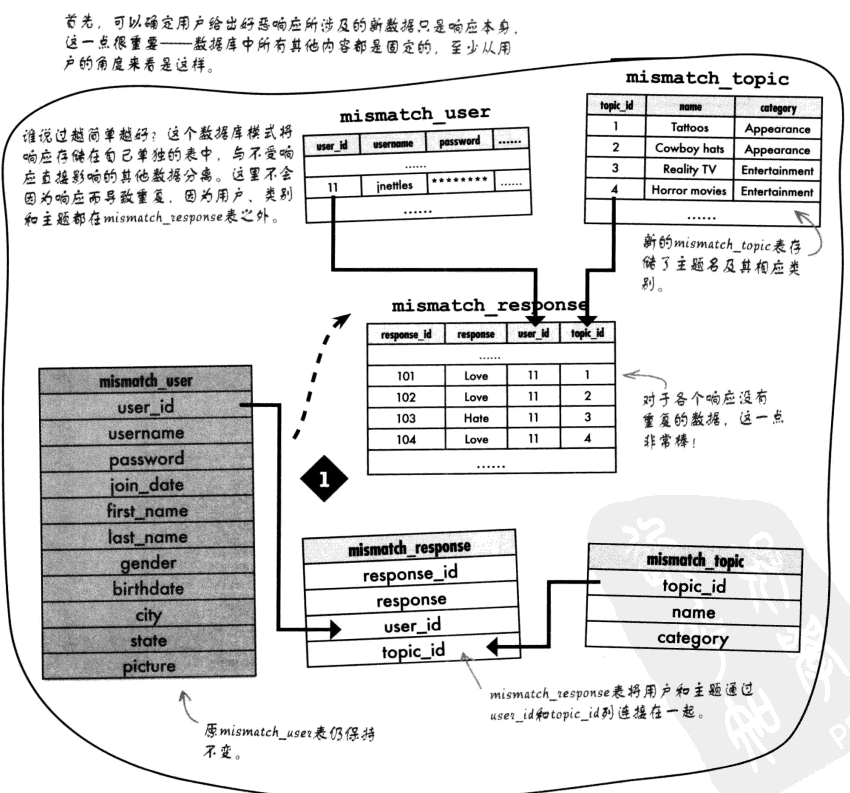
### 第七之二章:消除重復代碼-分享就是關愛

1. 將網站重新組織並模版化，雖然會很費功夫，但絕對是值得的，將來如果需要更改代碼，就只要更改一個文件就可以傳播到整個應用。

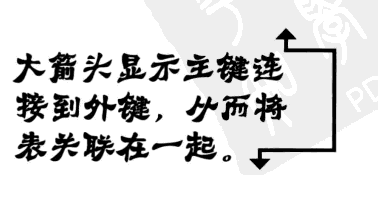
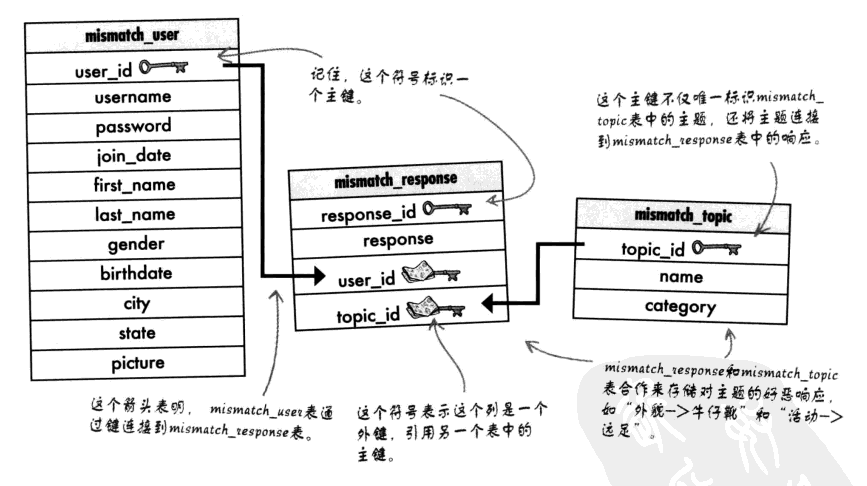


