PHP＆MariaDB教學手冊

目錄

[第一章 下載及安裝XAMPP說明 1](#_Toc144205519)

[第二章 設定與測試PHP、MariaDB環境 4](#_Toc144205520)

[第三章 MariaDB資料庫與資料表基本操作 9](#_Toc144205521)

[第四章 圖形介面操作 14](#_Toc144205522)

[第五章 撰寫PHP程式 24](#_Toc144205523)

[第六章 網頁程式範例 31](#_Toc144205524)

[第七章 檢查資料庫回傳的錯誤訊息 36](#_Toc144205525)

[第八章 日期的處理 38](#_Toc144205526)

[第九章 Session的應用與介紹 40](#_Toc144205527)

[第十章 Q&A 41](#_Toc144205528)

[附錄A: 編輯工具 43](#_Toc144205529)

[附錄B: 安裝XAMPP的問題和解決方法 44](#_Toc144205530)

# 下載及安裝XAMPP說明

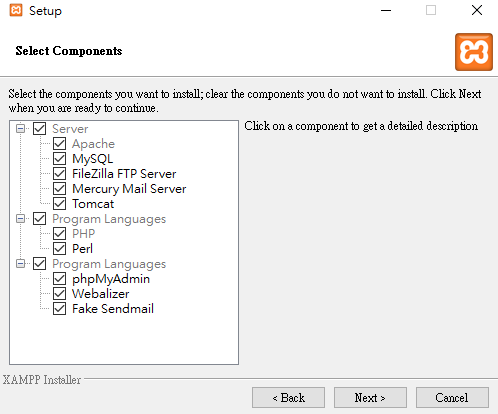
1. 連結至<https://www.apachefriends.org/zh_tw/download.html>，下載整合PHP、MariaDB的架站工具**XAMPP之64bits**版本。

目前的最新版本與本份文件不同，本份教學文件使用的為編號8.2.4的安裝檔，此版本包含之PHP為8.2.4，MariaDB為10.4.28。若需下載與本份文件相同版本執行檔，請點選上頁的 “更多下載”，然後如下兩頁依序點選 “XAMPP Windows”和 “8.2.4”。



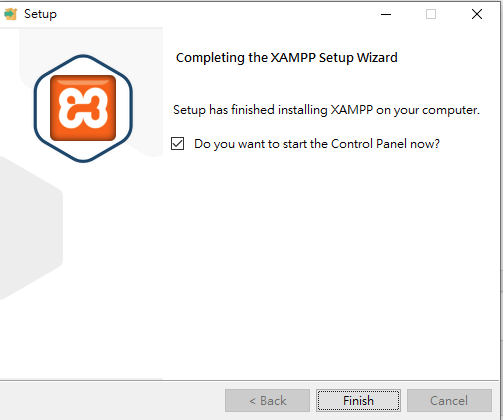
1. XAMPP的安裝過程，原則上皆勾選使用預設值、預設路徑即可(本學期主要是用到Apache, MariaDB(MySQL), PHP, phpMyAdmin等)。

註:部分畫面仍使用「MySQL」一字，而尚未改成「MariaDB」

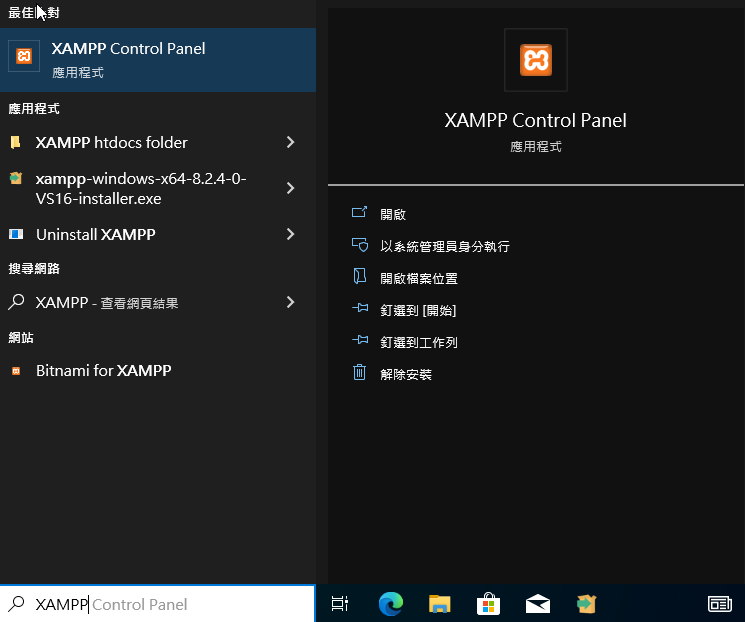


。

安裝成功的畫面如下



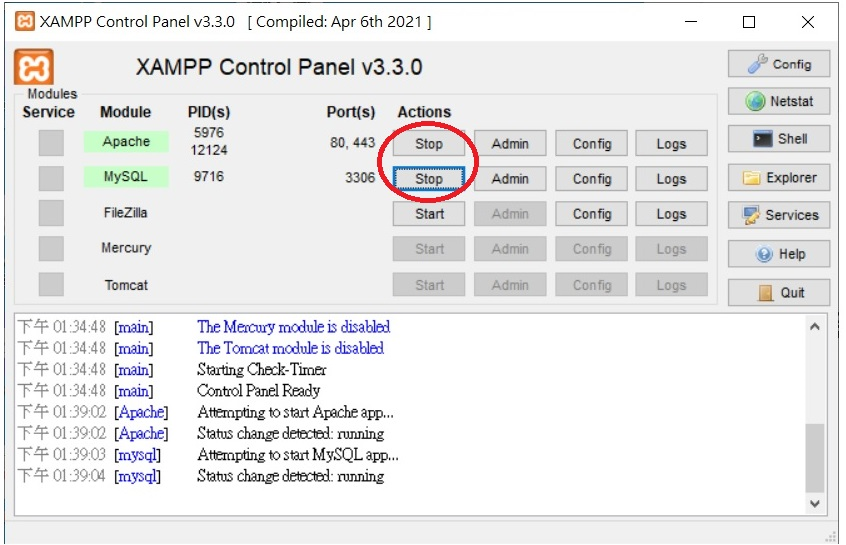
1. 安裝完成後，即可啟用XAMPP Control Panel。



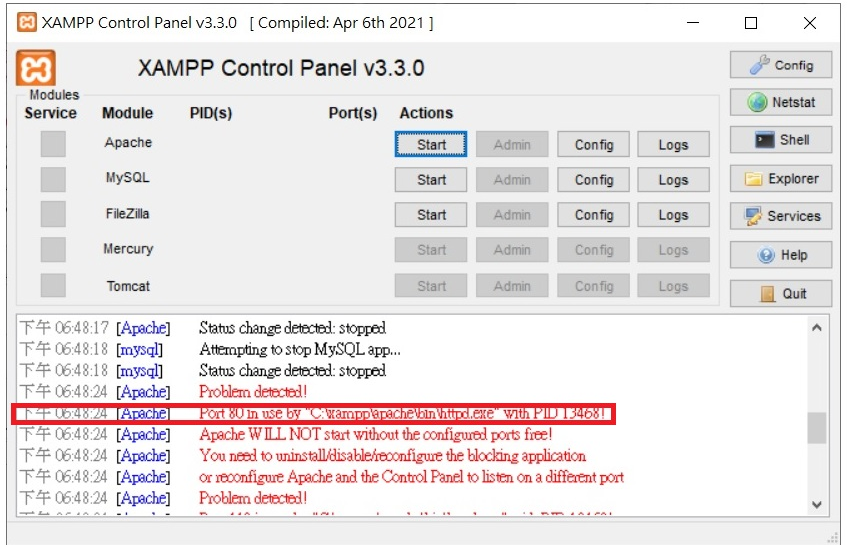
1. 無論是html或是php文件，都是純文字文件格式，同學可自行使用Windows所附的記事本或是使用NotePad++ 等文件編輯器。(參見[附錄A](#_附錄A:_編輯工具))

# 設定與測試PHP、MariaDB環境

接下來我們要對PHP及MariaDB做一些必要的設定，才可使將來編寫的PHP網頁正常工作。

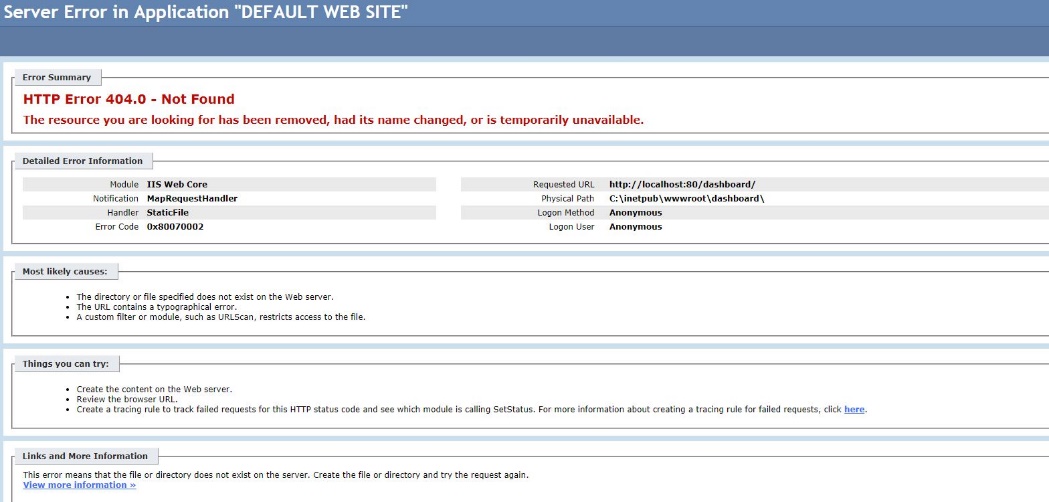
1. 安裝完後，要使用phpMyadmin前，每次開機都必須先開啟XAMPP Control Panel，再手動開啟**Apache**及**MariDB(MySQL)**。

若因Port: 80被占據使得Apache無法啟動，Control Panel會出現以下錯誤訊息，解決方法請參閱[**附錄B**](#_附錄B:_安裝XAMPP的問題和解決方法)。



