

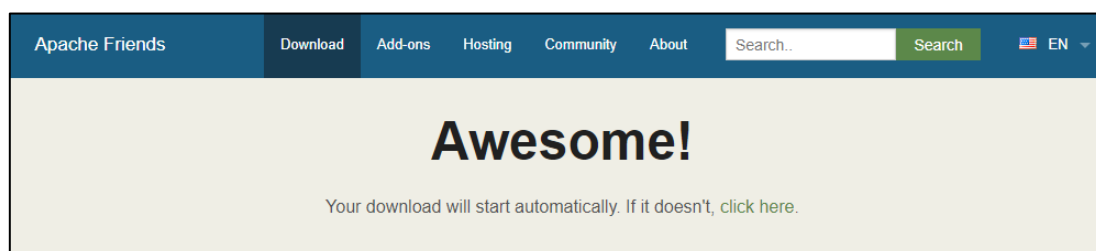
一、XAMPP 安裝

課程中，我們原則上使用 XAMPP 來進行環境安裝與操作。XAMPP 是 Apache、MariaDB、PHP、Perl 等整合在一起的工具，提供 Web、FTP、phpMyAdmin 等服務，Windows、Linux、Mac 環境都能使用。

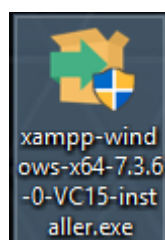
網址：<https://www.apachefriends.org>



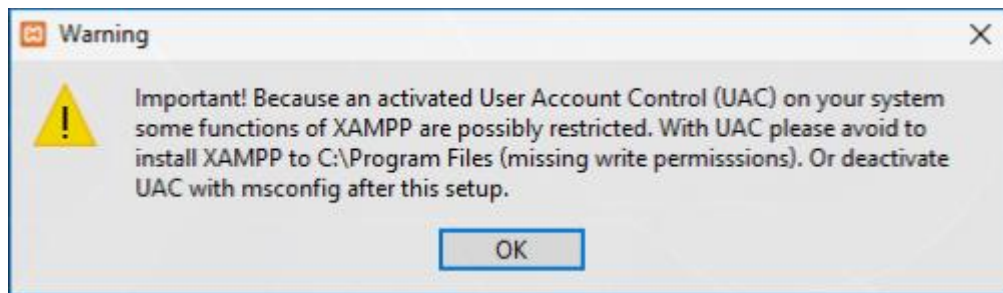
（圖）可下載不同作業系統版本，也可以切換語系



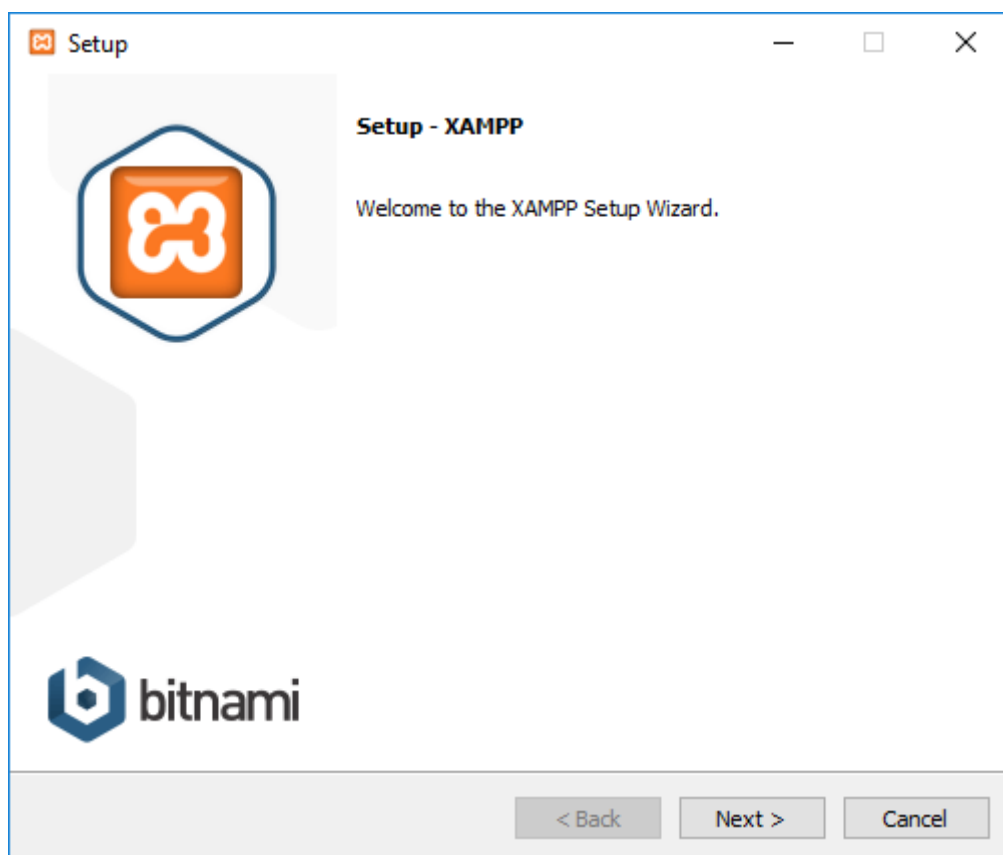
（圖）等待下載畫面



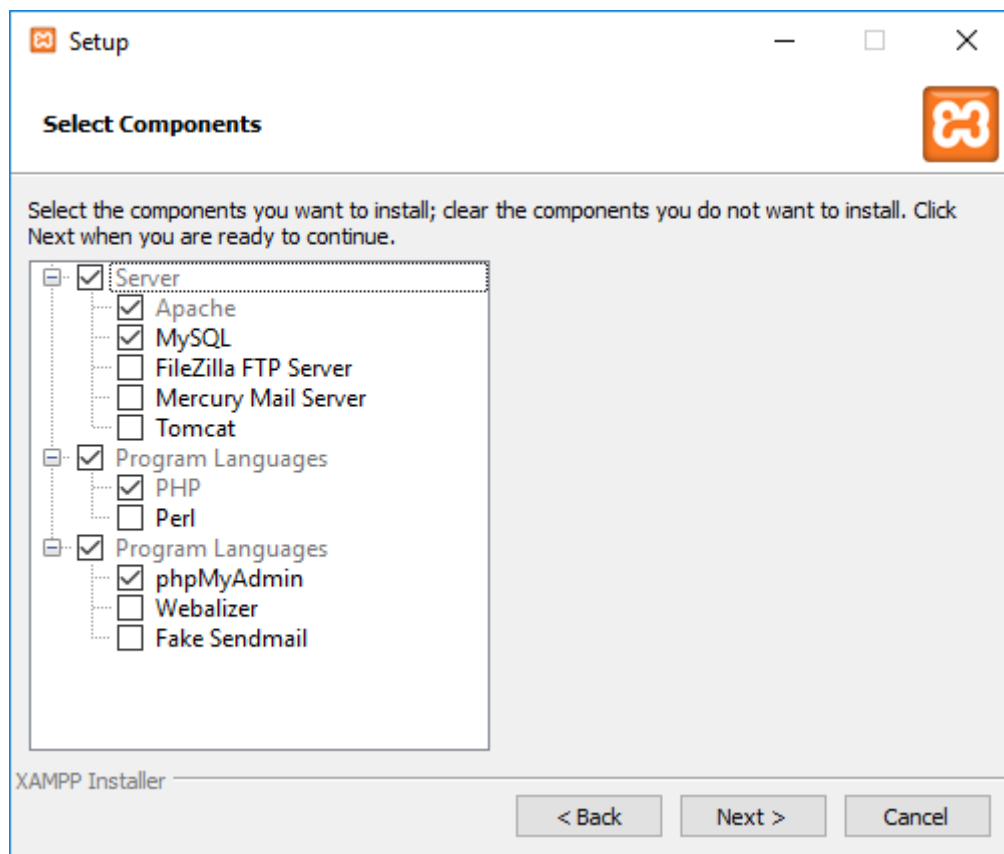
(圖) 下載後的檔案



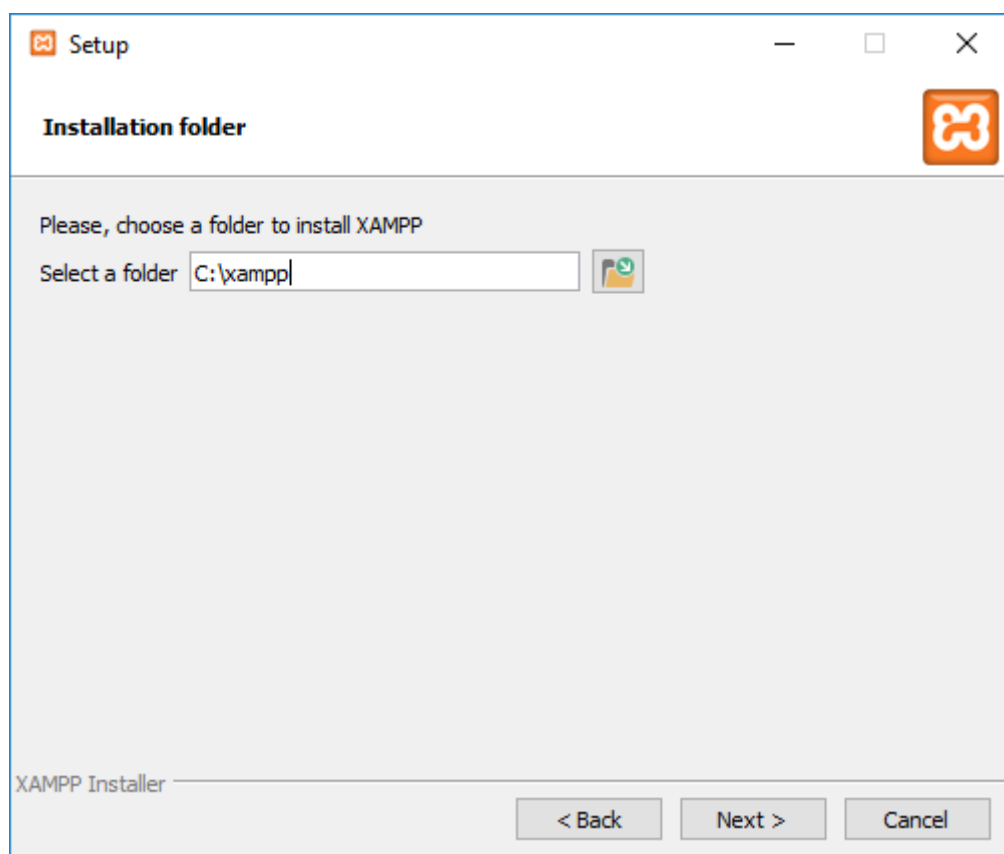
(圖) 不會安裝到 Program Files 路徑下，可以直接按下 OK



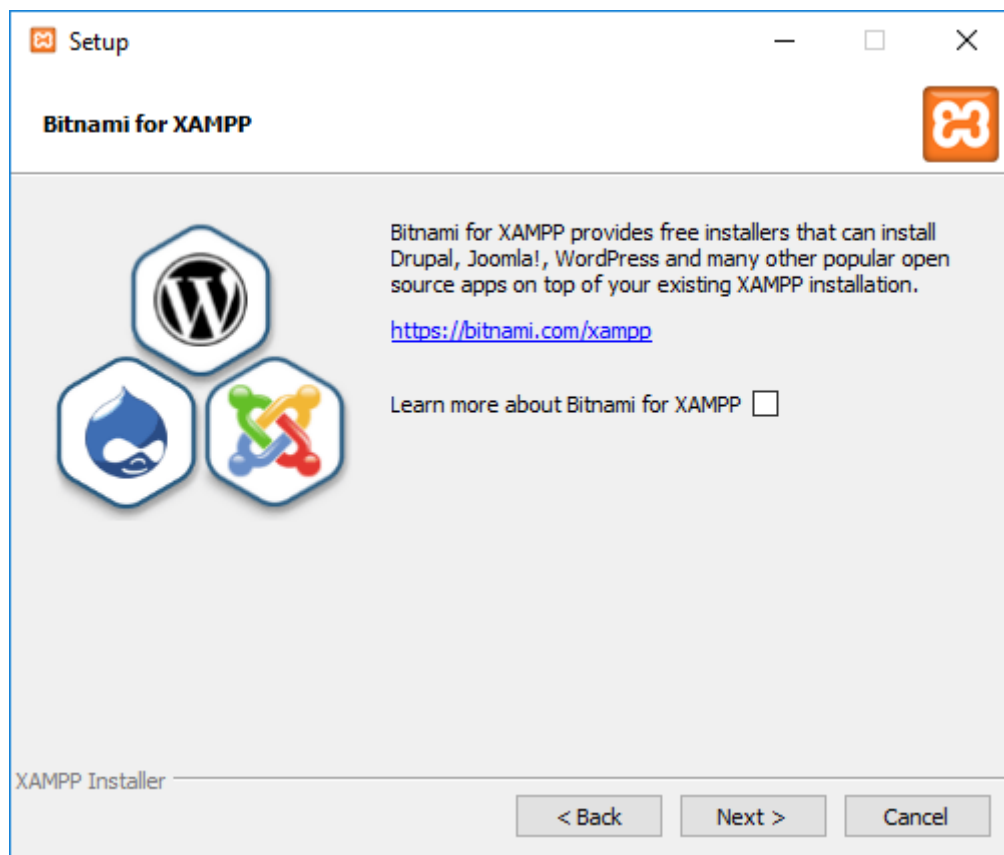
(圖) 按下一步



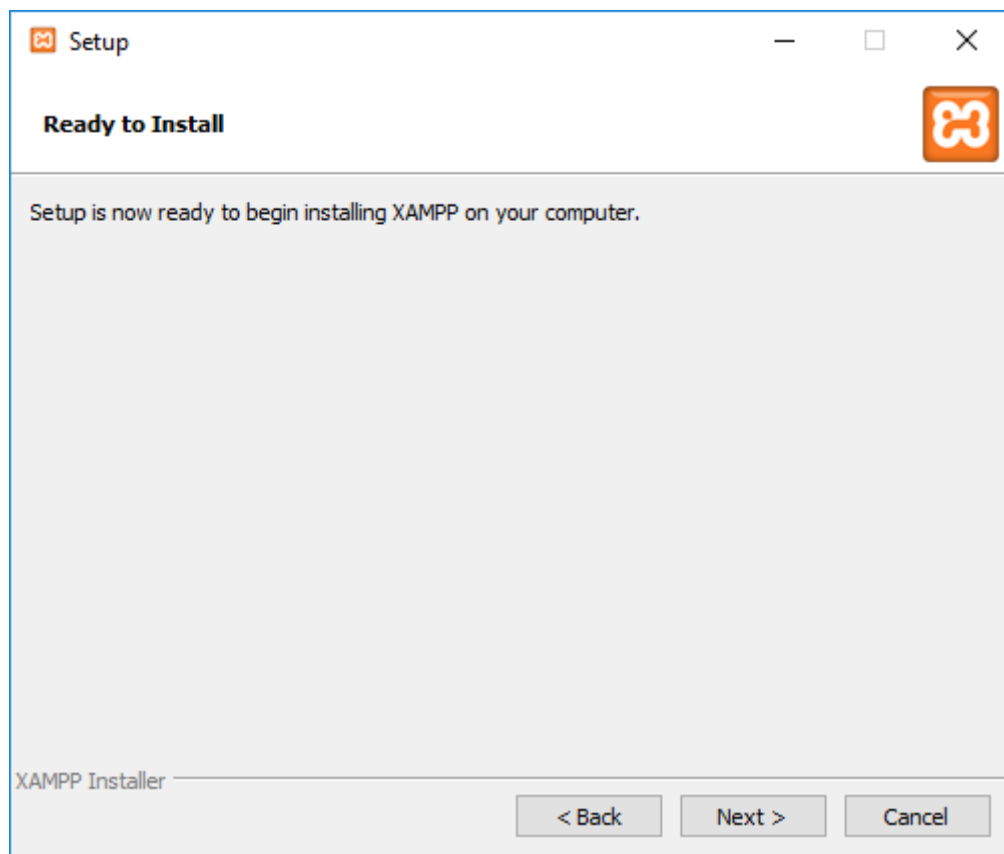
(圖) 除了預設的 Apache、PHP，僅留下 MySQL、phpMyAdmin 後，按下下一步



