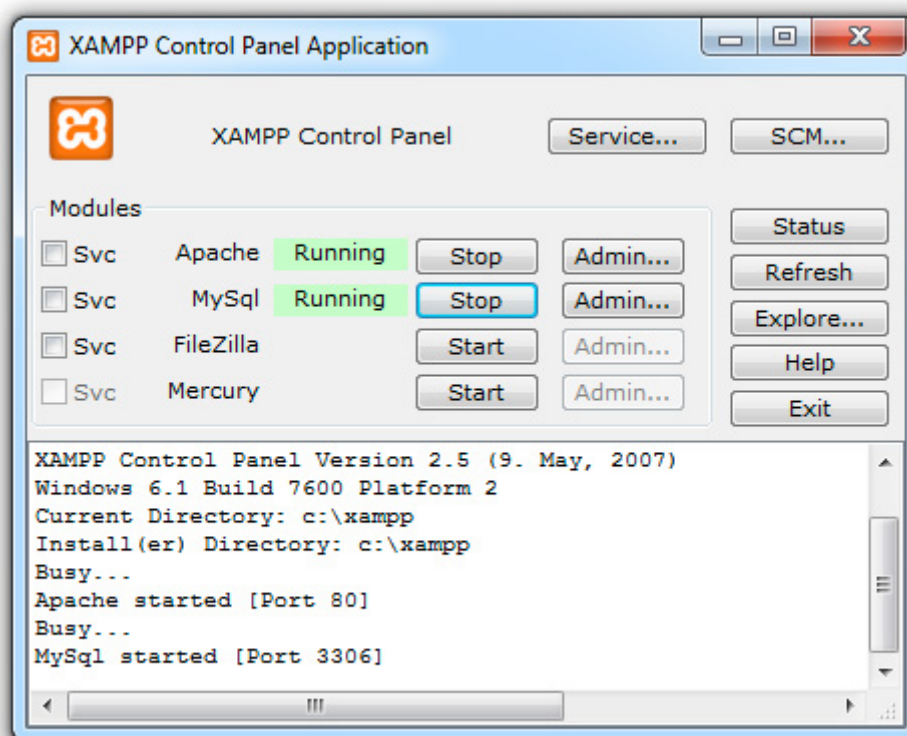


一. 下載及安裝 XAMPP 說明

1. 連結http://www.apachefriends.org/zh_tw/xampp-windows.html#1934下載 XAMPP 程式。
2. 亦可參考 XAMPP 網站http://www.apachefriends.org/zh_tw/xampp.html。
3. 原則上所有安裝過程皆使用預設值即可，管理者的名稱及密碼，同學可自行輸入。依照網頁所給予的指示安裝即可。
4. 安裝完成後，直接啟動 Apache 及 MySQL 即可，啟動成功後，右下角工作列會有橙色圖示。



XAMPP win32 版本的安裝小助手安裝完成後，你可以在[開始/程式集/XAMPP]下找到 XAMPP。在 XAMPP 的控制面板中，你可以啟動/停止所有的伺服器，以及安裝/移除這些伺服器的系統服務。

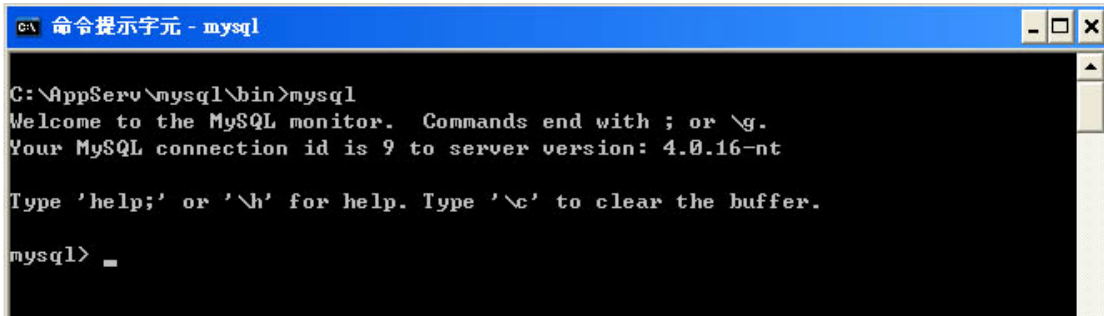


5. 無論是 html 或是 php 文件，都是純文字文件格式，同學可自行使用 Windows 所附的記事本或是使用 UltraEdit 等文件編輯器。

二. 設定與測試 PHP、MySQL 環境

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

1. 至 C:\xampp\mysql\bin(預設路徑)，並打開 mysql.exe 進入以下畫面。
2. 也可用命令提示字元開啓。首先，打開命令提示字元，並到達 C:\xampp\mysql\bin(預設路徑)，執行 MySQL 指令，即可進入 SQL 指令模式。

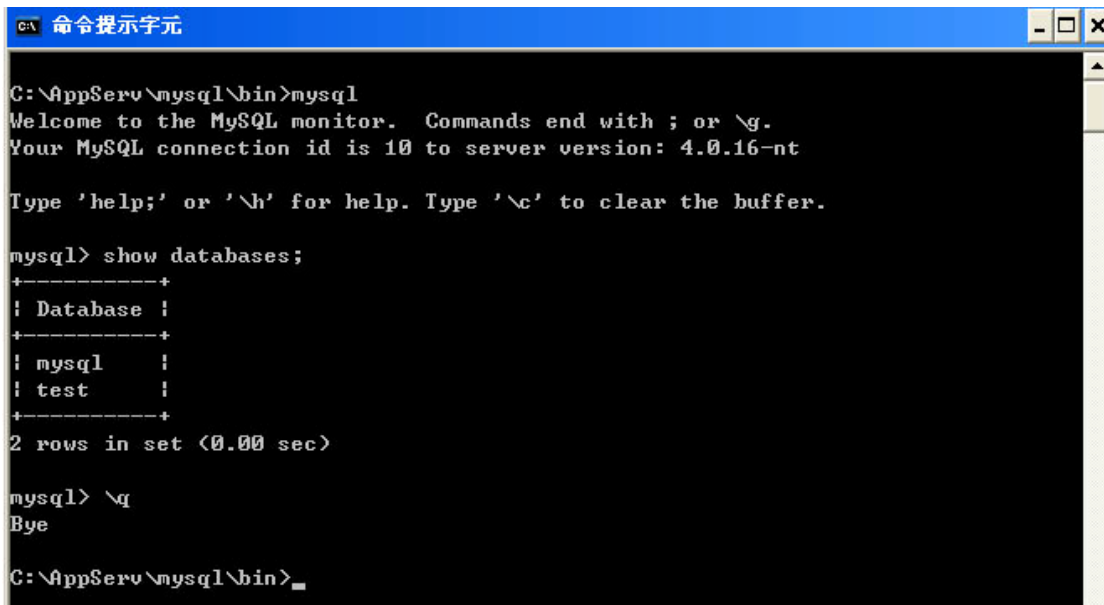


```
C:\AppServ\mysql\bin>mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9 to server version: 4.0.16-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> _
```