1. XAMPP預設是將網頁放置於 **C:\xampp\htdocs**目錄下，同學可直接在瀏覽器中輸入 <http://localhost> 位址，若看到下面畫面表示PHP環境安裝成功。

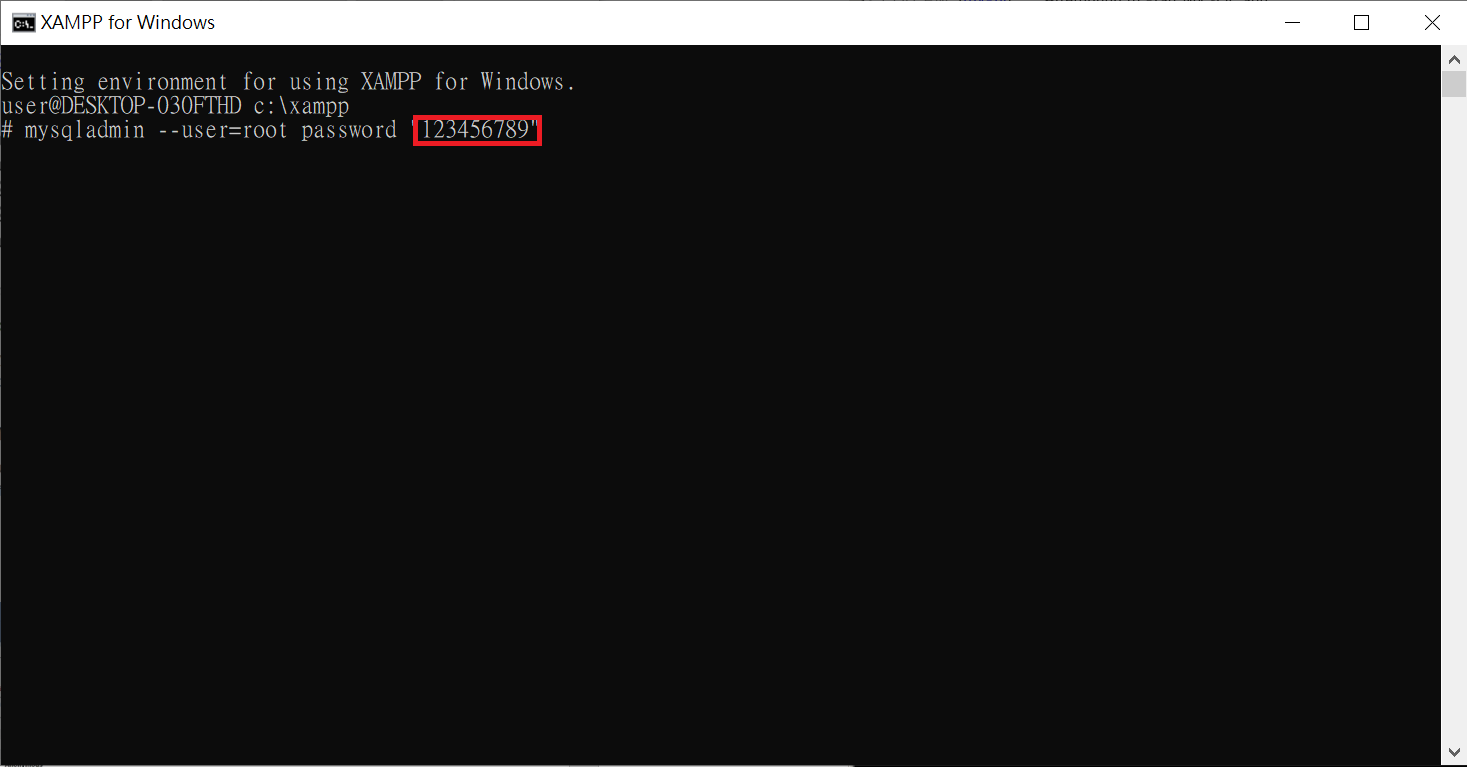
若出現下面畫面，則表示Apache無法啟動，解決方法同樣參照**附錄B**



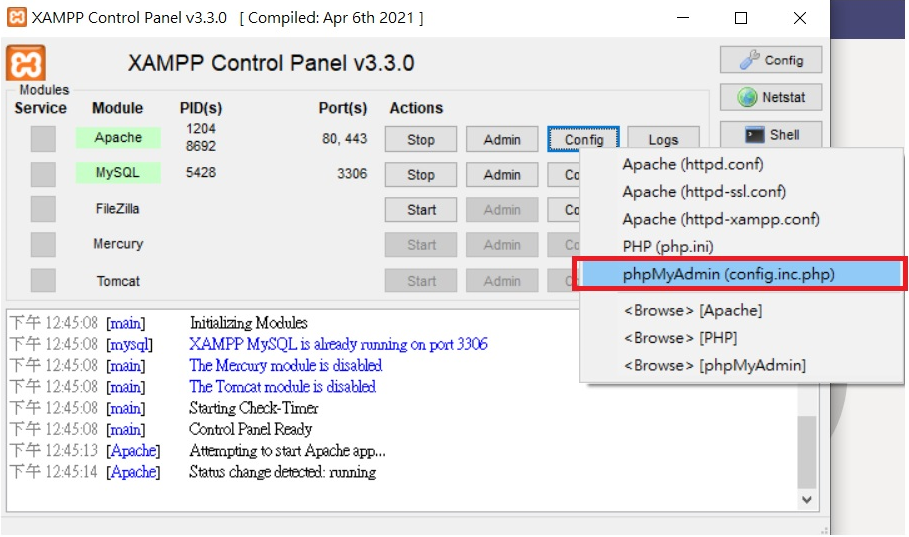
1. 點選Shell



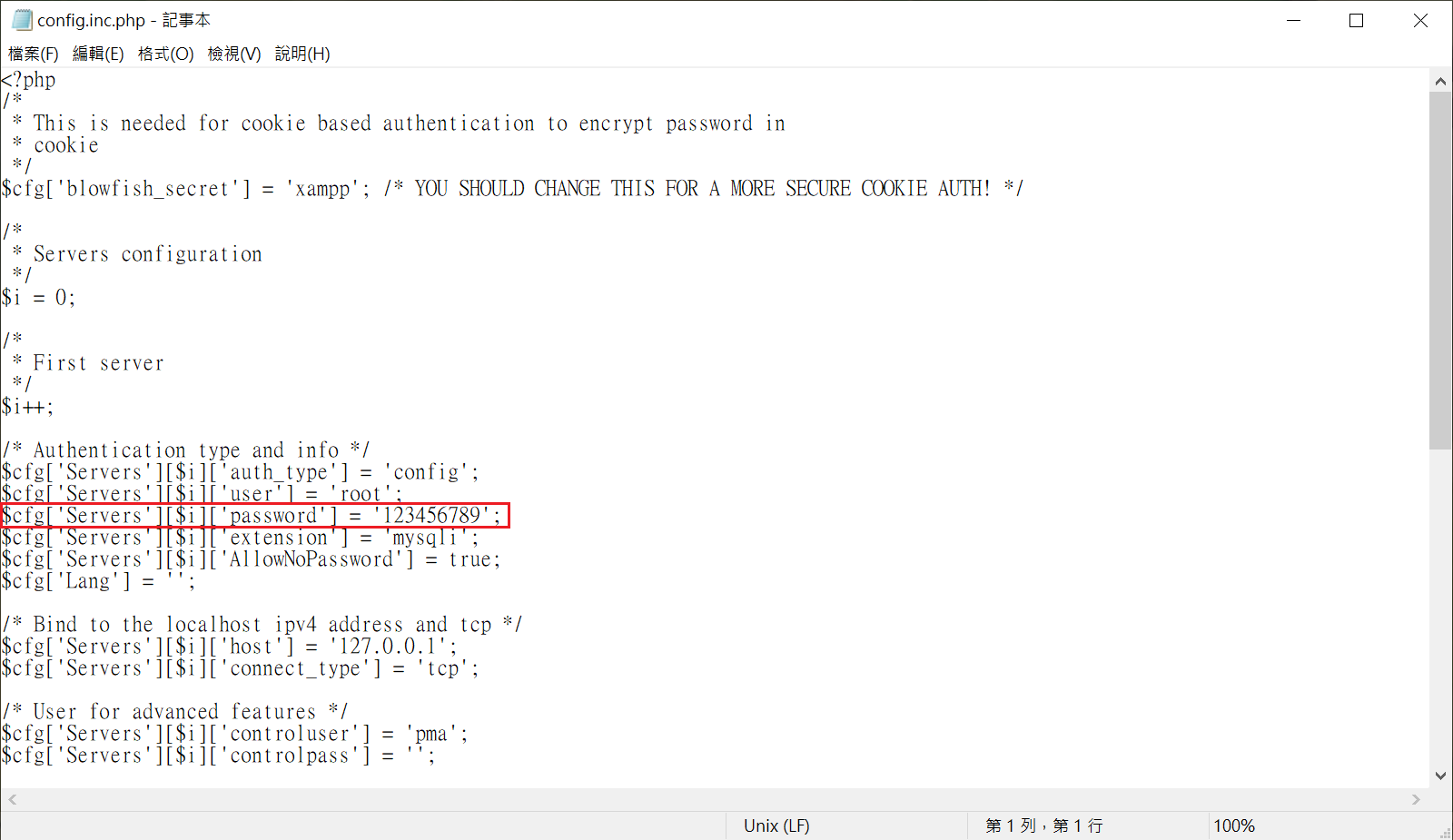
1. 輸入mysqladmin --user=root password "12345789"，此為MariaDB帳號"root"的密碼。 附註:紅字部分可自行設定。



1. 點選Apache的[Config]/[phpMyAdmin(config.inc.php)進行密碼設定，以便使phpMyAdmin可以連到MariaDB。



找到['password'] = ''後，輸入你上一步設定的密碼



1. 將命令提示字元移至桌面當捷徑：

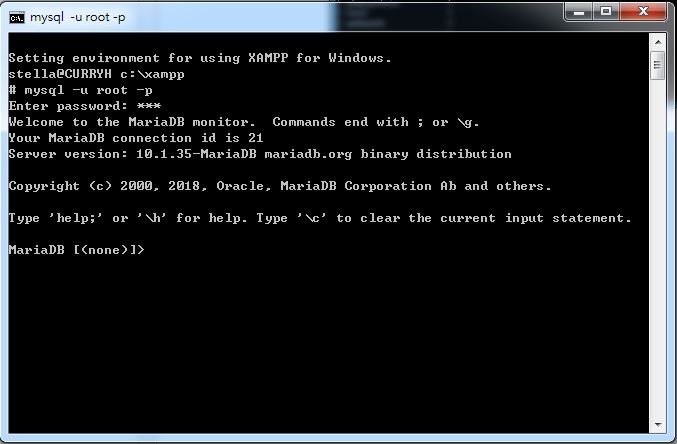


1. 在命令提示字元上點選右鍵，設定開始位置為**XAMPP**的安裝位置(預設路徑為： **C:\xampp\mysql\bin**，如下圖)，如此編輯程式時就不用辛苦在DOS模式下切換目錄。



1. 進入命令提示字元，輸入**mysql -u root -p**

跳出輸入password，輸入剛剛設定的密碼後即可進入MariaDB的SQL指令模式。



# MariaDB資料庫與資料表基本操作

本章介紹在MariaDB SQL指令模式下的重要指令，若要離開可輸入「\q」。

1. **資料庫與資料表相關指令：**
2. 在 MariaDB> 中輸入

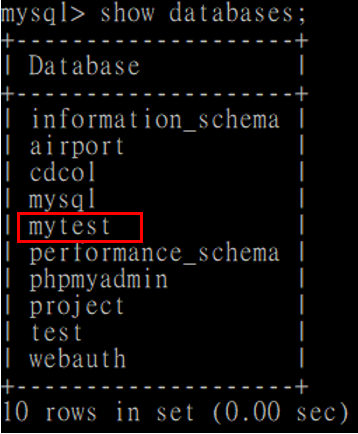
**create database mytest DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci;**