### 第八章:控制你的數據，世界在你手中-收獲數據

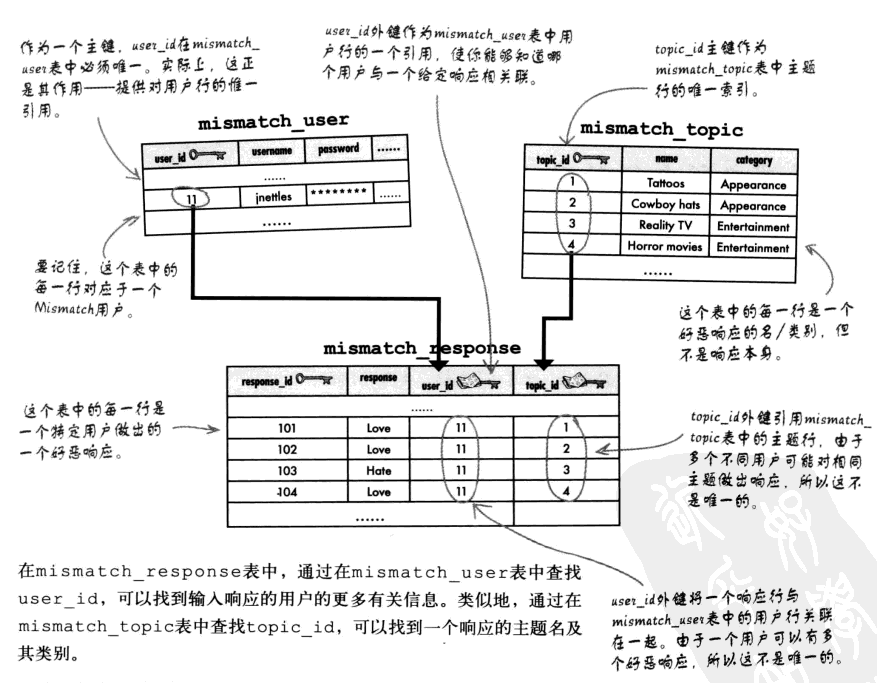
1. 使用模式為數據庫建模：
   * 對數據庫中的數據(表和列)以及所有其他相關對象和它們如何連接的描述稱為一個模式。
   * 創建表的一個結構化圖表可以保証表的設計與表中的數據分離



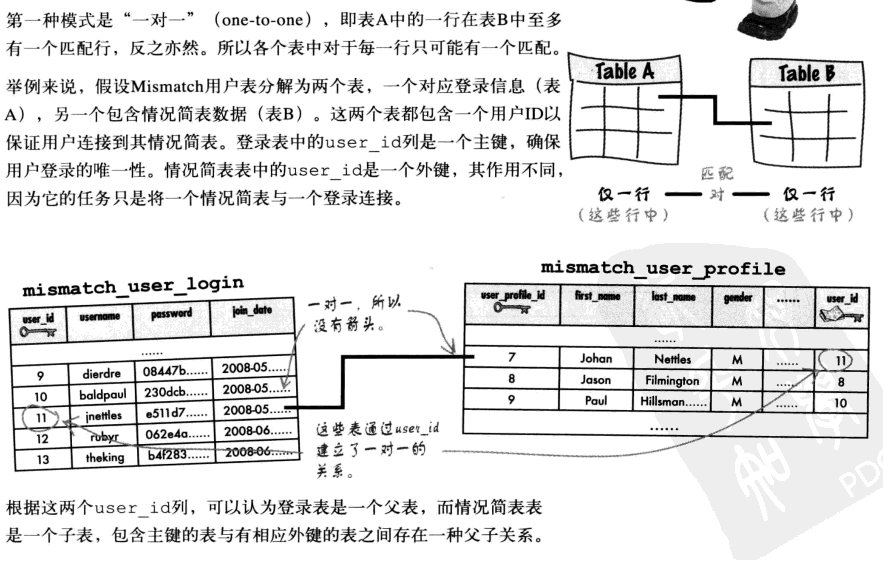
1. 外鍵是一個表中的一列，它引用了另一個表的主鍵。即使是最簡單的數據庫模式中，外鍵也有著很重要的地位。