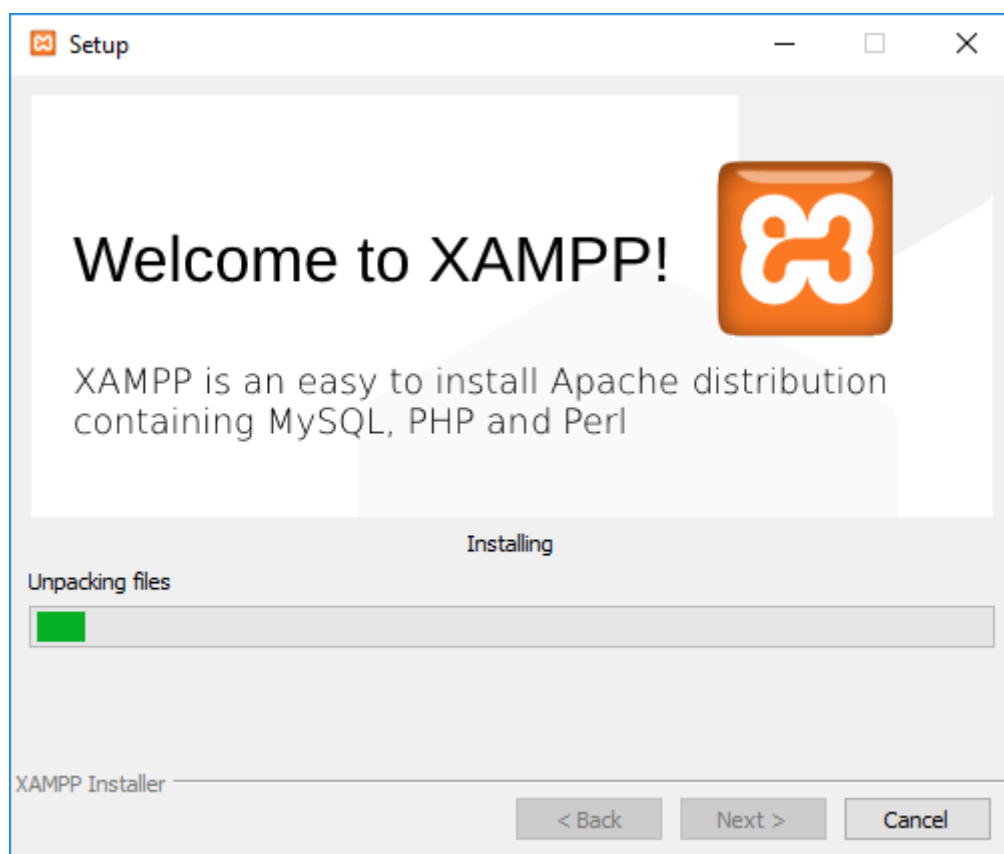
(圖) 使用預設路徑，按下一步



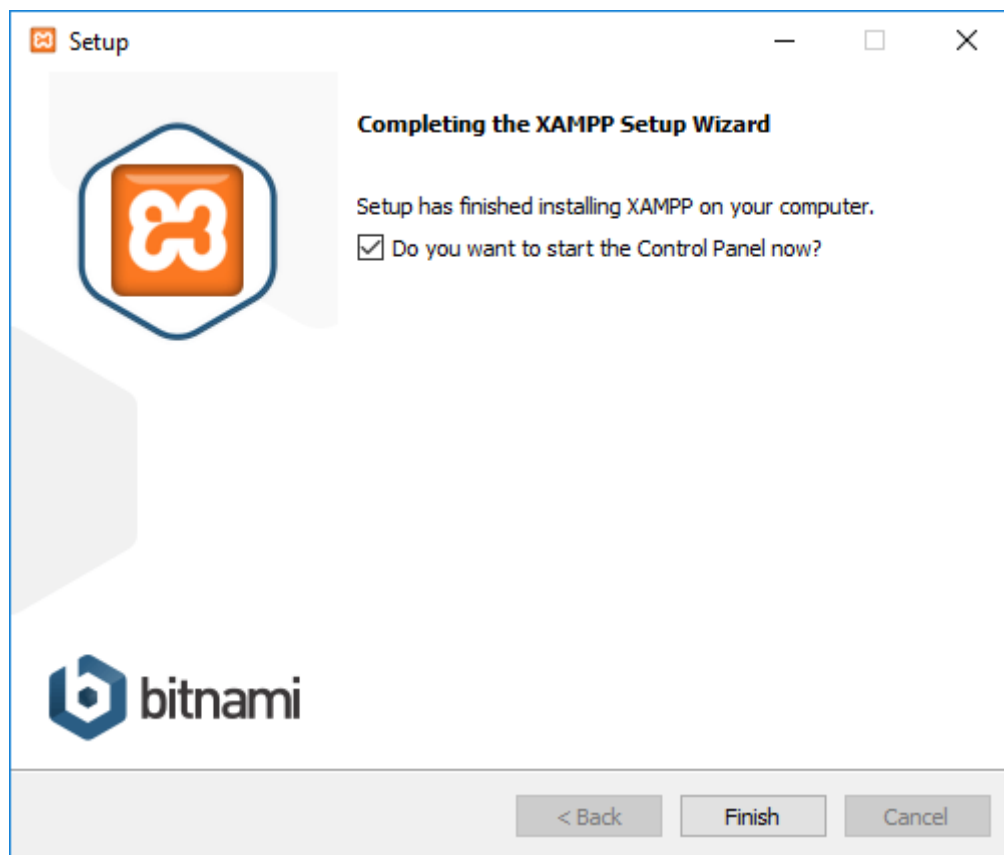
(圖) 課程中不會用到架站工具，所以取消上方勾選，按下一步



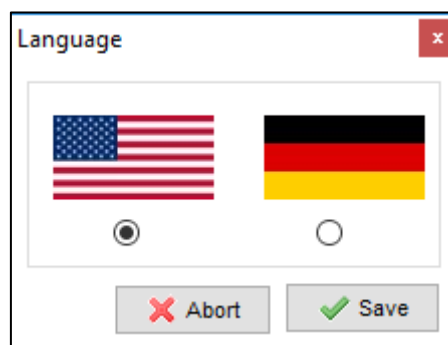
(圖) 按下一步，開始安裝



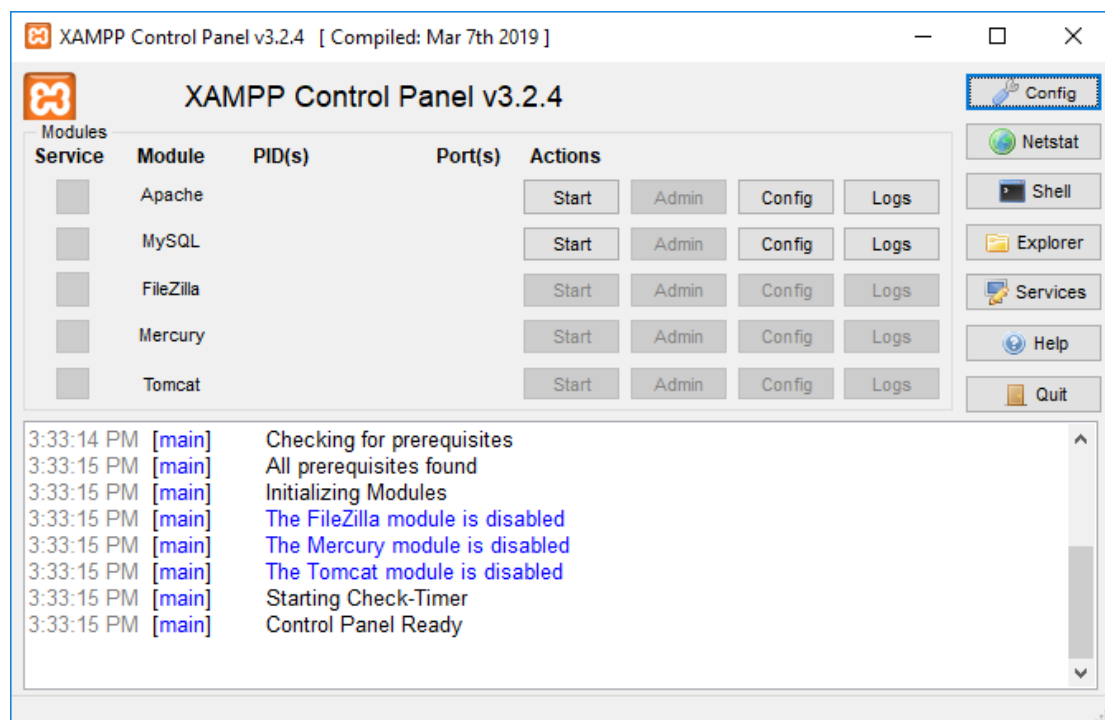
(圖) 安裝過程



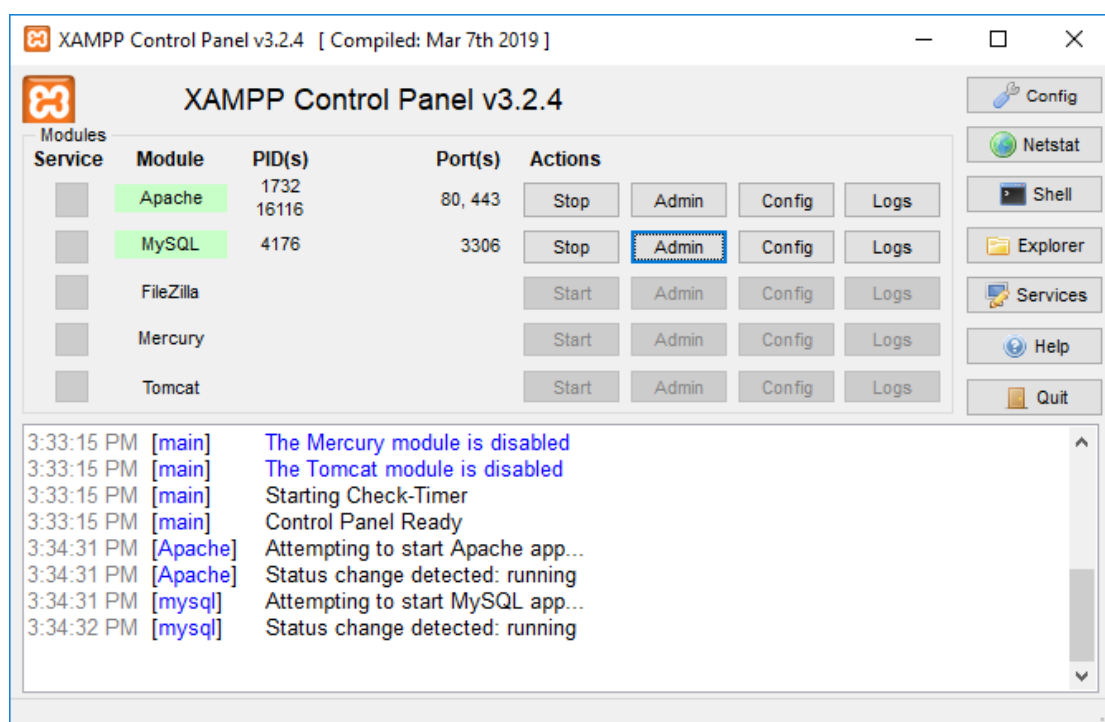
(圖) 勾選上方選項，開啟控制台，按下完成



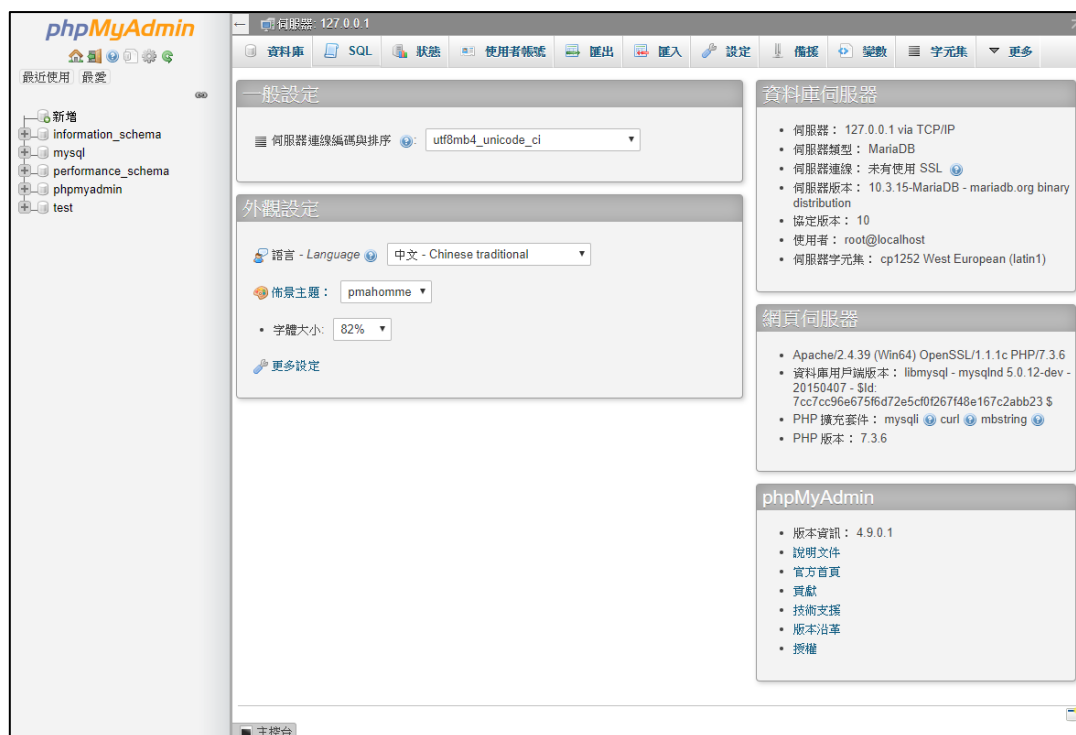
(圖) 選擇美國（英文）後，按儲存



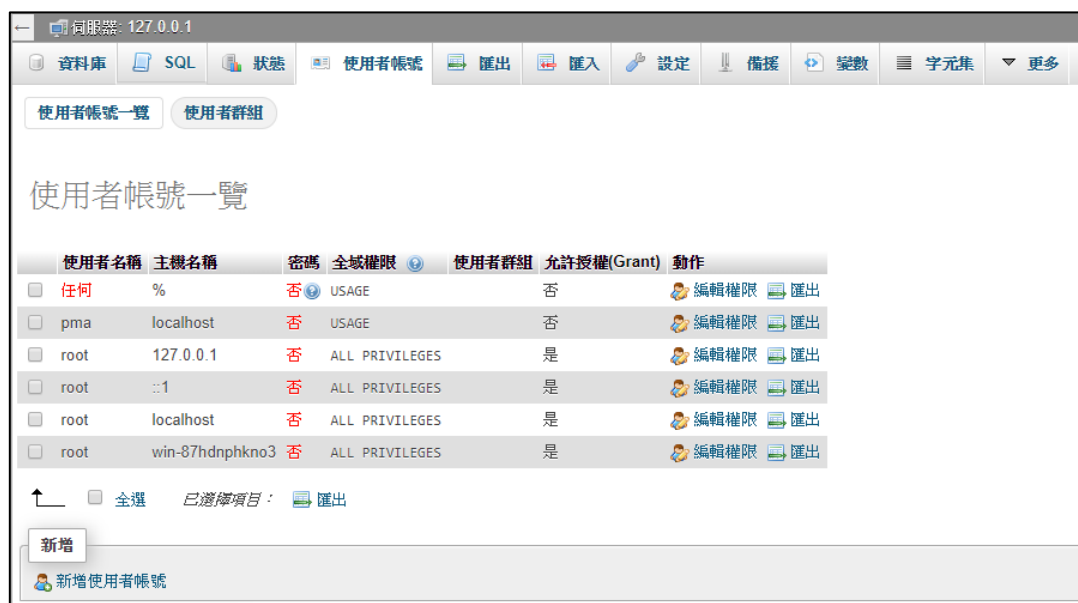
(圖) 控制台畫面，請按下 Apache、MySQL 右側的 Start 鍵，啟動服務