以上指令可建立一個 “mytest＂資料庫。注意，MariaDB的預設語系是 “latin1”，加入“DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci”的指令，可將資料表語系設為utf8且其排序規則為utf8\_general\_ci，以避免中文於資料表中顯示為亂碼。

1. 輸入以下命令可顯示目前所建立的所有資料庫

**show databases;**



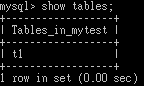
1. 由於在MariaDB資料庫管理系統中，可能會有許多個資料庫存在，因此我們在使用時，要先指定預設的資料庫為何，如下指令:

**use mytest;**



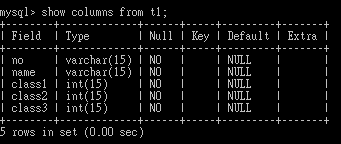
1. 假設資料表t1已建立，查詢資料庫中所定義的所有資料表:

**show tables;**



1. 假設資料表t1已建立，查詢資料表t1的schema定義

**show columns from t1;**



1. 刪除資料表

**drop table t1;**



1. 要刪除「mytest」資料庫，可輸入:

**drop database mytest;**

****

1. **Check語法**

Check: 在建立一筆資料時，會檢查該筆資料的某個欄位的值是否在一開始定義的範圍之內。假設我們在Persons table裡，限定P\_Id必須大於2，也就是若是輸入的值小於等於2，則會導致失敗。語法如下:

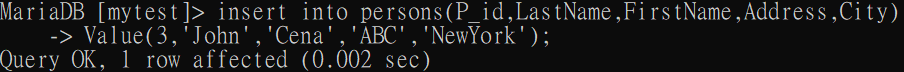
CREATE TABLE Persons

(P\_Id int NOT NULL,  
LastName varchar(255) NOT NULL,  
FirstName varchar(255),  
Address varchar(255),  
City varchar(255),

Primary key(P\_id),  
CHECK (P\_Id>2));

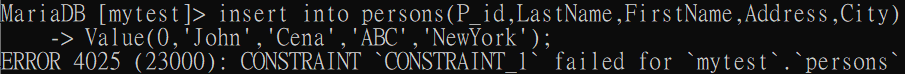
**執行結果如下:**

1. 新增成功:

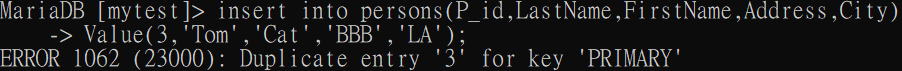
****

1. 新增失敗，因為P\_Id必須大於2:

(->符號在輸入完指令後按enter會出現)

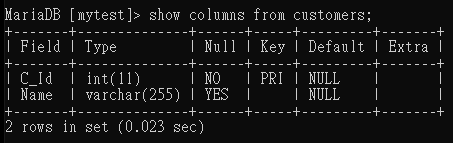
****

1. 補充Primary key的錯誤訊息，Primary key為P\_id，在(1)已存在P\_id為3的欄位，Primary key不可重複:

****

1. **FOREIGN KEY 於CREATE TABLE中的語法**

假設現有customers 資料表定義如下：



若要建立 orders 資料表並將其 C\_Id 欄位設為外鍵關聯到 customers 資料表中的 C\_Id，語法如下:

CREATE TABLE orders (

O\_Id INT NOT NULL,

Order\_No INT NOT NULL,

C\_Id INT,

PRIMARY KEY (O\_Id),

FOREIGN KEY (C\_Id) REFERENCES customers(C\_Id));

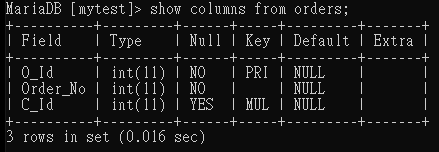
**執行結果如下:**

1. 新增成功

**一張含有 文字 的圖片

自動產生的描述**

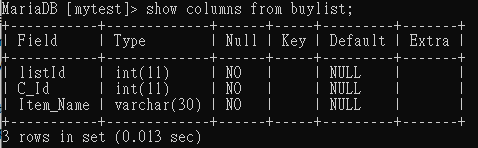
1. Order資料表定義如下:

****

附註：C\_Id的Key顯示MUL代表其值為Multiple(可重複)。

1. **FOREIGN KEY 於ALTER TABLE中的語法：**

假設現有buylist 資料表定義如下：

****

若要將現有 buylist 資料表中的C\_Id 欄位，更改設定為外鍵並將其關聯到 customers 資料表中的 C\_Id，語法如下:

ALTER TABLE `buylist` ADD FOREIGN KEY (`C\_Id`) REFERENCES `customers`(`C\_Id`);

**執行結果如下:**

1. 執行成功:



1. 更改後的buylist資料表定義如下:

一張含有 文字 的圖片

自動產生的描述

若由圖形介面設定foreign key並選擇 “restrict”選項，該項操作轉換出的SQL如下，其效果與上面的語法效果相同：

ALTER TABLE `buylist` ADD FOREIGN KEY (`C\_Id`) REFERENCES `customers`(`C\_Id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

圖形式介面的詳細說明將於第四章進行介紹。

# 圖形介面操作

1. 在安裝完XAMPP後，在網頁上輸入<http://localhost>就可登入自身的網頁中。點選右上phpMyAdmin，即可進入資料庫當中( **在這裡使用者名稱預設為root，密碼為當初設定的密碼** )





1. **新建資料庫**: 在左側按下「新增」

**\*\*注意 : 資料庫名稱必須為英文、建議全部為英文小寫(減少錯誤率)**

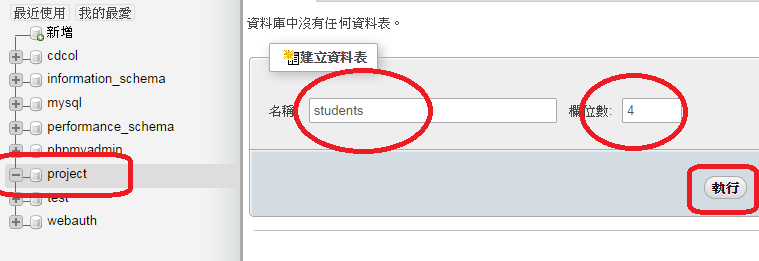
輸入新的資料庫名稱以及編碼排序方式後，按下「建立」鍵，

編碼選取方式如下圖所示，一般建議選取**utf8\_general\_ci**以便於輸入中文之欄位值，其他詳細說明可以參考以下連結:

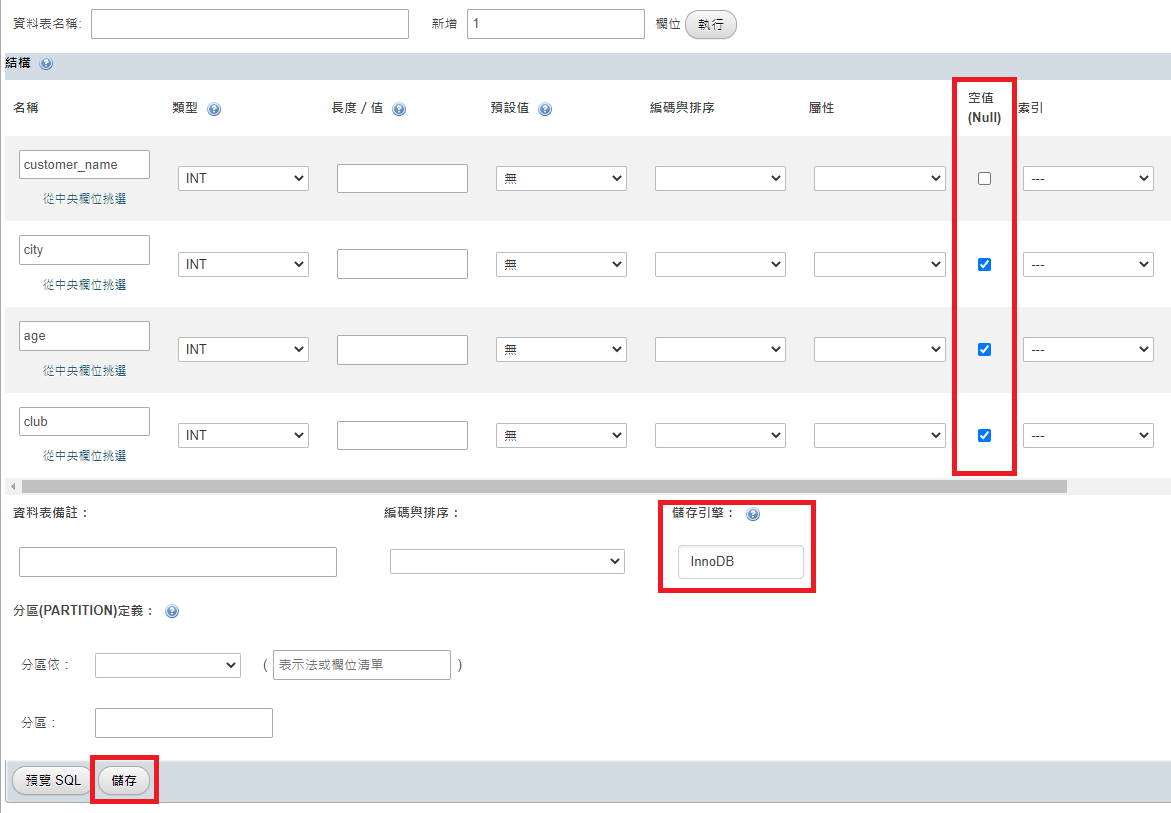
<https://kknews.cc/zh-tw/code/5rak5v8.html>

[](http://www.geocities.jp/jpcutes/mysqlcheck/index.htm)

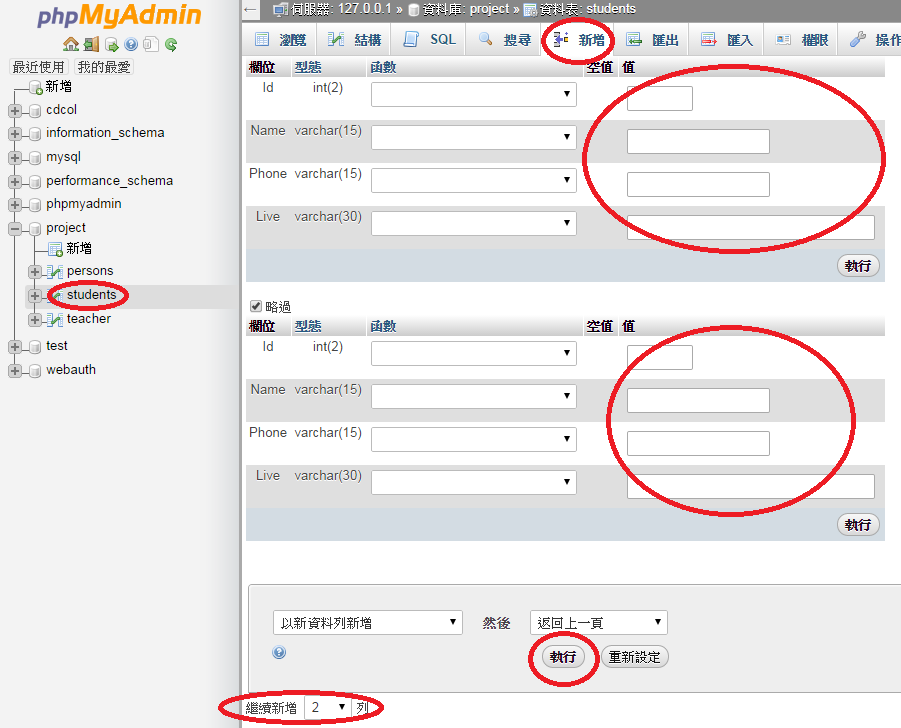
1. **新建資料表**: 在左側資料庫選單，先點選資料庫，如「project」，輸入新的資料表名稱以及欄位數量，再按右下方的「執行」。



