

# 第十三章 建立資料庫表與資料匯出入

# 閱讀本章之前



- 修改了管理者密碼及瞭解了MariaDB/MySQL連線方式後，這一章將瞭解兩個主題：
  - 第一個主題是「建立資料庫與資料表欄位規劃」
  - 第二個主題為「如何匯出入資料」
- MariaDB/MySQL中每個指令都是以分號來做結束。如果沒有輸入分號，指令不會執行，這也代表著您可以在指令當中換行。
- SQL 的敘述是不分大小寫的，但資料庫和資料表名稱則有大小寫的差異。

# 閱讀本章之前



- 建立資料表後就可以逐筆新增資料，但如果在其他機器上已有MariaDB/MySQL，資料是否需要重新輸入呢？
- MariaDB / MySQL支援資料的匯出與匯入作業，讓資料庫方便備份與匯入資料。
- MariaDB/MySQL提供的匯出入檔案預設副檔名為sql，檔案本身並沒有加密，所以請妥善保管您的資料檔案。

# 閱讀本章之前



- 本章會使用到終端機登入MariaDB/MySQL系統：
  - 您使用的是Windows環境，請您先確認是否依照12-2-1的說明開啟終端機進入MariaDB/MySQL的bin目錄。
  - 您使用的是Linux環境，請您先確認是否依照12-4-1的說明進入Linux主機。

# 建立資料庫



- 我們必須自己建立資料庫，才能在資料庫內建立資料表、建立或者匯入資料。
- 本節將分別以終端機與phpMyAdmin網頁兩種登入系統後建立資料庫。

# 以指令方式建立資料庫





- 請以「mysql -u帳號 -p密碼」方式登入到系統裡，例如以「mysql -uroot -pphpmysql」方式登入到系統。
- 建立資料庫之前可先使用「show databases;」瞭解系統目前有幾個資料庫。
- 而建立一個資料庫的語法為「create database 資料庫名稱 default character set 編碼;」。
- 欲建立一個名為「test3」的資料庫，請用「create database test3 default character set utf8;」語法建立。

# 以網頁方式建立MySQL資料庫



- 進入phpMyAdmin後可點選上方「資料庫」選項，再於「建立新資料庫」下方的文字框輸入資料庫名稱，並請挑選排序規則後點選「建立」鈕建立資料庫。
- 如果沒有挑選排序規則，預設是拉丁碼，這會對於後續資料表的規劃時預設編碼會有影響。

## 資料庫

 建立新資料庫 

# 欄位資料型態



- 建立了資料庫之後就可建立資料表，而建立資料表時也必須建立欄位。
- 我們在建立資料表之前，得先瞭解欄位的資料型態，否則資料表就無法順利的建立。
- 欄位資料形態可分為幾大類：
  - 數字
  - 文字
  - 日期時間
  - 列舉



# 建立資料表的三個原則



- 資料表規劃欄位時的三個原則：
  - 單位最小化
  - 需計算取得的資料不要成為欄位
  - 一個資料表只能有一個決定其他欄位資料的主索引欄位
- 「單位最小化」是指欄位內容必須是單一內容，例如學生資料表內的「科系班別」就應該要拆為「科系」與「班別」兩個欄位，當「科系」名稱變更(例如保險學系改為風險管理學系)或者「班別」名稱變更(例如A改為甲)時資料修改不容易出錯。

# 建立資料表的三個原則



- 「需計算取得的資料不要成為欄位」代表您可用MySQL的函數於查詢時計算取得即可，例如「年齡」就不用成為資料表的欄位，資料表內應該放的是「生日」，生日是固定的，但年齡會每年變動。
- 「一個資料表只能有一個決定其他欄位資料的主索引欄位」這一點是為了避免日後資料新增、修改與刪除時產生不必要的資料異常，建議您建立欄位時必須思考欄位之間的相依性，一個資料表只能有一個決定其他欄位資料的主索引欄位。如果有兩個以上的欄位可以決定其他欄位資料，建議您區分為兩個資料表。

