

# 快速活用 MySQL

## 精準設計關聯式資料庫

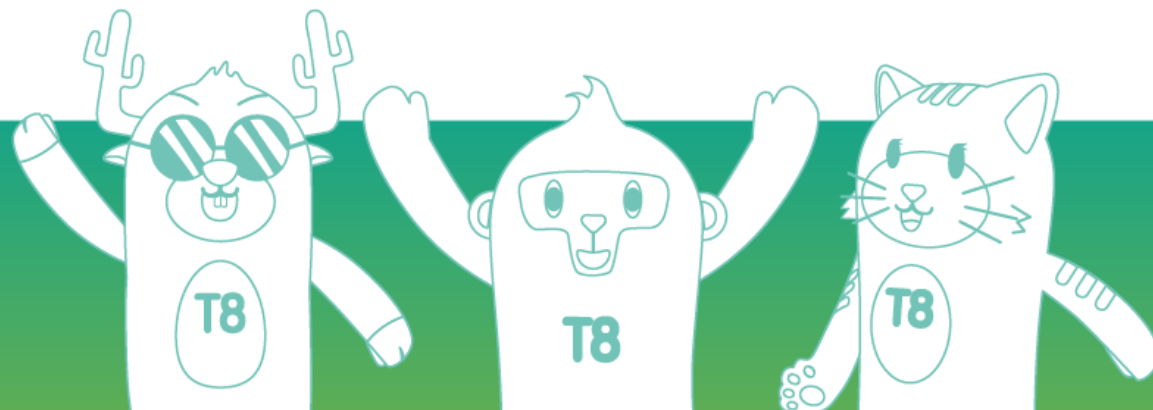
■ 授課講師 黃彬華

■ 教材編寫 黃彬華

緯育 *TibaMe*

即學・即戰・即就業

<https://www.tibame.com/>



學習目標：

1. 認識 MySQL

# Module 1.

## MySQL 資料庫 介紹與安裝

認識 MySQL 資料庫  
歷史與特色

## 歷史



- 1995年 – MySQL 初版發布，開發者為瑞典的 MySQL AB 公司
- 2008年 - Sun Microsystems 以 10 億美元收購 MySQL AB
- 2009年 - Oracle以 74 億美元收購 Sun Microsystems
- MySQL 最終歸 Oracle 所有

## 特色



- 關聯式資料庫 ( relational database )
- 使用 SQL 語法
- 表格內的值不區別大小寫
- 一個表格的最大欄位數限制為 4,096 個欄位
- 一筆資料大小限制為 65,535 bytes
  - BLOB, TEXT 儲存大量資料的欄位只會佔 9~12 bytes ,  
因為資料內容會儲存在其他資料列。
  - 參看 [Limits on Table Column Count and Row Size](#)

# Module 1.

## MySQL 資料庫 介紹與安裝

學習目標：

1. 了解 MySQL 安裝步驟  
與軟體設定的注意事項

了解 Windows 版  
MySQL 安裝與設定

## 下載

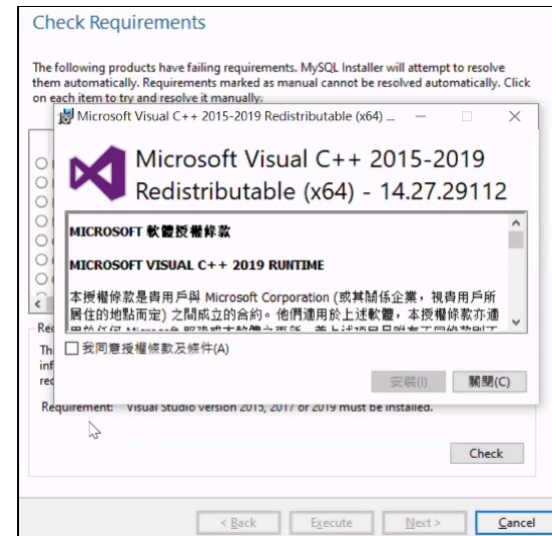
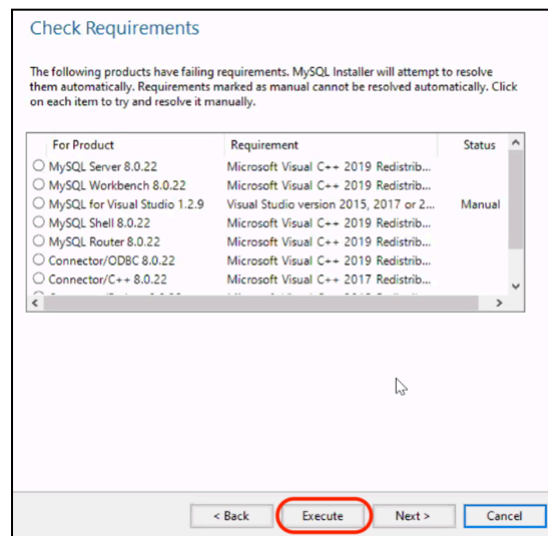


- 進入 [MySQL Community Server](#) 下載頁面（社群版開源且免費）
- Windows可選擇下載 [MySQL Installer](#)（完整安裝版，包含 Workbench 管理工具）
- 下載時會提示要登入 Oracle 帳號。向下捲動並點擊「No thanks, just start my download.」可跳過登入而直接下載

## 安裝

●●●●●

- Choosing a Setup Type : 選擇 Developer Default
- Check Requirements : 需安裝 Visual C++ Redistributable Package
  - 點擊 Execute 按鈕 > 勾選同意條款 > 點擊「安裝」按鈕
  - 沒顯示 Execute 按鈕代表已經安裝，點擊 Next 按鈕繼續
- Installation : 點擊 Execute 按鈕開始安裝



