

有鑑於目前市面上大部分同類書籍都缺乏相關理論基礎的說明，讓讀者就算學會了 SQL 語法和操作，仍然缺乏理論基礎的支援，而無法真正融會貫通。所以本書在內容上完美結合理論與實務，不只提供實際正規化和資料庫設計範例，更使用大量圖例和表格來說明相關理論和觀念，讓讀者不只能夠輕鬆學習資料庫系統的相關理論，更可以實際在 MySQL/MariaDB 建立資料庫設計成果的資料庫來驗證所學。

如何閱讀本書

本書章節架構上，廣泛參閱國內外資料庫設計與開發的相關書籍，以符合國內實際資料庫環境來規劃本書內容，全書共分為五篇 17 個章節，以循序漸進方式來詳細說明 MySQL/MariaDB 資料庫系統設計與開發。

第一篇：資料庫理論與 MySQL/MariaDB 的基礎

在第 1 章說明資料庫定義、ANSI/SPARC 三層資料庫系統架構的資料庫系統、資料庫綱要、資料庫管理師負責的工作和處理架構，第 2 章說明資料庫模型和關聯式資料庫模型，然後在第 3 章說明實體關聯模型與正規化，以便讀者擁有完整資料庫理論的基礎，第 4 章說明 MySQL/MariaDB 資料庫系統的安裝和基本操作，並且詳細說明 MySQL Workbench 圖形介面管理工具的使用。

第二篇：建立 MySQL/MariaDB 資料庫與資料表

第 5 章說明如何使用 MySQL Workbench 內建的資料庫設計工具，可以幫助我們繪製實體關聯圖，在第 6 章說明什麼是 SQL 語言後，詳細說明 MySQL/MariaDB 的字元集和定序，接著分別使用 MySQL Workbench 或 SQL 指令來建立、修改和刪除使用者資料庫，最後是資料庫的備份與還原，第 7 章說明資料類型後，開始建立資料表、資料庫儲存引擎和完整性限制條件，最後說明暫存資料表。

第三篇：SQL 語言的 DML 指令

在第 8 章是單一資料表的查詢和群組查詢，第 9 章是多資料表查詢的合併、集合和子查詢，第 10 章是 DML 語言的 INSERT、UPDATE 和 DELETE 指令，說明如何在資料表新增、更新和刪除記錄資料。

第四篇：MySQL/MariaDB 檢視表與索引

第 11 章說明如何在 MySQL/MariaDB 資料庫建立檢視表，第 12 章是資料表索引規劃和建立，包含索引結構、MySQL/MariaDB 自動建立的索引、如何建立資料表的索引和分析索引效率。

第五篇：ChatGPT X MySQL/MariaDB 程式設計與用戶端 程式開發

第 13~15 章是 MySQL/MariaDB 資料庫的 SQL 程式設計，詳細說明 SQL 程式化功能的語法、如何建立預存程序、函數、觸發程序、資料指標、參數化查詢和進行交易處理。第 16 章說明如何使用 Python 和 PHP 語言來建立資料庫的用戶端程式。第 17 章說明如何使用 ChatGPT 學習 MySQL、寫出 SQL 指令敘述與建立資料庫程式。

附錄 A 詳細說明 MySQL 內建函數和 JSON 欄位處理，附錄 B 說明如何安裝 MariaDB 資料庫系統和基本使用。

編著本書雖力求完美，但學識與經驗不足，謬誤難免，尚祈讀者不吝指正。

陳會安 Joe Chen 於台北

hueyan@ms2.hinet.net

2023.7.20



Chapter

6

SQL 語言與資料庫建置

6-1 | SQL 語言的基礎

SQL 語言是一種第四代程式語言，可以用來查詢或編輯關聯式資料庫的記錄資料，它是 70 年代由 IBM 公司研發，並且在 1986 年成為 ANSI 標準的一種關聯式資料庫語言。

6-1-1 SQL 結構化查詢語言

「SQL」（Structured Query Language）的全名是結構化查詢語言，在本書簡稱為 SQL 語言。SQL 語言是在 1980 年成為「ISO」（International Organization for Standardization）和「ANSI」（American National Standards Institute）的標準資料庫語言，其版本分為 1989 年的 ANSI-SQL 89 和 1992 年制定的 ANSI-SQL 92，也稱為 SQL 2，這是目前關聯式資料庫的標準語言，ANSI-SQL 99 稱為 SQL 3，適用在物件關聯式資料庫的 SQL 語言。

早在 1970 年，E. F. Codd 建立關聯式資料庫模型時，就提出一種構想的資料庫語言，一種完整和通用的資料庫存取語言，雖然當時並沒有真正建立語法，但這便是 SQL 的起源。



- 識別名稱不建議使用 SQL 關鍵字或保留字，所以不建議包含任何大小寫的關鍵字，如果一定需要使用關鍵字，請使用反引號（`）括起，這是鍵盤位在 **Tab** 鍵上方的符號，如下：

```
SELECT * FROM `select` WHERE `select`.id > 100;
```

- 識別名稱不允許數字開頭、特殊字元開頭、內嵌空格或使用關鍵字，否則需要使用反引號括起。

6-1-3 SQL 語言的指令種類

SQL 語言的指令依功能分成 DDL、DML 和 DCL 三種。一般來說，資料庫管理師最常使用 DDL 和 DCL 指令，DDL 指令是用來建立資料庫、資料表和相關物件；DCL 是資料庫的權限管理。SQL 程式設計者主要是使用 DML 指令來查詢和更新記錄資料，其說明如下：

- 資料定義語言 DDL（Data Definition Language）：DDL 指令是用來建立、修改、刪除資料庫物件的資料表、檢視表、索引、預存程序、觸發程序和函數等，如下表：

DDL 指令	說明
CREATE/ALTER/DROP DATABASE CREATE/ALTER/DROP SCHEMA	建立、更改和刪除資料庫，MySQL 的 DATABASE 和 SCHEMA 代表相同的資料庫
CREATE/ALTER/DROP TABLE	建立、更改和刪除資料表
CREATE/ALTER/DROP VIEW	建立、更改和刪除檢視表
CREATE/DROP INDEX	建立和刪除索引
CREATE/DROP PROCEDURE	建立和刪除預存程序
CREATE/DROP TRIGGER	建立和刪除觸發程序
CREATE/DROP FUNCTION	建立和刪除函數

- 資料操作語言 DML (Data Manipulation Language)：DML 指令是針對資料表儲存記錄的指令，可以插入、刪除、更新和查詢記錄資料，如下表：

DML 指令	說明
INSERT	在資料表插入一筆新記錄
UPDATE	更新資料表已經存在的記錄
DELETE	刪除資料表已經存在的記錄
SELECT	使用條件查詢資料表符合條件的記錄