# 資料表欄位：數字型態



- 數字類型裡標示「L」指的是「整數位數」，最大為255，也就是設定欄位時，若寫int代表依預設值可到10位數，若指定寫成int(8)則僅顯示8位數的數字。
- 數字類型裡標示「D」指的是「小數位數」，除了decimal(L,D)型態外，其他的資料型態沒有強制規定要加上L或D。
- unsigned屬性代表可設定為正整數，若設定時加上「zerofill」代表表示位數不足補0，如int(5)話，且設為zerofill，則存49這個數字，資料庫會將之存成00049。

# 資料表欄位：數字型態



- 數字型態欄位

類型	範圍	屬性選項
tinyint(L)	-128 到 127，unsigned 狀態則為 0 到 255	unsigned、zerofill
smallint(L)	-32768 到 32767，unsigned 狀態則為 0 到 65535	unsigned、zerofill
mediumint(L)	-8388608 到 8388607，unsigned 狀態則為 0 到 16777215	unsigned、zerofill
int(L)	-2147483648 到 2147483647，unsigned 狀態則為 0 到 4294967295	unsigned、zerofill
bigint(L)	-9223372036854775808 到 9223372036854775807，unsigned 狀態則為 0 到 18446744073709551615	unsigned、zerofill
float(L,D)	最小非零值： $\pm 1.175494351E - 38$	zerofill
double(L)	最小非零值： $\pm 2.2250738585072014E - 308$	zerofill
decimal(L,D)	可變範圍；其值的範圍依賴於 L 和 D	zerofill

# 資料表欄位：數字型態



- tinyint若為正整數型態也只能到255，所以適合超小數值資料，例如：成績。
- smallint就比tinyint大多了，正整數型態可到六萬多，所以適合小數值資料，例如：物品價錢。
- mediumint 的正整數可以計算到1600萬左右的數字，適合中型數值，例如：公司銷售金額統計。
- int的正整數已經可以用到42億左右的數字，您可評估網站資料是否需要用到「億」這樣的單位。
- 如果int的範圍還不夠，bigint的正整數已經可以用到18萬兆這麼大的數字，除非您的資料有使用到這麼大的單位，否則建議建立資料表時不要設計成這樣的欄位。

# 資料表欄位：數字型態



- float能夠記錄小數點，例如成績計算就可用此欄位設計。
- double和float的用途一樣，不過double所用掉的空間是float的兩倍，除非特別需要高精度或範圍極大的值，一般來說用float來儲存資料就夠了。
- decimal類型不同於float和double，其有效的取值範圍由L和D的值決定。如果改變L而固定D，則其取值範圍將隨L的變大而變大。

# 資料表欄位：數字型態



- 整數型態的欄位可以加上auto\_increment屬性。
- auto\_increment屬性具有以下特性：
  - 1.這個屬性在「Not Null」、「Primary Key」或「Unique」這三種屬性至少具備一種的時候才能使用
  - 2.一個資料表只能有一個auto\_increment屬性
  - 3.只有在新增紀錄時，在沒有明確指定數值及沒有指定Null值情況下才會自動產生新編號。您可以手動為新紀錄指定數值，只要您指定的數值沒有被用掉就可以

# 資料表欄位：日期時間型態



- 日期時間型態的欄位可表示年、月、日、時間：
  - Y代表「年」
  - YY代表「2位數的年」，如08年
  - YYYY代表「4位數的年」，如2008年
  - M代表「月」
  - D代表「日」
  - h代表「小時」
  - m代表「分」
  - s代表「秒」



# 資料表欄位：日期時間型態



- 日期時間型態欄位

類型	儲存方式
date	YYYY-MM-DD
datetime	YYYY-MM-DD hh:mm:ss
timestamp(L)	YYYYMMDDhhmmss
time	hh:mm:ss
year	YYYY

# 資料表欄位：日期時間型態



- date類型可儲存日期資料，若欲輸入的日期為「2013年10月12日」，以下的輸入方式均可接受：
  - 「2013/10/12」
  - 「2013-13-12」
  - 「13/10/12」
- 只要年月日之間有可以辨識的分隔符號就可以接受，或請依序輸入完整的年月日資料，例如「20131012」。