## 設定



- 設定過程大致上都可採用預設值
- 輸入最高管理者帳號 root 的密碼後請謹記該密碼
- 建議養成安全習慣：增加一個 DBA 帳號方便之後管理，以避免經常使用最高管理者帳號 - root，而遭他人窺探密碼、盜用



# Module 1.

## MySQL 資料庫 介紹與安裝

學習目標：

1. 了解 MySQL 安裝步驟  
與軟體設定的注意事項

了解 MacOS 版  
MySQL 安裝與設定

## Server 下載



- MacOS 不像 Windows 有完整安裝版，所以需要分別下載與安裝 server 與 Workbench 管理介面
- 進入 [MySQL Community Server](#) 下載頁面 (社群版開源且免費)
- 下載 macOS DMG Archive 檔案
- 下載時會提示要登入 Oracle 帳號。向下捲動並點擊「No thanks, just start my download.」可跳過登入而直接下載

## Server 安裝與設定



- 安裝與設定過程大致上都可採用預設值
- 輸入最高管理者帳號 root 的密碼後請謹記該密碼

# Module 2.

## 命令列啟動 MySQL

學習目標：

1. 認識命令提示字元
2. 學會啟動 MySQL server
3. 學會停止 MySQL server

了解 Windows  
啟動與停止 MySQL

- 開啟 Windows 命令提示字元
- 啟動 MySQL server

```
C:\> "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld"
```

因為 `mysqld` 的檔案路徑有空白字元，所以指令要加上雙引號

- 停止 MySQL server

```
C:\> "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqladmin" -u root -p shutdown
```

- 參看 [MySQL文件](#)

# Module 2.

## 命令列啟動 MySQL

學習目標：

1. 認識命令提示字元
2. 學會啟動與重新啟動 MySQL server
3. 學會停止 MySQL server

了解 MacOS  
啟動與停止 MySQL

- 開啟 MySQL 命令提示字元
- 啟動與重新啟動 MySQL server

```
shell> sudo /usr/local/mysql/support-files/mysql.server start
```

```
shell> sudo /usr/local/mysql/support-files/mysql.server restart
```

- 停止 MySQL server

```
shell> sudo /usr/local/mysql/support-files/mysql.server stop
```

- 以上指令都會要求輸入作業系統管理者密碼，而非資料庫管理者密碼

學習目標：

1. 了解資料庫資訊
2. 學會運用  
其他常見指令

## Module 2.

# 命令列啟動 MySQL

了解其他常用指令



## 進入與離開 MySQL 提示字元



- 進入 ( Windows 與 MacOS )

Windows shell> "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql" -u root -p

MacOS shell> /usr/local/mysql/bin/mysql -u root -p

- 離開

shell> quit

## 顯示資料庫資訊



- 顯示所有資料庫

```
mysql> show databases;
```

- 切換到指定資料庫

```
mysql> use sys;
```

- 將現行資料庫名稱顯示

```
mysql> select database();
```

- 將現行資料庫的所有表格名稱顯示

```
mysql> show tables;
```

- 將指定表格的結構顯示

```
mysql> describe user_summary;
```

# Module 3. Workbench 管理工具安裝

學習目標：

1. 了解 Workbench 安裝注意事項

了解 Windows 版  
Workbench 下載與安裝

- 如果安裝前述 MySQL Installer（完整安裝版），就已經包含 Workbench 管理工具，而無需再下載安裝。否則就參看下列步驟：
- 進入 [MySQL Workbench](#) 下載頁面
- 點擊下載 Windows MSI Installer 檔案
- 下載時會提示要登入 Oracle 帳號。向下捲動並點擊「No thanks, just start my download.」可跳過登入而直接下載
- 下載完畢後雙點該檔案，按照指示安裝即可
- 安裝完畢後無須設定，之後會說明如何建立連線

# Module 3. Workbench 管理工具安裝

學習目標：

1. 了解 Workbench 安裝注意事項

了解 MacOS 版  
Workbench 下載與安裝

- 進入 [MySQL Workbench](#) 下載頁面
- 點擊下載 macOS DMG Archive 檔案
- 下載時會提示要登入Oracle帳號。向下捲動並點擊「No thanks, just start my download.」可跳過登入而直接下載
- 下載完畢後雙點該檔案，按照指示安裝即可
- 安裝完畢後無須設定，之後會說明如何建立連線

# Module 4. Workbench 連線與偏好設定

學習目標：

1. 了解 Workbench  
建立連線步驟

學會使用  
Workbench 建立連線

- 預設會建立一個連線到本機 MySQL 的連線
- 也可建立新連線：點擊「+」按鈕
  - ✓ Connection Name：連線名稱
  - ✓ Connection Method：Standard (TCP/IP)
  - ✓ Hostname：主機名稱或 IP 位址
  - ✓ 本機連線：localhost ( 或 127.0.0.1 )
  - ✓ 外部連線：DB 伺服器所在 IP 位址
  - ✓ Port：3306
  - ✓ Username：登入帳號
  - ✓ Password：點擊「Store in ...」按鈕後輸入密碼可以儲存起來以後無需再輸入
  - ✓ Default Schema：要連線的資料庫，可不輸入，待連線成功後再選定
  - ✓ 點擊 Test Connection 按鈕測試連線是否成功



# Module 4. Workbench 連線與偏好設定

學習目標：

1. 了解如何開放外部連線至 MySQL

認識 Windows  
防火牆設定

- Windows 版 MySQL 安裝設定時已經開啟 3306 埠號並允許本機連線；欲開放外部連線至 MySQL 需做以下設定：

Windows 10 控制台 >

搜尋防火牆與網路保護 >

進階設定 >

「輸入規則」右鍵 >

新增規則 >

連接埠 >

特定本機連接埠為 3306 >

允許連線

# Module 4. Workbench 連線與偏好設定

學習目標：

1. 認識 SQL Editor
2. 了解如何更改  
字型與大小

了解 Workbench  
偏好設定

## 開啟偏好設定

- Windows 版：主選單 > Edit > Preferences
- MacOS 版：主選單 > MySQLWorkbench > Preferences