3. 輸入 show databases; 可顯示目前資料庫有哪些，可看到 mysql 及 test 二個預設資料庫，則表示 MySQL 正常運作，輸入 \q 可以離開 MySQL 環境。



```
C:\AppServ\mysql\bin>mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10 to server version: 4.0.16-nt

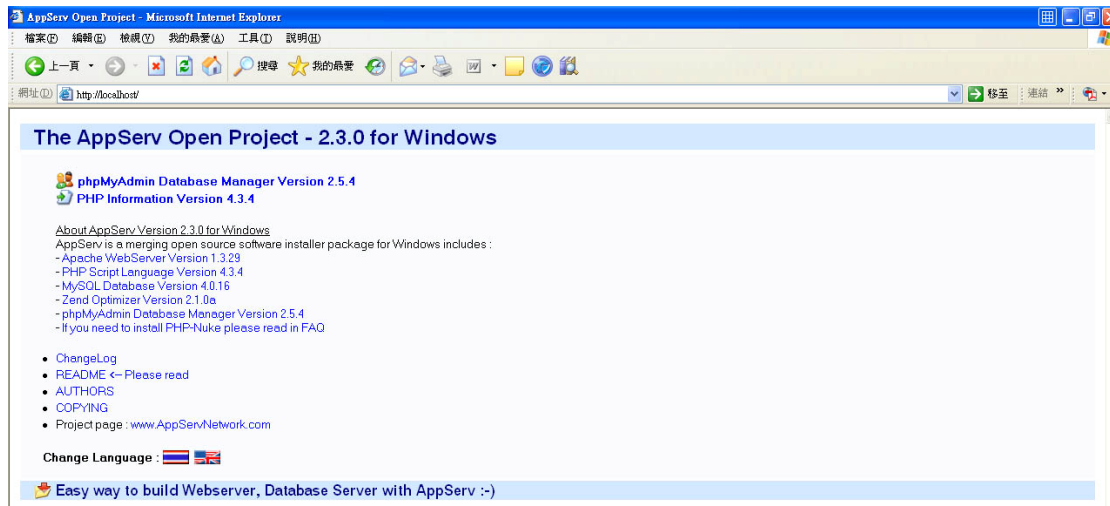
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| mysql    |
| test     |
+-----+
2 rows in set (0.00 sec)

mysql> \q
Bye

C:\AppServ\mysql\bin> _
```

4. 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。



5. 我們亦可使用文書編輯器撰寫一簡單的 PHP 網頁，置放於本機的 localhost 中執行，如此同學就能夠架設資料庫的網站了。

接下來我們做個簡單的網頁：

(1) 打開任意文書編輯程式，輸入下列程式碼

```
<head>

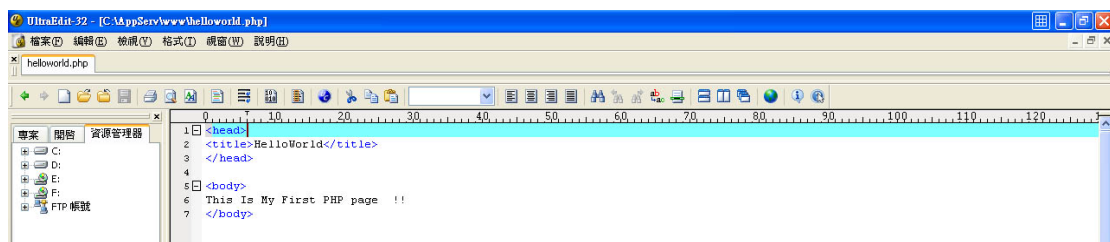
<title>HelloWorld</title>

</head>

<body>

This Is My First PHP page !!

</body>
```



(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。



三. MySQL 資料庫基本操作

1. 建立、顯示、刪除與使用資料庫：

(1) 在 `mysql>` 中輸入

```
create database mytest;
```

即可建立一個 “mytest” 資料庫

(2) 在 `mysql>` 中輸入

```
show databases;
```

即可顯示目前資料庫狀況

A screenshot of a Windows command prompt window titled "命令提示字元 - mysql". The prompt shows the following text:

```
C:\AppServ\mysql\bin>mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 4.0.16-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database mytest;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| mysql    |
| mytest   |
| test     |
+-----+
3 rows in set (0.00 sec)
```

(3)若我們再建立一個 mytset2 資料庫，並要刪除該資料庫，則輸入
drop database mytest2;

即可刪除目前 mytest2 資料庫

```
mysql> create database mytest2;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| mysql    |
| mytest   |
| mytest2  |
| test     |
+-----+
4 rows in set (0.00 sec)

mysql> drop database mytest2;
Query OK, 0 rows affected (0.08 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| mysql    |
| mytest   |
| test     |
+-----+
3 rows in set (0.00 sec)
```

(4)由於在 MySQL 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，現在假設我們使用已經建立好名為“mytest”的資料庫，則輸入。

Use mytest;

```
mysql> use mytest;
Database changed
mysql>
```

四. 資料表基本操作

1. 建立資料表的描述如下：

```
Create table 資料表名稱(  
欄位名稱 1 資料型別,           //加上『,』逗號分隔其他欄位  
欄位名稱 2 資料型別,  
.....  
欄位名稱 n 資料型別           //最後一個欄位記得不要加『,』逗號  
);
```

```
mysql> use mytest;  
Database changed  
mysql> create table t1< //建立t1資料表  
-> no integer,         //[座號],integer型別  
-> name char(8),       //[姓名],char型別,8個字元  
-> class1 integer,     //[科目1],integer型別  
-> class2 integer,     //[科目2],integer型別  
-> class3 integer      //[科目3],integer型別  
-> );  
Query OK, 0 rows affected (0.03 sec)
```

2. 資料表的查詢

使用

show tables; 可查詢資料庫中的資料表狀態

show columns from t1; 可顯示資料表定義

```
mysql> show tables;  
+-----+  
| Tables_in_mytest |  
+-----+  
| t1                |  
+-----+  
1 row in set (0.00 sec)  
  
mysql> show columns from t1;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type  | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| no    | int(11) | YES |     | NULL    |       |  
| name  | char(8) | YES |     | NULL    |       |  
| class1 | int(11) | YES |     | NULL    |       |  
| class2 | int(11) | YES |     | NULL    |       |  
| class3 | int(11) | YES |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.02 sec)
```

3. 資料表的刪除

執行

drop table t2;

可以刪除 t2 資料表(t1 資料表先不要刪除)

```
mysql> create table t2(name char(8));
Query OK, 0 rows affected (0.02 sec)

mysql> show columns from t2;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name  | char(8) | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> drop table t2;
Query OK, 0 rows affected (0.00 sec)
mysql> show tables;
+-----+
| Tables_in_mytest |
+-----+
| t1                |
+-----+
1 row in set (0.00 sec)
```

可看到只剩下 t1 資料表

4.Foreign Key 與 Check 語法

(1). Foreign Key: 在建立新的一筆資料時，會 reference 到其他 table 的 primary key。若是沒有該筆資料在 reference 的 table 裡，建立新的資料將會失敗。

如：create table account

```
(account_number char(10),
branch_name char(15),
balance integer,
primary key (account_number),
foreign key (branch_name) references branch )
```

在 account table 中，branch_name 這個欄位 reference 到 branch table 裡的 primary key。假設現在新增一筆資料(A, B, 10)在 account table 裡，由於 B 這個欄位的資料與 branch table 裡的資料有關，若是 B 這筆資料並不存在於 branch table 裡。則新增(A, B, 10)這筆資料將會失敗。