# 資料表欄位：日期時間型態



- timestamp與datetime類型相似，但timestamp可以設定時間顯示的寬度。

類型	顯示
timestamp(14)	yyyymmddhhmmss
timestamp(12)	yymmddhhmmss
timestamp(10)	yymmddhh
timestamp(8)	yyyymmdd
timestamp(6)	yymmdd
timestamp(4)	Yymm
timestamp(2)	Yy

# 資料表欄位：日期時間型態



- time類型只儲存時間資料，若欲輸入的時間為「15點7分20秒」，以下的輸入方式均可接受：
  - 「15:07:20」
  - 「15.07.20」
  - 「150720」
- 只要時分秒之間有可以辨識的分隔符號就可以接受。

# 資料表欄位：文字型態



- 字串型態的欄位設定均蠻相似的，這裡先列出基本規格，稍後再做比較。

類型	儲存範圍	用途
char(L)	$1 \leq L \leq 255$	固定長度字元
varchar(L)	$1 \leq L \leq 255$	變動長度字元
tinytext	255 個字元	微小型字串
text	65535 個字元	小型字串
mediumtext	16777215 個字元	中型字串
longtext	4294967295 個字元	大型字串
tinyblob	不超過 255bytes	微小型 blob
blob	不超過 65KB	小型 blob
mediumblob	不超過 16MB	中型 blob
longblob	不超過 4G	大型 blob

# 資料表欄位：文字型態



- char是固定長度字元資料型態。
- char(5)則只能儲存5個字元資料，若超出範圍則無法儲存。
- char(5)的欄位存入「pcschoo1」，則只剩下「pcsch」這五個字。
- 如果資料內容少於設定儲存的長度，char會自動補上空白字元。
- 無論存入的內容長短為何，char所需的儲存空間都是固定的。

# 資料表欄位：文字型態



- varchar是個長度可變的型態。
- varchar(5)的欄位遇到「abcdefg」的字串，那麼該欄位一樣只能存「abcde」，而且，他還要多用一個位元組來儲存資料長度，導致儲存「abcdef」會用到6個位元組。
- 如果儲存「abc」這樣的字串，那麼他會自動縮小所需空間，只要4個bytes就夠了。
- varchar與char最多只能設定到255。

# 資料表欄位：列舉型態

- 列舉（enumeration）型態可分為enum與set。
- 列舉（enumeration）型態值只能從固定的項目中挑選，不能隨心所欲的存入資料。
- 列舉型態共有兩種欄位格式如下：

類型	範圍	用途
enum	最多 65535 個選項	單選選項
set	最多 64 個選項	複選選項



# 資料表欄位：索引



- 資料表欄位若設定為index，可加快資料搜尋速度，但index建立後會占用儲存空間，資料增刪修時會跟著異動。
- 建議設定index的欄位長度是越短越好，且設定index的欄位長度若是固定長度會比變動長度效率更好。
- 資料表內欄位可依需求設定多個欄位為index欄位。

# 資料表欄位：索引



- Unique與Primary Key均具有index的特性，而兩者與index有若干不同的地方。
- Unique代表不重複索引或唯一索引，一個資料表內可以有許多個欄位設定為Unique。
- Primary Key代表主索引，一個資料表只能有一個Primary Key，設定為Primary Key欄位的資料不能重複且不能空白。

# 資料表的儲存引擎



- Mariadb/MySQL資料庫引擎主要有兩種常用的類型：MyISAM與InnoDB。
- MySQL 5.5版之後以及Mariadb預設資料表儲存引擎為InnoDB，而舊版MySQL預設資料表儲存引擎為MyISAM。

# MyISAM 特點



- MyISAM引擎不支援外鍵功能，也不支援交易功能。
- 由於MyISAM忽略了外鍵的限制，不用擔心資料表之間的關係，很適合初學者學習使用。
- MyISAM引擎資料表可執行全文檢索功能，但這功能目前不適合中文資料檢索。