- **SQL Editor**
  - SQL Editor > Other > Safe Updates
    - 勾選則 UPDATE 與 DELETE 語法必須加上條件限制方能執行
  - SQL Execution > General > New connections use auto commit mode
    - 執行異動語法會即刻確定生效，而無需加上 commit 指令
- **Fonts：更改字型與大小（需要重啟 Workbench）**
  - SQL Editor：修改 SQL 指令的字型大小（直接修改字型名稱或數字即可）
    - 也可直接在 SQL Editor 編輯區直接使用縮放手勢，就不需要設定
  - Resultset Grid：修改執行結果的字型大小

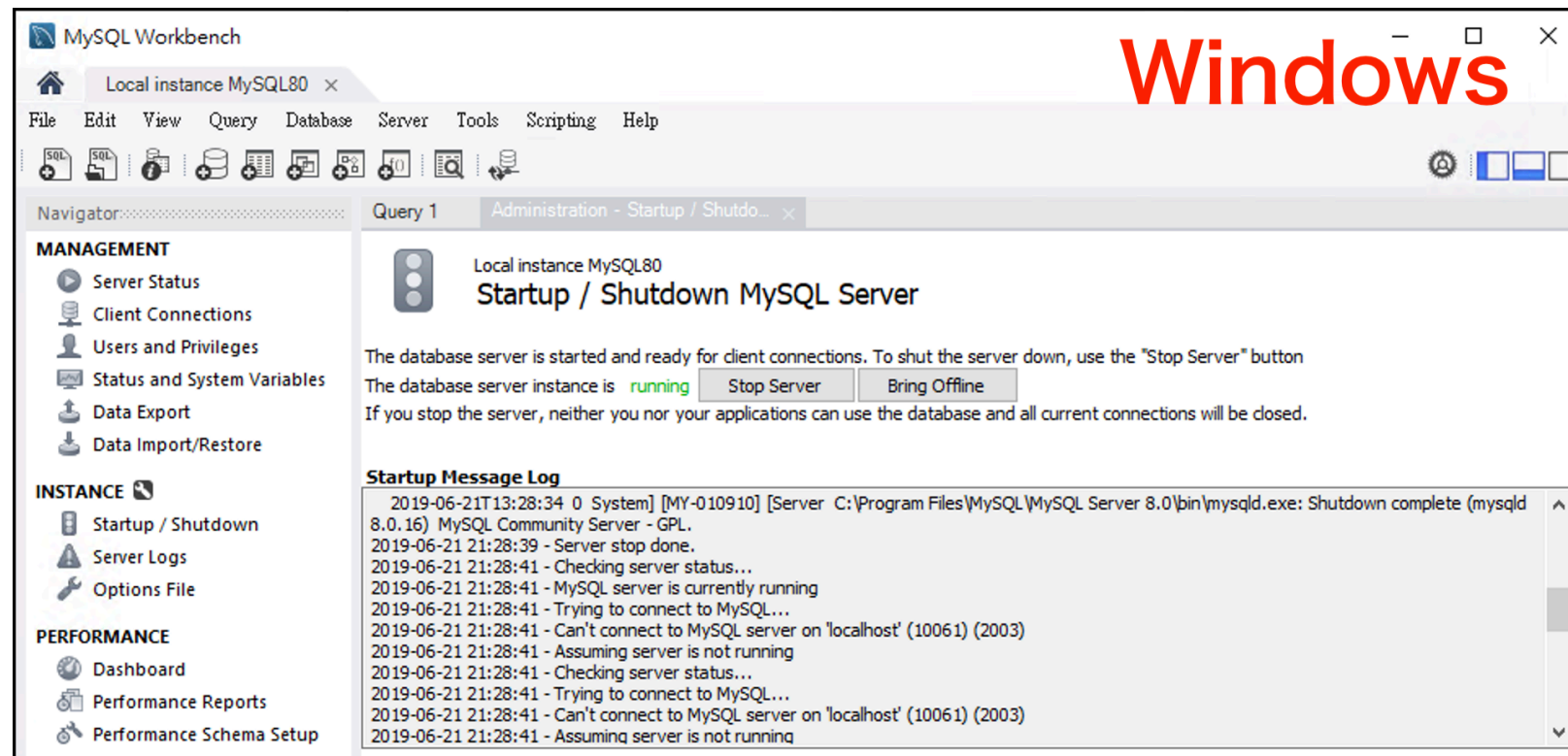
學習目標：

1. 學會操作 Workbench  
啟動與停止 MySQL

# Module 5. 圖形介面管理 MySQL

了解 Windows  
啟動與停止 MySQL

- MySQL Workbench 連線成功 > Administration 頁籤 > Instance > Startup / Shutdown



# Module 5. 圖形介面管理 MySQL

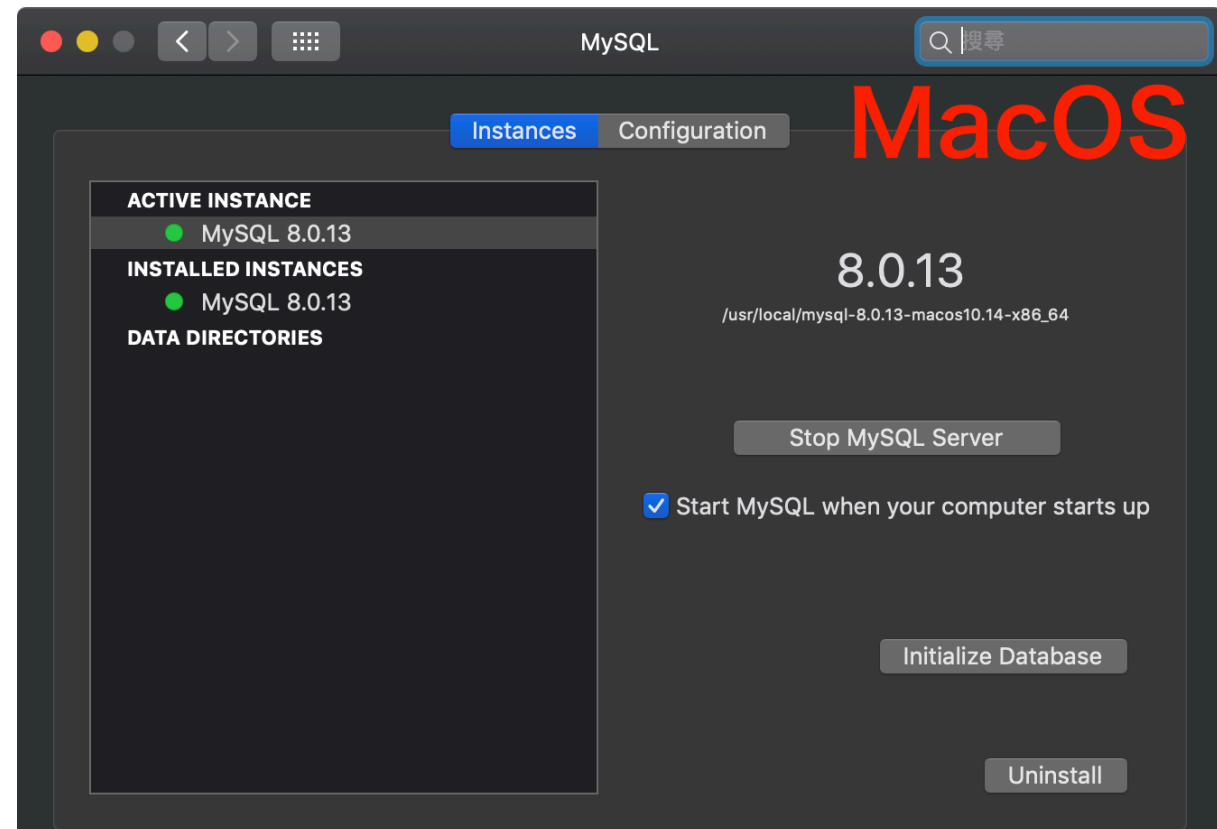
學習目標：

1. 學會操作系統偏好設定  
啟動與停止 MySQL

了解 MacOS  
啟動與停止 MySQL



- MacOS 系統偏好設定 > MySQL > Start / Stop MySQL Server
- Initialize Database：回復原廠設定，所有資料會被刪除
- Uninstall：移除 MySQL



# Module 5. 圖形介面管理 MySQL

## 學習目標：

1. 了解 MySQL 相關系統狀態
2. 了解連線使用者資訊
3. 了解如何強制斷線
4. 了解如何管理使用者

學會查詢  
MySQL 相關資訊

- 查詢 MySQL 系統狀態
  - 主選單 > Server > Server Status
- 查看連線資訊
  - 主選單 > Server > Client Connections > 點擊 Refresh 按鈕
- 對連線加以斷線
  - 選擇指定連線後點擊 Kill Connection
- 查看 user 帳號資訊
  - 主選單 > Server > Users and Privileges

學習目標：

1. 認識 Table
2. 認識 Database

# Module 6.

## 資料庫介紹

認識表格與資料庫

- **表格 (Table)**