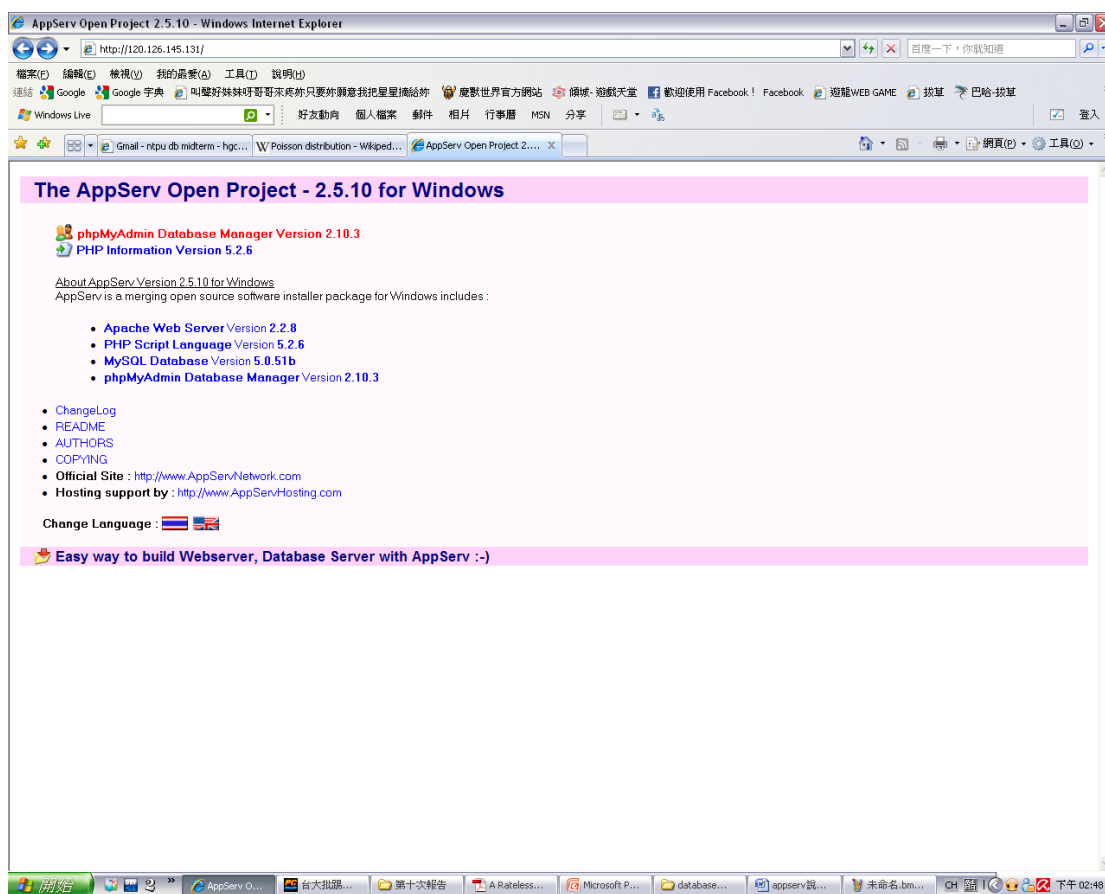
(2). Check: 在建立一筆資料時，會檢查該比資料的某個欄位的值是否在一開始定義的範圍之內。

如：CREATE TABLE Persons

```
(P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CHECK (P_Id>0))
```