# MyISAM 特點



- MyISAM引擎執行時整個資料表會鎖住(Table Lock)，因此新增或更新資料速度會較比較緩慢。
- 不支援外鍵功能也代表資料庫設計時您必須留意資料表之間的關係，以避免之後資料異常情形發生MyISAM遇到錯誤時必須完整掃描後才能重建索引或修正未寫入硬碟的錯誤，修復時間則與資料量的多寡成正比。

# InnoDB特點



- InnoDB設計資料表時可加入外鍵的設計，資料就可大幅簡化，以避免資料重複。
- 相對的InnoDB引擎資料表資料增減修改時必須遵守原則，若違反原則將產生錯誤。
- 執行時會鎖住正在編輯的該行資料(Row Lock)，所以在不考慮與其他資料表連結情況下，InnoDB引擎於大量新增與更新資料速度比MyISAM引擎快。
- InnoDB支援交易 (Transaction) 功能，並可藉由交易記錄檔來恢復程式崩潰或非預期結束所造成的資料錯誤。

# InnoDB特點



- 外鍵(Foreign key)是指資料表欄位的資料是來自於另外一個資料表的主鍵資料。
- 交易功能是指具備 ACID四項特性，可以用來確保資料庫系統的一致性以及正確性，也是資料庫實現並行控制、故障回復控制的基礎。
- ACID指Atomicity（不可分割性）、Consistency（一致性）、Isolation（隔離性）與Durability（持久性）。

# 以終端機方式建立資料表

- 建立資料表的指令為「Create Table 資料表名稱 (欄位定義) engine=資料表引擎 default charset=預設編碼;」。
- 欄位定義內的資料為「欄位名稱、資料型態、欄位寬度、[Null | NOT Null]、[Default 值]」，欄位間以「,」間隔。請依照以下範例輸入：

```
create table db2 (  
test1 varchar(10) not null ,  
test2 tinyint not null ,  
test3 date not null ,  
test4 blob not null  
) engine = myisam default charset=utf8;
```



# 以網頁方式建立資料表



- 進入資料庫後可於建立資料表下方輸入資料表名字與欄位數量，再請按下「執行」鈕。
- 按下「執行」鈕後可逐一規劃每一個欄位的名字、類型等項目，再請按下「儲存」鈕建立資料表。
- 我們可於test1資料庫內建立資料表：

欄位名稱	欄位型態	長度
test1	varchar	10
test2	tinyint	3
test3	date	
test4	blog	

# 以網頁方式建立資料表



- 以網頁方式規劃資料表內欄位屬性

The image shows the phpMyAdmin web interface. On the left, a sidebar lists databases: 'malaki', 'mysql', 'performance\_schema', 'phpmyadmin', 'record', 'record2', 'test', 'test1', 'test2', and 'test3'. The main area displays the 'db2' database structure. A blue arrow points from the 'test1' database entry in the sidebar to the 'db2' database structure view.

The 'db2' database structure view shows a table with the following columns:

名稱	型態	長度/值	預設值	索引與排序	屬性
test1	VARCHAR	10	無		
test2	TINYINT	3	無		
test3	DATE		無		
test4	BLOB		無		

Below the table, there are options for '資料表備註' (Table Comment), '編碼與排序' (Encoding and Collation), and '儲存引擎' (Storage Engine). The '編碼與排序' dropdown is set to 'utf8\_general\_ci', and the '儲存引擎' dropdown is set to 'InnoDB'. There is also a 'PARTITION 定義' (Partition Definition) section.

At the bottom, there are buttons for '預覽 SQL' (Preview SQL) and '儲存' (Save).