  - 以欄 (columns) 與列 (rows) 方式來儲存資料
- **資料庫 (Database, DB)**
  - 儲存著大量表格以及相關資料庫物件 (tables, views...) 的綜合儲存體。每個資料庫都會轉化成特定檔案並儲存在電腦內

學習目標：

1. 認識 DBMS

# Module 6.

## 資料庫介紹

認識資料庫管理系統

- 資料庫管理系統 (Database Management System , DBMS)
  - 管理資料庫的大型軟體
  - 允許資料庫管理師 (DBA) 透過命令列或是視窗化介面來完成資料庫相關事務，例如：資料存取與權限管理

# Module 6.

## 資料庫介紹

學習目標：

1. 認識資料庫設計師
2. 認識資料庫管理師
3. 認識應用程式設計師
4. 認識一般使用者

認識資料庫管理系統  
使用者



- **資料庫設計師 (Database Designer)**
  - 負責設計整個資料庫的架構，即為定義資料庫的綱要
- **資料庫管理師 (Database Administrator, DBA)**
  - 資料庫建好後交由資料庫管理師來維護資料庫的正常運作與定期備份
- **應用程式設計師 (Application Designer)**
  - 負責撰寫client端應用程式，讓使用者透過該應用程式與資料庫互動。幾乎所有程式語言都可以連接資料庫系統，例如：Java、C#、C++等
- **一般使用者 (End User)**
  - 透過client端的應用程式存取資料庫

# Module 7.

## 關聯式資料庫概論

學習目標：

1. 認識關聯式資料庫歷史
2. 認識關聯式資料庫之父

認識關聯式資料庫起源

- 關聯式資料庫 (Relational Database) ， 1970 年由IBM研究員 Edgar Frank Codd 提出
- 以表格 (table) 方式來儲存與呈現資料，再以數學集合論為基礎將表格與表格之間建立關聯以處理複雜的資料關係
- SQL 指令就是用於關聯式資料庫

# Module 7.

## 關聯式資料庫概論

學習目標：

1. 認識 DB-Engines
2. 了解關聯式資料庫系統市占率

認識關聯式資料庫管理  
系統與市佔率

- 關聯式資料庫管理系統 (Relational Database Management System, RDBMS) ，專門用來管理關聯式資料庫的系統
- 現今資料庫管理系統大多屬之
- DB-Engines 最受歡迎資料庫管理系統前4名皆屬之：
  - Oracle Database
  - MySQL Database
  - Microsoft SQL Server
  - PostgreSQL