- 資料控制語言 DCL (Data Control Language)：資料庫安全管理的權限設定指令，主要有 GRANT、DENY 和 REVOKE 指令。

本書內容主要是說明 SQL 語言的 DDL 和 DML 指令。對於書附的 SQL 指令碼檔案，請參閱第 4-3-3 節的說明啟動 MySQL Workbench，就可以開啟和執行書附 SQL 指令碼檔案。

6-2 | MySQL 字元集與定序

我們在 MySQL 建立資料庫之前，需要先了解什麼是字元集與定序。

6-2-1 字元集

「字元」(Character) 就是我們使用語言文字的最基本單位，例如：英文字母、中文字、阿拉伯數字和標點符號等。「字元集」(Character Set) 就是指某種語言文字的全部字元集合。例如：「ASCII」(American Standard Code for Information Interchange) 是英文字母和阿拉伯數字的字元集，英文字母 A 是 65；B 是 66 等。



因為中文字比英文字母多很多，至少需要使用 2 個位元組數值來代表常用的中文字，繁體中文的字元集是 Big5；簡體中文有 GB 和 HZ。也就是說，1 個中文字佔用 2 個以上位元組，至少是 2 個英文字母。

「統一字碼」（Unicode）是由 Unicode Consortium 組織制定的一個能包括全世界文字的字元集，包含 GB2312 和 Big5 的所有中文字，即 ISO 10646 字元集。Unicode 常用的編碼方式有兩種：UTF-8 為 8 位元編碼；UTF-16 為 16 位元編碼。

簡單的說，在 MySQL 資料庫指定字元集可以決定資料庫儲存記錄資料的語言文字種類，我們可以使用 SHOW CHARACTER SET 指令來顯示 MySQL 支援的 Charset 字元集和其 Default collation 預設定序（SQL 指令碼檔：Ch6_2_1.sql），如下：

```
SHOW CHARACTER SET;
```

	Charset	Description	Default collation	Maxlen	
▶	armscii8	ARMSCII-8 Armenian	armscii8_general_ci	1	
	ascii	US ASCII	ascii_general_ci	1	
	big5	Big5 Traditional Chinese	big5_chinese_ci	2	
	binary	Binary pseudo charset	binary	1	
	cp1250	Windows Central European	cp1250_general_ci	1	
	cp1251	Windows Cyrillic	cp1251_general_ci	1	
	cp1256	Windows Arabic	cp1256_general_ci	1	▼

因為字元集有很多種，以中文內容來說，我們是使用 Unicode 統一字碼，請在 SHOW CHARACTER SET 指令後加上 LIKE 子句，可以篩選出 utf 開頭的字元集（SQL 指令碼檔：Ch6_2_1a.sql），如下：

```
SHOW CHARACTER SET LIKE 'utf%';
```

	Charset	Description	Default collation	Maxlen
▶	utf16	UTF-16 Unicode	utf16_general_ci	4
	utf16le	UTF-16LE Unicode	utf16le_general_ci	4
	utf32	UTF-32 Unicode	utf32_general_ci	4
	utf8mb3	UTF-8 Unicode	utf8mb3_general_ci	3
	utf8mb4	UTF-8 Unicode	utf8mb4_0900_ai_ci	4

上述 utf8 字元集有 2 種 utf8mb3 和 utf8mb4，在國際上使用的 UTF-8 是對應 MySQL 的 utf8mb4，mb4 是指最多使用 4 個位元組來儲存；mb3 是使用 3 個位元組。

Memo

請注意！在 MySQL 8.x 版選 utf8 字元集就是 utf8mb3，在未來版本會遵循國際標準改為 utf8mb4，所以我們在建立 MySQL 資料庫時請使用 utf8mb4 字元集。

6-2-2 定序

「定序」(Collation)是指定字元集的排序規則，因為定序是依據字元集來定義排序規則，所以一種字元集可以擁有多種定序(可指定預設定序)，但一種定序一定只能基於一種字元集。基本上，在排序規則上，我們常需區分英文字母大小寫和發音時腔調的差異等，其簡單說明如下：

- Case sensitivity(CS)：區分英文字母大小寫，英文字母的大寫 A 和小寫 a 是不同的。
- Case Insensitive(CI)：不區分英文字母大小寫，也就是當查詢英文字母的大寫 A，連小寫 a 也一併查詢到。
- Binary(BIN)：使用二進位值進行比較。
- Accent sensitivity(AS)：區分發音時腔調的差異。
- Accent Insensitive(AI)：不區分發音時腔調的差異。

例如：utf8 (目前版本的 MySQL 是對應 utf8mb3) 字元集預設定序是 utf8_general_ci (對應 utf8mb3_general_ci)，最後的 ci 是不區分英文字母大小寫。簡單的說，定序方式可以決定這些記錄資料如何進行比較和排序，我們可以使用 SHOW COLLATION 指令來顯示 MySQL 支援的定序種類，第 2 欄是針對的字元集 (SQL 指令碼檔：Ch6_2_2.sql)，如下：

```
SHOW COLLATION;
```



	Collation	Charset	Id	Default	Compiled	Sortlen	Pad_attribute	^
▶	armscii8_bin	armscii8	64		Yes	1	PAD SPACE	
	armscii8_general_ci	armscii8	32	Yes	Yes	1	PAD SPACE	
	ascii_bin	ascii	65		Yes	1	PAD SPACE	
	ascii_general_ci	ascii	11	Yes	Yes	1	PAD SPACE	
	big5_bin	big5	84		Yes	1	PAD SPACE	
	big5_chinese_ci	big5	1	Yes	Yes	1	PAD SPACE	▼

對於指定字元集的定序，例如：`utf8mb4`，我們可以在 `SHOW COLLATION` 指令後加上 `LIKE` 子句，篩選出 `utf8mb4` 開頭的定序（SQL 指令碼檔：`Ch6_2_2a.sql`），如下：

```
SHOW COLLATION LIKE 'utf8mb4%';
```

	Collation	Charset	Id	Default	Compiled	Sortlen	Pad_attribute	^
	utf8mb4_swedish_ci	utf8mb4	232		Yes	8	PAD SPACE	
	utf8mb4_tr_0900_ai_ci	utf8mb4	265		Yes	0	NO PAD	
	utf8mb4_tr_0900_as_cs	utf8mb4	288		Yes	0	NO PAD	
	utf8mb4_turkish_ci	utf8mb4	233		Yes	8	PAD SPACE	
	utf8mb4_unicode_520_ci	utf8mb4	246		Yes	8	PAD SPACE	
	utf8mb4_unicode_ci	utf8mb4	224		Yes	8	PAD SPACE	
	utf8mb4_vietnamese_ci	utf8mb4	247		Yes	8	PAD SPACE	
	utf8mb4_vi_0900_ai_ci	utf8mb4	277		Yes	0	NO PAD	▼

上述 `utf8mb4` 字元集的定序支援多種語言，以中文來說，主要是使用 `utf8mb4_general_ci`、`utf8mb4_unicode_ci` 和 `utf8mb4_0900_ai_ci`，在 MySQL 8.0 之前版本的預設定序是 `utf8mb4_general_ci`；之後版本是 `utf8mb4_0900_ai_ci`，其說明如下：