# 以網頁方式建立資料表



- 您也可於進入資料庫後按下「SQL」鈕後於多行文字框輸入建立資料表語法後按下「執行」鈕建立資料表。
- 這裡的語法請參考13-4-1節說明，請先確認是否於13-1-2節建立test2資料庫，請選擇test2資料庫後建立資料表。

A screenshot of a web-based database management interface. The browser address bar shows "伺服器: 127.0.0.1" and "資料庫: test2". The interface has a top navigation bar with buttons: "結構", "SQL", "搜尋", "查詢", "匯出", "匯入", "操作", "權限", "預存程序", and "更多". Below this is a section titled "在資料庫 test2 執行 SQL 查詢:". It contains a multi-line text area with the following SQL code:

```
1 create table db2 (  
2 test1 varchar(10) not null ,  
3 test2 tinyint(3) not null ,  
4 test3 date not null ,  
5 test4 blob not null  
6 ) engine = myisam default charset=utf8;
```

Below the text area are three buttons: "清除", "格式", and "取得自動儲存的查詢". There is a checkbox labeled "Bind 參數" with a help icon. Below that is a text input field labeled "將此 SQL 查詢加為書籤:". At the bottom, there is a row of checkboxes: "[ 指令分割符號 : ]" (with a text input field), "在此再次顯示此查詢" (checked), "繼續顯示查詢框", "當完成後還原", and "開啟外鍵(Foreign Key)檢查" (checked). A "執行" button is located at the bottom right.

# 資料匯出與匯入



- 在PHP存取資料庫之前，請先規劃好資料庫及資料表。資料表設計建議可依循以下的原則：
  - 1. 資料表內不應該包涵多餘或重複的資料，如果您需要反覆輸入相同的資料，代表資料表設計需做調整。
  - 2. 資料表命名應該有意義，不應該是class1、class2等方式命名。
  - 3. 資料表需要的儲存空間應該愈小愈好。
  - 4. 建議在正式上線之前先做各種資料的新增刪除更新測試，確保資料在更動時不會產生資料遺失的風險。

# 資料匯出與匯入



- 資料庫必須有資料才有其意義，資料來源有自行輸入或匯入資料，Mariadb/MySQL預設匯出入檔案副檔名為sql，檔案本身編碼必須與資料表設定的編碼相同。
- 檔案本身編碼若與資料表設定的編碼不同，匯出入資料後的操作容易出現亂碼或資料空白的情況。
- phpMyAdmin網頁提供了資料提供了輸入與匯出功能，匯出時有多種檔案格式可以挑選。

# 資料匯出與匯入



- 網頁操作雖然方便，但匯入資料是透過HTTP網路通訊協定處理檔案上傳，所以檔案上傳時受到Apache與PHP設定影響，會有兩個限制：
  - 連線時逾300秒沒有回應則視為斷線
  - 上傳檔案不能超過2MB
- phpMyAdmin網頁只能手動方式備份，無法作排程備份。
- 我們進行資料匯出入操作將採兩種方式：
  - 第一種是採用網頁方式匯出入資料
  - 第二種為終端機方式，這是主要匯出入資料使用的方式

# 以網頁方式登入進行資料輸入



- 請點選資料庫後再點選右方網頁的「輸入」選項後，選擇檔案進行資料的輸入。
- 請將list1.sql輸入到test1及test2兩個資料庫內，這個檔案內有三個資料表，匯入後可觀察這些資料表有何不同：
  - list資料表預設引擎為InnoDB及編碼設定為utf8
  - list2資料表預設引擎為MyISAM及編碼設定為utf8
  - list3資料表預設引擎為MyISAM但不設定編碼

# 以網頁方式登入進行資料輸入



- test1 資料庫預設編碼是utf8

The screenshot shows a web-based database management interface for a server at 127.0.0.1, specifically for the 'test1' database. The interface includes a top navigation bar with tabs for '結構' (Structure), 'SQL', '搜尋' (Search), '查詢' (Query), '匯出' (Export), '匯入' (Import), '操作' (Operations), '權限' (Permissions), '預存程序' (Stored Procedures), '事件' (Events), and '觸發器' (Triggers). Below this, a table lists the database's tables and their properties.