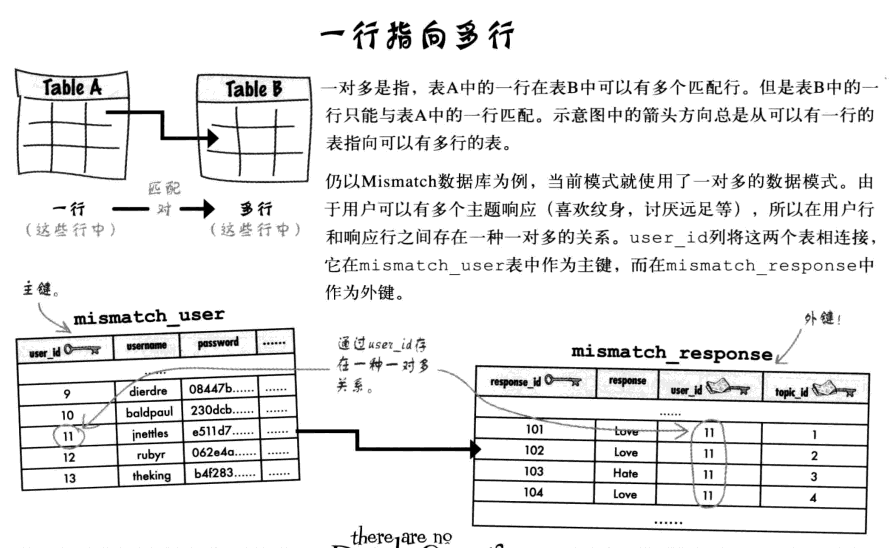
1. 使用外鍵：採用可視化方式將數據流程描述為表，并通過主鍵和外健將表相互連接，這通常很有幫助。再來更仔細地查看包含一些具體數據的mismatch表，這有助於揭示主鍵和外鍵相互之間如何關聯。