- `utf8mb4_general_ci`：不區分英文字母大小寫，沒有實作完整 Unicode 標準，排序速度快，但有可能出錯。
- `utf8mb4_unicode_ci`：不區分英文字母大小寫，完整實作 Unicode 標準，為了和 MariaDB 相容，在建立 MySQL 資料庫時建議使用此定序。
- `utf8mb4_0900_ai_ci`：在 MySQL 8.0 版新增支援 Unicode 9.0.0 標準的定序，不區分英文字母大小寫；也不區分腔調。

Memo

請注意！MariaDB 10.x.x 版資料庫系統並不支援 MySQL 8.x 版的 utf8mb4_0900_ai_ci 定序，如果在 SQL 指令碼檔有使用此定序，我們需修改成 utf8mb4_unicode_ci 或 utf8mb4_general_ci 定序，才能在 MariaDB 伺服器正確的執行。

6-3 | 建立使用者資料庫

在完成第 5 章的資料庫設計後，我們就可以使用 MySQL Workbench 或直接執行 SQL 的 CREATE DATABASE/SCHEMA 指令來建立 MySQL 使用者資料庫。

6-3-1 在 MySQL Workbench 建立資料庫

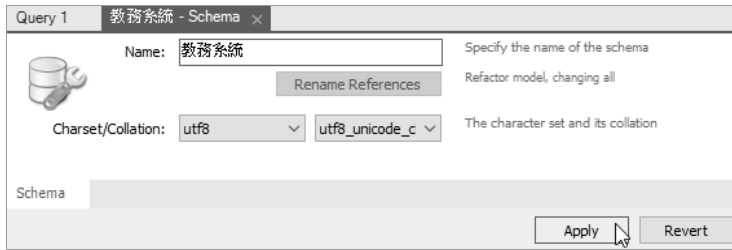
MySQL Workbench 提供相關圖形介面，我們只需在相關欄位輸入資料就可以建立資料庫。例如：建立名為【教務系統】的資料庫，其步驟如下：

- 1 請啟動 MySQL Workbench 連線 MySQL 伺服器後，在「Navigator」視窗選【Schemas】標籤，然後請在空白區域執行【右】鍵快顯功能表的【Create Schema...】命令。

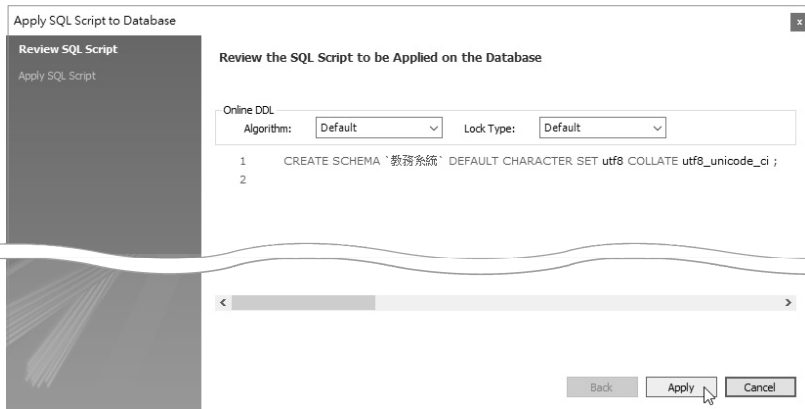




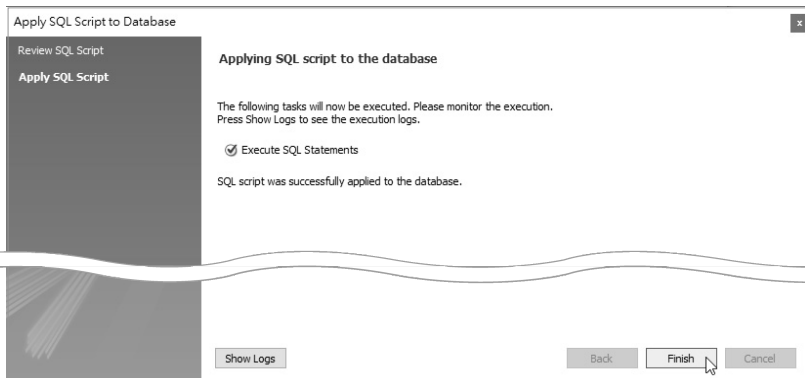
- 2 在【Name】欄輸入資料庫名稱【教務系統】，下方字元集和定序指定可儲存中文的【utf8】字元集和【utf8_unicode_ci】定序，按【Apply】鈕。



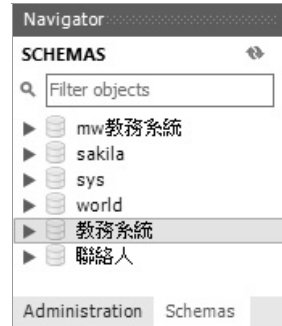
- 3 可以看到建立資料庫的 SQL 指令敘述，請按【Apply】鈕。



- 4 預設勾選【Execute SQL Statements】，按【Finish】鈕建立教務系統資料庫。



- 5 當建立資料庫後，在「Navigator」視窗的【Schemas】標籤，可以看到新建立的【教務系統】資料庫（如果沒有看到，請執行【右】鍵快顯功能表的【Refresh All】命令），如右圖：



- 6 在【教務系統】資料庫上，執行【右】鍵快顯功能表的【Schema Inspector】命令，可以檢視資料庫資訊的字元集是 utf8mb3；定序是 utf8mb3_unicode_ci，如下圖：



6-3-2 使用 SQL 指令建立資料庫

SQL 語言是使用 CREATE DATABASE 或 CREATE SCHEMA 指令來建立資料庫，其基本語法如下：

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] 資料庫名稱
[[DEFAULT] CHARACTER SET 字元集名稱]
[[DEFAULT] COLLATE 定序名稱];
```

在上述語法使用「[]」方括號括起的子句表示可有可無；在「{}」大括號是使用「|」分隔的多種關鍵字，可選其中之一。此語法可以建立名為【資料庫名稱】的資料庫，在之前的 IF NOT EXISTS 可以判斷當資料庫不存在時，才建立資料庫，存在會顯示警告訊息。



在 CHARACTER SET 子句指定資料庫的字元集；COLLATE 子句指定使用的定序，如果沒有指定，就是使用 MySQL 預設的設定，在這 2 個子句之前可加上 DEFAULT，表示設定成資料庫預設的字元集和定序。



SQL 指令碼檔：Ch6_3_2.sql

請使用 MySQL 預設字元集和定序來建立名為【圖書】的資料庫，如下：