資料表	動作	資料列數	型態	編碼與排序	大小	資料分散
db2	★ 瀏覽 結構 搜尋 新增 清空 刪除	0	InnoDB	utf8_general_ci	16 KB	-
list	★ 瀏覽 結構 搜尋 新增 清空 刪除	2	InnoDB	utf8_general_ci	16 KB	-
list2	★ 瀏覽 結構 搜尋 新增 清空 刪除	2	MyISAM	utf8_general_ci	2.6 KB	-
list3	★ 瀏覽 結構 搜尋 新增 清空 刪除	2	MyISAM	utf8_general_ci	2.6 KB	-
4 張資料表 總計		6	InnoDB	utf8_general_ci	37.1 KB	0 B



# 以網頁方式登入進行資料輸入



- test2 資料庫預設編碼是拉丁

伺服器: 127.0.0.1 » 資料庫: test2

結構

SQL

搜尋

查詢

匯出

匯入

操作

權限

預存程序

事件

觸發器

資料表	動作	資料列數	型態	編碼與排序	大小	資料分散
list	<div><div>★</div><div>瀏覽</div><div>結構</div><div>搜尋</div><div>新增</div><div>清空</div><div>刪除</div></div>	2	InnoDB	utf8_general_ci	16 KB	-
list2	<div><div>★</div><div>瀏覽</div><div>結構</div><div>搜尋</div><div>新增</div><div>清空</div><div>刪除</div></div>	2	MyISAM	utf8_general_ci	2.6 KB	-
list3	<div><div>★</div><div>瀏覽</div><div>結構</div><div>搜尋</div><div>新增</div><div>清空</div><div>刪除</div></div>	2	MyISAM	latin1_swedish_ci	2.6 KB	-
3 張資料表 總計		6	InnoDB	latin1_swedish_ci	21.1 KB	0 B

# 以網頁方式登入進行資料輸入



- test2 資料庫的list3 資料成為亂碼

伺服器: 127.0.0.1 » 資料庫: test2 » 資料表: list3

瀏覽 結構 SQL 搜尋 新增

⚠ 目前選擇的項目不含唯一(Unique)的資料欄位，將無法：

✓ 顯示第 0 - 1 列 (總計 2 筆, 查詢花費 0.0004 秒。)

```
SELECT * FROM `list3`
```

☐ 全部顯示 | 資料列數: 25 | 搜尋資料列: 搜尋

依主鍵排序: 無

+ 選項

username	email	sex
??	jiannrong@gmail.com	?
???	jiannrongkimohk@yahoo.com.hk	?

# 以網頁方式登入進行資料輸入



- 請將record.sql輸入test1資料庫，record.sql是utf8編碼的SQL檔案，但這個SQL檔案超過2MB，所以進行輸入時將會產生錯誤。
- 另請匯入pcschool.sql檔案至test3資料庫內(資料表來源為光碟片內的「13」資料夾內的「pcschool.sql」)，稍後的資料匯出將會利用到test3資料庫內的資料。

# 以網頁方式登入進行資料匯出



- 登入phpMyAdmin後請選取欲輸出的資料庫或資料表，接著請選取右邊視窗上方的「匯出」，就可將選取的資料庫或資料表備份出來。
- 匯出檔案格式共有CodeGen、CSV、MS Excel的CSV格式、Microsoft Word 2000、JSON、LaTeX、MediaWiki表、OpenOffice表格、OpenOffice檔案、PDF、PHP陣列、SQL、Texy!文字、XML與YAML。

# 以網頁方式登入進行資料匯出



- 匯出時可選擇要儲存為檔案或者直接於網頁上顯示。
- 儲存為檔案時也可以選擇不壓縮(SQL檔案)或者zip檔案壓縮(Windows環境)、gzip或bzip檔案壓縮(Linux環境)方式下載。

正在匯出資料表 "list" 中的資料列

**匯出樣板：**

新樣板：  
樣板名稱

已存在的樣板：  
樣板： -- 選擇一個樣板 --

**匯出方式：**

☐ 快速 - 僅顯示必要的選項  
☐ 自訂 - 顯示所有可用的選項

**格式：**

CodeGen  
CSV  
CSV for MS Excel  
Microsoft Word 2000  
JSON  
LaTeX  
MediaWiki Table  
OpenDocument Spreadsheet  
OpenDocument Text  
PDF  
PHP array  
SQL  
Text/Text  
XML  
YAML