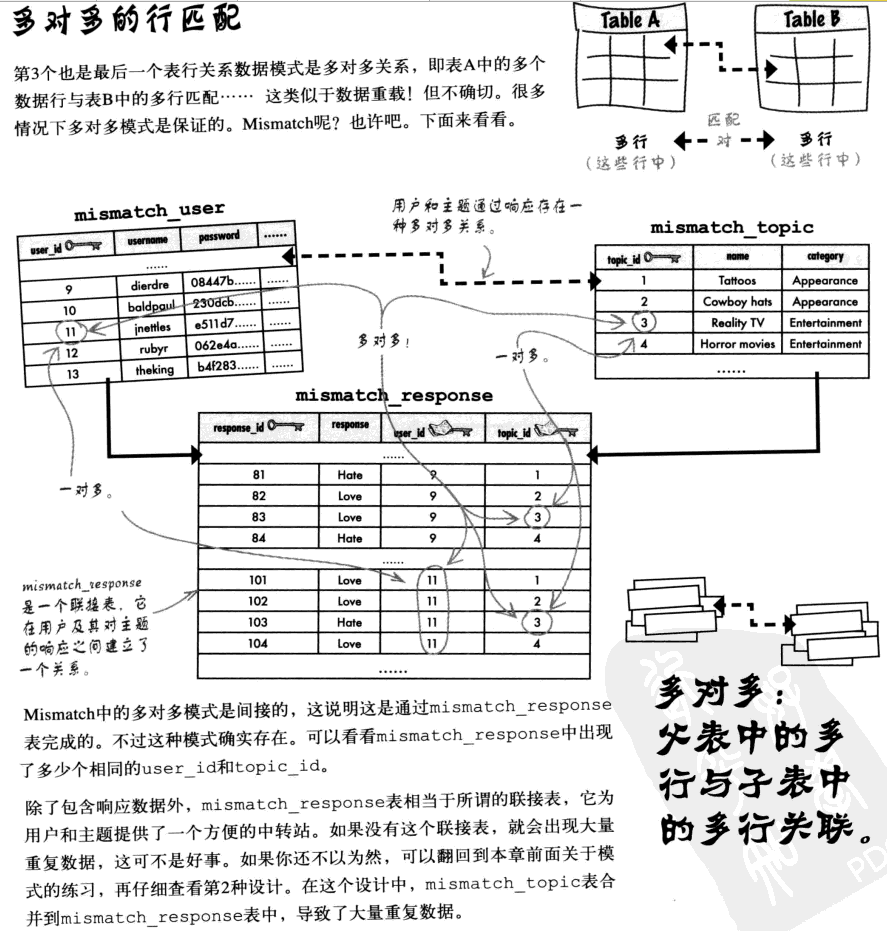
1. 表之間的關系類型：
   * 一對一：父表中的一行與子表中的一行關聯。



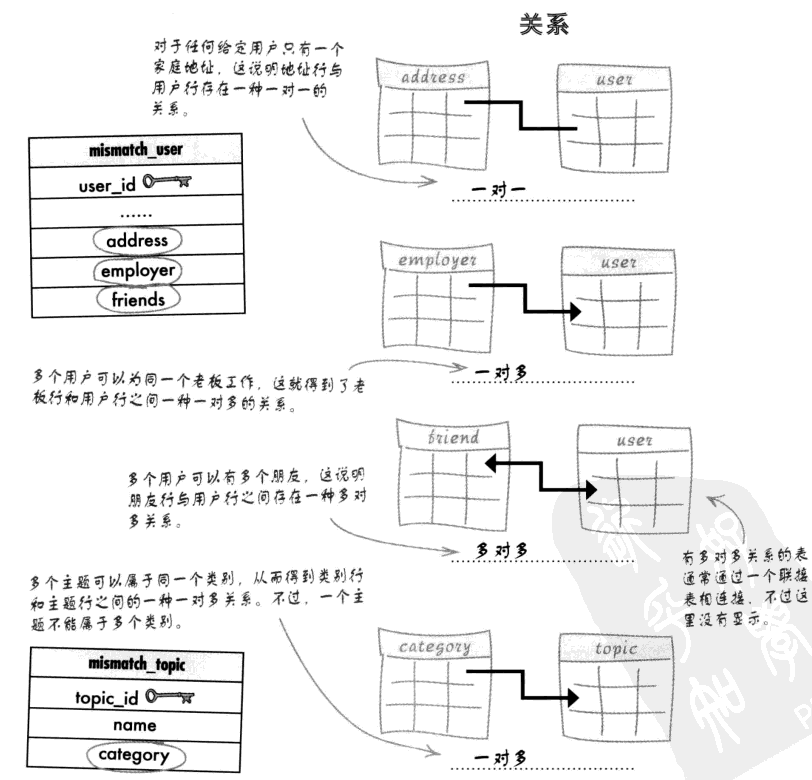
* + 一對多：父表中的一行與子表中的多行關聯。



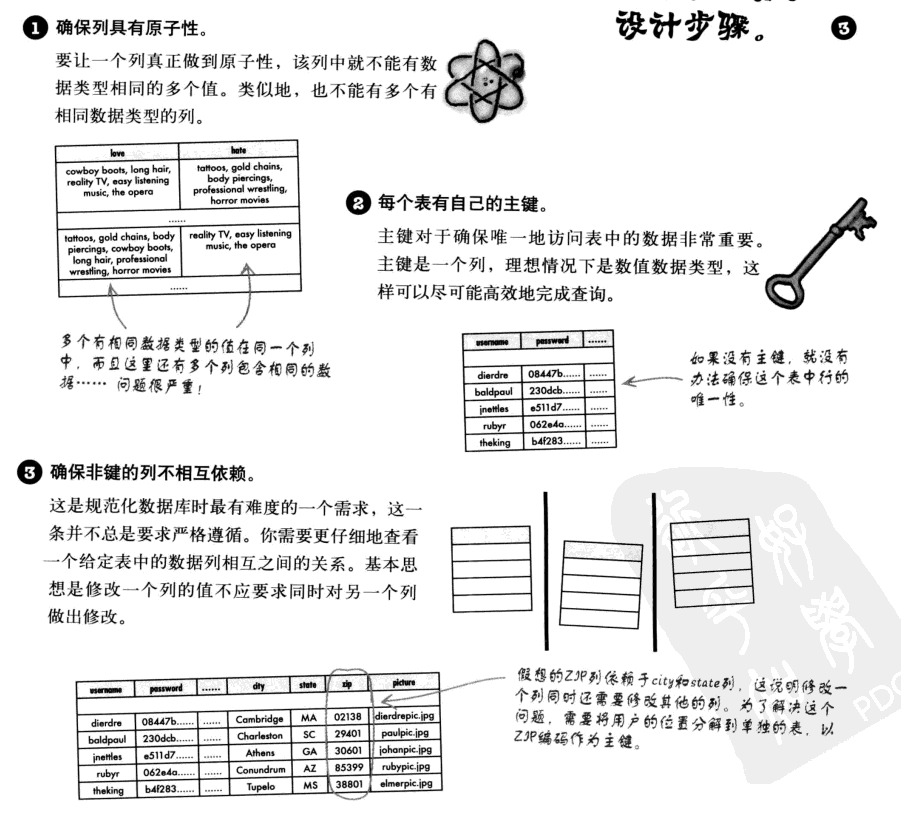
* + 多對多：父表中的多行與子表中的多行關聯。



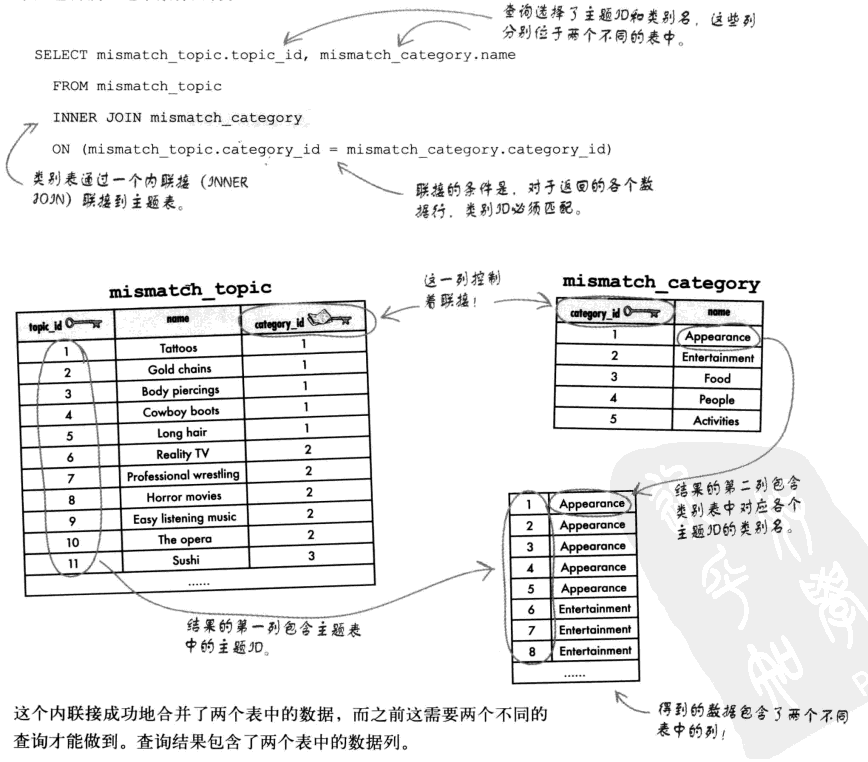
* + 關系判定：



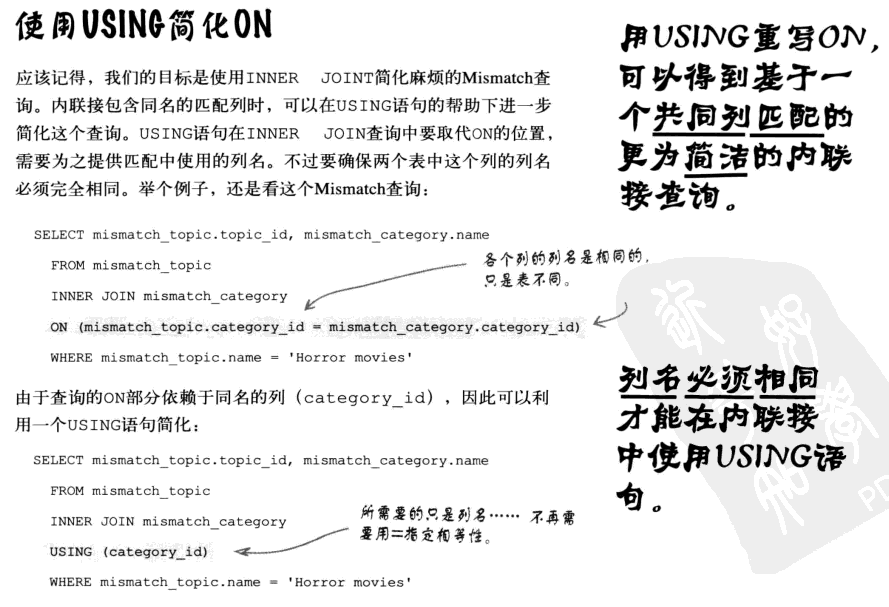
1. 問卷表單數據處理：
   1. 用戶第一次訪問表單時使用INSERT在數據庫中增加空的響應行。
   2. 根據用戶在表單上做出的響應使用UPDATE修改響應行。
   3. 使用SELECT獲取生成問卷表單所需的響應數據。
   4. 根據響應數據生成HTML問卷表單。
2. array\_push() 函数向第一个参数的数组尾部添加一个或多个元素（入栈），然后返回新数组的长度。注意，後面加入的元素，只會建立數字鍵。
   * 例1：array\_push(array,value1,value2)-將值一、值二接到陣列後方
3. 盡可能使用最高效的數據類型在MYSQL數據庫中存儲數據非常重要。
4. 規範化是指設計數據庫來減少重復數據，全改進數據之間的關系。規範化很有好處，具體來說就是數據庫的規模和速度會得到改進。
   * 選擇對象，即希望用表來描述的對象。🡨你希望表描述的主要對象是什麼？
   * 建立使用表時關於對象所需了解的一個信息列表。🡨你如何使用這個表？