在 Persons table 裡，限定 P_Id 必須大於 0，若是輸入的值小於零，則會導致失敗的情況。

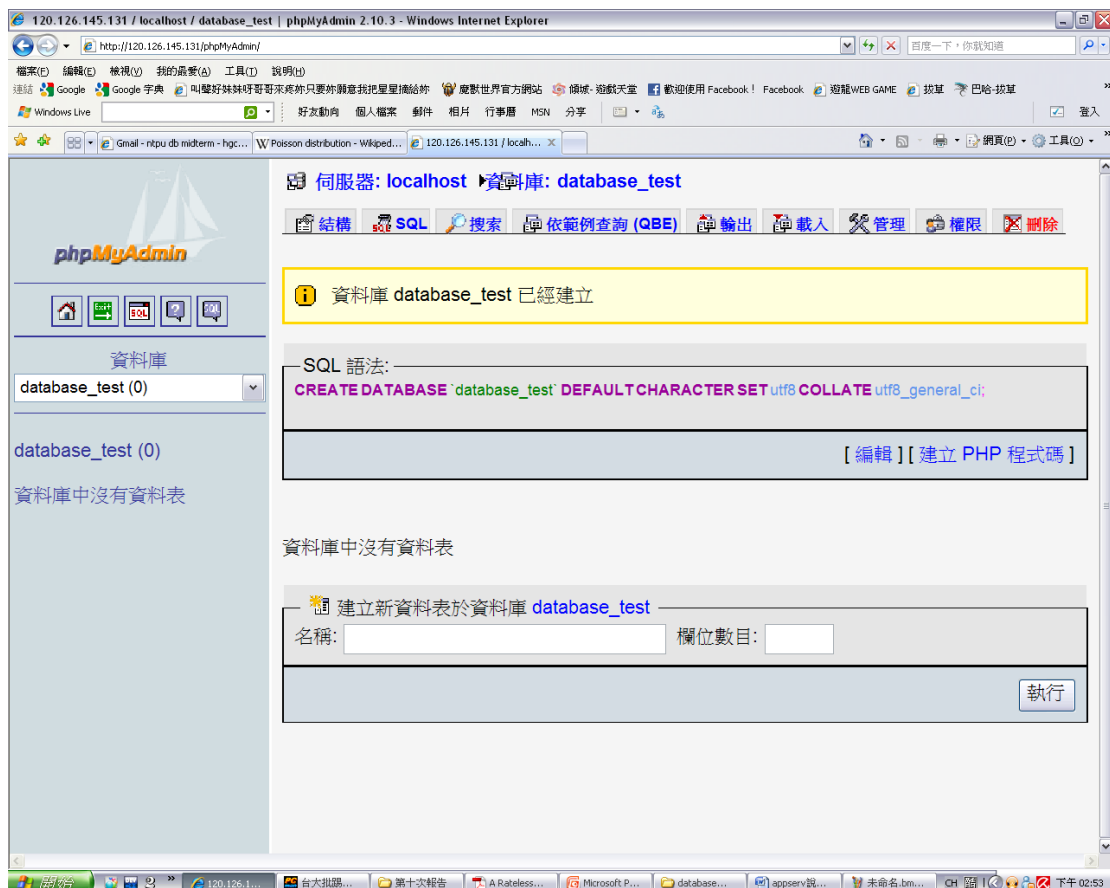
5.使用圖形介面操作



在安裝完 XAMPP 後，在網頁上輸入 localhost 就可登入自身的網頁中。點選紅色區塊部份的連結即可輸入帳號密碼進入資料庫當中。

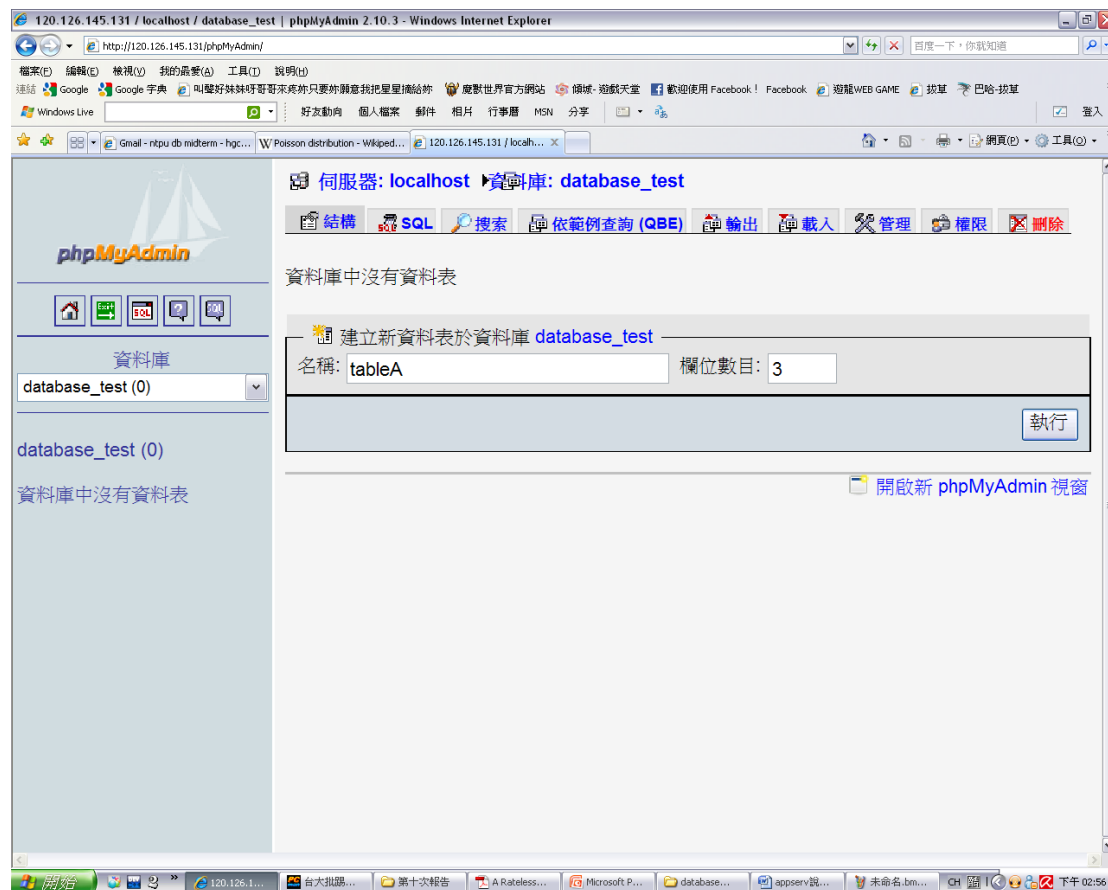


在建立資料庫的地方輸入新的資料庫名稱以及校對的項目。按下建立鍵

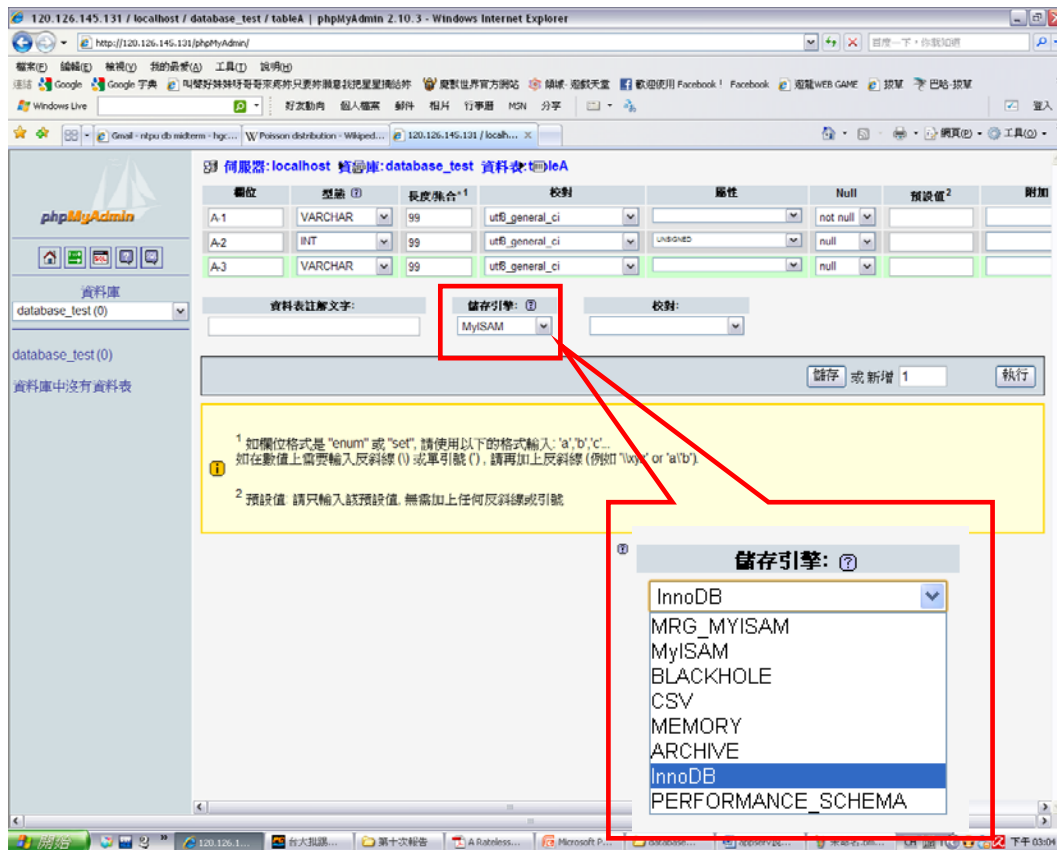


出現資料庫已經建立的畫面。使用者目前就處於新資料庫下。若是想要使用其他資料庫也可以藉由左邊的下拉式選單選取不同的資料庫。

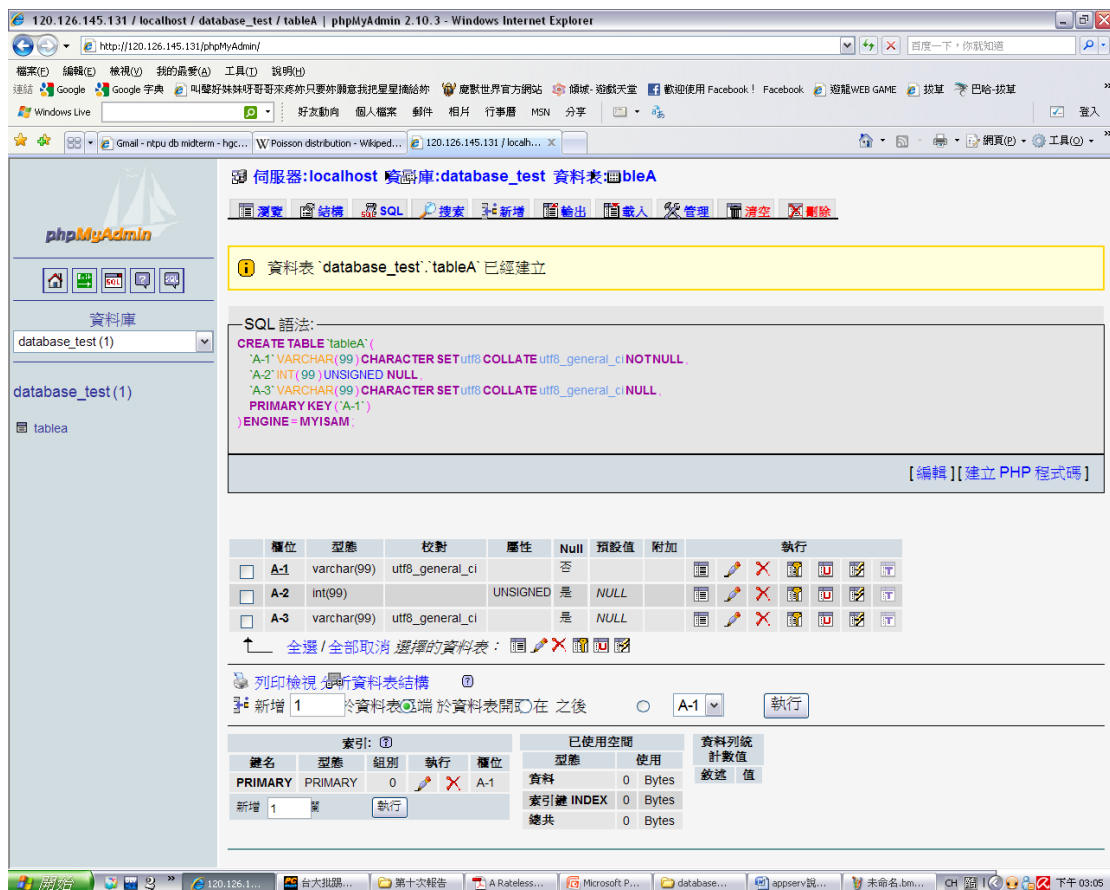
接下來就是在資料庫當中建立新的資料表。選擇結構輸入資料表的名稱以及所需要的欄位數目，接下來按下執行。