**輸出：**

☐ 重新命名已匯出的資料庫/資料表/欄位  
☐ 使用 LOCK TABLES 指令  
☒ 儲存為檔案

檔案名稱樣板:  ☒ 以後也使用此設定

檔案編碼:

壓縮:

☐ 直接顯示在文字區塊

略過資料表超過  MB

# 以網頁方式登入進行資料匯出



- 匯出時也可針對SQL檔案格式進行挑選：
  - 只匯出結構
  - 只匯出資料
  - 匯出資料與結構
- 您也可以選擇是否要與其他資料庫系統或舊版本MySQL最大程度相容，使您下載的SQL檔案可用於舊版MySQL或其他資料庫。

# 終端機方式進行資料匯出入之前



- phpMyAdmin雖可備份各種資料庫，但只能手動方式備份，無法作排程備份，而且匯入資料時會有檔案大小限制，Mariadb/MySQL有提供其他匯出與匯入的指令嗎？
- mysqldump是Mariadb/MySQL的終端機匯出指令，而mysql是終端機匯入指令，不論是Windows或Linux環境都有這個指令。

# 終端機方式進行資料匯出入之前

- 執行mysqldump或mysql指令時不需登入至MySQL終端機操作環境，所以可搭配Windows或Linux相關工具，做定時的備份或還原。
- 關於終端機開啟與目錄切換等動作請參考12-2-1節介紹。
- 匯入資料之前若需刪除原有的資料表，終端機登入系統並挑選資料庫後，可輸入以下指令就可以刪除資料表：

```
drop table 資料表名;
```



# mysqldump進行資料表資料匯出



- mysqldump是資料庫內資料匯出的語法，基本的語法結構：

```
mysqldump -u 帳號 -p 密碼 [參數] 資料庫名稱 [資料表名稱] > SQL 檔案
```

# mysqldump進行資料表資料匯出



- [參數]內「設定資料匯出入編碼」是必要的項目，其他參數與資料表名稱均可挑選。

常用參數	說明
--skip-opt	匯出資料時資料將會分行。
--no-data	只匯出資料表結構。
--no-create-info	只匯出資料表內容。
--default-character-set=utf8	設定編碼為 utf8。
--quick	匯出很多資料時可以使用，強制 mysqldump 從伺服器查詢取得記錄後直接輸出而不是取得所有記錄後將它們暫存起來後再做輸出。
--single-transaction	備份 InnoDB 引擎資料表資料。
--flush-logs	把目前 InnoDB 的 Log 匯出，請搭配--single-transaction 參數使用。
--master-data	匯出資料的 SQL 語法中加入匯出的時間點，請搭配--single-transaction 參數使用。

# mysqldump進行資料表資料匯出



- 匯出「test3」這一個資料庫內所有資料表欄位與內容，請於終端機輸入「**mysqldump -uroot -pphpmysql --default-character-set=utf8 test3 >test1.sql**」後，接著請以NotePad++開啟test1.sql，將可看到所有資料表欄位與內容。
- 上述匯出資料的方式，資料表內容不會分行論述，日後維護頗不方便的，因此您可以加上「--skip-opt」參數，讓資料都能逐筆分行列出：「**mysqldump -uroot -pphpmysql --skip-opt --default-character-set=utf8 test3 >test2.sql**」。

# mysqldump進行資料表資料匯出



- 若只需要匯出一個資料表，請在資料庫名稱後面以空格方式作間隔加上資料表名稱。例如匯出「test3」這一個資料庫內「list」資料表：「**mysqldump -uroot -pphpmysql --default-character-set=utf8 test3 list >test3.sql**」。
- 雖然只匯出指定資料表的欄位與內容，但資料表內容仍只寫在一行敘述內，所以請加上「**--skip-opt**」參數，讓資料都能逐筆分行列出：「**mysqldump -uroot -pphpmysql --skip-opt --default-character-set=utf8 test3 list >test4.sql**」。