```
CREATE DATABASE 圖書;
```

當成功建立資料庫後，可以看到字元集是 utf8mb4；定序是 utf8mb4_0900_ai_ci，如下圖：

Info	Tables	Columns	Indexes	Triggers	Views	Stored Procedures	Functions	Grants	Events
Local instance MySQL80									
圖書									
Schema Details									
Default collation:		utf8mb4_0900_ai_ci							
Default character set:		utf8mb4							
Table count:		0							
Database size (rough estimate):		0.0 bytes							

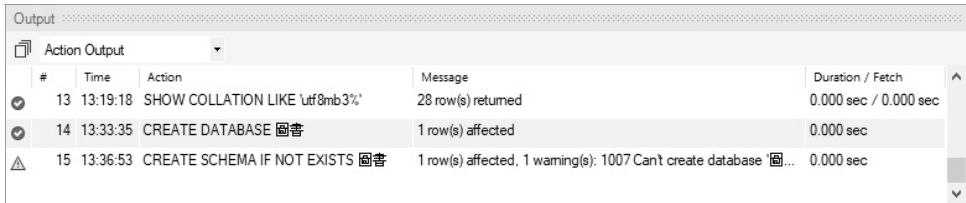


SQL 指令碼檔：Ch6_3_2a.sql

我們準備改用 CREATE SCHEMA 指令來建立名為【圖書】的資料庫，並且加上 IF NOT EXISTS，如下：

```
CREATE SCHEMA IF NOT EXISTS 圖書;
```

上述指令碼的執行結果，可以看到因為【圖書】資料庫已經存在，所以在「Output」視窗的最後顯示一個三角形圖示的警告訊息，無法建立資料庫，如下圖：



#	Time	Action	Message	Duration / Fetch
13	13:19:18	SHOW COLLATION LIKE 'utf8mb3%'	28 row(s) returned	0.000 sec / 0.000 sec
14	13:33:35	CREATE DATABASE 圖書	1 row(s) affected	0.000 sec
15	13:36:53	CREATE SCHEMA IF NOT EXISTS 圖書	1 row(s) affected, 1 warning(s): 1007 Can't create database '圖書...'	0.000 sec



SQL 指令碼檔：Ch6_3_2b.sql

我們準備建立名為【學校】的資料庫，指定字元集是 `utf8mb4`；定序是 `utf8mb4_general_ci`，在【學校】識別名稱有使用反引號括起（位在 `Tab` 鍵上方的按鍵），如下：

```
CREATE SCHEMA IF NOT EXISTS `學校`  
CHARACTER SET utf8mb4  
COLLATE utf8mb4_general_ci;
```



SQL 指令碼檔：Ch6_3_2c.sql

我們準備建立名為【銷售管理】的資料庫，只有指定字元集是 `big5`，此時的定序就是預設定序 `big5_chinese_ci`，如下：

```
CREATE DATABASE `銷售管理`  
CHARACTER SET big5;
```

6-4 | 修改使用者資料庫

在 MySQL 成功建立使用者資料庫後，如果資料庫結構有變更，我們可以使用 MySQL Workbench 或 SQL 指令來修改使用者資料庫，而不用重新建立資料庫。



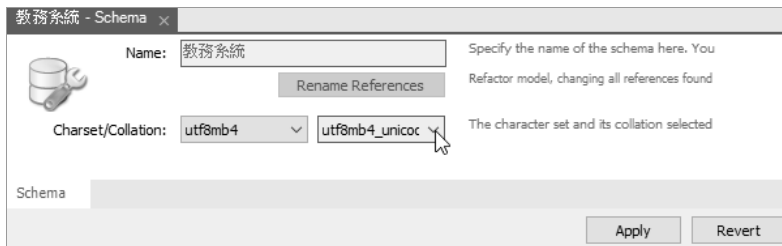
6-4-1 使用 MySQL Workbench 修改使用者資料庫

MySQL Workbench 提供相關圖形介面來修改使用者資料庫。例如：修改【教務系統】資料庫的字元集和定序，分別改成 utf8mb4 和 utf8mb4_unicode_ci，其步驟如下：

- 1 請啟動 MySQL Workbench 連線 MySQL 伺服器後，在「Navigator」視窗【Schemas】標籤的【教務系統】資料庫上，執行【右】鍵快顯功能表的【Alter Schema...】命令。



- 2 在【Charset/Collation】欄分別選【utf8mb4】字元集和【utf8mb4_unicode_ci】定序（如果介面無法完整顯示名稱，在 2 個 unicode 是選第 2 個），按【Apply】鈕。



- 3 可以看到 SQL 指令碼，請按【Apply】鈕，再按【Finish】鈕完成資料庫更改，如下：

```
ALTER SCHEMA `教務系統`  
DEFAULT CHARACTER SET utf8mb4  
DEFAULT COLLATE utf8mb4_unicode_ci;
```

上述 DEFAULT 指定資料庫預設的字元集和定序。請執行【右】鍵快顯功能表的【Schema Inspector】命令來檢視字元集與定序是否已經成功修改。

6-4-2 使用 SQL 指令修改使用者資料庫

SQL 語言可以使用 ALTER DATABASE 或 ALTER SCHEMA 指令來修改使用者資料庫，其基本語法如下：

```
ALTER {DATABASE | SCHEMA} 資料庫名稱  
[CHARACTER SET 字元集名稱]  
[COLLATE 定序名稱];
```

上述語法是修改名為【資料庫名稱】的資料庫，可以單獨修改字元集或定序，也可以同時修改字元集和定序。



SQL 指令碼檔：Ch6_4_2.sql

請更改【圖書】資料庫的定序，改成 utf8mb4_unicode_ci，如下：

```
ALTER DATABASE 圖書  
COLLATE utf8mb4_unicode_ci;
```



SQL 指令碼檔：Ch6_4_2a.sql

請更改【學校】資料庫的字元集和定序，分別改成 big5 和 big5_chinese_ci，如下：