(圖) 按下 MySQL 右側的 Admin，開啟 phpMyAdmin 網頁



(圖) 看到 phpMyAdmin 的畫面，代表安裝成功



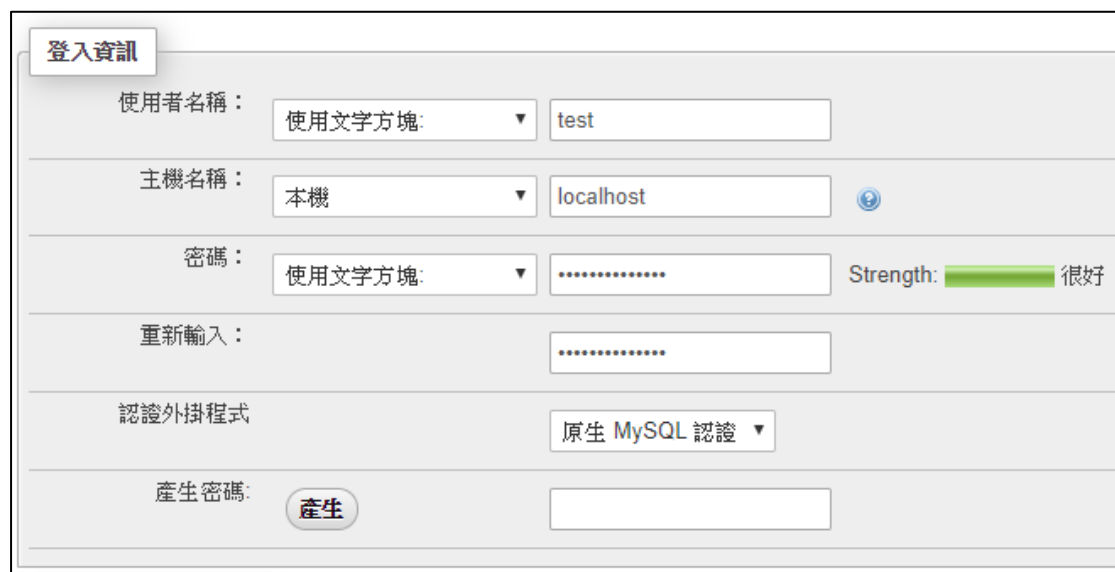
(圖) 選擇上面的使用者帳號，並按下圖片左下角的連結，新增使用者帳號

範例用資料庫帳號、密碼

課堂中所使用的範例，大部分會需要寫入資料庫，我們需要建立一組資料庫的帳號、密碼。

帳號：test

密碼：T1st@localhost



(圖) 登入資訊



(圖) 全域權限，請勾選「SELECT, INSERT, UPDATE, DELETE」



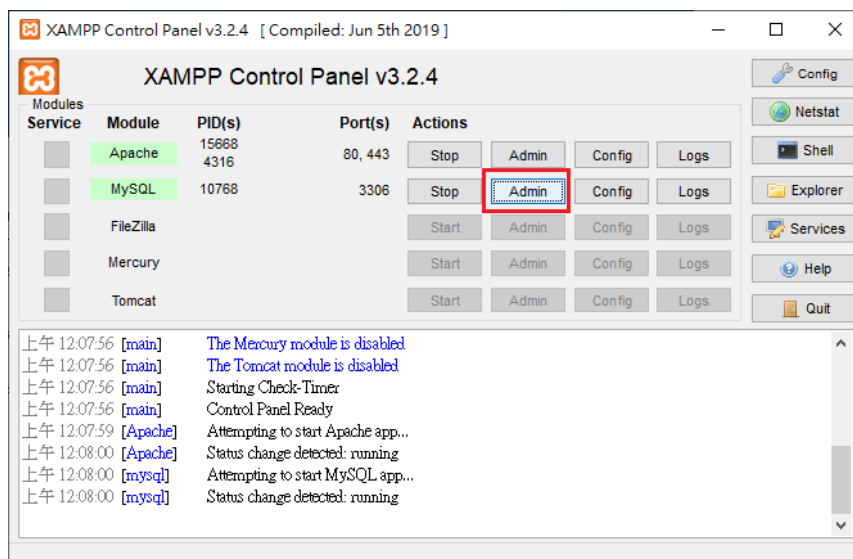
(圖) 最後按下網頁右下角的執行，建立新帳號

補充說明

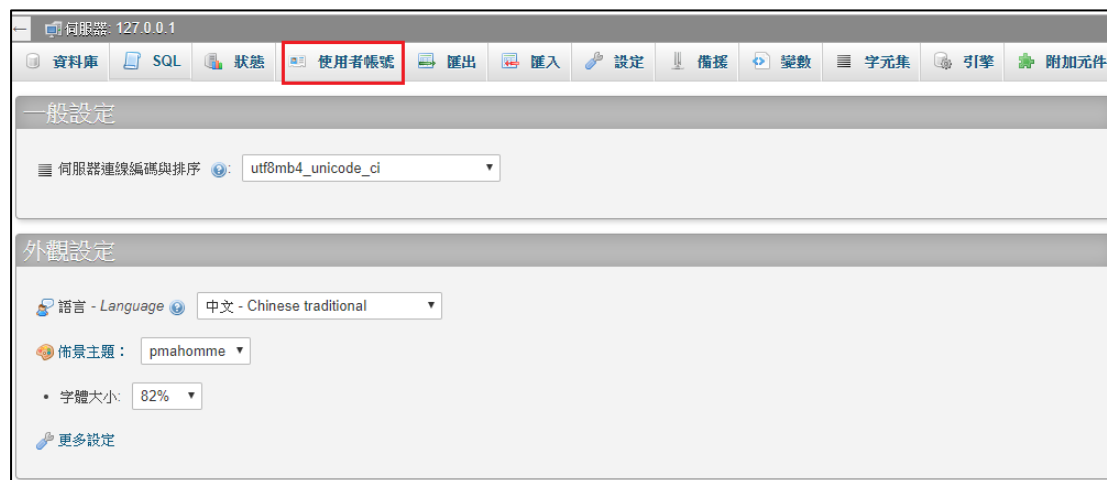
資料庫使用者帳號的增修，有很大的學問。課程中，為了教學方便，可以按照前述的流程建立使用者；若是公司有使用到資料庫，帳號管理不能草率，一定要請教有經驗的資料庫管理人員，協助帳號新增與權限設定。

設定 Root 密碼的方式，有兩種，一個是透過 phpMyAdmin 的 SQL 介面、一個是使用 mysql 執行檔，透過命令視窗來執行 SQL 語法。

phpMyAdmin



(圖) 先開啟 Apache、MySQL，再按下 MySQL 的 Admin 來開啟



(圖) 點選「使用者帳號」連結



(圖) 使用者帳號一覽

這裡有 3 個 root，我們先修改主機名稱為「localhost」的那一個帳號。



(圖) 按下 SQL 連結



(圖) 輸入 / 執行 SQL 的文字欄位

修改特定帳號的密碼：

SQL 語法

選擇資料庫

use mysql;

設定 特定帳號@主機名稱 = PASSWORD('你的密碼');

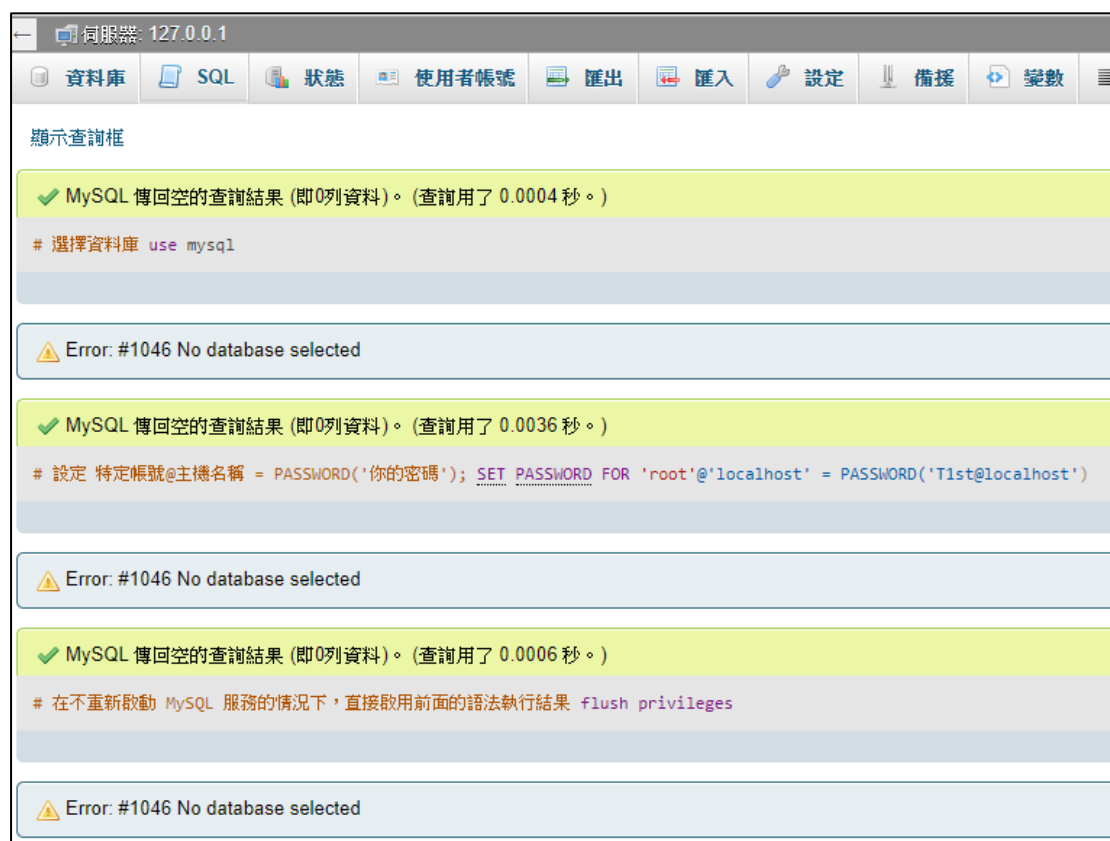
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('T1st@localhost');

在不重新啟動 MySQL 服務的情況下，直接啟用前面的語法執行結果

flush privileges;



(圖) 修改 root 密碼的 SQL 語法



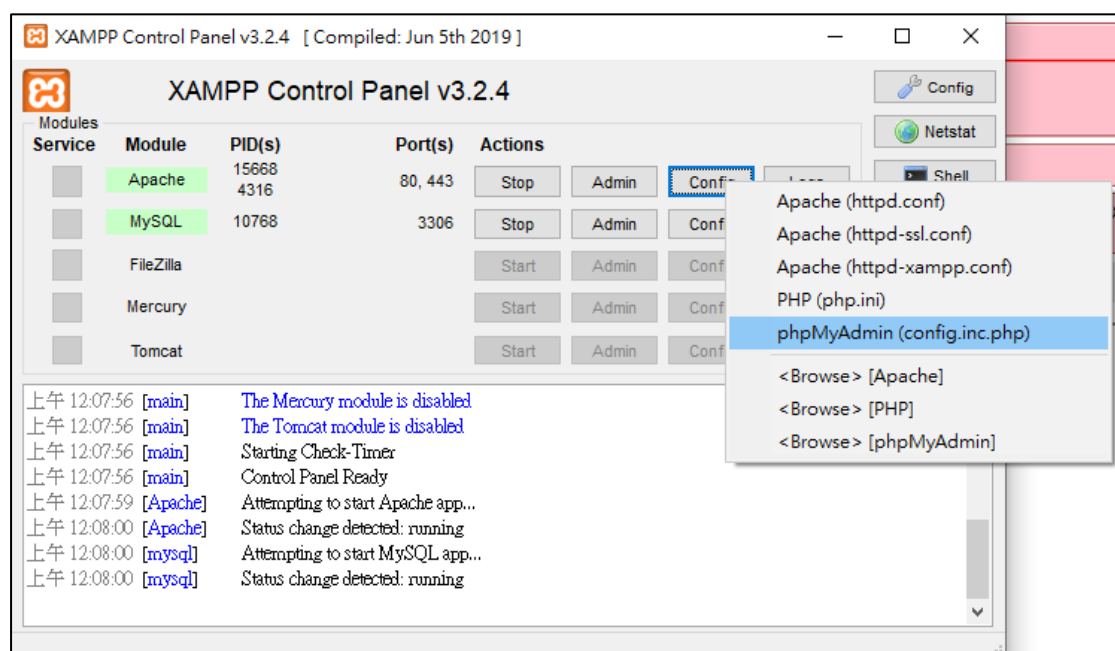
(圖) root 密碼修正完畢

改完密碼後，我們重新整理 phpMyAdmin 的頁面，會發現連線失敗的畫面。



(圖) 先前的 root 密已經無法使用

我們需要到 phpMyAdmin 裡面的 config.inc.php 去修正登入密碼。



(圖) 選擇 phpMyAdmin (config.inc.php)

我們接下來要將新密碼放到 config.inc.php 裡面。

```
/* Authentication type and info */
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = '';
$cfg['Servers'][$i]['extension'] = 'mysql';
$cfg['Servers'][$i]['AllowNoPassword'] = true;
$cfg['Lang'] = '';
```

(圖) 將新密碼放到 \$cfg['Servers'][\$i]['password'] = '你的新密碼'

```
/* Authentication type and info */
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'T1st@localhost';
$cfg['Servers'][$i]['extension'] = 'mysql';
$cfg['Servers'][$i]['AllowNoPassword'] = true;
$cfg['Lang'] = '';
```

(圖) 放入新密碼



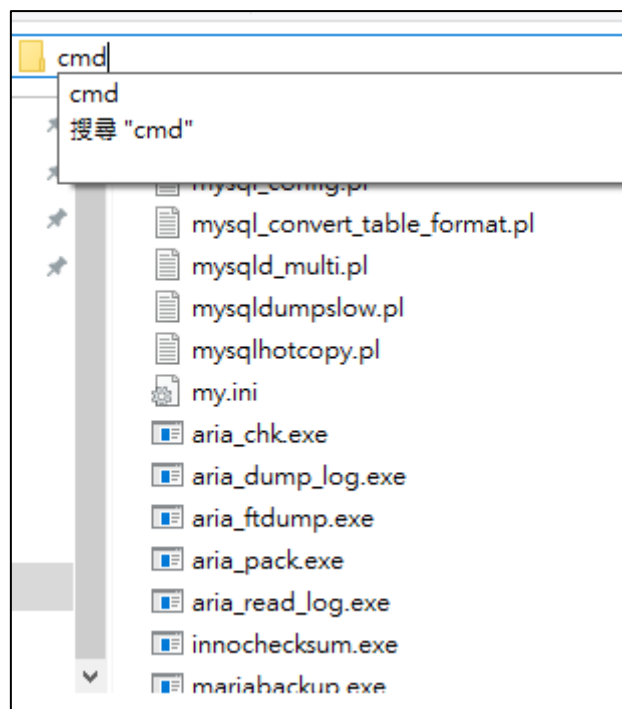
(圖) 再刷新 phpMyAdmin 網頁後，恢復到原先的樣子，一樣是 root 帳號

mysql 執行檔

確認兩件事：

- 有「C:\xampp\mysql\bin」路徑
- 路徑裡面，有幾個執行檔，其中一個叫作「mysql」

首先，我們先在「C:\xampp\mysql\bin」上頭的路徑列，輸入「cmd」。



(圖) 路徑列上，輸入 cmd



(圖) 出現命令提示字元

接下來，我們要輸入剛才修改的 root 帳號與密碼。

說明

使用 mysql 執行檔，使用者名稱為 root，密碼之後會詢問

> mysql.exe -u root -p

Enter password: *****


```
ca 選取 C:\Windows\System32\cmd.exe - mysql.exe -u root -p
Microsoft Windows [版本 10.0.17763.864]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\xampp\mysql\bin>mysql.exe -u root -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 592
Server version: 10.4.8-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> _
```

(圖) 登入成功的畫面

指令流程

```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> use mysql;
MariaDB [mysql]> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('use2@localhost');
Query OK, 0 rows affected (0.004 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.000 sec)

MariaDB [mysql]>
```

此時我們回到 phpMyAdmin，重新刷新頁面，便會出現錯誤訊息。

歡迎使用 phpMyAdmin

錯誤

MySQL 回應：

無法連線：設定無效。

mysqli_real_connect(): (HY000/1045): Access denied for user 'root'@'localhost' (using password: YES)

phpMyAdmin 曾經嘗試連線到 MySQL 伺服器，但遭伺服器拒絕。請檢查設定檔案中的主機、帳號和密碼，並確認這些資訊與 MySQL 伺服器管理員所提供的資訊一致。

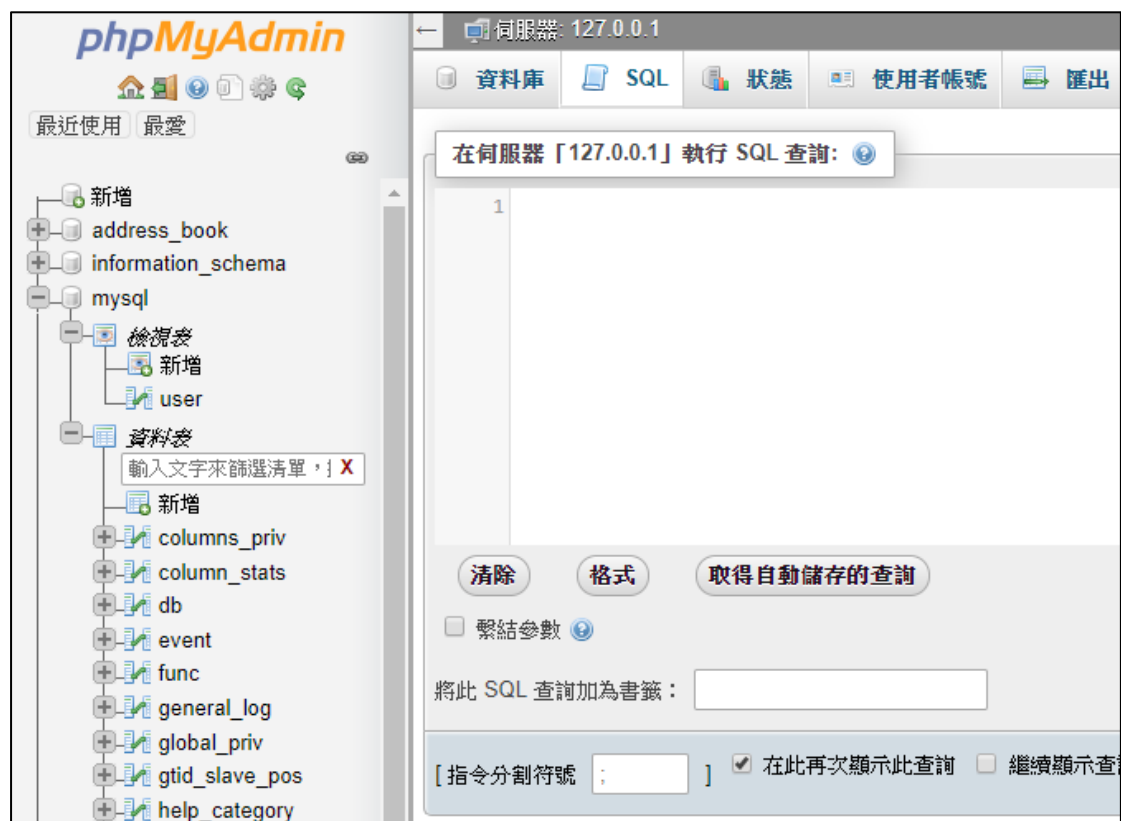
[重試連接](#)