1. 在「欄位」部份輸入每一個欄位的名稱( **\*\*注意： 欄位名稱必須謹慎輸入，下圖欄位名稱為範例** )，型態若是純數字則可選擇INT型態，若是字元型態則選擇VARCHAR。屬性的部份INT可選擇是正整數或是包含負數。如果要設定可**null**就把紅框區塊中的選項打勾，**not null**就維持預設不打勾。設定完成後按下左下方的「儲存」，就完成資料表的建立。

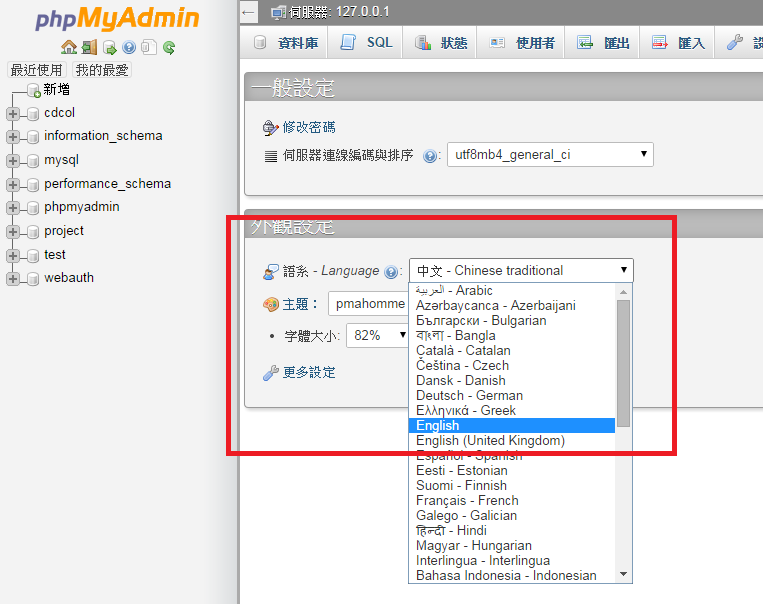


1. 新增資料: 點選左側一個資料表後，直接點選上排的「新增」即可開始新增資料。將下方的欄位值填寫完後，按下最下方的執行，即可新增所有資料。若要新增更多筆資料可以在最下方的下拉式選單選擇「繼續新增 ?? 列」。

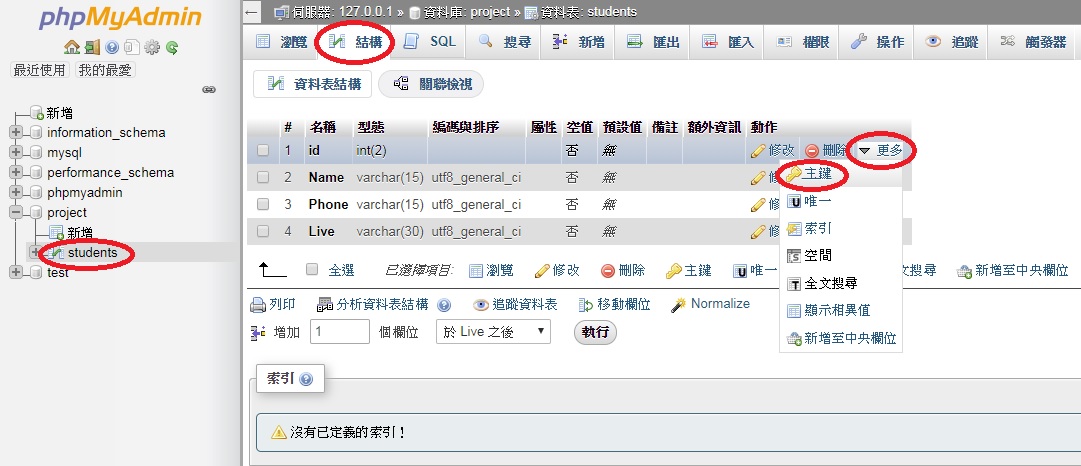


1. 瀏覽資料表內容: 點選左側一個資料表後，直接點選上排的「瀏覽」即可看到資料表的內容。
2. 顯示資料表結構: 點選左側一個資料表後，直接點選上排的「結構」即可看到資料表的結構。(如果點選「結構」後有當掉的情形，可以在每次要選資料表的「結構」時，在結構上按右鍵，開啟新分頁或新視窗，即可解決該問題。另一個方法是，點選如下圖左側的房子圖回到首頁，然後將語系改為English，即可正常。)





1. 設定主鍵: 點選「結構」後，直接點選右方「更多」就能設定「主鍵」。( 建構好資料表可以不定義主鍵，但定義了之後就一定要有一個為主鍵，不能取消，只能更換 )



若選擇Id當主鍵，按下主鍵的鑰匙圖示後，如下圖所示。

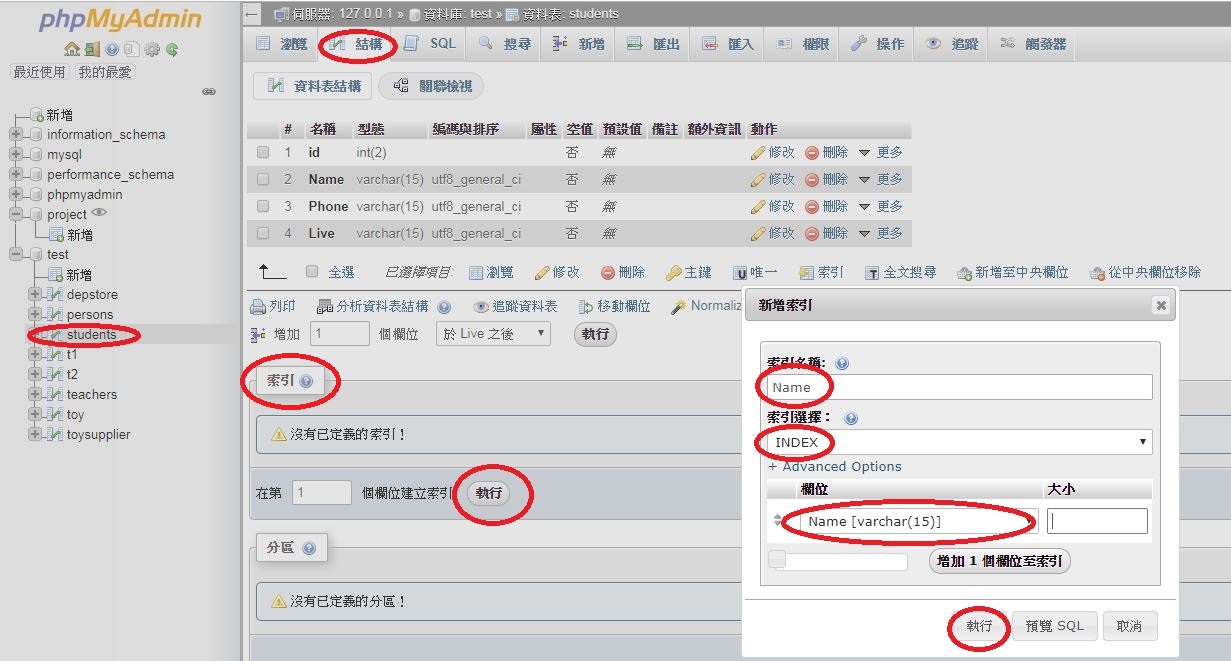


**\*\*注意 :** 若一個主鍵包含多個屬性，要先同時選取那些屬性，再按下主鍵圖示。

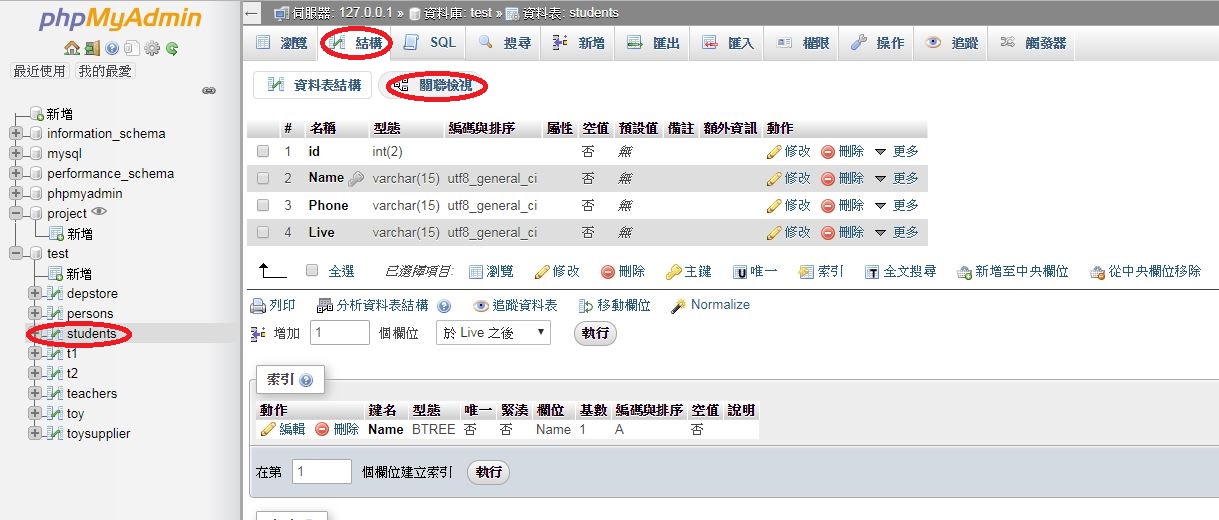
1. Foreign key的操作方式

在下面的例子，我們設定資料表students的Name要reference到teachers的Name，先確認被referenced的屬性Name在teachers表格是primary key。另外，在圖形化的操作上，如果一個欄位本身不是primary key但要reference到其他表格，需先把他建成索引(index)，如下圖所示：