```
ALTER SCHEMA 學校  
CHARACTER SET big5  
COLLATE big5_chinese_ci;
```



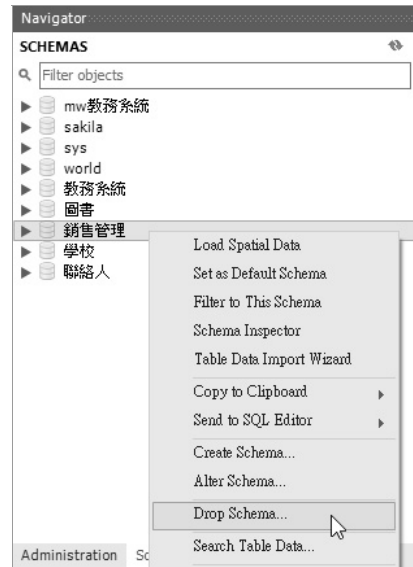
6-5 | 刪除使用者資料庫

對於不再需要的使用者資料庫，我們可以使用 MySQL Workbench 或 SQL 的 DROP DATABASE/SCHEMA 指令來刪除使用者資料庫。

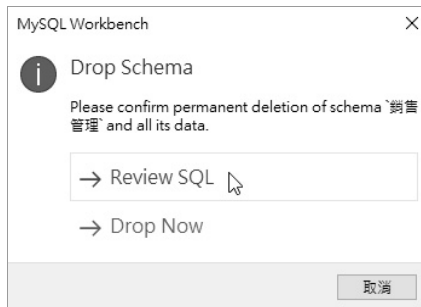
使用 MySQL Workbench 刪除使用者資料庫

在 MySQL Workbench 刪除 Ch6_3_2c.sql 建立的【銷售管理】資料庫，其步驟如下：

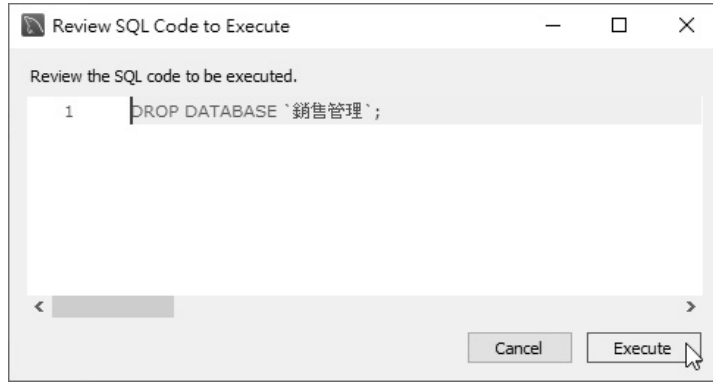
- 1 請啟動 MySQL Workbench 連線 MySQL 伺服器後，在「Navigator」視窗【Schemas】標籤的【銷售管理】資料庫上，執行【右】鍵快顯功能表的【Drop Schema...】命令。



- 2 在確認對話方塊選【Drop Now】是馬上刪除，請選【Review SQL】可以在檢視 SQL 指令碼後再執行刪除。



- 3 可以看到 SQL 指令 DROP DATABASE，請按【Execute】鈕執行 SQL 指令來刪除資料庫，【Cancel】鈕是取消刪除。



使用 SQL 指令刪除使用者資料庫

在 SQL 語言刪除資料庫是使用 DROP DATABASE 或 DROP SCHEMA 指令，其基本語法如下：

```
DROP {DATABASE | SCHEMA} [IF EXISTS] 資料庫名稱;
```

上述語法的【資料庫名稱】就是欲刪除的資料庫名稱，在之前的 IF EXISTS 是當資料庫存在時才刪除，否則會顯示警告訊息。



SQL 指令碼檔：Ch6_5.sql

使用 DROP DATABASE 指令刪除第 4-3-3 節建立的【聯絡人】資料庫，如下：

```
DROP DATABASE 聯絡人;
```



SQL 指令碼檔：Ch6_5a.sql

改用 DROP SCHEMA 指令刪除第 4-3-3 節建立的【聯絡人】資料庫，並且加上 IF EXISTS，如下：