(圖) 連線錯誤的畫面

我們要再一次將新密碼放到 config.inc.php 裡面。

```
/* Authentication type and info */  
$cfg['Servers'][$i]['auth_type'] = 'config';  
$cfg['Servers'][$i]['user'] = 'root';  
$cfg['Servers'][$i]['password'] = 'use2@localhost';  
$cfg['Servers'][$i]['extension'] = 'mysqli';  
$cfg['Servers'][$i]['AllowNoPassword'] = true;  
$cfg['Lang'] = '';
```

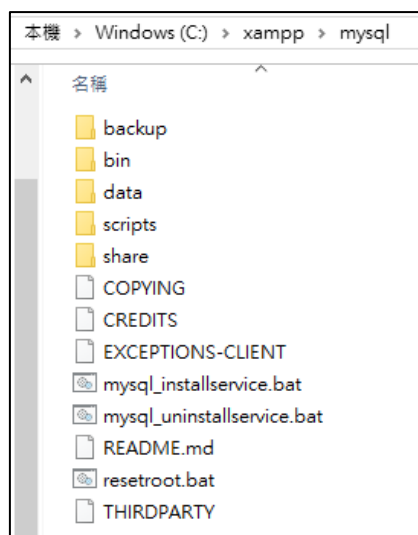
(圖) 輸入新密碼



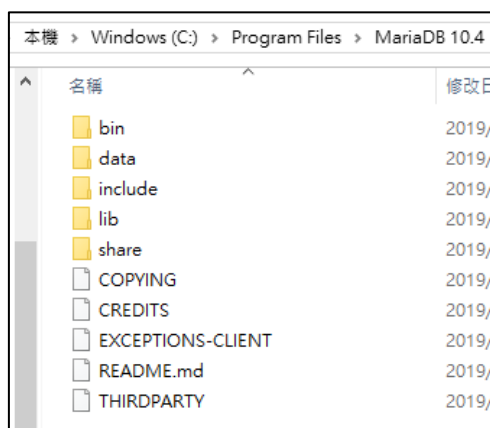
(圖) 此時又回到正常畫面

二、快速導覽

以 XAMPP 的 MySQL 資料夾為例，路徑在「C:\xampp\mysql」。



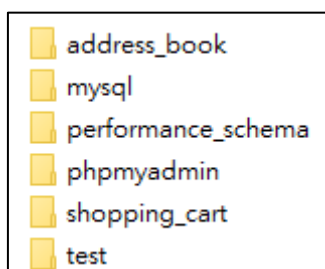
(圖) XAMPP 的 MySQL 資料夾結構



(圖) 直接官方網站下載 MariaDB 的資料夾結構

C:\xampp\mysql\data

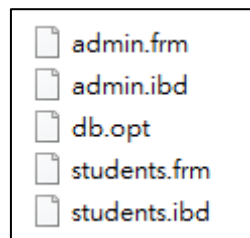
MySQL 放置「資料」的資料夾，也是日誌（log）被儲存放置的地方。在 data 中，子資料夾（mysql、performance_schema、phpmyadmin、test）也是放置資料庫（database）的地方，在放置資料庫的資料夾中，也存放著這些資料庫所擁有的資料表（tables）。



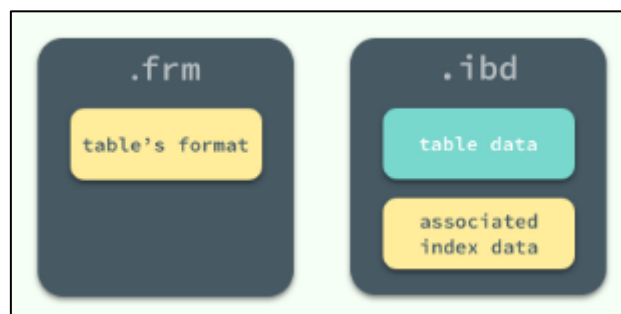
(圖) 以資料庫(database)為名的資料夾，裡面是資料表(tables)的檔案

一個新建立的資料庫，可能會有以下的檔案格式（以 InnoDB 為例）：

- frm：描述資料表的格式，或許說定義（table definition）
- ibd：放置資料表中的資料，以及關聯的索引資料
- db.opt：設定預設文字編碼和文字排序規則



(圖) address_book



(圖) .frm 和 .ibd 檔案結構

備註：ibdata1 是儲存 MySQL InnoDB 引擎的結構、資料、索引和快取

C:\xampp\mysql\data\mysql

放置主要的 mysql 資料庫資料夾，也是管理 MySQL 用的資料庫。

C:\xampp\mysql\share\charset

放置支援字元集的檔案

```

MariaDB [(none)]> use mysql;
Database changed
MariaDB [mysql]> show character set;
+-----+-----+-----+-----+
| Charset | Description | Default collation | Maxlen |
+-----+-----+-----+-----+
| big5 | Big5 Traditional Chinese | big5_chinese_ci | 2 |
| dec8 | DEC West European | dec8_swedish_ci | 1 |
| cp850 | DOS West European | cp850_general_ci | 1 |
| hp8 | HP West European | hp8_english_ci | 1 |
| koi8r | KOI8-R Relcom Russian | koi8r_general_ci | 1 |
| latin1 | cp1252 West European | latin1_swedish_ci | 1 |
| latin2 | ISO 8859-2 Central European | latin2_general_ci | 1 |
| swe7 | 7bit Swedish | swe7_swedish_ci | 1 |
| ascii | US ASCII | ascii_general_ci | 1 |
| ujis | EUC-JP Japanese | ujis_japanese_ci | 3 |
| sjis | Shift-JIS Japanese | sjis_japanese_ci | 2 |
| hebrew | ISO 8859-8 Hebrew | hebrew_general_ci | 1 |
| tis620 | TIS620 Thai | tis620_thai_ci | 1 |
| euckr | EUC-KR Korean | euckr_korean_ci | 2 |
| koi8u | KOI8-U Ukrainian | koi8u_general_ci | 1 |
| gb2312 | GB2312 Simplified Chinese | gb2312_chinese_ci | 2 |
| greek | ISO 8859-7 Greek | greek_general_ci | 1 |
| cp1250 | Windows Central European | cp1250_general_ci | 1 |
| gbk | GBK Simplified Chinese | gbk_chinese_ci | 2 |
| latin5 | ISO 8859-9 Turkish | latin5_turkish_ci | 1 |
| armSCII8 | ARMSCII-8 Armenian | armSCII8_general_ci | 1 |
| utf8 | UTF-8 Unicode | utf8_general_ci | 3 |
| ucs2 | UCS-2 Unicode | ucs2_general_ci | 2 |
| cp866 | DOS Russian | cp866_general_ci | 1 |
| keybcs2 | DOS Kamenicky Czech-Slovak | keybcs2_general_ci | 1 |
| macce | Mac Central European | macce_general_ci | 1 |
| macroman | Mac West European | macroman_general_ci | 1 |
| cp852 | DOS Central European | cp852_general_ci | 1 |
| latin7 | ISO 8859-13 Baltic | latin7_general_ci | 1 |
| utf8mb4 | UTF-8 Unicode | utf8mb4_general_ci | 4 |
| cp1251 | Windows Cyrillic | cp1251_general_ci | 1 |
| utf16 | UTF-16 Unicode | utf16_general_ci | 4 |
| utf16le | UTF-16LE Unicode | utf16le_general_ci | 4 |
| cp1256 | Windows Arabic | cp1256_general_ci | 1 |
| cp1257 | Windows Baltic | cp1257_general_ci | 1 |
| utf32 | UTF-32 Unicode | utf32_general_ci | 4 |
| binary | Binary pseudo charset | binary | 1 |
| geostd8 | GEOSTD8 Georgian | geostd8_general_ci | 1 |
| cp932 | SJIS for Windows Japanese | cp932_japanese_ci | 2 |
| eucjpms | UJIS for Windows Japanese | eucjpms_japanese_ci | 3 |
+-----+-----+-----+-----+
40 rows in set (0.007 sec)

MariaDB [mysql]> _

```

(圖) 使用 SHOW CHARACTER SET; 來看字元集列表



(圖) 從 phpMyAdmin 中一覽字元集

使用指令查詢特定字元集所支援的排序規則					
MariaDB [mysql]> SHOW COLLATION LIKE 'utf8%';					
+-----+-----+-----+-----+-----+					
Collation	Charset	Id	Default	Compiled	Sortlen
+-----+-----+-----+-----+-----+					
utf8_general_ci	utf8	33	Yes	Yes	1
utf8_bin	utf8	83		Yes	1
utf8_unicode_ci	utf8	192		Yes	8
utf8_icelandic_ci	utf8	193		Yes	8
utf8_latvian_ci	utf8	194		Yes	8
utf8_romanian_ci	utf8	195		Yes	8
utf8_slovenian_ci	utf8	196		Yes	8
utf8_polish_ci	utf8	197		Yes	8
utf8_estonian_ci	utf8	198		Yes	8
utf8_spanish_ci	utf8	199		Yes	8
utf8_swedish_ci	utf8	200		Yes	8
utf8_turkish_ci	utf8	201		Yes	8
utf8_czech_ci	utf8	202		Yes	8
utf8_danish_ci	utf8	203		Yes	8

utf8_lithuanian_ci	utf8	204	Yes		8
utf8_slovak_ci	utf8	205	Yes		8
utf8_spanish2_ci	utf8	206	Yes		8
utf8_roman_ci	utf8	207	Yes		8
utf8_persian_ci	utf8	208	Yes		8
utf8_esperanto_ci	utf8	209	Yes		8
utf8_hungarian_ci	utf8	210	Yes		8
utf8_sinhala_ci	utf8	211	Yes		8
utf8_german2_ci	utf8	212	Yes		8
utf8_croatian_mysql561_ci	utf8	213	Yes		8
utf8_unicode_520_ci	utf8	214	Yes		8
utf8_vietnamese_ci	utf8	215	Yes		8
utf8_general_mysql500_ci	utf8	223	Yes		1
utf8_croatian_ci	utf8	576	Yes		8
utf8_myanmar_ci	utf8	577	Yes		8
utf8_thai_520_w2	utf8	578	Yes		4
utf8_general_nopad_ci	utf8	1057	Yes		1
utf8_nopad_bin	utf8	1107	Yes		1
utf8_unicode_nopad_ci	utf8	1216	Yes		8
utf8_unicode_520_nopad_ci	utf8	1238	Yes		8
utf8mb4_general_ci	utf8mb4	45	Yes		1
utf8mb4_bin	utf8mb4	46	Yes		1
utf8mb4_unicode_ci	utf8mb4	224	Yes		8
utf8mb4_icelandic_ci	utf8mb4	225	Yes		8
utf8mb4_latvian_ci	utf8mb4	226	Yes		8
utf8mb4_romanian_ci	utf8mb4	227	Yes		8
utf8mb4_slovenian_ci	utf8mb4	228	Yes		8
utf8mb4_polish_ci	utf8mb4	229	Yes		8
utf8mb4_estonian_ci	utf8mb4	230	Yes		8
utf8mb4_spanish_ci	utf8mb4	231	Yes		8
utf8mb4_swedish_ci	utf8mb4	232	Yes		8
utf8mb4_turkish_ci	utf8mb4	233	Yes		8
utf8mb4_czech_ci	utf8mb4	234	Yes		8
utf8mb4_danish_ci	utf8mb4	235	Yes		8
utf8mb4_lithuanian_ci	utf8mb4	236	Yes		8
utf8mb4_slovak_ci	utf8mb4	237	Yes		8
utf8mb4_spanish2_ci	utf8mb4	238	Yes		8
utf8mb4_roman_ci	utf8mb4	239	Yes		8

utf8mb4_persian_ci	utf8mb4	240	Yes		8
utf8mb4_esperanto_ci	utf8mb4	241	Yes		8
utf8mb4_hungarian_ci	utf8mb4	242	Yes		8
utf8mb4_sinhala_ci	utf8mb4	243	Yes		8
utf8mb4_german2_ci	utf8mb4	244	Yes		8
utf8mb4_croatian_mysql561_ci	utf8mb4	245	Yes		8
utf8mb4_unicode_520_ci	utf8mb4	246	Yes		8
utf8mb4_vietnamese_ci	utf8mb4	247	Yes		8
utf8mb4_croatian_ci	utf8mb4	608	Yes		8
utf8mb4_myanmar_ci	utf8mb4	609	Yes		8
utf8mb4_thai_520_w2	utf8mb4	610	Yes		4
utf8mb4_general_nopad_ci	utf8mb4	1069	Yes		1
utf8mb4_nopad_bin	utf8mb4	1070	Yes		1
utf8mb4_unicode_nopad_ci	utf8mb4	1248	Yes		8
utf8mb4_unicode_520_nopad_ci	utf8mb4	1270	Yes		8
+-----+-----+-----+-----+-----+					
67 rows in set (0.000 sec)					

C:\xampp\mysql\share/{語系}

資料夾包含了各種語系的錯誤訊息

執行檔放置路徑為「C:\xampp\mysql\bin」我們介紹幾個常用執行檔的簡易使用範例。

mysqldump

資料庫匯出/備份

1. 備份單一資料庫:

\$ mysqldump -u root -p 你的資料庫名稱 > 你的備份名稱.sql;

2. 僅備份特定資料庫下的資料表:

\$ mysqldump -u root -p 你的資料庫名稱 Table1 Table2... > 你的備份名稱.sql;

3. 僅匯出資料表結構:

\$ mysqldump -u root -p --no-data 你的資料庫名稱 > 你的備份名稱.sql;

4. 備份全部的資料庫:

\$ mysqldump -u root -p -A > 你的備份名稱.sql;

mysql/mysqldump/mysqladmin 與「登入」有關的參數說明:

-u Username

用來存取 mysql 資料表的用戶名稱。此參數可省略，若省略，則表示為目前

登入 linux 的用戶名稱。

-p

於執行時詢問密碼。

mysql

登入

> mysql -u 使用者帳號 -p 要求輸入密碼

> mysql -u root -p

Enter Password:

資料庫匯入/還原

1. 建立資料庫，再從備份檔匯入單一資料庫:

\$ mysqladmin -u root -p create 你的資料庫名稱;

\$ mysql -u root -p 你的資料庫名稱 < 你的備份名稱..sql

2. 從備份檔匯入全部的資料庫 (不需先建立資料庫):

\$ mysql -u root -p < 所有資料庫備份檔.sql;

```
C:\xampp\mysql\bin>mysql -u root -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 296
Server version: 10.4.8-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]>
```

(圖) 登入 mysql 後，出現 Welcome to the MariaDB monitor

安裝完 mysql 後，系統預設會建立一個不需要密碼的 root 用戶，和一個無用戶名、無密碼的匿名用戶 (Anonymous Account)。進行下面的初始化操作以合理授權，增強安全。我們可以直接在 phpMyAdmin 裡面，使用 SQL 語法，來刪除匿名帳號。

SQL 語法

選擇資料庫

use mysql;

移除匿名使用者

DROP USER ''@'localhost';

或是

```
DELETE FROM `User` WHERE `User`=";
```

在不重新啟動 MySQL 服務的情況下，直接啟用前面的語法執行結果
flush privileges;

三、資料庫設計議題

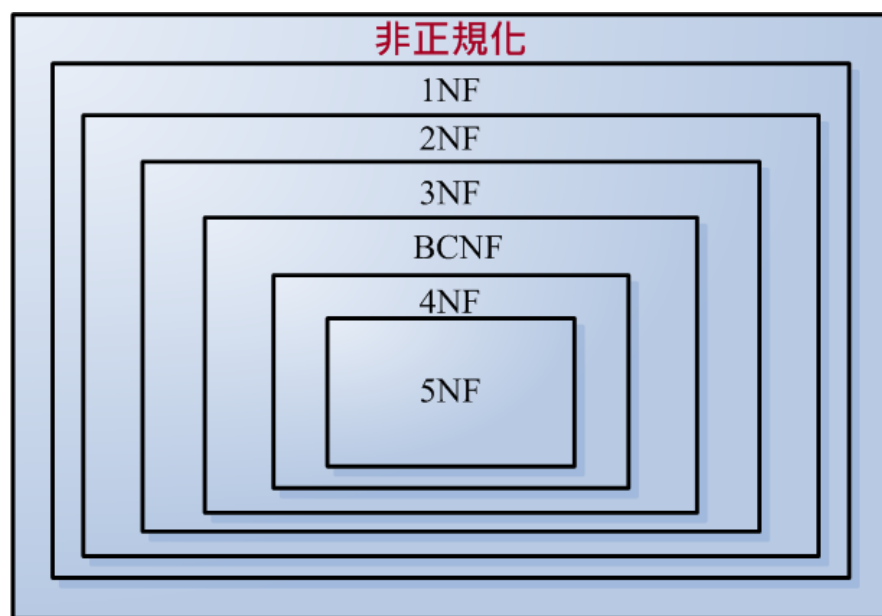
資料庫的設計原則如下：

- 確認綱要（schema）中屬性（attributes）的語意（semantic）清楚無誤。
- 減少值組中的重覆資訊。
- 減少值組中的 NULL 值。
- 不允許產生假值組（spurious tuple）