# mysqldump進行資料表資料匯出



- 若要匯出多個資料表，請在資料庫名稱後面以空格方式作間隔加上資料表名稱即可。
- 例如我們匯出「test3」這一個資料庫內「customers」與「list」兩個資料表：「**mysqldump -uroot -pphpmysql --default-character-set=utf8 test3 customers list >test5.sql**」。

# mysqldump只匯出資料表結構或內容



- 資料表結構與內容可分別匯出，若只想匯出資料表結構，請加入「--no-data」參數，例如匯出「test3」這一個資料庫內所有資料表的結構：「**mysqldump -uroot -pphpmysql --default-character-set=utf8 --no-data test3 >test6.sql**」。
- 若只想匯出特定的資料表結構，除加入「--no-data」參數外，也請在資料庫名稱之後加上資料表名稱即可。例如匯出「test3」這一個資料庫內「list」資料表的結構：「**mysqldump -uroot -pphpmysql --default-character-set=utf8 --no-data test3 list>test7.sql**」。

# mysqldump只匯出資料表結構或內容



- 若只要匯出資料表的內容，請加入「--no-create-info」參數，例如匯出「test3」這一個資料庫內所有資料表的內容：  
「**mysqldump -uroot -pphpmysql --default-character-set=utf8 --no-create-info test3 >test8.sql**」。
- 而上述參數只匯出資料表內容，而每一筆資料並未進行分行，所以還可加上「--skip-opt」參數，讓資料都能逐筆分行列出：「**mysqldump -uroot -pphpmysql --default-character-set=utf8 --no-create-info --skip-opt test3 >test9.sql**」。
- 「--no-create-info」與「--skip-opt」兩個參數彼此執行的結果並沒有衝突，所以兩個參數的順序不影響執行的結果。

# mysql匯入資料



- mysql是資料庫內資料匯入的語法，基本的語法結構為

```
mysql -u 帳號 -p 密碼 --default-character-set=編碼 資料庫名稱<SQL 檔案
```



# mysql匯入資料



- 匯入資料時需指定資料庫名稱，所以匯入前必須建立資料庫。
- 建立資料庫的語法請參考12-1節介紹。
- 請察看「record.sql」檔案，這個SQL檔案有50MB之多，無法透過phpMyAdmin的方式匯入，只能使用終端機的方式將檔案匯入到系統內，請輸入以下的指令：「**mysql -uroot -pphpmysql --default-character-set=utf8 test1 < record.sql**」，就可將資料匯入到test1資料庫內。
- 匯入後您可利用phpMyAdmin察看，這個資料表內有1,286,638筆資料。

# 範例實作



- 第4章我們建立了BMI分析、班級通訊錄系統資料輸入、名片管理系統資料輸入三個表單網頁，現在請依前面各節介紹，建立新的資料庫，再於資料庫內新增幾個資料表，儲存這三個表單網頁輸入資料後的內容。
- 我們想要設計一個留言版系統，可以儲存留言者主旨、內容、留言者姓名、留言者電子信箱與留言者的留言日期時間與ip位址，請思考並規劃這個系統所需的資料表欄位。

# 問題與討論



- 1.mysql終端機登入方式有哪幾種？
- 2.請使用終端機與網頁登入方式分別建立資料庫pcschool、pcschool2。
- 3.請說明欄位的auto\_increment屬性特性。

# 問題與討論



- 4.請分別以終端機與網頁登入方式進入MySQL系統，分別於pcschool與pcschool2資料庫內建立一個資料表，名稱為「gboard」，內有六個欄位，分別為

欄位名稱	欄位型態	欄位長度	預設值
boardid	tinyint		NOT NULL
yourname	varchar	20	NOT NULL
youremail	varchar	25	NOT NULL
ip	char	15	NOT NULL
ctime	datetim		NOT NULL
content	text		NOT NULL

# 問題與討論



- 5.請將pcschool.sql匯入pcschool資料庫內。
- 6.匯出pcschool資料庫內employees資料表，且資料會逐筆分行列出。
- 7.請匯出pcschool資料庫內「orders」與「employees」兩個資料表結構。
- 8.請匯出pcschool資料庫內「orders」與「employees」兩個資料表內容。