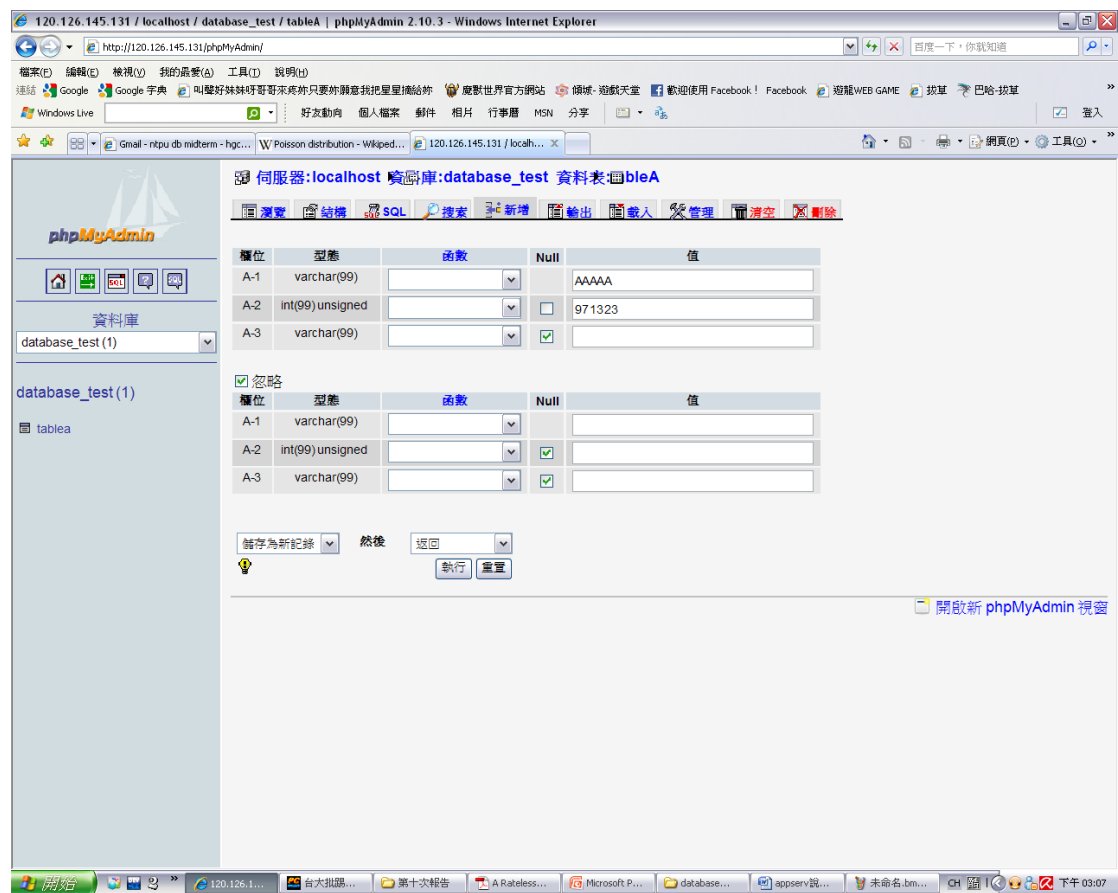
接下來在欄位部份就輸入每一個欄位的名稱(請勿像助教一樣隨意輸入)，型態若是純數字則選擇 INT 型態，若是字元型態則選擇 VARCHAR。屬性的部份 INT 可選擇是正整數或是包含負數。後面也可選擇哪個欄位作為主鍵。