異常（anomalies）原則上可以分成三種：

- 新增異常（insertion anomaly）
- 刪除異常（deletion anomaly）
- 修改異常（modification anomaly）

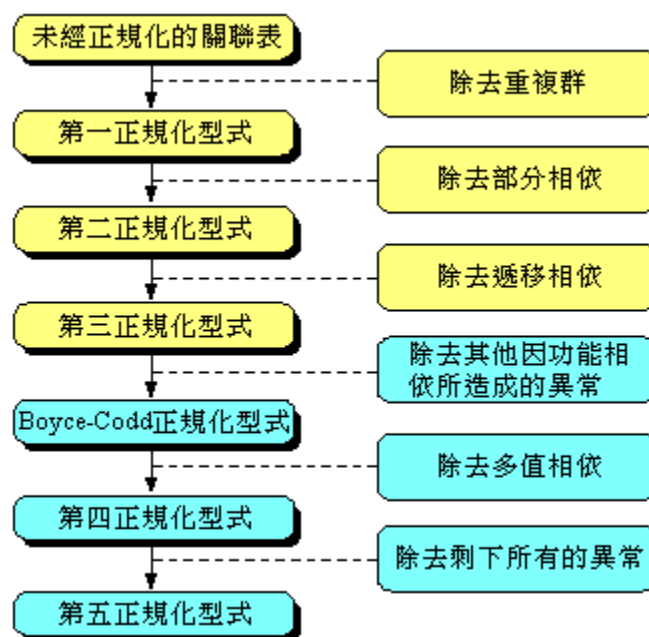
正規化（normalization）是在結構化分析與設計中，建立資料模式所用的技術，一種資料表分割的法則，目的是為了降低資料的重覆性，同時避免異常（anomalies）情況的發生。



(圖) 正規化的常見形式

正規化步驟	規則
1NF	<ul style="list-style-type: none"> ● 每一個欄位只能有一個基元值。 ● 沒有任何兩筆以上的資料完全重覆。 ● 資料表中有主鍵，其它欄位相依與主鍵。
2NF	<ul style="list-style-type: none"> ● 符合 1NF。 ● 每一個非鍵的屬性（例如姓名、性別）必須<u>完全相依</u>於主鍵（例如學號）。 ● 也就是不可以部分功能相依於主鍵。
3NF	<ul style="list-style-type: none"> ● 符合 2NF。 ● 各欄位與主鍵之間，沒有<u>遞移相依</u>的關係。 ● 若要找出資料表中的遞移相依性，簡單的方法就正從左到右掃描各個欄位，有沒有<u>與主鍵無關的相依性</u>存在。
BCNF	<ul style="list-style-type: none"> ● 符合 3NF。 ● 主鍵中的各個欄位（一張表有多個主鍵），不可以相依於其它非主鍵的欄位。

(表) 實務上最多進行到 3NF



(圖) 正規化步驟

尚未正規化的資料表：學生選課資料表

學號	姓名	性別	暱稱	課程 代號	課程 名稱	學分數	必選修	成績	老師 編號	老師 姓名
087	楊○○	男	好人	C001	程式設計	4	必	74	T001	曾○○

				C002	網頁設計	3	選	93	T002	林○○
088	陳○○	女	小白	C002	網頁設計	3	選	63	T002	林○○
				C003	視覺設計	2	必	82	T003	王○○
				C004	網路教學	4	選	94	T005	謝○○

1NF 後的資料表

學號	姓名	性別	暱稱	課程 代號	課程 名稱	學分數	必選修	成績	老師 編號	老師 姓名
087	楊○○	男	好人	C001	程式設計	4	必	74	T001	曾○○
087	楊○○	男	好人	C002	網頁設計	3	選	93	T002	林○○
088	陳○○	女	小白	C002	網頁設計	3	選	63	T002	林○○
088	陳○○	女	小白	C003	視覺設計	2	必	82	T003	王○○
088	陳○○	女	小白	C004	網路教學	4	選	94	T005	謝○○

新增異常：

學號	姓名	性別	暱稱	課程 代號	課程 名稱	學分數	必選修	成績	老師 編號	老師 姓名
NULL				C005	系統分析				NULL	

若要新增一門課程，要等學生選課、確定授課老師之後，才能新增。

修改異常：

學號	姓名	性別	暱稱	課程 代號	課程 名稱	學分數	必選修	成績	老師 編號	老師 姓名
087	楊○○	男	好人	C001	程式設計	4	必	74	T001	曾○○
087	楊○○	男	好人	C002	網頁設計	3	選	93 98	T002	林○○
088	陳○○	女	小白	C002	網頁設計	3	選	63	T002	林○○
088	陳○○	女	小白	C003	視覺設計	2	必	82	T003	王○○
088	陳○○	女	小白	C004	網路教學	4	選	94	T005	謝○○

若是希望有修「網頁設計」的同學，全班加 5 分，卻可能因為加分行為重覆多次，有些人加到、有些人沒加到，而造成資料不一致的異常現象。

刪除異常：

學號	姓名	性別	暱稱	課程 代號	課程 名稱	學分數	必選修	成績	老師 編號	老師 姓名
087	楊○○	男	好人	C001	程式設計	4	必	74	T001	曾○○
087	楊○○	男	好人	C002	網頁設計	3	選	93	T002	林○○
088	陳○○	女	小白	C002	網頁設計	3	選	63	T002	林○○

088	陳○○	女	小白	C003	視覺設計	2	必	82	T003	王○○
088	陳○○	女	小白	C004	網路教學	4	選	94	T005	謝○○

當我們刪除學生記錄時，同時也會刪除課程、老師的資訊，造成資料遺失的現象。

2NF 後的資料表

將部分功能相依的欄位分割出去，組成新的資料表。

學號	姓名	性別	暱稱	課程代號	課程名稱	學分數	必選修	成績	老師編號	老師姓名
087	楊○○	男	好人	C001	程式設計	4	必	74	T001	曾○○
087	楊○○	男	好人	C002	網頁設計	3	選	93	T002	林○○
088	陳○○	女	小白	C002	網頁設計	3	選	63	T002	林○○
088	陳○○	女	小白	C003	視覺設計	2	必	82	T003	王○○
088	陳○○	女	小白	C004	網路教學	4	選	94	T005	謝○○

分割出去的結果：

學生資料表

學號	姓名	性別	暱稱
087	楊○○	男	好人
088	陳○○	女	小白

成績資料表

學號	課程代號	成績
087	C001	74
087	C002	93
088	C002	63
088	C003	82
088	C004	94

課程資料表

課程代號	課程名稱	學分數	必選修	老師編號	老師姓名
C001	程式設計	4	必	T001	曾○○
C002	網頁設計	3	選	T002	林○○
C002	網頁設計	3	選	T002	林○○
C003	視覺設計	2	必	T003	王○○

C004	網路教學	4	選	T005	謝○○
------	------	---	---	------	-----

老師編號相依於課程代號

3NF 後的資料表

從左到右掃描各個欄位，有沒有與主鍵無關的相依性存在。

課程代號	課程名稱	學分數	必選修	老師編號	老師姓名
C001	程式設計	4	必	T001	曾○○
C002	網頁設計	3	選	T002	林○○
C002	網頁設計	3	選	T002	林○○
C003	視覺設計	2	必	T003	王○○
C004	網路教學	4	選	T005	謝○○

老師姓名相依於老師編號

老師姓名遞移相依於課程代號

將「老師編號、老師姓名」獨立成一個資料表

老師資料表

老師編號	老師姓名
T001	曾○○
T002	林○○
T003	王○○
T005	謝○○

課程資料表

課程代號	課程名稱	學分數	必選修	老師編號
C001	程式設計	4	必	T001
C002	網頁設計	3	選	T002
C002	網頁設計	3	選	T002
C003	視覺設計	2	必	T003
C004	網路教學	4	選	T005

資料表正規化完成結果

學生資料表

學號	姓名	性別	暱稱
087	楊○○	男	好人
088	陳○○	女	小白

成績資料表

學號	課程代號	成績
087	C001	74
087	C002	93
088	C002	63
088	C003	82
088	C004	94

老師資料表

老師編號	老師姓名
T001	曾○○
T002	林○○
T003	王○○
T005	謝○○

課程資料表

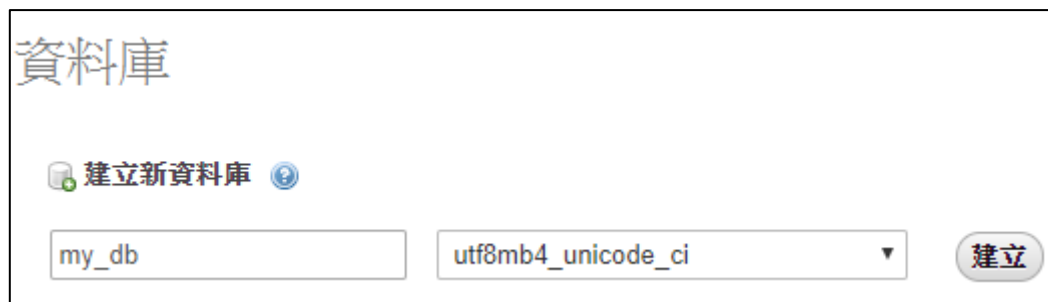
課程代號	課程名稱	學分數	必選修	老師編號
C001	程式設計	4	必	T001
C002	網頁設計	3	選	T002
C002	網頁設計	3	選	T002
C003	視覺設計	2	必	T003
C004	網路教學	4	選	T005

四、建立資料庫、資料表

我們將透過 phpMyAdmin 的管理介面，進行資料庫建置。



(圖) 點選資料庫



(圖) 輸入自訂的資料庫名稱，選擇排序規則

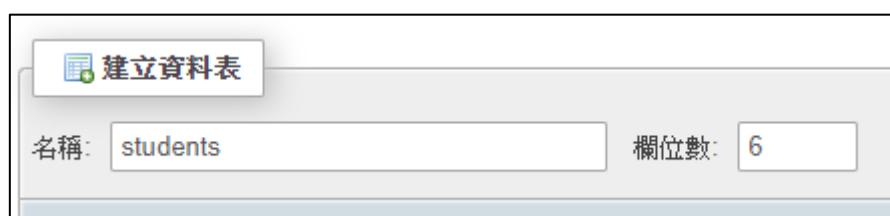
SQL
<pre>DROP DATABASE IF EXISTS `my_db`; CREATE DATABASE IF NOT EXISTS `my_db` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci; USE `my_db`;</pre>



(圖) 建立資料表，選擇欄位

依照我們先前的範例，我們可以建立正規化後的那四張表。

- students（學生資料表）
- scores（成績資料表）
- teachers（老師資料表）
- courses（課程資料表）



(圖) 學生資料表，6 個欄位

資料表名稱: students 新增 1 欄位 執行

名稱	類型	長度/值	預設值	編碼與排序	屬性	空值 (Null)	索引	A_I	備註
sId	VARCHAR	3	無			<input type="checkbox"/>	PRIMARY	<input type="checkbox"/>	學生編號
sName	VARCHAR	20	無			<input type="checkbox"/>	---	<input type="checkbox"/>	學生姓名
sGender	VARCHAR	1	無			<input type="checkbox"/>	---	<input type="checkbox"/>	學生性別
sNickname	VARCHAR	50	無			<input type="checkbox"/>	---	<input type="checkbox"/>	學生暱稱
created_at	DATETIME		CURRENT_TIME			<input type="checkbox"/>	---	<input type="checkbox"/>	新增時間
updated_at	DATETIME		CURRENT_TIME		on update CURR	<input type="checkbox"/>	---	<input type="checkbox"/>	更新時間

資料表備註: 學生資料表

編碼與排序: InnoDB

(圖) 學生資料表設計規格

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊
1	<u>sId</u>	varchar(3)	utf8mb4_unicode_ci	否	無		學生編號	
2	<u>sName</u>	varchar(20)	utf8mb4_unicode_ci	否	無		學生姓名	
3	<u>sGender</u>	varchar(1)	utf8mb4_unicode_ci	否	無		學生性別	
4	<u>sNickname</u>	varchar(50)	utf8mb4_unicode_ci	否	無		學生暱稱	
5	<u>created_at</u>	datetime		否		current_timestamp()	新增時間	
6	<u>updated_at</u>	datetime		否		current_timestamp()	更新時間	ON UPDATE CURRENT_TIMESTAMP()

(圖) 學生資料表結構

SQL 語法

```
CREATE TABLE `my_db`.`students` ( `sId` VARCHAR(3) NOT NULL COMMENT '學生編號', `sName` VARCHAR(20) NOT NULL COMMENT '學生姓名', `sGender` VARCHAR(1) NOT NULL COMMENT '學生性別', `sNickname` VARCHAR(50) NOT NULL COMMENT '學生暱稱', `created_at` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '新增時間', `updated_at` DATETIME on update CURRENT_TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '更新時間', PRIMARY KEY (`sId`)) ENGINE = InnoDB COMMENT = '學生資料表';
```

```
ALTER TABLE `students`
  ADD PRIMARY KEY (`sId`);
COMMIT;
```

建立資料表

名稱: scores 欄位數: 5

(圖) 成績資料表，5 個欄位

資料表名稱: scores 新增 1 欄位 執行

名稱	類型	長度/值	預設值	編碼與排序	屬性	空值 (Null)	索引	備註
sId	VARCHAR	3	無			<input type="checkbox"/>	---	學生編號
cId	VARCHAR	4	無			<input type="checkbox"/>	---	課程編號
score	TINYINT	3	無			<input type="checkbox"/>	---	成績
created_at	DATETIME		CURRENT_TIME			<input type="checkbox"/>	---	新增時間
updated_at	DATETIME		CURRENT_TIME		on update CURR	<input type="checkbox"/>	---	更新時間

資料表備註: 成績資料表 編碼與排序: 儲存引擎: InnoDB

(圖) 成績資料表設計規格

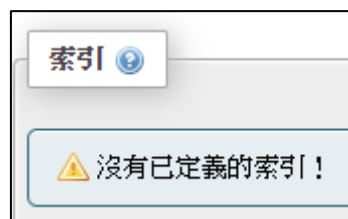
#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊
1	sId	varchar(3)	utf8mb4_unicode_ci		否	無	學生編號	
2	cId	varchar(4)	utf8mb4_unicode_ci		否	無	課程編號	
3	score	tinyint(3)			否	無	成績	
4	created_at	datetime			否	current_timestamp()	新增時間	
5	updated_at	datetime			否	current_timestamp()	更新時間	ON UPDATE CURRENT_TIMESTAMP()

(圖) 成績資料表結構

SQL

```
CREATE TABLE `scores` (
  `sId` varchar(3) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '學生編號',
  `cId` varchar(4) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '課程編號',
  `score` tinyint(3) NOT NULL COMMENT '成績',
  `created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
  `updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() COMMENT '更新時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='成績資料表';
```

在這裡，我們尚未在「索引」的下拉式選單選擇主鍵（Primary Key），待執行完成後，額外設定。



(圖) 在這裡會看見沒有定義索引的提示



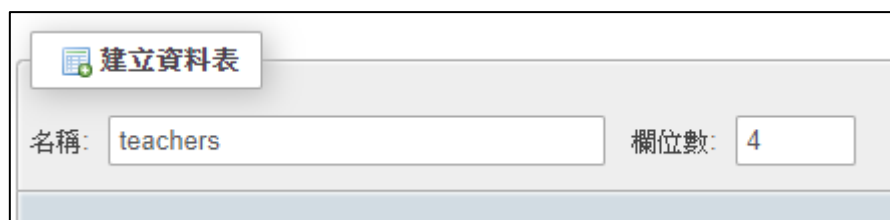
(圖) 選擇學生編號、課程編號，按下「主鍵」圖示



(圖) 看到 sId 和 cId 右側，各有一把主鍵圖示

SQL

```
ALTER TABLE `scores` ADD PRIMARY KEY( `sId`, `cId`);
```



建立資料表

名稱: teachers 欄位數: 4

(圖) 老師資料表，4 個欄位



資料表名稱: teachers 新增 1 欄位 執行

名稱	類型	長度/值	預設值	編碼與排序	屬性	空值 (Null)	索引	A_I	備註
tid	VARCHAR	4	無			<input type="checkbox"/>	PRIMARY	<input type="checkbox"/>	老師編號
tName	INT		無			<input type="checkbox"/>	---	<input type="checkbox"/>	老師姓名
created_at	DATETIME		無			<input type="checkbox"/>	---	<input type="checkbox"/>	新增時間
updated_at	DATETIME		無			<input type="checkbox"/>	---	<input type="checkbox"/>	更新時間