學習目標：

1. 認識關聯式設計  
基本觀念

# Module 7.

## 關聯式資料庫概論


了解關聯式設計

- 所有資料儲存在一個表格內會有資料重複問題，資料量大時會造成
  - 浪費儲存空間
  - 資料錯誤率提高
  - 增加修改上困難

BOOK							
ISBN	BOOK_NAME	PRICE	AUTHOR	PUBLISHER_ID	PUBLISHER_NAME	CONTACT	PHONE
9780596009205	Head First Java	1186	Kathy Sierra	P001	O'Reilly	Ocean	02-23456789
9780596809157	R Cookbook	935	Paul Teetor	P001	O'Reilly	Ocean	02-23456789
9781491936696	iOS 9 Swift Programming Cookbook	1443	Vandad Nahavandipoor	P001	O'Reilly	Ocean	02-23456789
9781118407813	Programming with Java For Dummies	550	Barry Burd	P002	John Wiley, Sons Inc	Don	03-36962869
9781617291999	Java 8 in Action	936	Raoul-gabriel Urma	P003	Manning Publications	Mary	04-43456789
9781430237174	Pro IOS Apps Performance Optimization	1151	Khang Vo	P004	Apress	Allen	05-59876543
9781484226766	Python Unit Test Automation	731	Ashwin Pajankar	P004	Apress	Allen	05-59876543
9780071751292	Oracle Database 11g	1139	Jinyu Wang	P005	McGraw-Hill	Mike	06-69876543
9781259587405	Programming the Raspberry Pi	440	Simon Monk	P005	McGraw-Hill	Mike	06-69876543
9780134543666	Starting Out with Python	5451	Tony Gaddis	P006	Pearson	Paul	09-98767867

- 要解決資料重複的問題，可將前頁表格分割成「BOOK」與「PUBLISHER」2個表格；「PUBLISHER\_ID」即為關聯欄位

BOOK				
ISBN	BOOK_NAME	PRICE	AUTHOR	PUBLISHER_ID
9780596009205	Head First Java	1186	Kathy Sierra	P001
9780596809157	R Cookbook	935	Paul Teetor	P001
9781491936696	iOS 9 Swift Programming Cookbook	1443	Vandad Nahavandipoor	P001
9781118407813	Programming with Java For Dummies	550	Barry Burd	P002
9781617291999	Java 8 in Action	936	Raoul-gabriel Urma	P003
9781430237174	Pro IOS Apps Performance Optimization	1151	Khang Vo	P004
9781484226766	Python Unit Test Automation	731	Ashwin Pajankar	P004
9780071751292	Oracle Database 11g	1139	Jinyu Wang	P005
9781259587405	Programming the Raspberry Pi	440	Simon Monk	P005
9780134543666	Starting Out with Python	5451	Tony Gaddis	P006



PUBLISHER			
PUBLISHER_ID	PUBLISHER_NAME	CONTACT	PHONE
P001	O'Reilly	Ocean	02-23456789
P002	John Wiley, Sons Inc	Don	03-36962869
P003	Manning Publications	Mary	04-43456789
P004	Apress	Allen	05-59876543
P005	McGraw-Hill	Mike	06-69876543
P006	Pearson	Paul	09-98767867



# Module 7.

## 關聯式資料庫概論

學習目標：

1. 了解主鍵特性
2. 了解如何辨識主鍵
3. 了解主鍵建立過程

### 認識主鍵

- 主鍵 (Primary Key, PK) 欄位內的值是唯一不可重複的
- 主鍵欄位最好是個重要的辨識欄位
- 主鍵欄位不可為空值 (NOT NULL)
- 主鍵是一種 constraint，稱為限制或約束，也就是加上主鍵設定的欄位，值就不可重複

- 主鍵建立過程

- 超鍵 (Super Key)

- 符合唯一性
    - 假設除了 ISBN 書號，再新增書籍編號，方便自己使用
    - ISBN 書號、書籍編號、ISBN 書號 + 書籍編號 皆為超鍵

- 候選鍵 (Candidate Key)

- 符合唯一性與最小性
    - ISBN 書號、書籍編號符合最小性 (單一欄位較多個欄位符合最小性)
    - 主鍵是由候選鍵中挑選出來

- 次要鍵 (Alternate Key)

- 沒被選為主鍵的其他候選鍵
    - ISBN 書號被選為主鍵，書籍編號即為次要鍵

學習目標：

1. 認識外來鍵特性

# Module 7.

## 關聯式資料庫概論

認識外來鍵

- 外來鍵 (Foreign Key, FK) 欄位是為了關聯而存在的
- 外來鍵欄位必須參照到另一個表格的具有唯一性的欄位 (例如主鍵)，以避免多頭馬車
- 被參照的欄位必須有值，外來鍵欄位方可填寫該值
- 所以外來鍵也是一種 constraint，預防斷頭資料

BOOK				
ISBN	BOOK_NAME	PRICE	AUTHOR	PUBLISHER_ID
9780596009205	Head First Java	1186	Kathy Sierra	P001
9781118407813	Java For Dummies	550	Barry Burd	P002
9781617291999	Java 8 in Action	936	Raoul-gabriel Urma	P003 X



PUBLISHER			
PUBLISHER_ID	PUBLISHER_NAME	CONTACT	PHONE
P001	O'Reilly	Ocean	02-23456789
P002	John Wiley, Sons Inc	Don	03-36962869
P003 ?			