```
DROP SCHEMA IF EXISTS 聯絡人;
```



因為上述資料庫已經在之前刪除了，所以在「Output」視窗顯示警告訊息，如下圖：

#	Time	Action	Message	Duration / Fetch
26	14:52:38	DROP DATABASE '銷售管理'	0 row(s) affected	0.000 sec
27	15:02:15	DROP DATABASE 聯絡人	1 row(s) affected	0.015 sec
28	15:02:43	DROP SCHEMA IF EXISTS 聯絡人	0 row(s) affected, 1 warning(s): 1008 Can't drop database 聯絡...	0.000 sec

6-6 | MySQL 資料庫的備份與還原

MySQL 可以使用 MySQL Workbench 圖形介面來幫助我們備份資料庫，以便當資料庫系統故障或錯誤時，能夠還原資料庫。MariaDB 並不支援 MySQL Workbench，請改用附錄 B-4 節的 phpMyAdmin。

Memo

請注意！MySQL Workbench 備份與還原資料庫是執行 `mysqldump` 命令列工具，此工具因為 MySQL 資料庫檔案名稱是 `big5` 編碼，和資料表記錄資料使用 `utf-8` 編碼的衝突問題，並無法備份與還原中文命名的資料庫，所以本書從第 7 章開始的【教務系統】資料庫會改成英文名稱【school】。

6-6-1 備份資料庫

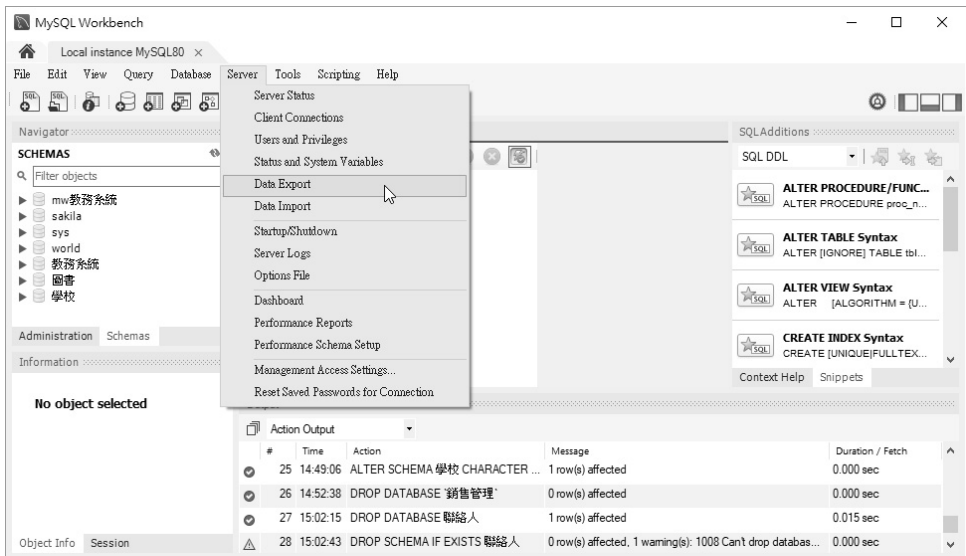
當資料庫系統故障或錯誤而造成資料遺失時，我們就需要借助資料庫備份與還原，才能夠快速恢復資料庫系統的正常運作。對於資料庫管理師來說，一項很重要的工作，就是記得定時備份資料庫。

備份 (Backup) 是使用備份工具 (可能是資料庫管理系統的內建功能或其他廠商開發的工具程式)，將資料庫儲存的資料儲存在備份裝置的儲存媒體。基本上，備份資料庫的時機共有三種，如下：

- 即時備份：當有新記錄資料產生時，就立即執行備份。
- 定期備份：在固定時間周期的執行備份，例如：每日固定時間執行一次，或每間隔 12 小時備份一次等。
- 手動備份：當有需要或發生特殊情況時執行備份，我們可以自行決定在何時執行所需的備份作業。

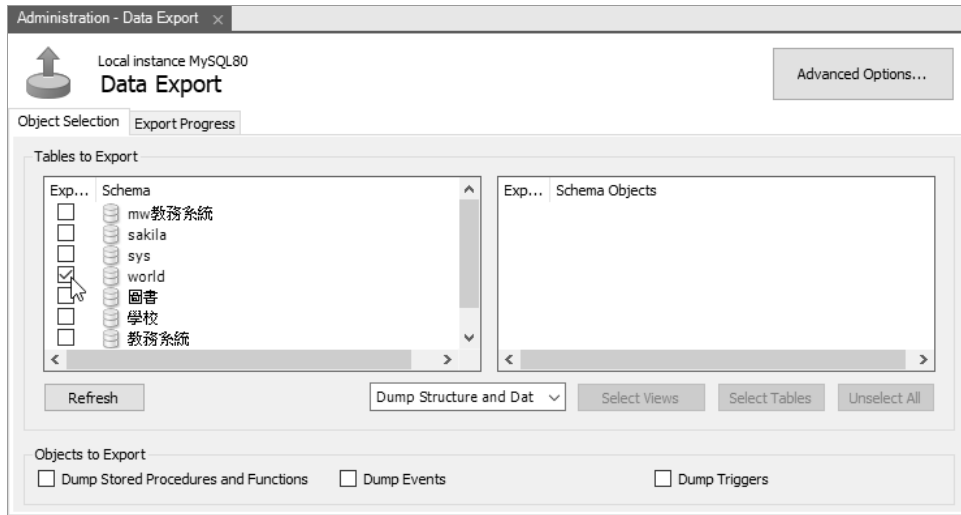
在本節準備使用 MySQL 安裝的範例資料庫 world 為例，說明如何使用 MySQL Workbench 來手動備份 world 資料庫，其步驟如下：

- 1 請啟動 MySQL Workbench 連線 MySQL 伺服器後，執行「Server>Data Export」命令。

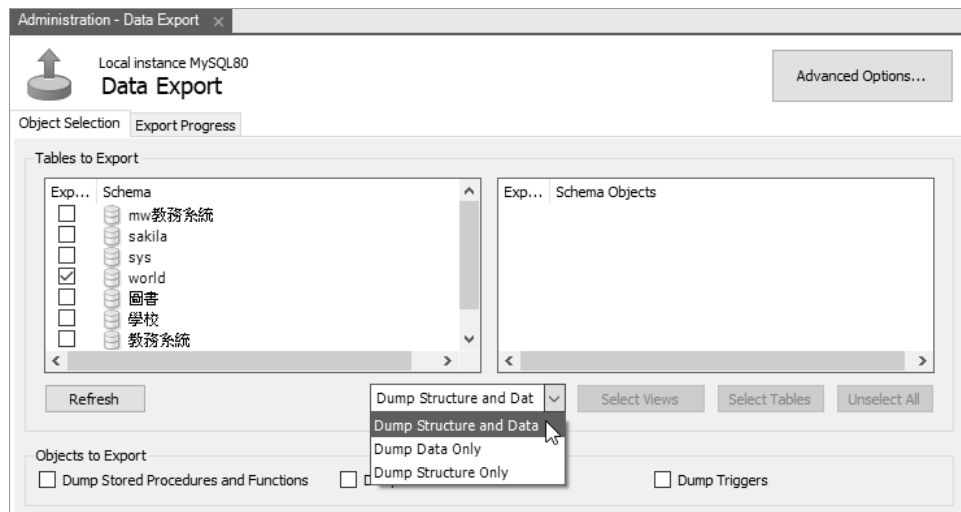




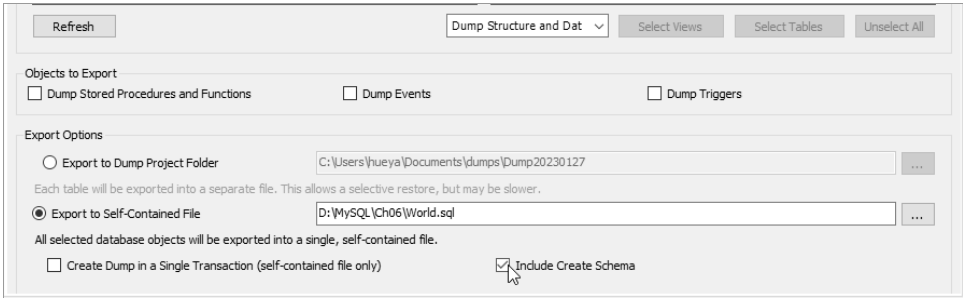
- 2 在【Administration – Data Export】標籤頁的「Table to Export」框勾選欲備份的資料庫，可以勾選多個資料庫，以此例勾選【world】。



- 3 在下拉式清單選擇備份類型是 Dump Structure and Data 備份資料庫結構和記錄資料；Dump Data Only 只備份記錄資料；Dump Structure Only 只備份資料庫結構。



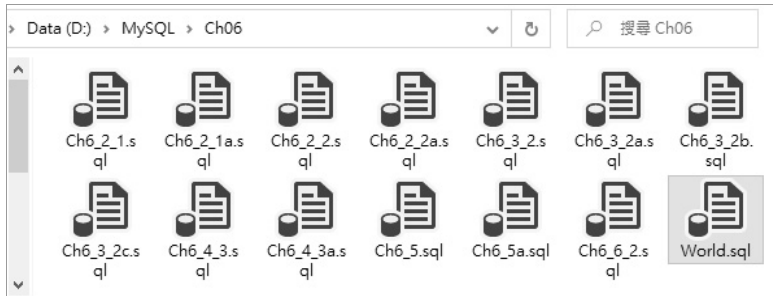