資料表備註: 老師資料表 編碼與排序: 儲存引擎: InnoDB

(圖) 老師資料表設計規格

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註
1	<u>tid</u>	varchar(4)	utf8mb4_unicode_ci		否	無	老師編號
2	<u>tName</u>	varchar(20)	utf8mb4_unicode_ci		否	無	老師姓名
3	<u>created_at</u>	datetime			否	無	新增時間
4	<u>updated_at</u>	datetime			否	無	更新時間

(圖) 老師資料表結構

SQL

```

CREATE TABLE `teachers` (
  `tid` varchar(4) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '老師編號',
  `tName` int(11) NOT NULL COMMENT '老師姓名',
  `created_at` datetime NOT NULL COMMENT '新增時間',
  `updated_at` datetime NOT NULL COMMENT '更新時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='老師資料表';

ALTER TABLE `teachers`
  ADD PRIMARY KEY (`tid`);

```

COMMIT;

建立資料表

名稱: courses 欄位數: 7

(圖) 課程資料表，7 個欄位

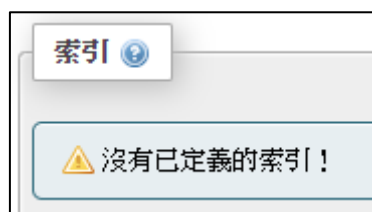
名稱	類型	長度/值	預設值	編碼與排序	屬性	空值 (Null)	索引	A.I.	備註
cId	VARCHAR	4	無			<input type="checkbox"/>	---	<input type="checkbox"/>	課程編號
cName	VARCHAR	10	無			<input type="checkbox"/>	---	<input type="checkbox"/>	課程名稱
credit	TINYINT	1	無			<input type="checkbox"/>	---	<input type="checkbox"/>	學分數
isCompulsory	BOOLEAN		無			<input type="checkbox"/>	---	<input type="checkbox"/>	必修
tId	VARCHAR	4	無			<input type="checkbox"/>	---	<input type="checkbox"/>	老師編號
created_at	INT		無			<input type="checkbox"/>	---	<input type="checkbox"/>	新增時間
updated_at	INT		無			<input type="checkbox"/>	---	<input type="checkbox"/>	更新時間

資料表備註: 課程資料表 編碼與排序: 儲存引擎: InnoDB

(圖) 課程資料表設計規格

SQL

```
CREATE TABLE `my_db`.`courses` ( `cId` VARCHAR(4) NOT NULL COMMENT '課程編號', `cName` VARCHAR(10) NOT NULL COMMENT '課程名稱', `credit` TINYINT(1) NOT NULL COMMENT '學分數', `isCompulsory` BOOLEAN NOT NULL COMMENT '必修', `tId` VARCHAR(4) NOT NULL COMMENT '老師編號', `created_at` INT NOT NULL COMMENT '新增時間', `updated_at` INT NOT NULL COMMENT '更新時間' ) ENGINE = InnoDB COMMENT = '課程資料表';
```



(圖) 沒有索引的提示

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註
<input checked="" type="checkbox"/> 1	<u>cId</u>	varchar(4)	utf8mb4_unicode_ci	否	無	無	課程編號
<input type="checkbox"/> 2	<u>cName</u>	varchar(10)	utf8mb4_unicode_ci	否	無	無	課程名稱
<input type="checkbox"/> 3	<u>credit</u>	tinyint(1)		否	無	無	學分數
<input type="checkbox"/> 4	<u>isCompulsory</u>	tinyint(1)		否	無	無	必選修
<input checked="" type="checkbox"/> 5	<u>tId</u>	varchar(4)	utf8mb4_unicode_ci	否	無	無	老師編號
<input type="checkbox"/> 6	<u>created_at</u>	datetime		否	無	無	新增時間
<input type="checkbox"/> 7	<u>updated_at</u>	datetime		否	無	無	更新時間

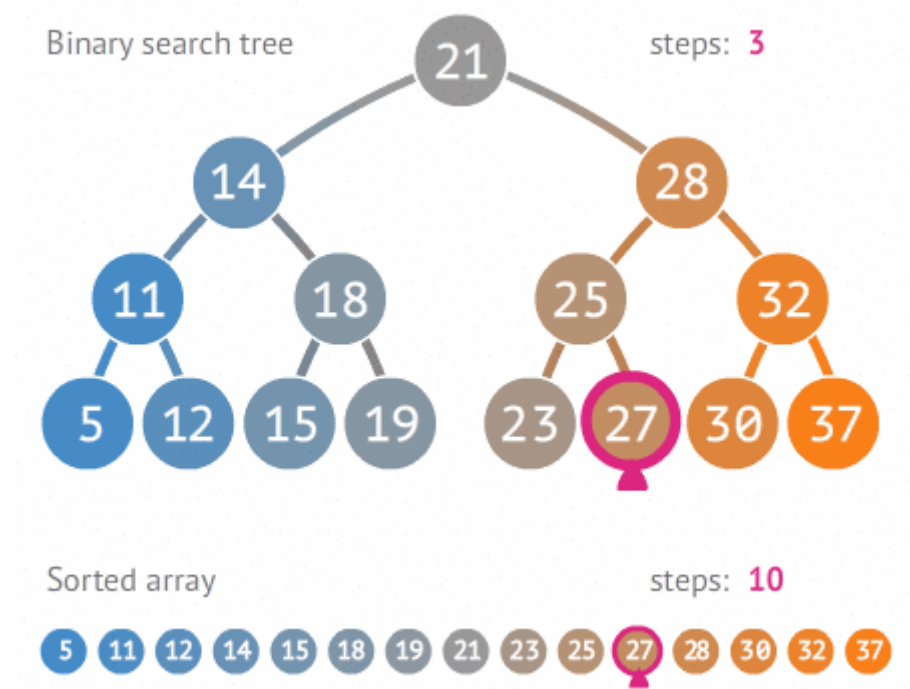
全選
 已選擇項目：
 瀏覽
 修改
 刪除
 主鍵
 獨一
 索引

(圖) 勾選 cId、tId 後，按下「主鍵」圖示

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註
1	<u>cId</u>	varchar(4)	utf8mb4_unicode_ci	否	無	無	課程編號
2	<u>cName</u>	varchar(10)	utf8mb4_unicode_ci	否	無	無	課程名稱
3	<u>credit</u>	tinyint(1)		否	無	無	學分數
4	<u>isCompulsory</u>	tinyint(1)		否	無	無	必選修
5	<u>tId</u>	varchar(4)	utf8mb4_unicode_ci	否	無	無	老師編號
6	<u>created_at</u>	datetime		否	無	無	新增時間
7	<u>updated_at</u>	datetime		否	無	無	更新時間

(圖) 課程資料表結構

建立索引的目的，在於提升資料查詢的效率，我們可以針對單獨的欄位建立索引，也可以勾選多欄位，成立索引。讓我們檢視學生資料表的結構，以及它的索引設定（主鍵也是一種索引）。



(圖) 建立索引後的資料儲存方式

students

資料表備註： 學生資料表

欄位	類型	空值 (Null)	預設值	連結到	備註	MIME
sId(主鍵)	varchar(3)	否			學生編號	
sName	varchar(20)	否			學生姓名	
sGender	varchar(1)	否			學生性別	
sNickname	varchar(50)	否			學生暱稱	
created_at	datetime	否	current_timestamp()		新增時間	
updated_at	datetime	否	current_timestamp()		更新時間	

索引

鍵名	類型	獨一	緊湊	欄位	基數	編碼與排序	空值(Null)	備註
PRIMARY	BTREE	是	否	sId	0	A	否	

我們可以設定單一欄位為索引：

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註
<input type="checkbox"/>	1 sId	varchar(3)	utf8mb4_unicode_ci	否	無		學生編號
<input type="checkbox"/>	2 sName	varchar(20)	utf8mb4_unicode_ci	否	無		學生姓名
<input type="checkbox"/>	3 sGender	varchar(1)	utf8mb4_unicode_ci	否	無		學生性別
<input checked="" type="checkbox"/>	4 sNickname	varchar(50)	utf8mb4_unicode_ci	否	無		學生暱稱
<input type="checkbox"/>	5 created_at	datetime		否	current_timestamp()		新增時間
<input type="checkbox"/>	6 updated_at	datetime		否	current_timestamp()		更新時間

已選擇項目：

(圖) 單一欄位建立索引

索引									
動作	鍵名	類型	獨一	緊湊	欄位	基數	編碼與排序	空值(Null)	備註
<input type="button" value="編輯"/> <input type="button" value="刪除"/>	PRIMARY	BTREE	是	否	sId	0	A	否	
<input type="button" value="編輯"/> <input type="button" value="刪除"/>	sNickname	BTREE	否	否	sNickname	0	A	否	

(圖) 除了主鍵，多了一個索引

SQL - 新增單一欄位索引

```
ALTER TABLE `students` ADD INDEX(`sNickname`);
```

SQL - 刪除單一欄位索引

```
ALTER TABLE `students` DROP INDEX `sNickname`;
```

我們也可以設定複數欄位為索引：

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註
<input type="checkbox"/>	1 sId	varchar(3)	utf8mb4_unicode_ci	否	無		學生編號
<input checked="" type="checkbox"/>	2 sName	varchar(20)	utf8mb4_unicode_ci	否	無		學生姓名
<input type="checkbox"/>	3 sGender	varchar(1)	utf8mb4_unicode_ci	否	無		學生性別
<input checked="" type="checkbox"/>	4 sNickname	varchar(50)	utf8mb4_unicode_ci	否	無		學生暱稱
<input type="checkbox"/>	5 created_at	datetime		否		current_timestamp()	新增時間
<input type="checkbox"/>	6 updated_at	datetime		否		current_timestamp()	更新時間

已選擇項目：

(圖) 複數欄位建立索引

索引									
動作	鍵名	類型	獨一	緊湊	欄位	基數	編碼與排序	空值(Null)	備註
<input type="button" value="編輯"/> <input type="button" value="刪除"/>	PRIMARY	BTREE	是	否	sId	0	A	否	
<input type="button" value="編輯"/> <input type="button" value="刪除"/>	sName	BTREE	否	否	sName	0	A	否	
					sNickname	0	A	否	

(圖) 除了主鍵，多了複數欄位索引

SQL - 新增複數欄位索引

ALTER TABLE `students` ADD INDEX(`sName`, `sNickname`);

SQL - 刪除複數欄位索引

ALTER TABLE `students` DROP INDEX `sName`;

五、新增、修改、刪除資料

格式

```
INSERT INTO `資料表名稱` (欄位 1, 欄位 2, 欄位 3, ..., 欄位 N)
VALUES (值 1, 值 2, 值 3, ..., 值 N),
      (值 1, 值 2, 值 3, ..., 值 N),
      (值 1, 值 2, 值 3, ..., 值 N);
```

sId 學生編號	sName 學生姓名	sGender 學生性別	sNickname 學生暱稱	created_at 新增時間	updated_at 更新時間
087	楊=	男	好人	2019-12-02 11:11:37	2019-12-02 11:11:37
088	陳=	女	小白	2019-12-02 11:11:37	2019-12-02 11:11:37

(圖) 學生資料表 - 新增前

SQL
<pre>INSERT INTO `students` (`sId`,`sName`,`sGender`,`sNickname`) VALUES ('003','王○○','男','小王'), ('004','江○○','女','小江'), ('005','周○○','女','小周'), ('006','黃○○','男','小黃'), ('007','丁○○','男','小丁'), ('008','鄭○○','男','小鄭');</pre>

sId 學生編號	sName 學生姓名	sGender 學生性別	sNickname 學生暱稱	created_at 新增時間	updated_at 更新時間
003	王○○	男	小王	2019-12-02 18:27:16	2019-12-02 18:27:16
004	江○○	女	小江	2019-12-02 18:27:16	2019-12-02 18:27:16
005	周○○	女	小周	2019-12-02 18:27:16	2019-12-02 18:27:16
006	黃○○	男	小黃	2019-12-02 18:27:16	2019-12-02 18:27:16
007	丁○○	男	小丁	2019-12-02 18:27:16	2019-12-02 18:27:16
008	鄭○○	男	小鄭	2019-12-02 18:27:16	2019-12-02 18:27:16
087	楊○○	男	好人	2019-12-02 11:11:37	2019-12-02 11:11:37
088	陳○○	女	小白	2019-12-02 11:11:37	2019-12-02 11:11:37

(圖) 學生資料表 - 新增後

格式
<pre>UPDATE `資料表名稱` SET `欄位 1` = 值 1, `欄位 2` = 值 2, `欄位 3` = 值 3 WHERE `欄位 A` = 值 A AND `欄位 B` = 值 B OR `欄位 C` = 值 C</pre>

tId 老師編號	tName 老師姓名	created_at 新增時間	updated_at 更新時間
T001	曾○○	2019-12-02 00:00:00	2019-12-02 00:00:00
T002	林○○	2019-12-02 00:00:00	2019-12-02 00:00:00
T003	王○○	2019-12-02 00:00:00	2019-12-02 00:00:00
T005	謝○○	2019-12-02 00:00:00	2019-12-02 00:00:00

(圖) 老師資料表 - 修改前

SQL

UPDATE

`teachers`

SET

`tName` = '黃○○'

WHERE

`tId` = 'T001'

tId 老師編號	tName 老師姓名
T001	黃○○
T002	林○○
T003	王○○
T005	謝○○

(圖) 老師料表 - 修改後

格式

DELETE FROM `資料表名稱`

WHERE `欄位 1` = 值 1,

AND `欄位 B` = 值 B

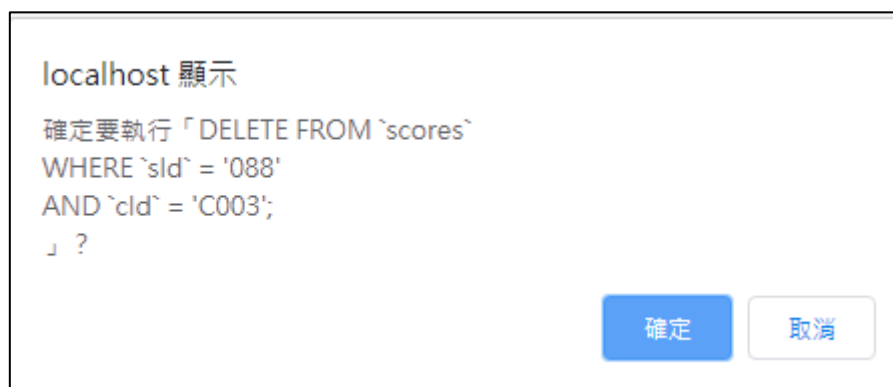
OR `欄位 C` = 值 C;

sId 學生編號	cId 課程編號	score 成績
087	C001	74
087	C002	93
088	C002	63
088	C003	82
088	C004	94

(圖) 成績資料表 - 刪除前

SQL

```
DELETE FROM `scores`
WHERE `sId` = '088'
AND `cId` = 'C003';
```



(圖) 跳出警示視窗

sId 學生編號	cId 課程編號	score 成績
087	C001	74
087	C002	93
088	C002	63
088	C004	94

(圖) 成績資料表 - 刪除後

六、資料庫基本查詢

格式
<pre>SELECT * FROM `資料表名稱` WHERE `欄位 A` = 值 1 AND `欄位 B` > 值 B, OR `欄位 C` != 值 C ORDER BY `欄位 D` ASC</pre>

格式
<pre>SELECT `欄位 1`,`欄位 2`,...`,`欄位 N` FROM `資料表名稱` WHERE `欄位 A` = 值 1 AND `欄位 B` > 值 B, OR `欄位 C` != 值 C ORDER BY `欄位 D` ASC</pre>

格式
<pre>SELECT `資料庫`.`資料表名稱`.`欄位 1`, `資料庫`.`資料表名稱`.`欄位 2`, ..., `資料庫`.`資料表名稱`.`欄位 N` FROM `資料庫`.`資料表名稱` WHERE `資料庫`.`資料表名稱`.`欄位 A` = 值 1 AND `資料庫`.`資料表名稱`.`欄位 B` > 值 B, OR `資料庫`.`資料表名稱`.`欄位 C` != 值 C ORDER BY `資料庫`.`資料表名稱`.`欄位 D` ASC</pre>

以上課的「my_db」資料為例：

M0: 列出所有的學生資料
<pre>SELECT * FROM `students`</pre>

sId 學生編號	sName 學生姓名	sGender 學生性別	sNickname 學生暱稱
003	王○○	男	小王
004	江○○	女	小江
005	周○○	女	小周
006	黃○○	男	小黃
007	丁○○	男	小丁
008	鄭○○	男	小鄭
087	楊○○	男	好人
088	陳○○	女	小白

(圖) M0 執行結果

M1: 列出學生的學號、姓名、暱稱

```
SELECT `sId`, `sName`, `sNickname`  
FROM `students`
```

sId 學生編號	sName 學生姓名	sNickname 學生暱稱
003	王○○	小王
004	江○○	小江
005	周○○	小周
006	黃○○	小黃
007	丁○○	小丁
008	鄭○○	小鄭
087	楊○○	好人
088	陳○○	小白