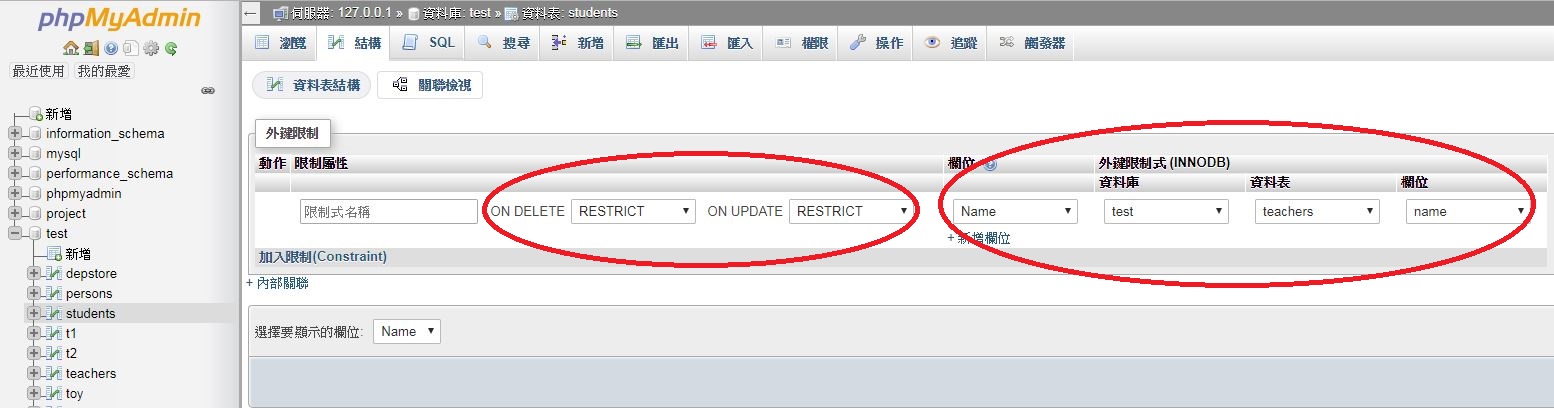
* 1. 索引的操作

點選資料表students，在「結構」下的「索引」點擊「執行」後，會跳出一個小視窗，自己給定一個索引名稱，索引類型選index，欄位是定義這個index是建在哪個欄位上，這裡是建在Name上。

* 1. 在students的「結構」下點擊「關聯檢視」，如下圖的方框所示。



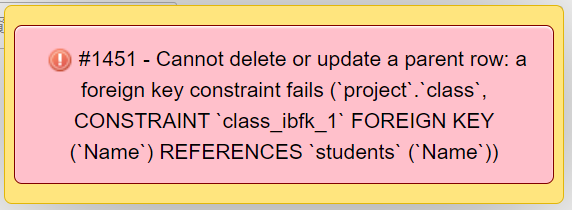
* 1. 按下「關聯檢視」後會跳出下圖畫面，選取要設定reference的兩個欄位，並針對delete/update設定restrict/cascade的限制，如下圖框框。 ( 如點選「關聯檢視」無反應，在「關聯檢視」上按右鍵開啟新分頁或新視窗即可 )。



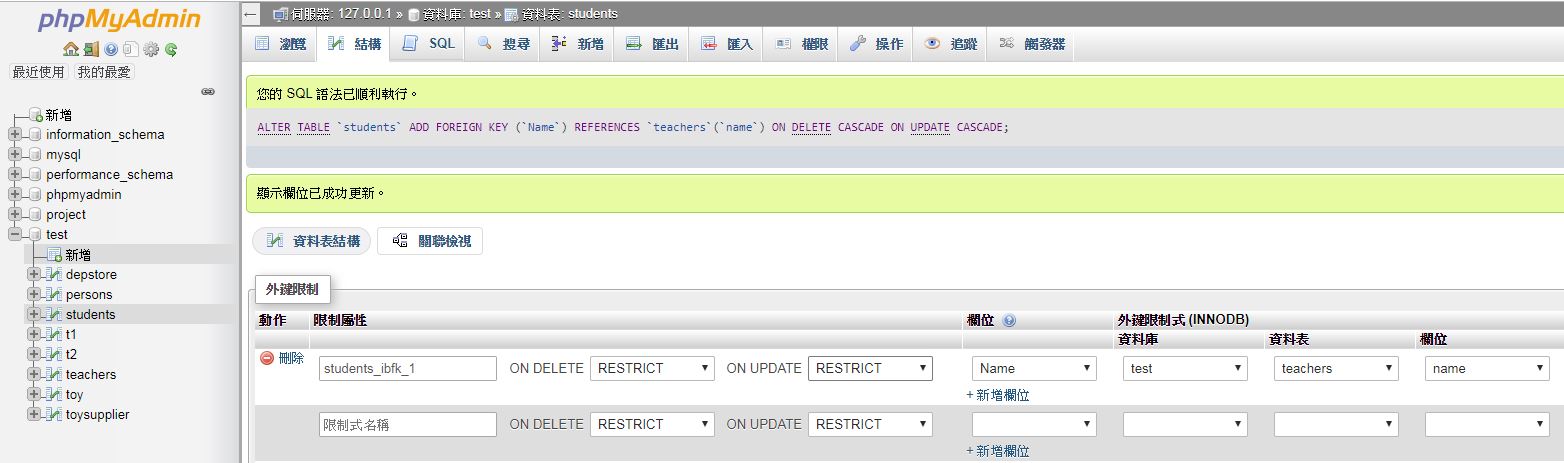
1. ON DELECT/ ON UPDATE 的四種模式如下:

* RESTRICT：預設模式。當在父表格刪除資料或修改主鍵欄位時，會比對子表格是否有對應的資料，如果有則不允許刪除或修改。
* CASCADE：當在父表格刪除資料或修改主鍵欄位時，會比對子表格是否有對應的資料，如果有則會一起刪除或修改。
* SET NULL：當在父表格刪除資料或修改主鍵欄位時，會比對子表格是否有對應的資料，如果有則將子表格的外鍵欄位值設為 NULL（注意：欄位需允許 NULL）。
* NO ACTION：與 RESTRICT 相同。

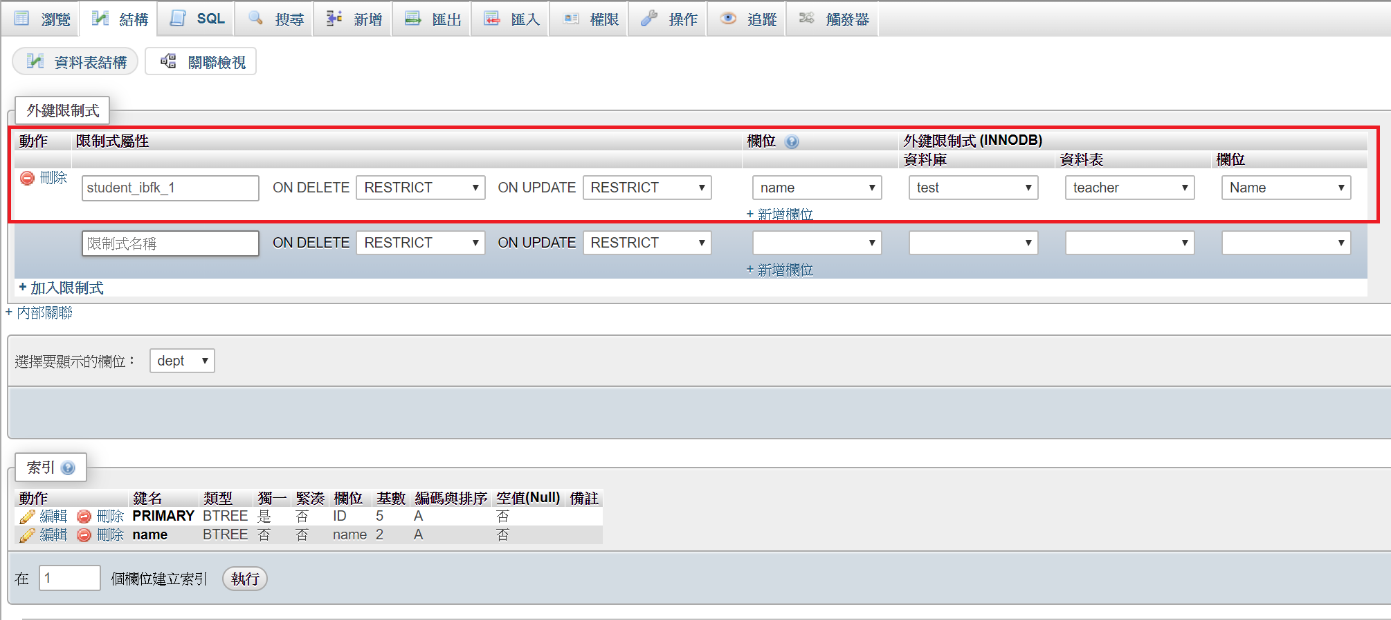
1. 若是delete/update設定為RESTRICT，針對父表格刪除資料時，若子表格有對應的資料，會出現以下錯誤訊息：



(4)完成後按「儲存」（Save），若成功即會顯示以下畫面。



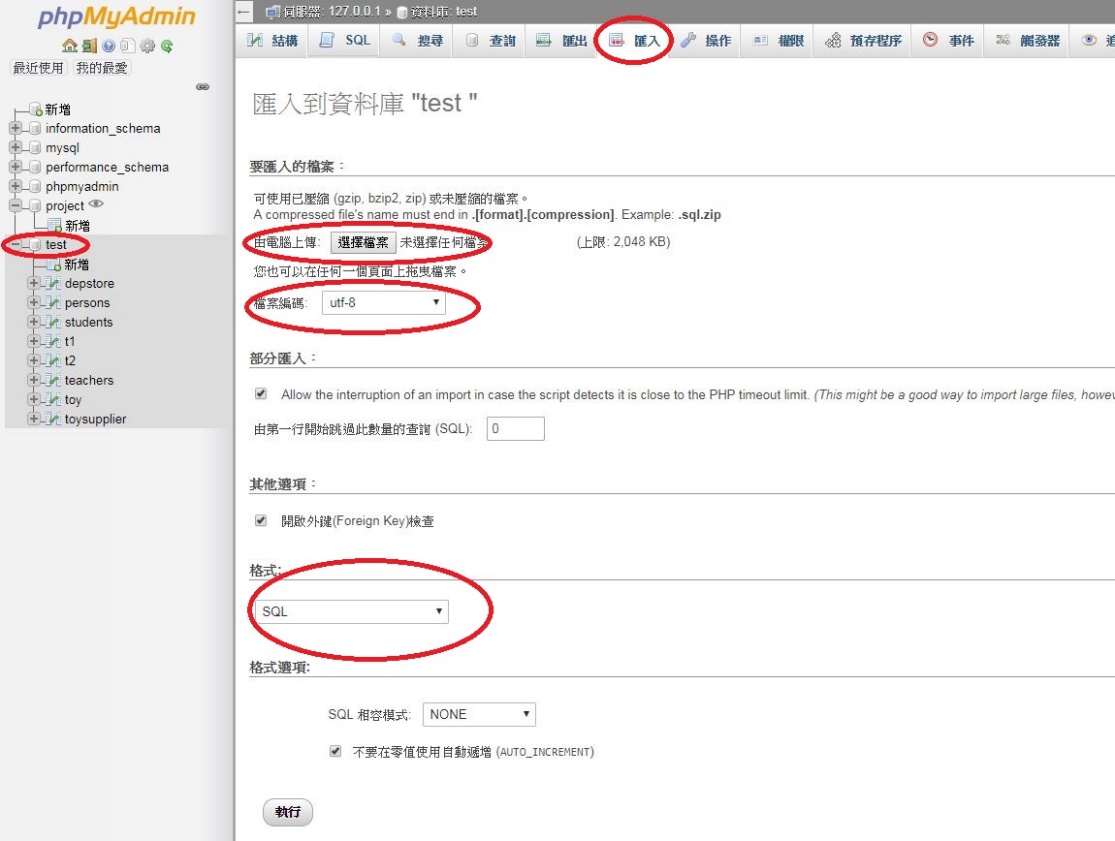
再次按下「關聯檢視」後會跳出下圖畫面，可以確認已經設定好的foreign key。