# Module 8.

## SQL介紹

學習目標：

1. 認識結構化查詢語言
2. 了解 SQL 特色

## 認識 SQL

- SQL (Structured Query Language , 結構化查詢語言) 是 IBM 公司於 1970 年初發展出來的一套專門用於關聯式資料庫存取的語法
- SQL 語法不區別大小寫
- ANSI 與 ISO 先後訂定了 SQL 標準
  - SQL-86, SQL-89, SQL-92, SQL:1999, SQL:2003, SQL:2006, SQL:2008, SQL:2011, SQL:2016

- 雖然 SQL 有前述標準，但資料庫廠商並沒有完全遵循，而對標準 SQL 做了一定程度的修改或擴充，而導致不一定相容情況
- 以 SQL 為基礎的擴展語言，常用於函式 (function) 或預存程序 (stored procedure)
  - Oracle DB 為 PL-SQL (Procedural Language - SQL)
  - MS SQL Server 為 Transact-SQL (T-SQL)
  - MySQL 也有自己語法，但沒有特別名稱，參看維基百科 [Stored procedure](#)



# Module 8.

## SQL介紹

學習目標：

1. 認識 DDL
2. 認識 DML
3. 認識 DQL
4. 認識 DCL

## SQL 語法種類

- **資料定義語言 (Data Definition Language, DDL)**
  - 建立、修改或移除資料庫或其他資料庫物件的語法
  - 以 CREATE、ALTER、DROP 語法為主
- **資料處理語言 (Data Manipulation Language, DML)**
  - 處理表格內資料的語法
  - 以 INSERT、UPDATE、DELETE 語法為主

- **資料查詢語言 (Data Query Language, DQL)**
  - SELECT 語法為主
  - 也有人將 SELECT 語法歸於 DML，而不細分出 DQL 語法
- **資料控制語言 (Data Control Language, DCL)**
  - 對使用者設定資料庫或其他資料庫物件使用權限的語法
  - 以 GRANT、REVOKE 語法為主

# Module 9.

## 建立/刪除資料庫

學習目標：

1. 學會使用 Workbench 建立資料庫
2. 學會使用 Workbench 刪除資料庫

使用 Workbench  
建立/刪除資料庫

- 建立資料庫
  - Schemas 列表空白處右鍵 > Create Schema
- 刪除資料庫
  - 指定 Database 右鍵 > Drop Schema

# Module 9.

## 建立/刪除資料庫

學習目標：

1. 學會使用 SQL 建立資料庫
2. 學會使用 SQL 刪除資料庫

使用 SQL  
建立/刪除資料庫

- 點擊工具列第1個按鈕開啟 SQL 頁籤
- 建立資料庫
  - `CREATE DATABASE (SCHEMA) db_name;`
- 刪除資料庫
  - `DROP DATABASE (SCHEMA) db_name;`
- 範例：M09 - CREATE / DROP DATABASE

# Module 9.

## 建立/刪除資料庫

學習目標：

1. 學會建立新的 SQL 檔案
2. 學會儲存 SQL 檔案
3. 學會開啟原有 SQL 檔案

### 開啟與儲存 SQL 檔案



- 建立新的 SQL 檔案
  - Workbench主選單 > File > New Query Tab
  - 點擊工具列第1個按鈕
- 儲存 SQL 檔案
  - Workbench 主選單 > File > Save Script
- 開啟原有 SQL 檔案
  - Workbench 主選單 > File > Open SQL Script
  - 點擊工具列第2個按鈕

# Module 10.

## 表格與欄位說明

學習目標：

1. 認識表格欄位制定
2. 認識正規化

## 了解表格設計流程

- 制定欄位

- 表格是主要儲存資料的地方，依照資訊系統功能來分析需要儲存哪些資料，並將其轉化成對應的欄位
- 蒐集資料
  - 查閱紙本表單或記錄
  - 與相關人員面談
  - 觀摩實際狀況
  - 例如：紙本訂單上需要記錄書籍名稱、價格、數量、送貨日期、送貨地址、聯絡人等，就可以先建立一個含有上述6個欄位的訂單表格

- 正規化

- 以資料結構的角度分析欄位的歸屬與表格的關聯
- 減少資料重複過冗、增進資料的一致性

# Module 10.

## 表格與欄位說明

學習目標：

1. 認識整數
2. 認識浮點數
3. 認識自訂精準位數

認識數字類型

- 整數
  - TINYINT (1 byte), SMALLINT (2 bytes), MEDIUMINT (3 bytes), INT (4 bytes), BIGINT (8 bytes) , signed / unsigned代表有 / 無正負號。參看下頁表格說明
- 浮點數
  - FLOAT (4 bytes), DOUBLE (8 bytes)
- 自訂精準位數
  - DECIMAL(M, D)
  - M標示總共的有效位數 ; scale 標示小數位數

Type	Storage (Bytes)	Minimum Value Signed	Minimum Value Unsigned	Maximum Value Signed	Maximum Value Unsigned
TINYINT	1	-128	0	127	255
SMALLINT	2	-32768	0	32767	65535
MEDIUMINT	3	-8388608	0	8388607	16777215
INT	4	-2147483648	0	2147483647	4294967295
BIGINT	8	$-2^{63}$	0	$2^{63}-1$	$2^{64}-1$

# Module 10.

## 表格與欄位說明

學習目標：

1. 認識 CHAR
2. 認識 VARCHAR
3. 認識 TEXT

認識文字類型



- **CHAR(M)**
  - 固定長度，空白字元填充
  - M標示字元數而非 bytes，M的值為 0 ~ 255
- **VARCHAR(M)**
  - 動態長度，用在不確定但有限度的字數上
  - M 代表字元數而非 bytes，M 的值為 0 ~ 65,535
  - VARCHAR 的最大長度不可超過前述一筆資料的長度限制 - 65,535 bytes
  - 不同字元集 ( character set ) 佔的空間不同，UTF-8 一個字元要佔用4 bytes，  
所以 VARCHAR(10) 會佔 40 bytes