   * 使用這個列表，將對象的有關信息分解為組織表時可用的多個部分。🡨列是否包含**原子數據**(原子數據就是分解為給定數據庫所需最小形式的數據)，以保證查詢簡短而且切中要點？
5. 為什麼我們要規範化？
   * 規範表不會有重復數據，這會縮小數據庫的規模。
   * 要搜索的數據更小，查詢也會更快。
6. 規範化數據庫的三大步驟：
   * 確保列具有原子性。
   * 每個表有自己的主鍵。
   * 確保非鍵的列不相互依賴。



1. 點記法：透過點來引用表中的列。
   * 例：SELECT mismatch\_topic.topic\_id, …
   * Mismatch\_topic：表名
   * Topic\_id：列名
2. 聯接(JOIN)：使用這個特性，我們只用一個查詢就可以從多個表獲取結果。
   * 內聯接(INNER JOIN)：它根據一個條件從兩個表中選擇行。在一個內聯接中，查詢結果只包含滿足這個條件的行。



1. 使用USING簡化ON：USING 語句在INNER JOIN查詢中要取代ON的位置，需要為之提供匹配中使用的列名。不過要確保兩個表中這個列的列名必須完全相同。



1. AS：利用別名允許在查詢中重命名一個表或列從而有助于從某種程度簡化查詢。
   * 例：SELECT mt.topic\_id,mc.name from mismatch\_topic as mt…

### 第九章:串與定制函數-通過函數改善生活

1. LIKE：在SQL中，LIKE可以為匹配類型增加靈活性，可以把LIKE認為是=操作符的一種更寬松的版本。另外，LIKE還支援正規表達式。
   * 例：WHERE title LIKE ‘%fighter%’
   * %：通配符，代表所有的字符。
   * \_：通配符，代表一個字符。
2. explode()：此函數能將一個串分解為一個子串數組。