1. 匯出資料庫: 選定左側的一個資料庫後，如「test」，點選上排的「匯出」後，匯出方式: 選擇 ●快速、格式: SQL，按下執行即可。



1. 匯入資料庫: 注意\*\*： 匯入前必須先創立一個資料庫，要與匯出的資料庫名稱相同。選定欲匯入的資料庫後，如「test」，點選上排的「匯入」後，匯入方式: 選擇資料庫sql檔，選擇檔案編碼utf-8，選擇格式SQL後，按執行即可進行匯入。



# 撰寫PHP程式

本章介紹如何以php架設網站。請注意，PHP程式碼必須寫在

<?php

…

?>

之內。

1. 首先我們做個簡單的網頁並使用PHP程式：

(1)打開任意文書編輯程式，輸入下列程式碼

**<html>**

**<head>**

**<title>HelloWorld</title>**

**</head>**

**<body>**

**<?php**

**echo "This Is My First PHP page !!";**

**?>**

**</body>**

**</html>**

(2) 將檔案存成 “helloworld.php”，置放路徑為 “**C:\xampp\htdocs**”

(3) 在瀏覽器中輸入http://localhost/helloworld.php位址，即可看到執行結果。



1. 接下來我們將結合PHP與MariaDB，將資料庫應用於網頁當中。

若在網路上找一些PHP連接資料庫的範例，大多會看到的寫法都是PHP最早期的API，例如mysql\_connect(),mysql\_query(),mysql\_select\_db()等等。由於PHP版本的更新，自從PHP7.0.0之後不會再支援以上的api。

目前在PHP7以上版本中提供了兩種連接資料庫的API: PDO\_MYSQL和MySQL Improved Extension（MySQLi）。PDO\_MYSQL和MySQLi差別主要在於PDO\_MySQL提供多種資料庫連接，例如：MySQL、PostgreSQL、MS SQL Server、SQLite等等都可以使用PDO對應並擴展。反之，MySQLi只支援MySQL/MariaDB，對只使用一種資料庫連線的使用者極為方便，但是要想使用別種資料庫改動會非常大。故在本章中我們使用PDO\_MYSQL的用法為例。

1. **建立範例所需的資料庫:**

新增資料庫test，並建立student資料表。

****

1. **連接資料庫的範例**(db\_conn.php)**:**

|  |
| --- |
| **<?php**  $user = 'root'; //資料庫使用者名稱  $password = 'pass'; //資料庫的密碼  try{  $db = new PDO('mysql:host=localhost;dbname=test;charset=utf8',$user,$password);  //之後若要結束與資料庫的連線，則使用「$db = null;」  $db->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);  $db->setAttribute(PDO::ATTR\_EMULATE\_PREPARES,false);  }catch(PDOException $e){ //若上述程式碼出現錯誤，便會執行以下動作  Print "ERROR!: " . $e->getMessage();  die();  }  **?>** |

* db為一個MariaDB連線的物件。
  + 1. $db = new PDO ('mysql: host=localhost;

dbname=database; charset=utf8', $user, $password);

* + - * 開啟和MariaDB的連線。

localhost為主機名或IP位址、database為Database名稱、user為MariaDB使用者名稱、password為MariaDB使用者密碼。

詳見<http://php.net/manual/en/pdo.construct.php>。

* + 1. $db->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);
       - 透過上述的db連線引出連線錯誤報告以及拋出exceptions異常。

詳見<http://php.net/manual/en/pdo.setattribute.php>。

* + 1. $db->setAttribute(PDO::ATTR\_EMULATE\_PREPARES,false);
       - 使用perpare和execute兩步驟，是先透過prepare讓database

Driver預先準備好query所需空間，然後再透過execute將值代入並執行，此方法也可以防止SQL injection的問題。若此處將值設為TRUE，則是強制由PDO模擬**預處理(prepare)**，無法真正防止SQL injection。若此處設為FALSE，才會真正交由database

Driver**預處理(prepare)**，除非driver不能成功預處理(prepare)，才會交由PDO模擬。為了安全性的考量，此處設為**FALSE**。

* + 1. $db = null;
       - 關閉與MariaDB連線。

**需要注意的是**，必須將1)、2)、3)放進try{}，另外catch exception，詳見範例。

1. **新增資料範例(insert.php):**

|  |
| --- |
| <?php  //連接資料庫  include\_once "db\_conn.php";  //設定想要新增入資料庫的資料內容如下  $dept = "資工";  $no = "00557888";  $name = "發發發";  $club = "桌遊社";  **//使用預處理寫法是為了防止「sql injection」**  //設定要使用的SQL指令  $query = ("insert into student values(?,?,?,?)");  $stmt = $db->prepare($query);  //執行SQL語法  $result = $stmt->execute(array($dept,$no,$name,$club));  **//以下為一般直接執行而沒有預處理的寫法，與上面的程式碼一起跑會因//為重複輸入而出現錯誤**  $full\_query = "insert into student values  ('資工','00557888','發發發','桌遊社')";  $stmt = $db->query($full\_query);  ?> |

新增資料所使用到的物件/API/變數如下:

* db為一個MariaDB連線的物件。
* query為SQL語法。
* stmt對應到一個欲執行SQL的物件。

1. $stmt = db->prepare($query);
   * + - 提前預處理SQL query，回傳statement。

詳見<http://php.net/manual/en/pdo.prepare.php>

1. $stmt->execute(array($dept,$no,$name,$club));
   * + - 執行SQL語法。

詳見<http://php.net/manual/en/pdostatement.execute.php>

1. $db->query($full\_query);
   * + - 直接執行SQL語法而沒有預處理(prepare)。

詳見<http://php.net/manual/en/pdo.query.php>

1. **查詢資料範例(inform.php):**

|  |
| --- |
| <?php  include "db\_conn.php";  $no = "00557888";  $query = ("select \* from student where no = ?");  $stmt = $db->prepare($query);  $error = $stmt->execute(array($no));  $result = $stmt->fetchAll();  **//以上寫法是為了防止「sql injection」**  for($i=0; $i<count($result); $i++){  echo "dept:".$result [$i]['dept'].' '.  "no:". $result [$i]['no'].' '.  "name:".$result[$i]['name'].' '.  "club:". $result[$i]['club'].' '.  '<br>';  　　}  **//一般直接執行而沒有預處理的寫法**  $query = ("select \* from student where no = ‘00557888’");  $stmt = $db->query($query);  $result = $stmt->fetchAll();  ?> |

查詢資料所使用到的物件/API/變數如下:

* + - $result為SQL執行結果集合。

1. $result = $stmt-> fetchAll();。

回傳物件，將$stmt的資料取出來存成陣列，若無資料則回傳NULL。

除了單純的fetchAll()之外，還有PDO::FETCH\_ASSOC、PDO::FETCH\_ASSOC、PDO::FETCH\_OBJ等其他用法，詳閱<http://php.net/manual/en/pdostatement.fetchall.php>。

1. $result[i]為結果的一列，之後的['dept']、 ['no'] 、['name'] 、['club']為資料表的欄位。
2. **刪除資料範例(delete.php):**

刪除或修改資料的方式大致如同insert，以下顯示兩種方式刪除資料。

|  |
| --- |
| <?php  include "db\_conn.php";    **//使用prepare的寫法**  $stmt = $db->prepare("delete from student where no=?");  $no = "00557888";  $result = $stmt->execute(array($no));  **//直接刪除的寫法**  $result = $db->exec  ("delete from student where no='00557888'");  ?> |

1. $db->exec("delete from student where no='00557888'")

* 直接執行SQL語法而沒有預處理(prepare)。

詳見<http://php.net/manual/en/pdo.exec.php>

1. **修改資料範例(update.php):**

|  |
| --- |
| <?php  include\_once "db\_conn.php";  $dept = "資工";  $no = "00557000";  $name = "孫小美";  $club = "熱音社";  $query = ("update student set dept=?,name=?,club=? where no=?";  $stmt = $db->prepare($query);  $result = $stmt->execute(array($dept,$name,$club,$no));  ?> |

修改資料的方式也大致如同insert，以下顯示prepare的寫法。

最後說明PDO::query()、PDO::exec()以及PDO::execute()的差別。PDO::query執行一條SQL語句，如果通過，則回傳一個PDOStatement物件，用在SELECT語句執行上較合適。PDO::exec執行一條SQL語句，並回傳受影響的行數，不會回傳結果集合，用在INSERT, DELETE,UPDATE語句執行上較合適。例如我們可以這樣用：

|  |
| --- |
| <?php  include\_once "db\_conn.php";  /\* Delete all rows from the FRUIT table \*/  $count = $dbh->**exec**("DELETE FROM fruit");  /\* Return number of rows that were deleted \*/  print("Deleted $count rows.\n");  ?> |

結果就會顯示：

|  |
| --- |
| Deleted **1** rows. |

至於PDOStatement::execute是用於執行已經**預處理**過的語句，回傳值為TRUE or FALSE，對應到執行結果成功或失敗。使用execute()與prepare()不但可以防止SQL injection，而且各種SQL語法皆可使用，是較為直觀又兼顧安全的方法。

# 網頁程式範例

1. 範例1：下圖為index.html的程式碼，在HTML中呼叫PHP程式



1. **[第6行]**center標籤: 在瀏覽器上執行會自動將網頁置中，通常放於<html>中的</head>之後。
2. **[第12-27行]**Form提供畫面讓使用者輸入，action代表動作，須將送出資料後要執行的php程式寫在action中，利用post方式將變數帶過去該php檔。
3. **[第17行]**在body中直接輸入文字 如「學號: 」，就會在網頁上直接顯示；而「name = “no”」是在指定對應輸入資料的變數為no (變數名稱可以自己取)，而type為text則代表輸入的是文字會存在no這個變數內。