- **TEXT**：需要儲存極大量文字時使用 (不確定字數且無限度的字數)
  - TEXT (0 ~ 65535 characters)
  - MEDIUMTEXT (0 ~ 16777215 characters)
  - LONGTEXT (0 ~ 4294967295 characters)

# Module 10.

## 表格與欄位說明

學習目標：

1. 認識 DATE
2. 認識 DATETIME
3. 認識 TIMESTAMP
4. 認識自動填入系統時間的型別

認識日期時間類型

- **DATE**
  - 格式：'YYYY-MM-DD'
  - 支援範圍：'1000-01-01' ~ '9999-12-31'
  - 官方文件說：超過支援範圍的日期應該也能處理，但不保證
- **DATETIME：**
  - 格式：'YYYY-MM-DD HH:MM:SS'
  - 支援範圍：'1000-01-01 00:00:00' ~ '9999-12-31 23:59:59'
  - 不儲存時區資訊：不將日期時間轉成UTC時間而直接儲存

- **TIMESTAMP**
  - 格式：'YYYY-MM-DD HH:MM:SS'
  - 支援範圍：'1970-01-01 00:00:01' UTC ~ '2038-01-19 03:14:07' UTC.
  - 會儲存時區資訊：會將日期時間轉成UTC時間再儲存
- **新增/修改時可以自動填入系統時間的型別**
  - **TIMESTAMP 與 DATETIME**
  - 新增時自動填入系統時間
    - **DEFAULT CURRENT\_TIMESTAMP** (Workbench建立不要加 Default，SQL 指令要加)
  - 修改時自動填入系統時間
    - **ON UPDATE CURRENT\_TIMESTAMP**

# Module 10.

## 表格與欄位說明

學習目標：

1. 認識二位元資料
2. 了解如何儲存大量資料

認識二位元資料

- 可儲存大量資料，例如圖片
  - BLOB (0 ~ 65535 bytes)
  - MEDIUMBLOB (0 ~ 16777215 bytes)
  - LONGBLOB (0 ~ 4294967295 bytes)

學習目標：

1. 認識主鍵建立
2. 認識 NOT NULL 限制

# Module 11.

## 表格建立、修改 與刪除

建立主鍵與  
NOT NULL 限制



- **Workbench**
  - Schemas > 指定資料庫 > Tables 右鍵 > Create Table

- SQL

- 指定要對哪個資料庫執行 SQL 語法
- 使用「USE db\_name;」指令，或雙點該資料庫
- 輸入建立表格的 SQL 語法

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    ....  
);
```

- 主鍵欄位後面直接加上 PRIMARY KEY 與 NOT NULL 限制
  - 也可在最後加上 PRIMARY KEY 設定
  - 新增時自動填入系統時間：DEFAULT CURRENT\_TIMESTAMP
  - 修改時自動填入系統時間：ON UPDATE CURRENT\_TIMESTAMP
- **範例：M11 - CREATE TABLE - PK**

# Module 11.

## 表格建立、修改 與刪除

學習目標：

1. 認識如何建立外來鍵
2. 認識外來鍵建立失敗可能原因

### 建立外來鍵限制

- 外來鍵與參照到的主鍵，資料類型必須一致
- Workbench
  - 點擊表格定義下方 Foreign Keys 頁籤
  - 填入 Foreign Key 名稱、選擇 Referenced Table (被參照表格)
  - 勾選 Column (外來鍵欄位)、Referenced Column (被參照欄位)
  - 點擊右下角 Apply 按鈕

- SQL

```
create table book (  
    isbn char(13) primary key not null,  
    book_name varchar(200) not null,  
    price decimal(8,2),  
    author varchar(200),  
    publication_date date,  
    publisher_id varchar(40),  
    constraint fk_book_publisher  
    foreign key (publisher_id) references publisher (publisher_id)  
);
```

- 範例：M11 - CREATE TABLE - FK

- 外來鍵建立失敗可能原因
  - 外來鍵與參照到的主鍵，資料類型不相同
  - 欲參照 (關聯) 到的欄位不是 PK (或不具有唯一性)
  - FK Constraint 的名稱重複

學習目標：

1. 學會修改表格定義

# Module 11.

## 表格建立、修改 與刪除

修改表格



- **Workbench**
  - 點選表格後滑向右邊第2個按鈕 (板手圖示)
  - 表格右鍵 > Alter Table
- **SQL 參看範例：M11 - ALTER TABLE**

學習目標：

1. 學會刪除表格定義

# Module 11.

## 表格建立、修改 與刪除

### 刪除表格

- **Workbench**
  - 指定表格右鍵 > Drop Table
- **SQL**
  - `DROP TABLE table_name;`
- **範例：M11 - DROP TABLE**

# Module 12.

## 新增資料與 自動編號

學習目標：

1. 學會使用 Workbench  
新增資料
2. 學會使用 SQL  
新增資料

### 新增資料

- **Workbench**
  - 點選表格後滑向右邊第3個按鈕
  - 表格右鍵 > Select Rows

- SQL

INSERT INTO table\_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

- 新增文字值要加上單引號「'」，如果值已經有單引號就再加上單引號跳脫，例如「O'Reilly」改成「'O''Reilly'」
- 新增日期/時間要使用'YYYY-MM-DD hh:mm:ss'格式，例如「'2005-02-19 21:02:44'」

- 範例：M12 - INSERT INTO PUBLISHER/BOOK

# Module 12.

## 新增資料與 自動編號

學習目標：

1. 認識自動編號功能
2. 了解自動編號建立方式

### 建立自動編號

- 建立表格時主鍵可以加上自動編號 (auto increment) 功能
  - 新增資料時，自動編號欄位會自動給值
- Workbench
  - PK 欄位勾選 AI
- SQL