若此資料表有 foreign key 的功能，請在儲存引擎的地方選擇 **innodb** 的選項。
 接下來，選擇儲存建立輸入的資料或是執行來新增新的欄位。

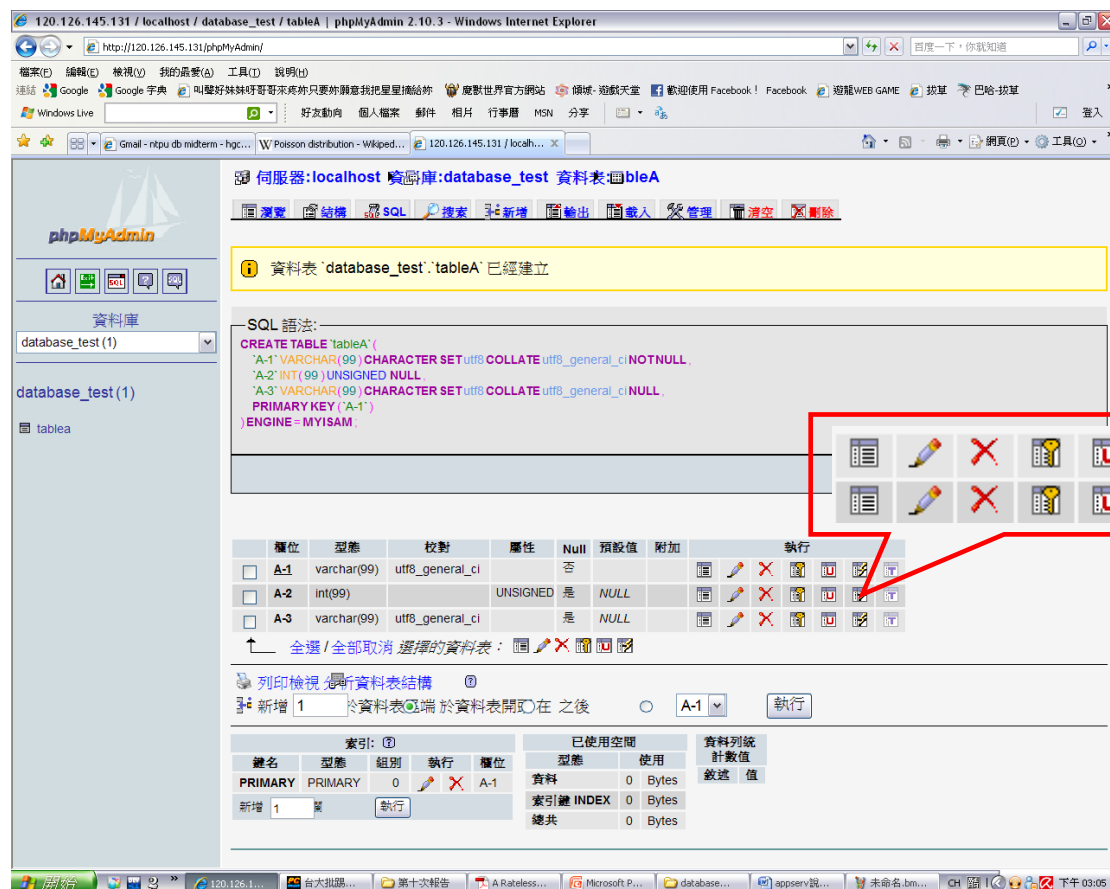


最後，表格 tableA 就順利建立完成。接下來，在這個表格內點選上方的新增按鈕，即可新增一筆資料至這個表格當中。

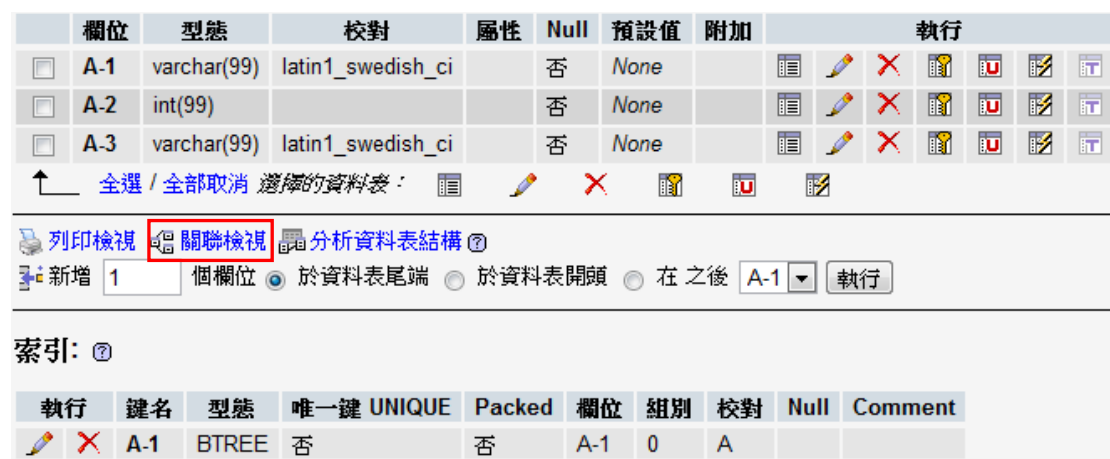


將三個欄位的值輸入完成後，點選執行的按鈕後。再點選上方的瀏覽，即可看見新增的資料已在資料表當中。重複這幾個步驟，即可建立全部的資料表以及資料在這個資料庫當中了。

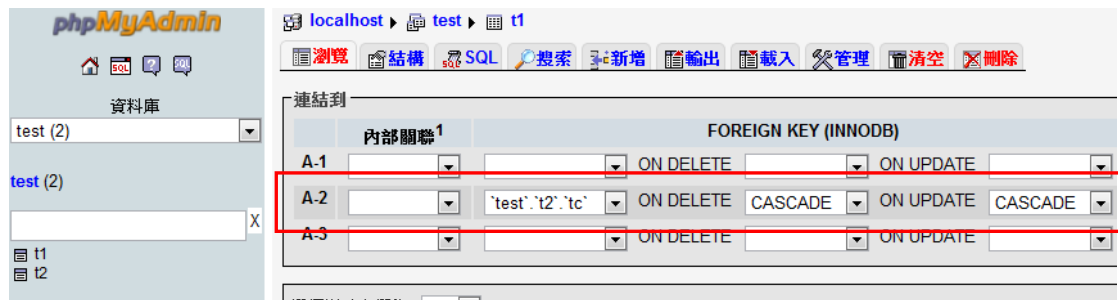
若要在圖形化介面使用 foreign key，先將該資料表的資料庫引擎選為 innodb，接著選定索引欄位，並點擊索引鍵(Index key)



接著點選下方的**關聯檢視**



設定想參考的 key。且設定更改為「CASCADE」，才能順利運作，若沒有設定，原本被連結的 KEY 不能刪除。



如上圖，t1 資料表裡 A-2 的欄位，索引到 t2 資料表中的 tc 欄位，點選儲存後，即完成 foreign key 的設置。

五. 基本資料維護語法

所謂紀錄，即表示資料表中的內容，其更改方式如下：

1. 新增紀錄

要新增紀錄的方式，使用以下語法

```
insert into 資料表名稱(欄位名 1, 欄位名 2, ..., 欄位名 n)
          values( 值 1, 值 2, ..., 值 3);
```

```
mysql> insert into t1 (no,name,class1,class2,class3)
-> values(1,'Tom',80,85,90);
Query OK, 1 row affected (0.03 sec)
```

其中因為姓名的部份是字串，所以前後加上『』單引號。
另外，我們亦可使用下列的精簡指令，以同時輸入多筆資料

```
mysql> insert into t1 values(2,'Mary',70,65,90),(3,'Joe',100,80,95);
Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

2. 查詢紀錄

使用 select 指令，可以查詢資料，語法格式為

```
select * from t1;
```

```
mysql> select * from t1
-> ;
+----+-----+-----+-----+-----+
| no  | name  | class1 | class2 | class3 |
+----+-----+-----+-----+-----+
| 1   | Tom   | 80     | 85     | 90     |
| 2   | Mary  | 70     | 65     | 90     |
| 3   | Joe   | 100    | 80     | 95     |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from t1;
+----+-----+-----+-----+-----+
| no  | name  | class1 | class2 | class3 |
+----+-----+-----+-----+-----+
| 1   | Tom   | 80     | 85     | 90     |
| 2   | Mary  | 70     | 65     | 90     |
| 3   | Joe   | 100    | 80     | 95     |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

而加上 where 條件判斷敘述，可以顯示出所要尋找的特定資料，例如我們要找 class1 成績大於 75 分，可輸入下列判斷式(編號 2 的 Mary class1<75 故被過濾掉了)：

```
mysql> select * from t1;
+----+-----+-----+-----+-----+
| no | name | class1 | class2 | class3 |
+----+-----+-----+-----+-----+
| 1 | Tom | 80 | 85 | 90 |
| 2 | Mary | 70 | 65 | 90 |
| 3 | Joe | 100 | 80 | 95 |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from t1 where class1>75;
+----+-----+-----+-----+-----+
| no | name | class1 | class2 | class3 |
+----+-----+-----+-----+-----+
| 1 | Tom | 80 | 85 | 90 |
| 3 | Joe | 100 | 80 | 95 |
+----+-----+-----+-----+-----+
2 rows in set (0.03 sec)
```

3.更新記錄

要更新記錄時，語法格式為

update 資料表名稱 set 欄位名稱=新的值

例如要將 Mary 成績中 class2 由 65 改成 70，我們可以結合上述的 where 指令作修正

```
mysql> update t1 set class2=70 where name='Mary';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from t1;
+----+-----+-----+-----+-----+
| no | name | class1 | class2 | class3 |
+----+-----+-----+-----+-----+
| 1 | Tom | 80 | 85 | 90 |
| 2 | Mary | 70 | 70 | 90 |
| 3 | Joe | 100 | 80 | 95 |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

(4)刪除記錄

刪除記錄的語法如下

delete from 資料表

不過上述語法會刪除資料表中所有資料，因此在刪除的時候，通常都會結合 where 指令作條件篩選，下例中，我們刪除座號為 3 的資料。

```
mysql> delete from t1 where no=3;
Query OK, 1 row affected (0.02 sec)

mysql> select * from t1;
+----+-----+-----+-----+-----+
| no | name | class1 | class2 | class3 |
+----+-----+-----+-----+-----+
| 1 | Tom | 80 | 85 | 90 |
| 2 | Mary | 70 | 70 | 90 |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```


六. Outer Join 語法

這裡要介紹的是 Left Outer Join 的語法。它的用法是以左邊的資料表作根據建立第一個資料集，並透過左邊資料表的其中一個欄位，逐筆取出右邊資料表某欄位擁有相同的值的資料成爲另一個資料集，最後結合兩個資料集。

(1)利用 Left Outer Join 串連兩個資料表。

標準語法：Select * from 資料表 1 Left Outer Join
資料表 2 On 資料表 1.欄位 1=資料表 2.欄位 2

實例語法：Select * from t1 Left Outer Join
t3 On t1.name=t3.name

假設現有 t1 及 t3 兩個資料表，內容如下：

```
mysql> select * from t1;
+----+-----+-----+-----+-----+
| no  | name  | class1 | class2 | class3 |
+----+-----+-----+-----+-----+
| 1   | Tom   | 80     | 85     | 90     |
| 2   | Mary  | 70     | 70     | 90     |
| 3   | Joe   | 100    | 80     | 95     |
| 4   | Amy   | 60     | 65     | 95     |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> select * from t3;
+-----+-----+
| name  | city  |
+-----+-----+
| Tom   | Taipei |
| Mary  | Tainan |
| Joe   | Sanshia |
| Alex  | Tainan |
+-----+-----+
```