(圖) M1 執行結果

M2: 找出老師編號為 T001 或 T003 的老師姓名

```
SELECT `tName`  
FROM `teachers`  
WHERE `tId` = 'T001'  
OR `tId` = 'T003'
```

tName 老師姓名
黃○○
王○○

(圖) M2 執行結果

M3: 將學生依學生排序，號碼最大的排前面，全部列出來

```
SELECT *
FROM `students`
ORDER BY `students`.`sId` DESC
```

sId 學生編號	sName 學生姓名	sGender 學生性別	sNickname 學生暱稱
088	陳○○	女	小白
087	楊○○	男	好人
008	鄭○○	男	小鄭
007	丁○○	男	小丁
006	黃○○	男	小黃
005	周○○	女	小周
004	江○○	女	小江
003	王○○	男	小王

(圖) M3 執行結果

M4: 找出性別為女、暱稱為小白的學生姓名

```
SELECT `sName`
FROM `students`
WHERE `sGender` = '女'
AND `sNickname` = '小白';
```

M4A: 找出性別為女、暱稱為小白的學生姓名 (用 AS 來替代資料表名稱)

```
SELECT `s`.`sName`
FROM `students` AS `s`
WHERE `s`.`sGender` = '女'
AND `s`.`sNickname` = '小白';
```

補充說明：

「資料表」AS「資料表代稱」，其中 AS 可以省略。

sName 學生姓名
陳==

(圖) M4、M4A 執行結果

M5: 找出暱稱是「小」開頭的學生姓名與性別

```
SELECT `sName`, `sGender`
FROM `students`
WHERE `sNickname` like '小%'
```

M5A: 找出暱稱是「人」結尾的學生姓名與性別

```
SELECT `sName`, `sGender`
FROM `students`
WHERE `sNickname` like '%人'
```

sName 學生姓名	sGender 學生性別
王==	男
江==	女
周==	女
黃==	男
丁==	男
鄭==	男
陳==	女

(圖) M5 執行結果

sName 學生姓名	sGender 學生性別
楊==	男

(圖) M5A 執行結果

M6: 找出學號（三位數）第二碼為 8 的學生姓名與暱稱

```
SELECT `sName`, `sNickname`
FROM `students`
WHERE `sId` like '_8_';
```

sName 學生姓名	sNickname 學生暱稱
楊==	好人
陳==	小白

(圖) M6 執行結果

下面的關聯式資料庫設計有些難度，只要理解即可，或是對可以理解的部分練習即可。

練習用資料庫建立語法

```
-- phpMyAdmin SQL Dump
-- version 4.9.1
-- https://www.phpmyadmin.net/
--
-- 主機： 127.0.0.1
-- 產生時間：
-- 伺服器版本： 10.4.8-MariaDB
-- PHP 版本： 7.3.11

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

--
-- 資料庫： `company`
--
-- -----
--
-- 資料表結構 `department`
--

CREATE TABLE `department` (
  `Dname` varchar(15) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Dnumber` int(11) NOT NULL,
  `Mgr_ssn` char(9) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Mgr_start_date` date DEFAULT NULL,
  `created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
  `updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() COMMENT '結束時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='department';
```

```
--
-- 傾印資料表的資料 `department`
--

INSERT INTO `department` (`Dname`, `Dnumber`, `Mgr_ssn`, `Mgr_start_date`, `created_at`, `updated_at`) VALUES
('Headquarters', 1, '888665555', '1981-06-19', '2019-12-02 21:23:56', '2019-12-02 21:23:56'),
('Administration', 4, '987654321', '1995-01-01', '2019-12-02 21:23:56', '2019-12-02 21:23:56'),
('Research', 5, '333445555', '1988-05-22', '2019-12-02 21:23:56', '2019-12-02 21:23:56');

-----

--
-- 資料表結構 `dependent`
--

CREATE TABLE `dependent` (
  `Essn` char(9) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Dependent_name` varchar(15) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Sex` char(1) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `Bdate` date DEFAULT NULL,
  `Relationship` varchar(8) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
  `updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() COMMENT '修改時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='dependent';

--
-- 傾印資料表的資料 `dependent`
--

INSERT INTO `dependent` (`Essn`, `Dependent_name`, `Sex`, `Bdate`, `Relationship`, `created_at`, `updated_at`) VALUES
('123456789', 'Alice', 'F', '1988-12-30', 'Daughter', '2019-12-02 21:45:14', '2019-12-02 21:45:14'),
('123456789', 'Elizabeth', 'F', '1967-05-05', 'spouse', '2019-12-02 21:45:14', '2019-12-02 21:45:14'),
('123456789', 'Michael', 'M', '1988-01-04', 'Son', '2019-12-02 21:45:14', '2019-12-02 21:45:14'),
('333445555', 'Alice', 'F', '1986-04-05', 'Daughter', '2019-12-02 21:41:37', '2019-12-02 21:41:37'),
('333445555', 'Joy', 'F', '1958-05-03', 'Spouse', '2019-12-02 21:45:14', '2019-12-02 21:45:14'),
('333445555', 'Theodore', 'M', '1983-10-25', 'Son', '2019-12-02 21:41:37', '2019-12-02 21:41:37'),
('987654321', 'Abner', 'M', '1942-02-28', 'Spouse', '2019-12-02 21:45:14', '2019-12-02 21:45:14');
```

```
-- -----  
  
--  
-- 資料表結構 `dept_locations`  
--  
  
CREATE TABLE `dept_locations` (  
  `Dnumber` int(11) NOT NULL,  
  `Dlocation` varchar(15) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',  
  `updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() COMMENT '更新時間'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='dept_locations';  
  
--  
-- 傾印資料表的資料 `dept_locations`  
--  
  
INSERT INTO `dept_locations` (`Dnumber`, `Dlocation`, `created_at`, `updated_at`) VALUES  
(1, 'Houston', '2019-12-02 21:25:12', '2019-12-02 21:25:12'),  
(4, 'Stafford', '2019-12-02 21:25:12', '2019-12-02 21:25:12'),  
(5, 'Bellaire', '2019-12-02 21:25:12', '2019-12-02 21:25:12'),  
(5, 'Houston', '2019-12-02 21:25:12', '2019-12-02 21:25:12'),  
(5, 'Sugarland', '2019-12-02 21:25:12', '2019-12-02 21:25:12');  
  
-- -----  
  
--  
-- 資料表結構 `employee`  
--  
  
CREATE TABLE `employee` (  
  `Fname` varchar(15) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '名',  
  `Minit` char(1) COLLATE utf8mb4_unicode_ci DEFAULT NULL,  
  `Lname` varchar(15) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '姓',  
  `Ssn` char(9) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `Bdate` date DEFAULT NULL COMMENT '生日',  
  `Address` varchar(30) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT '地址',  
  `Sex` char(1) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT '性別',
```

```
`Salary` decimal(10,2) DEFAULT NULL COMMENT '薪資',
`Super_ssn` char(9) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
`Dno` int(11) NOT NULL,
`created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
`updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() COMMENT '更新時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='EMPLOYEE';

--
-- 傾印資料表的資料 `employee`
--

INSERT INTO `employee` (`Fname`, `Minit`, `Lname`, `Ssn`, `Bdate`, `Address`, `Sex`, `Salary`, `Super_ssn`, `Dno`, `created_at`,
`updated_at`) VALUES
('John', 'B', 'Smith', '123456789', '1965-01-09', '731 Fondren, Houston, TX', 'M', '30000.00', '333445555', 5, '2019-12-02 21:11:36',
'2019-12-02 21:11:36'),
('Franklin', 'T', 'Wong', '333445555', '1955-12-08', '638 Voss, houston, TX', 'M', '40000.00', '888665555', 5, '2019-12-02 21:14:28',
'2019-12-02 21:14:28'),
('Joyce', 'A', 'English', '453453453', '1972-07-31', '5631 Rice, Houston, TX', 'F', '25000.00', '333445555', 5, '2019-12-02 21:22:14',
'2019-12-02 21:22:14'),
('Ramesh', 'K', 'Narayan', '666884444', '1962-09-15', '975 Fire Oak, Humble, TX', 'M', '38000.00', '333445555', 5, '2019-12-02
21:22:14', '2019-12-02 21:22:14'),
('James', 'E', 'Borg', '888665555', '1937-11-10', '450 Stone, Houston, TX', 'M', '55000.00', NULL, 1, '2019-12-02 21:22:14', '2019-12-
02 21:22:14'),
('Jennifer', 'S', 'Wallace', '987654321', '1941-06-20', '291 Berry, Bellaire, TX', 'F', '43000.00', '888665555', 4, '2019-12-02 21:22:14',
'2019-12-02 21:22:14'),
('Ahmad', 'V', 'Jabbar', '987987987', '1969-03-29', '980 Dallas, Houston, TX', 'M', '250000.00', '987654321', 4, '2019-12-02 21:22:14',
'2019-12-02 21:22:14'),
('Alicia', 'J', 'Zelaya', '999887777', '1968-01-19', '3321 Castle, Spring, TX', 'F', '25000.00', '987654321', 4, '2019-12-02 21:14:28',
'2019-12-02 21:14:28');

--
-----
--
-- 資料表結構 `project`
--

CREATE TABLE `project` (
  `Pname` varchar(15) COLLATE utf8mb4_unicode_ci NOT NULL,
```

```
`Pnumber` int(11) NOT NULL,
`Plocation` varchar(15) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
`Dnum` int(11) NOT NULL,
`created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
`updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() COMMENT '更新時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='project';

--
-- 傾印資料表的資料 `project`
--

INSERT INTO `project` (`Pname`, `Pnumber`, `Plocation`, `Dnum`, `created_at`, `updated_at`) VALUES
('ProductX', 1, 'Bellaire', 5, '2019-12-02 21:36:29', '2019-12-02 21:36:29'),
('ProductY', 2, 'Sugarland', 5, '2019-12-02 21:36:29', '2019-12-02 21:36:29'),
('ProductZ', 3, 'Houston', 5, '2019-12-02 21:36:29', '2019-12-02 21:36:29'),
('Computerization', 10, 'Stafford', 4, '2019-12-02 21:36:29', '2019-12-02 21:36:29'),
('Reorganization', 20, 'Houston', 1, '2019-12-02 21:36:29', '2019-12-02 21:36:29'),
('Newbenefits', 30, 'Stafford', 4, '2019-12-02 21:36:29', '2019-12-02 21:36:29');

--
-----
--
-- 資料表結構 `works_on`
--

CREATE TABLE `works_on` (
  `Essn` char(9) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Pno` int(11) NOT NULL,
  `Hours` decimal(3,1) DEFAULT NULL,
  `created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
  `updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() COMMENT '修改時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='works_on';

--
-- 傾印資料表的資料 `works_on`
--

INSERT INTO `works_on` (`Essn`, `Pno`, `Hours`, `created_at`, `updated_at`) VALUES
```

```
('123456789', 1, '32.5', '2019-12-02 21:26:47', '2019-12-02 21:26:47'),
('123456789', 2, '7.5', '2019-12-02 21:26:47', '2019-12-02 21:26:47'),
('333445555', 2, '10.0', '2019-12-02 21:29:38', '2019-12-02 21:29:38'),
('333445555', 3, '10.0', '2019-12-02 21:29:38', '2019-12-02 21:29:38'),
('333445555', 10, '10.0', '2019-12-02 21:29:38', '2019-12-02 21:29:38'),
('333445555', 20, '10.0', '2019-12-02 21:29:38', '2019-12-02 21:29:38'),
('453453453', 1, '20.0', '2019-12-02 21:26:47', '2019-12-02 21:26:47'),
('453453453', 2, '20.0', '2019-12-02 21:26:47', '2019-12-02 21:26:47'),
('666884444', 3, '40.0', '2019-12-02 21:26:47', '2019-12-02 21:26:47'),
('888665555', 20, '0.0', '2019-12-02 21:33:12', '2019-12-02 21:33:12'),
('987654321', 20, '15.0', '2019-12-02 21:33:12', '2019-12-02 21:33:12'),
('987654321', 30, '20.0', '2019-12-02 21:33:12', '2019-12-02 21:33:12'),
('987987987', 10, '35.0', '2019-12-02 21:33:12', '2019-12-02 21:33:12'),
('987987987', 30, '5.0', '2019-12-02 21:33:12', '2019-12-02 21:33:12'),
('99887777', 30, '30.0', '2019-12-02 21:29:38', '2019-12-02 21:29:38'),
('999887777', 10, '10.0', '2019-12-02 21:33:12', '2019-12-02 21:33:12');
```

```
--
```

```
-- 已傾印資料表的索引
```

```
--
```

```
--
```

```
-- 資料表索引 `department`
```

```
--
```

```
ALTER TABLE `department`
```

```
ADD PRIMARY KEY (`Dnumber`);
```

```
--
```

```
-- 資料表索引 `dependent`
```

```
--
```

```
ALTER TABLE `dependent`
```

```
ADD PRIMARY KEY (`Essn`,`Dependent_name`);
```

```
--
```

```
-- 資料表索引 `dept_locations`
```

```
--
```

```
ALTER TABLE `dept_locations`
```

```
ADD PRIMARY KEY (`Dnumber`,`Dlocation`);
```

```
--  
-- 資料表索引 `employee`  
--  
ALTER TABLE `employee`  
    ADD PRIMARY KEY (`Ssn`);  
  
--  
-- 資料表索引 `project`  
--  
ALTER TABLE `project`  
    ADD PRIMARY KEY (`Pnumber`),  
    ADD UNIQUE KEY `Pname` (`Pname`);  
  
--  
-- 資料表索引 `works_on`  
--  
ALTER TABLE `works_on`  
    ADD PRIMARY KEY (`Essn`,`Pno`);  
COMMIT;
```

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Q0: 查詢名叫「John B. Smith」的員工，他的生日與住址

```
SELECT `Bdate`, `Address`
FROM `employee`
WHERE `Fname` = 'John' AND `Minit` = 'B' AND `Lname` = 'Smith';
```

Q0A: 查詢名叫「John B. Smith」的員工，他的生日與住址 (欄位使用 * 號)

```
SELECT *
FROM `employee`
WHERE `Fname` = 'John' AND `Minit` = 'B' AND `Lname` = 'Smith';
```

Bdate 生日	Address 地址
1965-01-09	731 Fondren, Houston, TX

(圖) Q0 執行結果

Fname 名	Minit	Lname 姓	Ssn	Bdate 生日	Address 地址	Sex 性別	Salary 薪資	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000.00	333445555	5

(圖) Q0A 執行結果

Q1: 擷取在 Research 部門工作的所有員工姓名與住址

```
SELECT `Fname`, `Lname`, `Address`
FROM `employee`, `department`
WHERE `Dname` = 'Research' AND `Dnumber` = `Dno`;
```

*造成交叉乘積

Fname 名	Lname 姓	Address 地址
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, houston, TX
Joyce	English	5631 Rice, Houston, TX
Ramesh	Narayan	975 Fire Oak, Humble, TX

(圖) Q1 執行結果

Q2: 列出所有位在 Stafford 地點的計畫，其計畫編號、部門編號，以及部門經理的姓氏、住址和生日

```
SELECT `Pnumber`, `Dnum`, `Lname`, `Address`, `Bdate`
FROM `project`, `department`, `employee`
```

```
WHERE `Dnum` = `Dnumber`
AND `Mgr_ssn` = `Ssn`
AND `Plocation` = 'Stafford';
```

*造成交叉乘積

Pnumber	Dnum	Lname 姓	Address 地址	Bdate 生日
10	4	Wallace	291 Berry, Bellaire, TX	1941-06-20
30	4	Wallace	291 Berry, Bellaire, TX	1941-06-20

(圖) Q2 執行結果

處理模稜兩可的屬性名稱、別名、重新命名與值組變數：

Q1A: 擷取在 Research 部門工作的所有員工姓名與住址

```
SELECT `employee`.`Fname`, `employee`.`Lname`, `employee`.`Address`
FROM `employee`, `department`
WHERE `department`.`Dname` = 'Research'
AND `department`.`Dnumber` = `employee`.`Dno`;
```

*造成交叉乘積

Fname 名	Lname 姓	Address 地址
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, houston, TX
Joyce	English	5631 Rice, Houston, TX
Ramesh	Narayan	975 Fire Oak, Humble, TX

(圖) Q1A 執行結果

Q8: 擷取每一位員工的姓名與其直屬上司的姓名

```
SELECT `E`.`Fname`, `E`.`Lname`, `S`.`Fname`, `S`.`Lname`
FROM `employee` AS `E`, `employee` AS `S`
WHERE `E`.`Super_ssn` = `S`.`Ssn`
```

*造成交叉乘積

Q8: 擷取每一位員工的姓名與其直屬上司的姓名 (沒有 AS)

```
SELECT `E`.`Fname`, `E`.`Lname`, `S`.`Fname`, `S`.`Lname`
FROM `employee` `E`, `employee` `S`
WHERE `E`.`Super_ssn` = `S`.`Ssn`
```