- 4 然後在下方選【Export to Self-Contained File】備份成單一 SQL 指令碼檔案後，按之後按鈕選擇備份路徑和檔名，以此例是「D:\MySQL\Ch06\World.sql」，勾選【Include Create Schema】包含建立資料庫的 SQL 指令。



- 5 選【Export Progress】標籤，按右下方【Start Export】鈕開始備份資料庫（按鈕會改成【Export Again】鈕），在中間顯示備份訊息，如下圖：



上述訊息"Export of D:\MySQL\Ch06\World.sql has finished"指出已經完成資料庫備份，在「D:\MySQL\Ch06」資料夾可以看到匯出建立的 SQL 指令碼檔案：World.sql，如下圖：

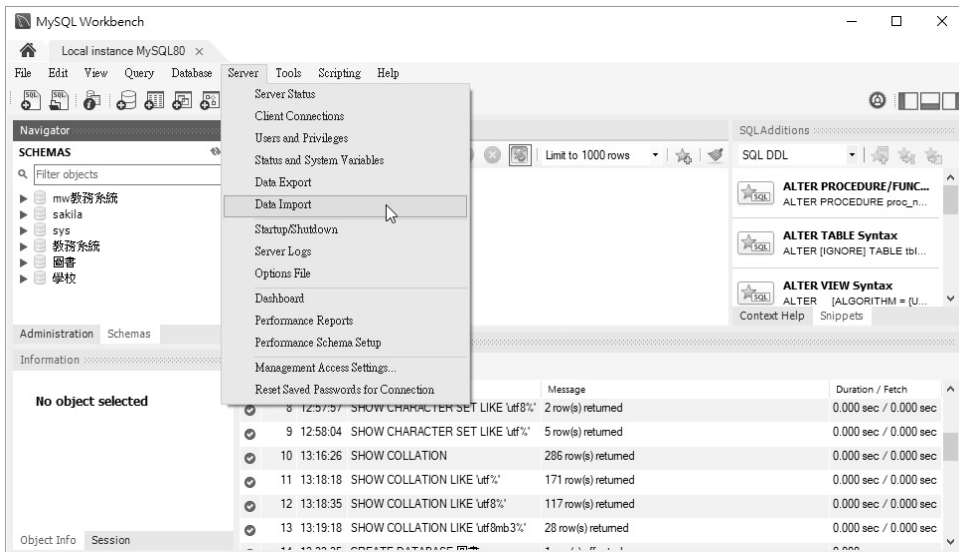


6-6-2 還原資料庫

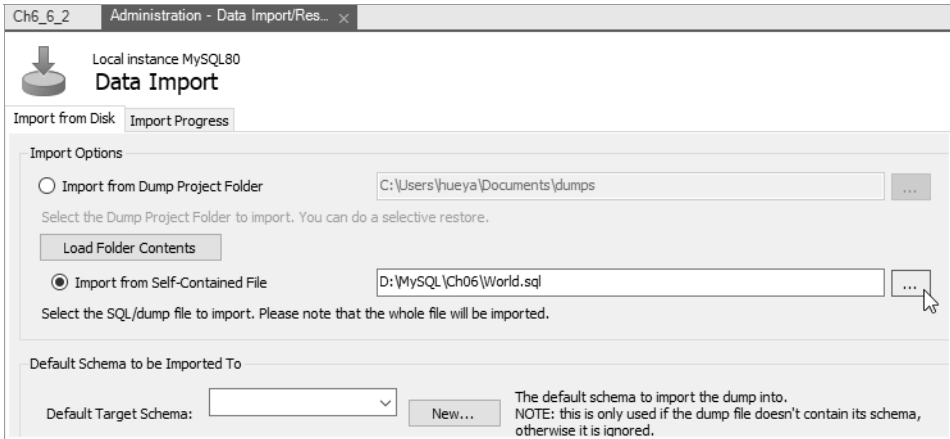
MySQL 的還原 (Restore) 作業就是一種回復處理，可以將備份作業建立的備份資料回存至資料庫系統，我們可以在資料庫系統發生錯誤或故障後，使用還原作業來恢復資料庫系統的正常運作。

例如：因為 world 資料庫發生故障，我們可以使用第 6-6-1 節備份資料庫的 World.sql 來還原 world 資料庫(請先執行 Ch6_6_2.sql 刪除 world 資料庫)，其步驟如下：

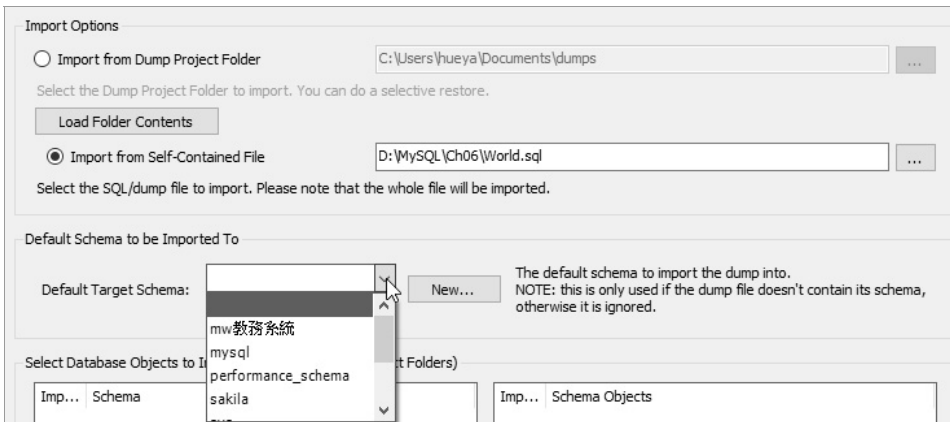
- 1 請啟動 MySQL Workbench 連線 MySQL 伺服器後，執行「Server>Data Import」命令。



- 2 選【Import from Self-Contained File】使用單一 SQL 指令碼檔案來還原資料庫後，按之後按鈕選擇路徑和檔名，以此例是「D:\MySQL\Ch06\World.sql」。

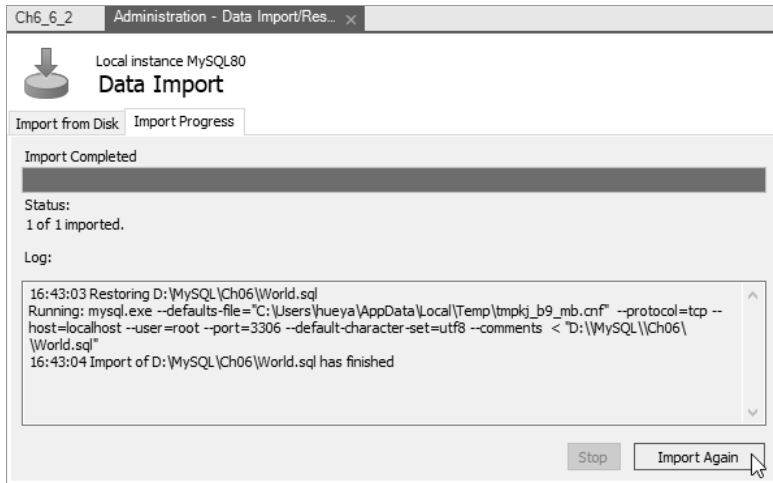


- 3 如果 SQL 指令碼檔內容不包含建立資料庫的指令，我們可以在下方「Default Schema to be Imported To」框選擇匯入哪一個資料庫，按【New...】鈕可以新增一個資料庫。





- 4 選【Import Progress】標籤，按右下方【Start Import】鈕開始還原資料庫（按鈕會改成【Import Again】鈕），在中間顯示還原訊息，如下圖：



上述訊息 "Import of D:\MySQL\Ch06\World.sql has finished" 指出已經完成資料庫還原，在「Navigator」視窗的【Schemas】標籤，可以看到還原的【world】資料庫（如果沒有看到，請執行【右】鍵快顯功能表的【Refresh All】命令），如右圖：



17-4 | 使用 ChatGPT 寫出 Python 和 PHP 資料庫程式

ChatGPT 不只可以輕鬆的寫出 SQL 指令敘述，還可以幫助我們寫出存取 MySQL 資料庫的 Python 與 PHP 程式，或將現成的 Python 程式改寫成 PHP 版本的資料庫程式。

寫出存取 MySQL 資料庫的 Python 程式

我們準備讓 ChatGPT 寫出存取 MySQL 資料庫的 Python 程式。ChatGPT 的詳細問題和功能描述（Ch17_4.txt），如下：

請使用 Python 語言和 PEP8 風格寫一個 Python 程式，並且加上繁體中文的註解文字，這是使用 pymysql 模組在 MySQL 資料庫下達 SQL 指令，可以顯示查詢結果的記錄資料。SQL 指令如下：