在此使用 Left Outer Join 指令實作如下：

```
mysql> select * from t1 Left Outer Join
-> t3 On t1.name=t3.name;
+----+-----+-----+-----+-----+-----+-----+
| no  | name  | class1 | class2 | class3 | name  | city  |
+----+-----+-----+-----+-----+-----+-----+
| 1   | Tom   | 80     | 85     | 90     | Tom   | Taipei |
| 2   | Mary  | 70     | 70     | 90     | Mary  | Tainan |
| 3   | Joe   | 100    | 80     | 95     | Joe   | Sanshia |
| 4   | Amy   | 60     | 65     | 95     | NULL  | NULL   |
+----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

(2)Right Outer Join，則是跟 Left Outer Join 剛好相反。Right Outer Join 是以右邊的資料表作為根據的，一樣是結合兩個資料集。

標準語法：Select * from 資料表 2 Right Outer Join

資料表 1 On 資料表 2.欄位 2=資料表 1.欄位 1

實例語法：Select * from t1 Right Outer Join

t3 On t1.name=t3.name

這個跟 Left Outer Join 非常相似，只是把關鍵字換成了 Right Outer Join。由於換了關鍵字，變成根據右邊的資料庫串連。

執行結果：

```
mysql> select * from t1 Right Outer Join
-> t3 On t1.name=t3.name;
+-----+-----+-----+-----+-----+-----+-----+
| no    | name  | class1 | class2 | class3 | name  | city   |
+-----+-----+-----+-----+-----+-----+-----+
| 1     | Tom   | 80     | 85     | 90     | Tom   | Taipei |
| 2     | Mary  | 70     | 70     | 90     | Mary  | Tainan |
| 3     | Joe   | 100    | 80     | 95     | Joe   | Sanshia |
| NULL  | NULL  | NULL   | NULL   | NULL   | Alex  | Tainan |
+-----+-----+-----+-----+-----+-----+-----+
```

七. 撰寫 PHP 程式

接下來我們將結合 PHP 與 MySQL，將資料庫應用於網頁當中，在這之前，我們將上述的 t1 資料表建構如下

```
mysql> select * from t1;
+----+-----+-----+-----+-----+
| no  | name  | class1 | class2 | class3 |
+----+-----+-----+-----+-----+
| 1   | Tom   | 80     | 85     | 90     |
| 2   | Mary  | 70     | 70     | 90     |
| 3   | Joe   | 100    | 80     | 95     |
| 4   | Amy   | 60     | 65     | 95     |
| 5   | Alex  | 55     | 70     | 80     |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

1. 新增資料

接下來打開文書編輯器，製作一個 insert.php 的檔案，功能為新增一筆 T1 資料表中的記錄，其內容如下

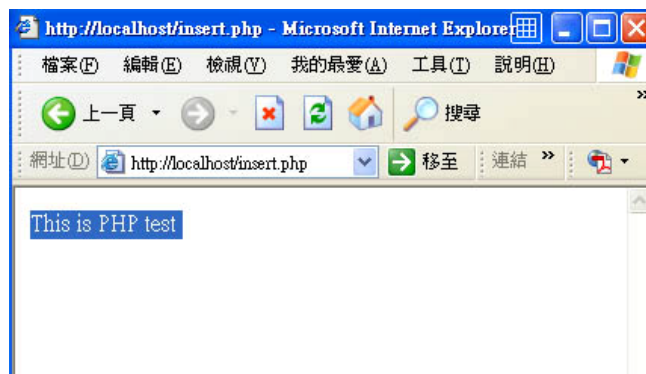
```
<?php
    mysql_select_db("mytest");
    mysql_query ("insert into t1 values (6,'ivan',90,95,100)");
?>
This is PHP test
```

其中 `mysql_select_db("mytest");` 表示所選擇的資料庫
`mysql_query ("insert into t1 values (6,'ivan',90,95,100)");`
為新增一資料的 PHP 語法，而特別注意的是，在撰寫 PHP 時，必須寫在

```
<?php
...
?>
```

的範圍當中，以便和 html 語法分開。

將此檔案存於 “C:\xampp\htdocs\” ，則我們在瀏覽器中輸入 `http://localhost/insert.php` 位址，會顯示以下畫面。



並且在 mysql 環境中查詢資料表，可發現已成功新增資料

```
mysql> select * from t1;
+----+-----+-----+-----+-----+
| no  | name  | class1 | class2 | class3 |
+----+-----+-----+-----+-----+
| 1   | Tom   | 80     | 85     | 90     |
| 2   | Mary  | 70     | 70     | 90     |
| 3   | Joe   | 100    | 80     | 95     |
| 4   | Amy   | 60     | 65     | 95     |
| 5   | Alex  | 55     | 70     | 80     |
| 6   | Ivan  | 90     | 95     | 100    |
+----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

2. 查詢資料

在 PHP 中查詢語法如下

```
$data=mysql_query ("select * from t1");
```

則會將所查詢的資料傳回於 \$data 當中，而要使傳回值能在網頁中顯示，則要使用 mysql_fetch_row() 函數，此函數會將紀錄讀取成陣列的資料格式，再配合 echo 指令(類似 C 語言中的 printf)將訊息列印於螢幕上，我們將此功能撰寫於 search.php 當中，語法如下：

```
<?php
mysql_select_db("mytest");
$data=mysql_query ("select * from t1");
$a=mysql_fetch_row($data);
echo"
    $a[0]
    $a[1]
    $a[2]
    $a[3]
    $a[4]
    ";
?>
```

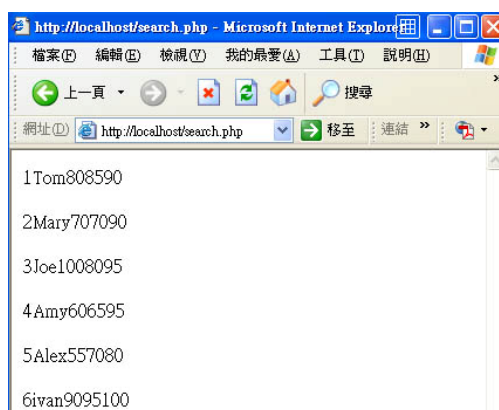
執行結果如下：



若搜尋到多筆資料時，我們可以使用 while 判斷式來執行

```
<?php
mysql_select_db("mytest");
$data=mysql_query ("select * from t1");
while($a=mysql_fetch_row($data))
{
echo"<p> $a[0]$a[1]$a[2]$a[3]$a[4]";
}
?>
```

while(\$a=mysql_fetch_row(\$data))表示若\$data 可以讀取到資料，就繼續執行迴圈，執行結果如下



3. 刪除資料

在 PHP 中刪除語法如下

mysql_query(“delete from 資料表名稱 where 條件式”);

其中條件式一樣可有可無，我們將此功能撰寫於 delete.php 當中，此功能為將座號為 6 之資料刪除：

```
<?php
mysql_select_db("mytest");
mysql_query("delete from t1 where no=6");
?>
```

執行後，MySQL 執行結果如下：

```
mysql> select * from t1;
+----+-----+-----+-----+-----+
| no | name | class1 | class2 | class3 |
+----+-----+-----+-----+-----+
| 1 | Tom | 80 | 85 | 90 |
| 2 | Mary | 70 | 70 | 90 |
| 3 | Joe | 100 | 80 | 95 |
| 4 | Amy | 60 | 65 | 95 |
| 5 | Alex | 55 | 70 | 80 |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

八. 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，MySQL 的時間欄位的資料型別如下：

資料型態	意義
DATE	日期以"YYYY-MM-DD"
TIME	時間以"HH:MM:SS"格式表示
DATETIME	日期與時間以"YYYY-MM-DD HH:MM:SS"格式表示
TIMESTAMP	時間紀錄，以 YYYYMMDDHHMMSS 格式表示
YEAR	以 YYYY 格式表示