*造成交叉乘積

Fname	Lname	Fname	Lname
John	Smith	Franklin	Wong
Franklin	Wong	James	Borg
Joyce	English	Franklin	Wong
Ramesh	Narayan	Franklin	Wong
Jennifer	Wallace	James	Borg
Ahmad	Jabbar	Jennifer	Wallace
Alicia	Zelaya	Jennifer	Wallace

(圖) Q8 執行結果

Q9: 在資料庫中選擇所有的 EMPLOYEE Ssn

```
SELECT `Ssn`
FROM `employee`
```

Ssn
123456789
333445555
453453453
666884444
888665555
987654321
987987987
999887777

(圖) Q9 執行結果

Q10: 選擇 EMPLOYEE Ssn 與 DEPARTMENT Dname 的所有組合

```
SELECT `Ssn`, `Dname`
FROM `employee`, `department`
```

*造成交叉乘積

Ssn	Dname
123456789	Headquarters
123456789	Administration
123456789	Research
333445555	Headquarters
333445555	Administration
333445555	Research
453453453	Headquarters
453453453	Administration
453453453	Research
666884444	Headquarters
666884444	Administration
666884444	Research
888665555	Headquarters
888665555	Administration
888665555	Research
987654321	Headquarters
987654321	Administration
987654321	Research
987987987	Headquarters
987987987	Administration
987987987	Research
999887777	Headquarters
999887777	Administration
999887777	Research

(圖) Q10 執行結果

Q11: 擷取每一位員工的薪資

```
SELECT ALL `Salary`
FROM `employee`;
```

Salary 薪資
30000.00
40000.00
25000.00
38000.00
55000.00
43000.00
25000.00
25000.00

(圖) Q11 執行結果

Q11A: 擷取所有不同的薪資值

```
SELECT DISTINCT `Salary`
FROM `employee`
```

Salary 薪資
30000.00
40000.00
25000.00
38000.00
55000.00
43000.00

(圖) Q11A 執行結果

Q4A: 列出姓氏為 Smith 的員工所參與的所有計畫編號，不論該員工是計畫的工人員，或是管理計畫的部門經理

```
(SELECT DISTINCT `Pnumber`
FROM `project`,`department`,`employee`
WHERE `Dnum` = `Dnumber`
AND `Mgr_ssn` = `Ssn`
AND `Lname` = 'Smith')
```

UNION

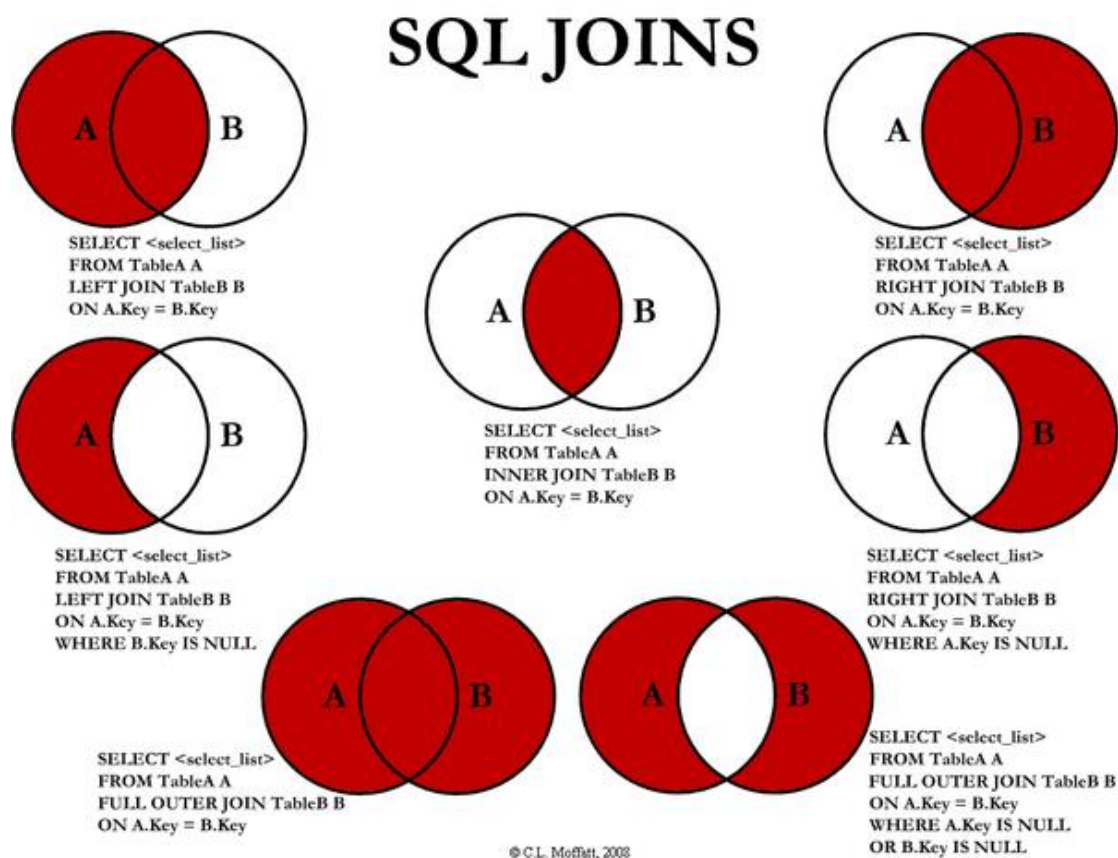
```
(SELECT DISTINCT `Pnumber`
FROM `project`,`works_on`,`employee`
WHERE `Pnumber` = `Pno`
AND `Essn` = `Ssn`
AND `Lname` = 'Smith')
```

*造成交叉乘積

Pnumber
1
2

(圖) Q4A 執行結果

七、進階查詢



(圖) 資料表結合的常見態樣

格式

```
SELECT `欄位 1`, `欄位 2`
FROM `資料表 1` INNER JOIN `資料表 2`
ON `資料表 1`.`欄位 A` = `資料表 2`.`欄位 A`
```

說明：

INNER JOIN 兩個資料表進行內部結合
 LEFT JOIN 以左邊資料表為主，進行結合
 RIGHT JOIN 以右邊資料表為主，進行結合
 OUTER JOIN

M0: 查詢各個課程中的授課老師姓名

```
SELECT `cName`, `tName`
FROM `courses` INNER JOIN `teachers`
```

```
ON `courses`.`tId` = `teachers`.`tId`
```

cName 課程名稱	tName
程式設計	黃○○
網頁設計	林○○
視覺設計	王○○
網路教學	謝○○

(圖) M0 執行結果

M1: 列出所有擁有成績的學生姓名與課程名稱

```
SELECT `sName`, `cName`, `score`
FROM `scores`
LEFT JOIN `students`
ON `scores`.`sId` = `students`.`sId`
LEFT JOIN `courses`
ON `scores`.`cId` = `courses`.`cId`
```

sName	cName	score 成績
楊○○	程式設計	74
楊○○	網頁設計	93
陳○○	網頁設計	63
陳○○	網路教學	94

(圖) M1 執行結果

接下來，讓我們使用 JOIN，來修改先前的範例。

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Q1: 擷取在 Research 部門工作的所有員工姓名與住址

```
SELECT `Fname`, `Lname`, `Address`
FROM `employee`, `department`
WHERE `Dname` = 'Research' AND `Dnumber` = `Dno`;

SELECT `employee`.`Fname`, `employee`.`Lname`, `employee`.`Address`
FROM `employee` INNER JOIN `department`
ON `employee`.`Dno` = `department`.`Dnumber`
WHERE `department`.`Dname` = 'Research'
```

Fname 名	Lname 姓	Address 地址
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, houston, TX
Joyce	English	5631 Rice, Houston, TX
Ramesh	Narayan	975 Fire Oak, Humble, TX

(圖) Q1 執行結果

Q2: 列出所有位在 Stafford 地點的計畫，其計畫編號、部門編號，以及部門經理的姓氏、住址和生日

```
SELECT `Pnumber`, `Dnum`, `Lname`, `Address`, `Bdate`
FROM `project`, `department`, `employee`
WHERE `Dnum` = `Dnumber`
AND `Mgr_ssn` = `Ssn`
AND `Plocation` = 'Stafford';

SELECT `project`.`Pnumber`, `project`.`Dnum`, `employee`.`Lname`,
`employee`.`Address`, `employee`.`Bdate`
FROM `project`
INNER JOIN `department`
ON `project`.`Dnum` = `department`.`Dnumber`
INNER JOIN `employee`
ON `department`.`Mgr_ssn` = `employee`.`Ssn`
AND `project`.`Plocation` = 'Stafford'
```

Pnumber	Dnum	Lname 姓	Address 地址	Bdate 生日
10	4	Wallace	291 Berry, Bellaire, TX	1941-06-20
30	4	Wallace	291 Berry, Bellaire, TX	1941-06-20

(圖) Q2 執行結果

處理模稜兩可的屬性名稱、別名、重新命名與值組變數：

Q1A: 擷取在 Research 部門工作的所有員工姓名與住址

```
SELECT `employee`.`Fname`, `employee`.`Lname`, `employee`.`Address`
FROM `employee`, `department`
WHERE `department`.`Dname` = 'Research'
AND `department`.`Dnumber` = `employee`.`Dno`;
```

```
SELECT `employee`.`Fname`, `employee`.`Lname`, `employee`.`Address`
FROM `employee` INNER JOIN `department`
ON `department`.`Dnumber` = `employee`.`Dno`
WHERE `department`.`Dname` = 'Research'
```

Fname 名	Lname 姓	Address 地址
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, houston, TX
Joyce	English	5631 Rice, Houston, TX
Ramesh	Narayan	975 Fire Oak, Humble, TX

(圖) Q1A 執行結果

Q8: 擷取每一位員工的姓名與其直屬上司的姓名

```
SELECT `E`.`Fname`, `E`.`Lname`, `S`.`Fname`, `S`.`Lname`
FROM `employee` AS `E`, `employee` AS `S`
WHERE `E`.`Super_ssn` = `S`.`Ssn`
```

*造成交叉乘積

Q8: 擷取每一位員工的姓名與其直屬上司的姓名 (沒有 AS)

```
SELECT `E`.`Fname`, `E`.`Lname`, `S`.`Fname`, `S`.`Lname`
FROM `employee` `E`, `employee` `S`
WHERE `E`.`Super_ssn` = `S`.`Ssn`
```

```
SELECT `E`.`Fname`, `E`.`Lname`, `S`.`Fname`, `S`.`Lname`
FROM `employee` AS `E` INNER JOIN `employee` AS `S`
ON `E`.`Super_ssn` = `S`.`Ssn`
```

Fname	Lname	Fname	Lname
John	Smith	Franklin	Wong
Franklin	Wong	James	Borg
Joyce	English	Franklin	Wong
Ramesh	Narayan	Franklin	Wong
Jennifer	Wallace	James	Borg
Ahmad	Jabbar	Jennifer	Wallace
Alicia	Zelaya	Jennifer	Wallace

(圖) Q8 執行結果

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

(圖) EMPLOYEE 資料表的內容

由於該資料表同時有員工的 Ssn 和上級 Superior 的 Ssn (Super_ssn)，我們可以試著查詢每一個員工的頂頭上司是誰。

Q1: 找查詢每一個員工的頂頭上司是誰，包括員工本身的姓名、Ssn，以及上級的姓名 Ssn

```
SELECT `InterE`.`Fname` AS `InterE_Fname`, `InterE`.`Lname` AS `InterE_Lname`,
`InterE`.`Ssn` AS `InterE_Ssn`, `SuperE`.`Fname` AS `SuperE_Fname`,
`SuperE`.`Lname` AS `SuperE_Lname`, `SuperE`.`Ssn` AS `SuperE_Ssn`
FROM `employee` AS `InterE`
LEFT JOIN `employee` AS `SuperE`
ON `InterE`.`Super_ssn` = `SuperE`.`Ssn`
```

InterE_Fname	InterE_Lname	InterE_Ssn	SuperE_Fname	SuperE_Lname	SuperE_Ssn
John	Smith	123456789	Franklin	Wong	333445555
Franklin	Wong	333445555	James	Borg	888665555
Joyce	English	453453453	Franklin	Wong	333445555
Ramesh	Narayan	666884444	Franklin	Wong	333445555
James	Borg	888665555	NULL	NULL	NULL
Jennifer	Wallace	987654321	James	Borg	888665555
Ahmad	Jabbar	987987987	Jennifer	Wallace	987654321
Alicia	Zelaya	999887777	Jennifer	Wallace	987654321

(圖) Q1 執行結果

補充說明

from 和 join 均是用於指定需要從哪些表查詢數據。

- from 可以是一個資料表或多個資料表，如果是多個資料表則是生成一個笛卡爾集，會涉及到大量數據。所以通常在涉及到多個資料表的查詢時，通常通過 join 來拼接多個資料表。故如果資料表的數據行多時，則數據量很多，造成巨大的磁碟、記憶體開銷，當然查詢速度也會很慢。
- join 主要是通過多個資料表之間的外鍵關聯來進行拼接，注意用於拼接的鍵需要加上索引，如果沒有則 MySQL 也會默認加上，不過前提是外鍵和引用的主鍵需要是相同的資料型態如數字類型需要是相同的長度和均是有符號或無符號數，字符串類型長度可以不一樣。

以下額外補充聚合（aggregation）的概念：

格式 1

```
SELECT 函式(欄位)`  
FROM `資料表`
```

備註：也可以加上 WHERE 條件來過濾資料

M1: 查詢所有課程中，最大的學分數是多少

```
SELECT MAX(`credit`)  
FROM `courses`
```

MAX(`credit`)

4

(圖) M1 執行結果

格式 2

```
SELECT `欄位 1`,`欄位 2`, 函式(欄位 3`)  
FROM `資料表`  
WHERE `欄位 A` = `欄位 B`  
GROUP BY `欄位 1`,`欄位 2`  
HAVING 函式(欄位 3`) > 值
```

M2: 查詢各個學分數，相對應的課程各有幾個

```
SELECT `credit`, COUNT(`cId`)
FROM `courses`
GROUP BY `credit`
```

credit 學分數	COUNT(`cId`)
2	1
3	1
4	2

(圖) M2 執行結果

八、使用 MySQL 內建函式

運算子	作用
==	等於
<=>	安全的等於
<> (!=)	不等於
<=	小於等於
>=	大於等於
>	大於
IS NULL	判斷一個值是否為 NULL
IS NOT NULL	判斷一個值是否不為 NULL
LEAST	在有兩個或多個引數時,返回最小值
GREATEST	當有 2 或多個引數時，返回最大值
BETWEEN AND	判斷一個值是否落在兩個值之間

ISNULL	與 IS NULL 作用相同
IN	判斷一個值是否落在兩個值之間
NOT IN	判斷一個值不是 IN 列表中的任意一個值
LIKE	萬用字元匹配
REGEXP	正規表示式匹配

運算子	作用
NOT 或者 !	邏輯非
AND 或者 &&	邏輯與
OR 或者 	邏輯或
XOR	邏輯互斥或

控制流函式	說明
CASE()	<p>SELECT CASE column WHEN a THEN b WHEN c THEN d ELSE e END</p> <p>若 column 等於 a，則返回 b 若 column 等於 c，則返回 d 否則返回 e 另外，a 與 c 可為運算式，例如 a<10，只要是 a 小於 10 都返回 b</p>
IF()	<p>SELECT IF(x, y, z) IF()返回一個數字或字符串值。 如果 x 是 TRUE (x 不為 0 且 x 不為 NULL)，那麼 IF() 返回 y，否則它返回 z。 類似 PHP 中的 三元運算子</p>

IFNULL()	<p>SELECT IFNULL(x, y)</p> <p>如果 x 是 NULL，IFNULL() 返回 y，否則它返回 x。</p> <p>IFNULL() 回傳一個數字或字串值。</p>
NULLIF()	<p>SELECT NULLIF(x, y)</p> <p>如果 x 等於 y，則結果回傳 NULL，否則回傳 x。</p> <p>效果等同於</p> <p>CASE WHEN expr1 = expr2 THEN NULL ELSE expr1 END</p>

CASE()
<pre>SELECT `sId`, `cId`, `score`, CASE WHEN `score` >= 90 THEN '成績在 90 分以上' WHEN `score` < 90 AND `score` >= 70 THEN '成績在 70 到 89 分之間' ELSE '成績低於 70' END AS `msg` FROM `scores`</pre>

sId 學生編號	cId 課程編號	score 成績	msg
087	C001	74	成績在70到89分之間
087	C002	93	成績在90分以上
088	C002	63	成績低於70
088	C004	94	成績在90分以上

(圖) CASE() 執行結果

IF()
<pre>SELECT `sId`, `cId`, `score`, IF(`score` >= 90, "大於 90 分", "低於 90 分") AS `msg` FROM `scores`;</pre>

sld 學生編號	cld 課程編號	score 成績	msg
087	C001	74	低於 90 分
087	C002	93	大於 90 分
088	C002	63	低於 90 分
088	C004	94	大於 90 分

(圖) IF() 執行結果

IFNULL()

SELECT IFNULL(NULL, "It's not NULL") AS `msg`;

msg
It's not NULL

(圖) IFNULL() 執行結果

NULLIF()

SELECT NULLIF(10, 10) AS `msg`;

msg
NULL

(圖) NULLIF() 執行結果