\*\*注意: HTML語法中，如果在屬性值和引號中間參雜空格，有可能會導致變數無法被正確宣告而出了錯誤。

舉例: name=" no " 這樣就會出錯。

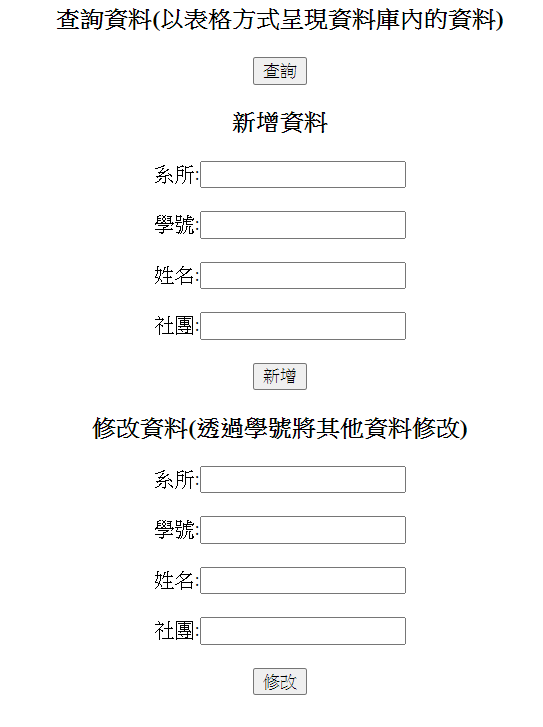
1. **[第26行]**<input type="submit" value="新增">

意思為一個按鈕，而value中就是放顯示在按鈕上的字串。

\*\*注意: HTML語言為巢狀式，先出現的標籤就必須最後結束，不可以把開始、結束標籤交叉擺放，會執行出錯誤結果。

1. **[第8行]**get 的用法與 post很類似，但有個差異是post透過表單傳值，get 透過網址傳値，也就是當送出表單後，會把資料透過網址傳遞到 PHP 取得資料的程式

執行結果：



1. 範例2：下圖為inform.php的程式碼，以表格方式將資料庫內的資料呈現在網頁上。

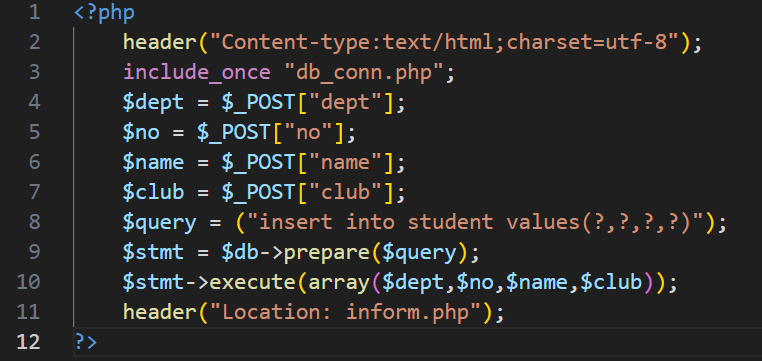


1. **[第26行]** history.back()為網頁中”Go Back”按鈕的觸發事件，觸發(即按下)後能使網頁回到上一頁。

執行結果 :



1. 範例3：下圖為insert.php的程式碼，此程式會接受使用者輸入的資料，並跳轉至inform.php。



* + - 1. **[第4行]**$\_POST["dept"]接收以POST方式傳送過來的資訊。此範例會將dept、no、name、club傳到insert.php中
      2. **[第11行]**header('Location: inform.php')在執行完畢後轉址至查詢頁面。

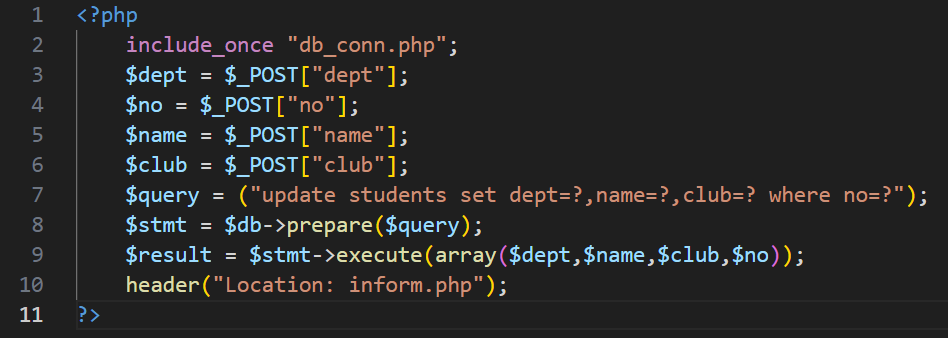
輸入畫面 :



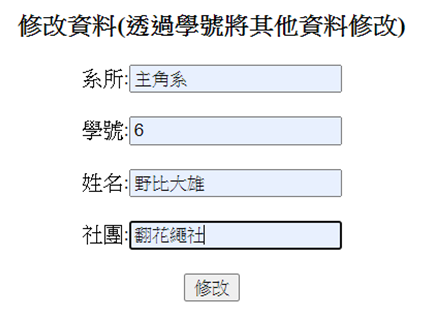
執行結果:

|  |  |
| --- | --- |
| 執行前 | 執行後 |
|  |  |

1. 範例4 :下圖為update.php的程式碼，以表格方式將資料庫內的資料呈現在網頁上。



輸入畫面:



執行結果:

|  |  |
| --- | --- |
| 執行前 | 執行後 |
|  |  |

更多PHP片段程式碼範例以及功能請參考以下網址:

<http://php.net/manual/en/book.pdo.php>

# 檢查資料庫回傳的錯誤訊息

在第五章的連接資料庫範例中，使用到

setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION)

這個設定可主動拋出 exceptions 異常。若發生異常後，異常資訊不會直接輸出，而是產生PDOException 物件，再透過PHP提供的一些函數來取得異常。在這章的範例中我們使用getMessage()來取得異常訊息內容，需要以try{}catch(){}輸出異常資訊。

更多處理異常的函數請參見以下網頁:

<https://www.php.net/manual/en/class.exception.php>或參見第十章。

我們沿用第四章的project資料庫裡students資料表(id為primary key)，Schema為：



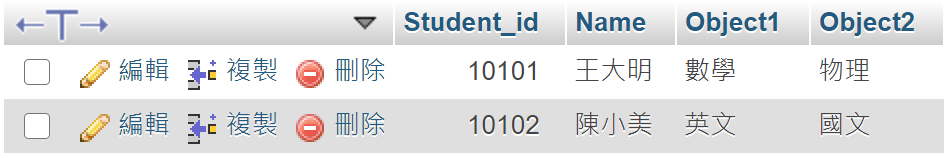
並新增兩列資訊:



另外在project資料庫下建立class資料表，student\_id為primary key， Name為foreign key，且Name要reference到students的Name(詳細設定方法請看第四章)，Schema為:



並新增兩列資訊:



假設我們的PHP程式如下：

|  |
| --- |
| **<?php**  //連接資料庫同第五章的範例  try{  $stmt = $db->prepare($query); // $query為要使用的SQL指令  $result = $stmt->execute();//執行SQL語法  echo "Success";  }catch(PDOException $e){ //若上述程式碼出現錯誤，便會執行以下動作  Print "getMessage(): " . $e->getMessage();  }  **?>** |

1. 假設students表單輸入相同的primary key的值。

EX：$query為以下命令：

|  |
| --- |
| "insert into students values('2','陳阿明','0955555555','Taipei')" |

則輸出的訊息如下：

getMessage(): SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry '2' for key 'PRIMARY'

1. 假設class表單的foreign key對應不到students的foreign key。

EX：$query為以下命令：

|  |
| --- |
| "insert into class values('10104','王大偉','化學','地球科學')" |

則輸出的訊息如下：

getMessage(): SQLSTATE[23000]: Integrity constraint violation: 1452 Cannot add or update a child row: a foreign key constraint fails (`project`.`class`, CONSTRAINT `class\_ibfk\_1` FOREIGN KEY (`Name`) REFERENCES `students` (`Name`

# 日期的處理

若想要在網頁中取得使用者登入或者留言的時間，則所設計的資料庫亦要有時間欄位，MariaDB的時間欄位的資料型別如下：

|  |  |
| --- | --- |
| 資料型態 | 意義 |
| DATE | 日期，以下列幾種方式表示：  YYYY-MM-DD、YY-MM-DD、YYYYMMDD、YYMMDD |
| TIME | 時間，以下列幾種方式表示：  HH:MM:SS.ssssss、HH:MM:SS，以及HHMMSS |
| DATETIME | 以YYYY-MM-DD HH:MM:SS或YY-MM-DD HH:MM:SS表示 |
| TIMESTAMP | 時間戳，以YYYY-MM-DD HH:MM:SS表示 |
| YEAR | 年，以YYYY表示 |

在此範例中，假設要建立一個留言版，能夠記錄發言時間、使用者名稱、主題、留言內容，以datetime為例，建立一新的資料表comment，其Schema為：



接下來，使用『date( )』函數抓取當下時間，所撰寫的php如下：

<?php

include "db\_conn.php";

date\_default\_timezone\_set('Asia/Taipei');

$dt = date("Y-m-d H:i:s");

//date的格式詳見下方說明

$query="insert into comment values(?,?,?,?)";

$stmt = $db->prepare($query);

$stmt->execute(array($dt,'Joe','DB Class','TEST123'));

?>

其中，我們先設定時區，然後將date的資料暫存於“$dt”變數中，$stmt=$db->prepare指令先預處理query，然後$stmt->execute(array(…))，執行完畢後查詢comment資料表的內容，可以發現，時間、留言者、主題、留言內容的欄位已被寫入了。



為了避免混淆，在此特別說明。表格中所列的date是MariaDB的格式，而程式碼中的

**$dt = date("Y-m-d H:i:s");**

是PHP的用法，Y年m月d日 H時i分s秒，而大小寫也代表不同意義。

關於date()的各種用法請參考<http://php.net/manual/en/function.date.php>

# Session的應用與介紹

Session 負責記錄在Server端上的使用者訊息，會在一個用戶完成身分認證後，存下所需的用戶資料在Server端，而不是Client端，即使當使用者的電腦遭竊取，也無法取得Session的相關資料。以下範例的SESSION只要伺服器維護方有做控管維護，使用者就不需擔心資料被竊取與盜用。

**第一步、啟用 session ： session\_start()**  
在你使用 session 來記錄用戶的資訊前，要先用 session\_start()這個函式，告訴系統準備開始使用，請記住 session\_start() 一定要放在網頁的最上方還沒有輸出任何東西之前，也就是類似下面的範例，如果前面出現了任何的輸出可是會出錯的唷！這也是一般人常碰到的問題。

**第二步、給變數值並記錄在伺服器上**  
我們設定了一個account的變數，並給他一個值為"james"，接著這個 account= james的session設定就存在伺服器上了。

<?php

/\* 假設先前已經有判斷是否為正確的帳號密碼 \*/

session\_start(); //此變數是存在Server端內

$\_SESSION['account']="james"; //此變數是存在Server端內

?>

**第三步、取得 session 的值，使用 $\_SESSION["變數名稱"];**

<?

if($\_SESSION['account']==True)

echo "你的帳號是".$\_SESSION ['account'];

else

echo "尚未登入";