```
SELECT 課程編號, 名稱, 學分 FROM 課程 WHERE 學分 >= 3;
```

連接 MySQL 伺服器的連線資訊，如下：

主機是 localhost。

使用者名稱是 root，密碼 Aa12345678。

資料庫名稱是 school。

字元集是 utf8。



```
python Copy code

import pymysql

# 連接MySQL伺服器的連線資訊
host = "localhost"
username = "root"
password = "Aa12345678"
database = "school"
charset = "utf8"

# 建立資料庫連線
db = pymysql.connect(host=host, user=username, password=password, db=database)

try:
    # 使用 cursor() 方法建立一個遊標物件 cursor
    cursor = db.cursor()

    # SQL 指令
    sql = "SELECT 課程編號, 名稱, 學分 FROM 課程 WHERE 學分 >= 3;"

    # 執行 SQL 指令
    cursor.execute(sql)

    # 取回所有查詢結果
    results = cursor.fetchall()

    # 顯示查詢結果
    for row in results:
        course_id = row[0]
        name = row[1]
        credit = row[2]
        print(f"課程編號: {course_id} 名稱: {name} 學分: {credit}")

except pymysql.Error as e:
    # 顯示錯誤訊息
    print(f"Error {e.args[0]}: {e.args[1]}")

finally:
    # 關閉資料庫連線
    db.close()
```

請點選程式框右上方【Copy code】，可以複製程式碼至剪貼簿，然後貼至 Python IDLE 後，儲存成為 ch17_4.py 檔案，其執行結果如下：

```
===== RESTART: D:\MySQL\Ch17\ch17_4.py =====
課程編號: CS101 名稱: 計算機概論 學分: 4
課程編號: CS111 名稱: 線性代數 學分: 4
課程編號: CS121 名稱: 離散數學 學分: 4
課程編號: CS203 名稱: 程式語言 學分: 3
課程編號: CS205 名稱: 網頁程式設計 學分: 3
課程編號: CS222 名稱: 資料庫管理系統 學分: 3
課程編號: CS349 名稱: 物件導向分析 學分: 3
```

寫出存取 MySQL 資料庫的 PHP 程式

請繼續目前的 ChatGPT 交談過程，我們準備改用 PHP 語言來寫出相同功能的程式。ChatGPT 的詳細問題和功能描述（Ch17_4a.txt），如下：

請改用 PHP 語言來寫出存取 MySQL 資料庫的 PHP 程式，PHP 是使用 ext/mysqli 擴充程式的傳統方式來存取 MySQL 資料庫，可以顯示[課程編號]、[名稱]和[學分]三個欄位的查詢結果。

```
php Copy code

<?php
// 連接MySQL伺服器的連線資訊
$host = "localhost";
$username = "root";
$password = "Aa12345678";
$database = "school";
$charset = "utf8";

// 建立資料庫連線
$conn = mysqli_connect($host, $username, $password, $database);

// 確認資料庫連線是否成功
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
```



```
// 執行 SQL 查詢
$sql = "SELECT 課程編號, 名稱, 學分 FROM 課程 WHERE 學分 >= 3;";
$result = mysqli_query($conn, $sql);

// 顯示查詢結果
if (mysqli_num_rows($result) > 0) {
    echo "<table>";
    echo "<tr>";
    echo "<th>課程編號</th>";
    echo "<th>名稱</th>";
    echo "<th>學分</th>";
    echo "</tr>";

    while ($row = mysqli_fetch_assoc($result)) {
        echo "<tr>";
        echo "<td>" . $row["課程編號"] . "</td>";
        echo "<td>" . $row["名稱"] . "</td>";
        echo "<td>" . $row["學分"] . "</td>";
        echo "</tr>";
    }

    echo "</table>";
} else {
    echo "0 results";
}

// 釋放記憶體並關閉資料庫連線
mysqli_free_result($result);
mysqli_close($conn);
?>
```

請點選程式框右上方【Copy code】，可以複製程式碼至剪貼簿，然後儲存成為 `ch17_4a.php` 檔案。請啟動 XAMPP 後，在瀏覽器輸入下列 URL 網址來執行 PHP 程式，如下：

- http://localhost:8080/ch17_4a.php



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/ch17_4a.php'. The page content is a table with three columns: '課程編號' (Course ID), '名稱' (Name), and '學分' (Credits). The table lists the following courses:

課程編號	名稱	學分
CS101	計算機概論	4
CS111	線性代數	4
CS121	離散數學	4
CS203	程式語言	3
CS205	網頁程式設計	3
CS222	資料庫管理系統	3
CS349	物件導向分析	3



將 Python 程式改寫成 PHP 語言的版本：ch14-5a_gpt.py

16 如果有現成的 Python 程式碼時，我們也可以直接讓 ChatGPT 將 Python 程式改寫成 PHP 語言的版本。ChatGPT 的詳細問題和功能描述（Ch17_4b.txt），如下：

請將下列 Python 程式碼改寫成 PHP 語言的版本，PHP 是使用 ext/mysql 擴充程式的傳統方式來存取 MySQL 資料庫，並且使用 HTML 表格標籤來顯示查詢結果的[學號]和[姓名]欄位。Python 程式碼如下：

```
import pymysql

db = pymysql.connect(host="localhost",
                    user="root",
                    password="Aa12345678",
                    database="school",
                    charset="utf8")

cursor = db.cursor()
sql = "SELECT * FROM 學生 WHERE 生日 <='2003/6/1'"
cursor.execute(sql)
data = cursor.fetchall()
for row in data:
    print(row[0], row[1])
db.close()
```



```
php Copy code

<?php
$servername = "localhost";
$username = "root";
$password = "Aa12345678";
$dbname = "school";

$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT * FROM 學生 WHERE 生日 <='2003/6/1'";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    echo "<table><tr><th>學號</th><th>姓名</th></tr>";
    while($row = mysqli_fetch_assoc($result)) {
        echo "<tr><td>" . $row["學號"]. "</td><td>" . $row["姓名"]. "</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}

mysqli_close($conn);
?>
```

請點選程式框右上方【Copy code】，可以複製程式碼至剪貼簿，然後儲存成為 `ch17_4b.php` 檔案。請啟動 XAMPP 後，在瀏覽器輸入下列 URL 網址來執行 PHP 程式，如下：

- http://localhost:8080/ch17_4b.php