在此範例中，假設要建立一個留言板，能夠紀錄發言時間、使用者名稱、主題，並以 datetime 為例，建立一新的資料表 t2

```
mysql> create table t2(  
  -> date datetime,  
  -> name char(60),  
  -> subject char(255))  
  -> ;  
Query OK, 0 rows affected (0.02 sec)
```

其內容為：

```
mysql> show columns from t2;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type      | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| date  | datetime  | YES  |     | NULL    |      |  
| name  | char(60)  | YES  |     | NULL    |      |  
| subject | char(255) | YES  |     | NULL    |      |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.01 sec)
```

接下來，使用『date()』函數撰寫 php：

```
<?php  
  mysql_select_db("mytest");  
  $dt=date("Y-m-d h:i:s");  
  mysql_query("insert into t2 values($dt , 'Joe' , 'DB class')");  
?>
```

其中，我們將 date 的資料暫存於 “\$dt” 變數中，mysql_query 指令裡加入資料，

執行後，查詢 t2 資料表的內容，可以發現，時間、留言者、主題的欄位已被寫入了。

```
mysql> select * from t2;
+-----+-----+-----+
| date           | name  | subject |
+-----+-----+-----+
| 2008-04-14 04:12:30 | Joe   | DB class |
+-----+-----+-----+
1 row in set (0.00 sec)
```

九. 作業上傳方式

PHP為伺服器端執行程式，同學一般使用<http://localhost/> 即是在本機電腦模擬伺服器執行，同學在上傳作業時亦是直接將資料依組別取名(如第一組其資料夾名稱為NO1)，直接傳至已架好的伺服器中，上傳帳號如下：

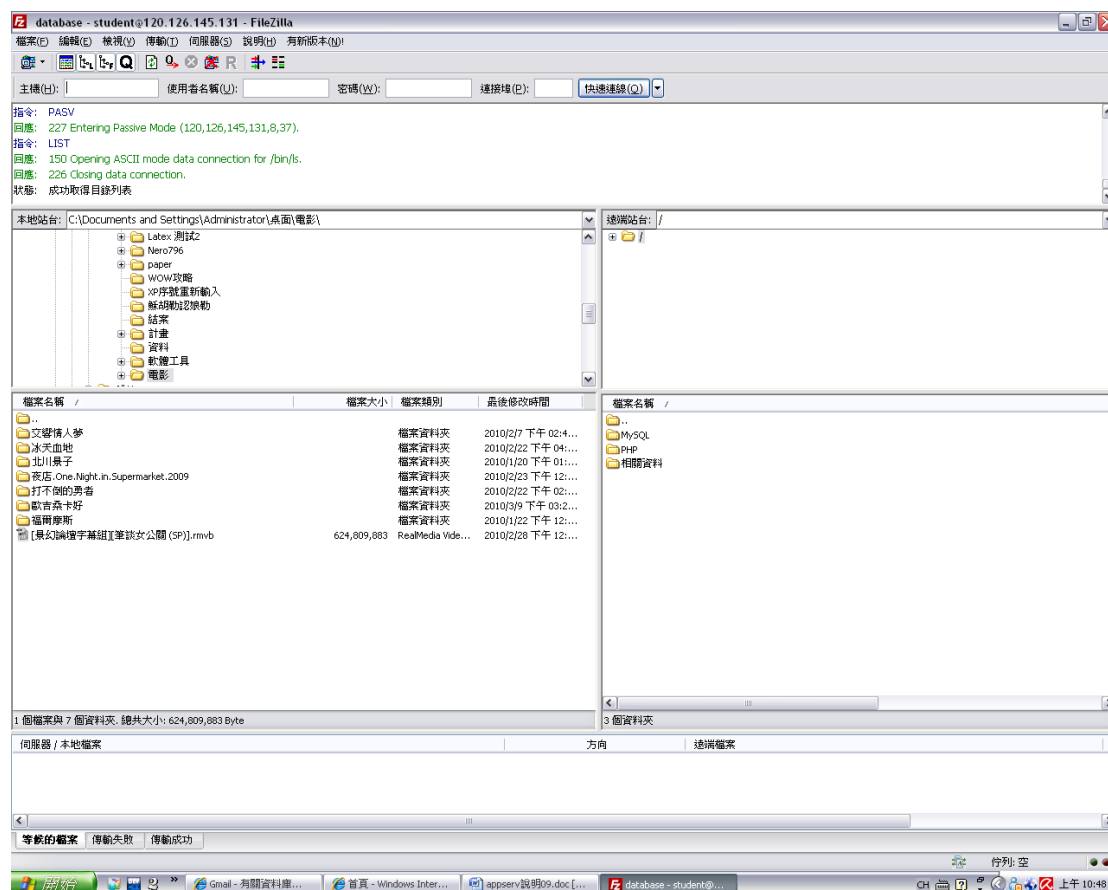
位址：120.126.145.64

帳號：student

密碼：student

port：21(預設)

使用 FTP 連線軟體登入後，裡面會有三個資料夾(在此使用 Filezilla 為例)



裡面會有 MySQL，PHP 以及相關資料三個資料夾，分述如下：

MySQL 資料夾：即為存放資料庫的地方，在同學的電腦中，路徑應該為 C:\xampp\mysql\data\ “資料庫名稱的資料夾”。

PHP 資料夾：即為存放所撰寫完成之 PHP 檔案，在同學的電腦中，路徑應該為 C:\xampp\htdocs\ “所撰寫完成之.php 檔”。

相關資料資料夾：裡面有 php 的範例程式壓縮檔案、appach 的檔案以及實做範例。

而爲了同學繳交作業及管理方便，上傳時請注意下列幾點事項：

1. 檔案上傳一定要資料庫及 php 檔，兩者皆上傳，否則無法正常瀏覽
2. 同學在建立資料庫時，請使用以組別爲名的資料庫名稱，否則班級上有 2 個以上相同名稱的資料庫將無法正常執行，上傳時則將整個資料庫的資料夾上傳。
3. 同學在上傳 php 檔之前，請先把檔案放在以組別爲名的資料夾內，上傳時亦是將整個以組別爲名的資料夾上傳
4. 上傳完成後，可至
<http://120.126.145.64/database/> “組別名稱” / “所撰寫之php檔.php”
查看資料是否正確

範例參考可至

<http://120.126.145.64/Database/php/dbshow.php>

範例下載

http://120.126.145.64:8080/database_sample.rar

將範例下載並解壓縮後，

1. 將 php 資料夾放至 C:\xampp\htdocs\ (預設路徑)
2. 開啓<http://localhost/phpmyadmin>，將 t1.sql 檔匯入至 MySQL 資料庫即可開啓<http://localhost/php/dbshow.php>，並看到網頁內容。

若 t1.sql 的檔案載入失敗，請自行創建資料庫 dbtest，資料表 t1，其它欄位如下圖



欄位	型態	校對	屬性	Null	預設值	附加	執行
<input type="checkbox"/> no	int(3)			否	None	auto_increment	     
<input type="checkbox"/> name	varchar(30)	utf8_bin		否	None		     
<input type="checkbox"/> class1	int(3)			否	None		     
<input type="checkbox"/> class2	int(3)			否	None		     
<input type="checkbox"/> class3	int(3)			否	None		     

MySQL 語法如下。

```
CREATE TABLE IF NOT EXISTS `t1` (  
  `no` int(3) NOT NULL AUTO_INCREMENT,  
  `name` varchar(30) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,  
  `class1` int(3) NOT NULL,  
  `class2` int(3) NOT NULL,  
  `class3` int(3) NOT NULL,  
  PRIMARY KEY (`no`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```