?>

**第四步、刪除 session**  
其實預設伺服器會自動刪除超過有效時間的 session，但有的情況是必須讓用戶操作的時候可以自己刪除，像是會員登出，希望做到主動清除後馬上變成登出狀態，這個時候可以使用以下兩種方式清除：

* **unset($\_SESSION['變數名稱']); //**單獨刪除該筆記錄
* **session\_destroy();** //將全部的session清除

# Q&A

1. 兩種不同的Web Server ，IIS和Apache有辦法併存嗎？

A：IIS和Apache都是以Port：80為預設值，只需要把Port值設定成不同值即可，不過通常都會選擇其中一個Web Server使用，請參照附錄B的說明。

(Ex：Apache的Port：80然後把IIS的Port設定成其他值即可)

1. PHP如何check error

A：PHP的語法比較自由、寬鬆，並不像其他程式語言那麼嚴謹，若是要檢查哪一行指令有語意錯誤(semantic error)或語法錯誤(syntax error)有下列三個方法來檢查錯誤。

1. 把PHP檔放到XAMPP路徑C:\xampp\底下的htdocs資料夾，再用瀏覽器去執行即可。
2. 使用dreamweaver編輯PHP，即可預覽顯示。
3. 關於處理資料庫的錯誤訊息、代碼，除了第七章的方法外，若不用Exception的做法也可使用以下API，如: PDO::errorinfo()、PDO::errorCode()、PDOStatement::errorInfo、PDOStatement::errorCode。非Statement的API都是使用在資料庫連線居多，而Statement則是用在SQL指令上。

詳細用法請見<http://php.net/manual/en/pdo.errorinfo.php>、

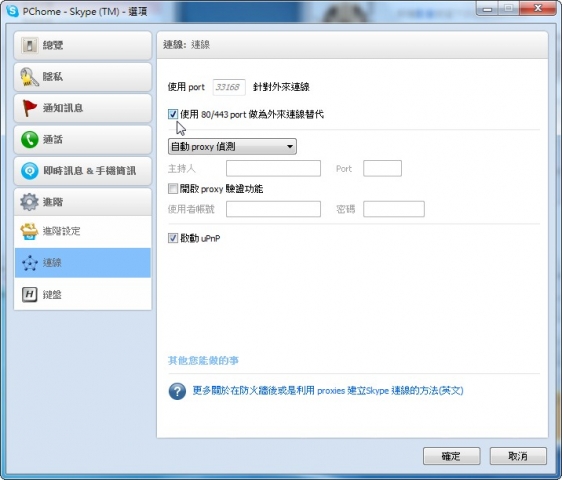
<http://php.net/manual/en/pdo.errorcode.php>、

<http://php.net/manual/en/pdostatement.errorinfo.php>、

<http://php.net/manual/en/pdostatement.errorcode.php>。

1. 開啟skype時會造成Apache的http://localhost無法連入使用?

A：因為skype會佔用80/443 Port來對外連線，把預設項目關閉即可同時使用skype及 XAMPP Control Panel。



# **附錄A: 編輯工具**

* Notepad++

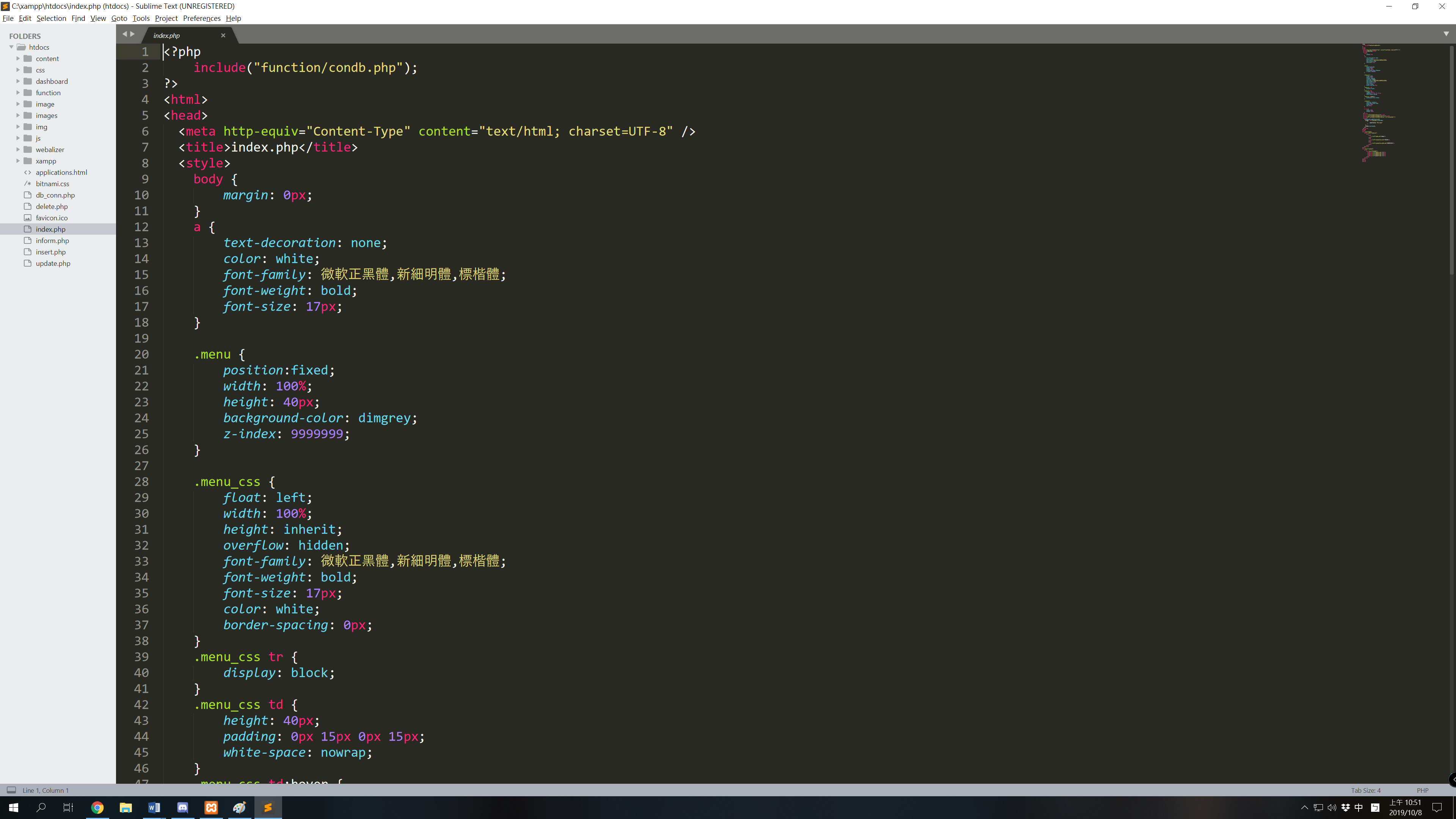
輔助小工具Notepad++，可取代記事本，可幫PHP縮排，標顏色，體積輕巧不佔系統記憶體，支援多分頁功能。

官方網站：<http://notepad-plus-plus.org/zh/>



* Sublime Text 3

為網頁開發者所使用的軟體，支援大部分的網頁語言編輯撰寫功能。

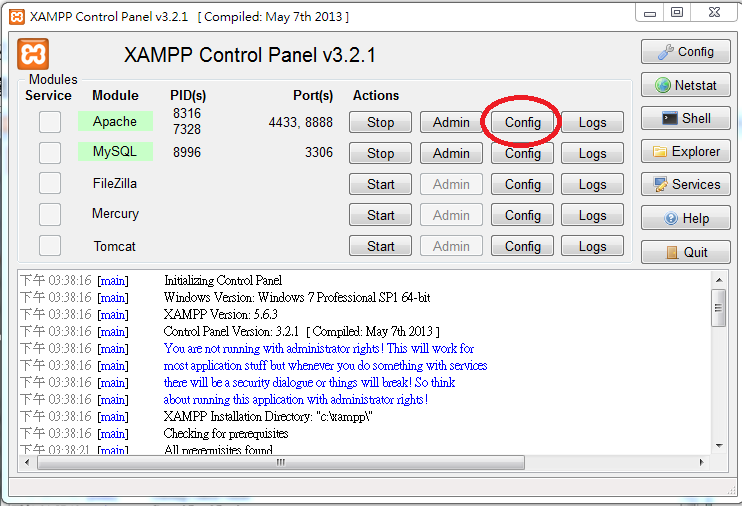
官方網站：<https://www.sublimetext.com/>

# **附錄B: 安裝XAMPP的問題和解決方法**

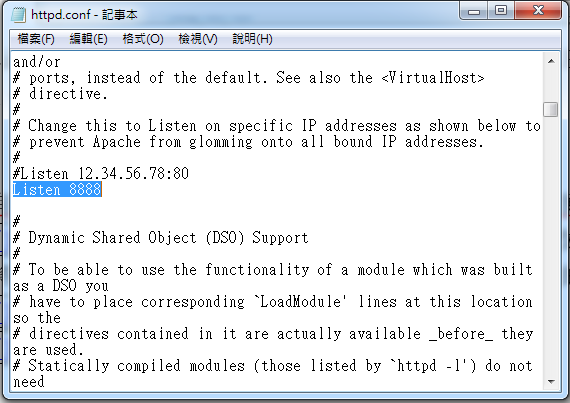
安裝時會有錯誤常常是因為Apache的Port:80被占據而無法啟動，請參照以下方式處理：

* 方法一：

1. 開啟XAMPP Control Panel，點選「Config」按鈕之後選擇「Apache(httpd.conf)」選項。



1. 將#Listen 12.34.56.78:80下的Listen改為8888或其他Port，就能排除問題。



注意:如有更改Port，連上系統網址皆須加上自己更改的Port，例如原網址為 <http://localhost>，若將Port改為8888，在此須將網址改為[http://localhost:8](http://localhost:8080)888，同樣，若要前往的網址為<http://localhost/phpmyadmin>，網址則須改為<http://localhost:8888/phpmyadmin>。

附註: 不能安裝多個xampp，會有錯誤。

* 方法二：若是和Windows內建的IIS衝突(都是80 port)，可至[控制台]->[系統管理工具]->[電腦管理]中的[服務與應用程式]執行以下步驟
  + 若是Windows 8以前的系統，選擇[Internet Information Service (IIS)管理員]，停止服務後重啟Apache。
  + 若是Windows 10出現以下錯誤訊息:

Error: Apache shutdown unexpectedly.  
This may be due to a blocked port, missing dependencies,  
improper privileges, a crash, or a shutdown by another method.Check the “/xampp/apache/logs/error.log” file  
and the Windows Event Viewer for more clues

則是在服務中選[World Wide Web Publishing Service Properties]，將此服務停止，或者至[控制台]->[程式集]->[程式和功能]->[開啟或關閉Windows功能]，關閉[Internet Information Service]功能，再重啟即可，如果沒有在服務中尋找到[World Wide Web Publishing Service Properties]，嘗試安裝Visual C++ Redistributable並且重新安裝XAMPP。