   * $search\_words = explode(‘ ’,’Tipper Cow’);
   * 第一個參數提供要以什麼字符來做為分割的依據，我們稱為定界符。
   * 第二個參數為要分解的字符串。
3. implode()：此函數提供了一個為explode()剛好相反的功能，它能將希望合併的一個串數組組合成一個子串。
   * $where\_clause=implode(‘ OR ’,$where\_list);
   * 結果會輸出一個由’ OR ’聯結的字串。
4. str\_replace()：此函數提供替換字串內容的功能。
   * $clean\_search=str\_replace(‘day’,’year’,’happy day’);
   * 結果：happy year。
5. substr(string,start,length)：此函數提供我們取得字串一部份的方法，string是要擷取的字串，start是要開始的位置(前面從0開始，後面從-1開始)，length是要擷取的長度。
   * $substr(‘are you’,1,2) 🡺結果為re
   * $substr(‘are you’,-2,2)🡺結果為ou
   * $substr(12345,1,2)🡺結果為23(註：表示是可以處理數值，但是會轉成字串來處理。)
6. SUBSTRING()：在MySQL中，SUBSTRING()也提供擷取字串的功能，和substr()不同的是，它的起始索引從1還不是從0開始。
   * SELECT SUBSTRING(job\_description,1,100) FROM riskyjobs;
7. 一致地使用PHP函數有一個好處：我們可以同時得到部份職位描述和完整的職位描述。如果使用MYSQL，只能得到部分職位描述，要得到完整的描述則需要另外建立一個查詢。
8. 函數的定制與使用：

Function replace\_commas($str) {

$new\_str=str\_replace(‘’’,’ ‘,$str);

return $new\_str;

}

$clean\_search=replace\_commas(‘tightrope,walker,circus’);

1. switch語句：switch語句包含一系列case標簽，根據一個變量的值執行不同的代碼塊。

switch($benefit\_code){

case 1:

$benefits=’Major medical, 10 sick days’;

break;

case 2:

$benefits=’Death and dismemberment only,one month paid leave’;

break;

case 3:

case 4:

$benefits=’Good luck!’;

break;

Default:

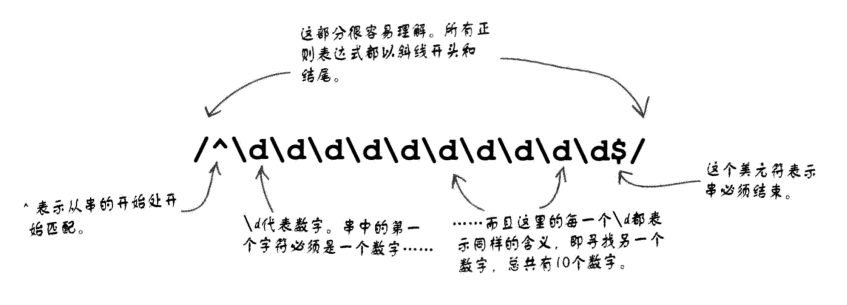
$benefits=’None.’;

}

* 其中case3和4執行的結果是一樣的。

1. 分頁功能：透過MYSQL中LIMIT子句可以很簡單的制作查詢結果的分頁功能。
2. floor(),ceil(),round()：小數進位方式函數。
   * floor(0.22)🡺0，向下取整。
   * ceil(0.22)🡺1，向上取整。
   * round(8.22)🡺8
   * round(8.62)🡺9，四捨五入。

### 第十章:正則表達式-替換規則



1. 正則表達式是一些規則，用于匹配一個或多個串中的模式。
2. 指示重覆，用兩個大括號表示：
   * /^\d{10}$/：將/d重覆10次。
3. 利用元字符可以在正則表達式中描述文本模式。
   * \d：查找一個數字。這個元字符會匹配0到9之間的任何數字，\d本身只匹配一位數字，所以如果希望匹配一個兩位數，則要使用\d\d或\d{2}。
   * \w：查找任何字母數字字符。範圍包括a-z、A-Z、0-9。
   * \s：查找空白符。包含空格鍵、制表符、換行或回車符。
   * ^：查找一個串的開始位置。用它來指示文本串的起始位置開始匹配。例如：正則表達式/^\d{3}/能匹配串 “300 applications”，但不能匹配”wereceived300 applications”。
   * .：點元字符可以匹配除換行符以外的任意一個字符。它能匹配字母或數字，還可以匹配空格或制表符。
   * $：查找串尾。可以結合^使用這個$元字符確定匹配的範圍，指定匹配究竟從哪里開始到哪里結束。例如：/^\w{5}\s\d{3}$/能匹配”nanny 411”,不能匹配”nanny 411 is great”或是”call nanny 411”。
4. 如果希望正則表達式中有一個特定字符該怎麼做？只需要直接在表達式中使用該字符。例如：如果希望匹配具體的電話號碼”707-827-7000”，可以使用正則表達式/707-827-7000/。