```
CREATE TABLE table_name(  
    id int unsigned primary key not null auto_increment  
);
```
- 範例：M12 - Auto Increment



- MySQL 的 auto increment 欄位一定要為整數類型，所以該欄位不可有任何文字
- 如果希望商品 ID 為自訂前綴字 + 自動編號，建議再新增一個欄位存放前綴字，查詢時使用 CONCAT(前綴字, 自動編號)
- 範例：M12 - Prefix + Auto Increment

學習目標：

1. 認識自動編號相關資訊  
查詢方法

# Module 12.

## 新增資料與 自動編號

### 查詢自動編號

- 查詢下一個準備產生的號碼

```
SELECT AUTO_INCREMENT  
FROM INFORMATION_SCHEMA.TABLES  
WHERE TABLE_SCHEMA = '資料庫名稱'  
AND TABLE_NAME = '表格名稱';
```

- 顯示表格資訊也會呈現自動編號資訊 ( Auto-increment欄位 )

```
SHOW TABLE STATUS FROM '資料庫名稱'  
WHERE name LIKE '表格名稱';
```

- 範例：M12 - Auto Increment Info

# Module 13.

## 修改與刪除資料

學習目標：

1. 學會使用 Workbench 修改資料
2. 學會使用 SQL 修改資料

修改資料

- **Workbench**

- 指定表格右鍵 > Select Rows > 修改完成後點擊 Apply 按鈕

- **SQL**

UPDATE table\_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

- 預設修改、刪除沒有加上唯一鍵條件會被禁止；如果要取消，先關閉 Safe Updates，然後重新連線
- Preferences > SQL Editor > Other > Safe Updates

- **範例：M13 - UPDATE**

# Module 13.

## 修改與刪除資料

學習目標：

1. 學會使用 Workbench 刪除資料
2. 學會使用 SQL 刪除資料

### 刪除資料

- **Workbench**

- 指定表格右鍵 > View/Edit Data > All Rows > 點選欲刪除資料列後點擊 Delete 按鈕

- **SQL**

DELETE FROM table\_name

WHERE condition;

- 預設修改、刪除沒有加上唯一鍵條件會被禁止；如果要取消，先關閉 Safe Updates，然後重新連線
- 被外來鍵參照到的資料無法刪除

- **範例：M13 - DELETE**

# Module 14.

## 交易管理

學習目標：

1. 認識交易管理基本觀念
2. 認識 ACID

認識交易管理



- 交易管理 (Transaction Management)
- 對資料庫執行一個交易可能包含一連串的新增、修改或刪除指令。  
為了保證交易的正確與可靠，必須符合ACID (Atomicity, Consistency, Isolation, Durability)

- **Atomicity (單元性)**
  - 一個交易所含有的所有運算不是完全做完就是完全不做
- **範例**
  - A 從自己帳戶匯款 1000 元到 B 的帳戶。A 帳戶扣除 1000 元，B 帳戶增加 1000 元必須都做完
  - 執行過程中發生錯誤必須回復 (roll back) 到尚未匯款前的狀態

- **Consistency (一致性)**

- 交易前與交易後的結果都沒有破壞資料庫的完整性，也就是完全符合資料庫設定的規則

- **範例**

- A 從自己帳戶匯款 1000 元到 B 的帳戶。A 帳戶扣除的金額，必須與 B 帳戶增加的金額一致

- **Isolation (隔離性)**

- 同一筆資料在一個交易尚未完成前不可讓其他交易執行更改，也就是預防同時性 (synchronization) 可能造成的錯誤

- **範例**

- A、B 同時從自己帳戶匯款 1000 元到 C 的帳戶。當 A 匯款至 C 帳戶時，必須鎖定 C 帳戶直到匯款結束才能讓 B 匯款至 C 帳戶

- **Durability (永久性)**
  - 一旦交易成功，對資料的變更即永久有效，即使系統故障也不影響
  - 須仰賴資料庫備份與交易日誌 (log)

# Module 14.

## 交易管理

學習目標：

1. 認識交易與 Auto-Commit 之間關係
2. 了解如何查詢或修改 auto-commit 狀態

了解交易  
與 Auto-Commit

- 交易 (transaction) 可將多個異動視為一體，通常會搭配
  - rollback還原執行過的異動
  - commit確定交易 (無法還原執行過的異動)
- Workbench > Query > Auto-Commit Transactions 預設為開啟
  - 每次異動都會視為一個確定交易，所以無法還原
  - Auto-Commit 關閉後需要搭配 commit 方能確定交易
- 可以下指令顯示或修改auto-commit狀態
  - 顯示：select @@autocommit;
  - 修改：set autocommit = 0; ( 0 - 關閉；1 - 開啟 )
- 範例：M14 - Transaction

# Module 14.

## 交易管理

學習目標：

1. 認識儲存點觀念
2. 了解 Savepoint 設定

## 了解 Savepoint



- 使用 savepoint 可以設定 rollback 欲回復的儲存點
- 範例：M14 - SAVEPOINT

# Module 15.

## 匯出/匯入資料庫

學習目標：

1. 認識 Workbench  
匯出資料庫功能

匯出資料庫

- Workbench > Administration > Data Export > 勾選匯出資料庫與表格 > 選擇 Export to Self-Contained File > 點選右邊瀏覽按鈕並指定匯出的路徑與檔案名稱 > 點擊右下方 Start Export 按鈕

學習目標：

1. 認識 Workbench  
匯入資料庫功能

# Module 15.

## 匯出/匯入資料庫

匯入資料庫

- Workbench > Administration > Data Import/Restore > Import from Self-Contained File > 點選右邊瀏覽按鈕並指定匯入的 SQL 檔案 > Default Target Schema 點擊右邊 New 按鈕建立資料庫名稱 > 點擊右下方 Start Import 按鈕

# Module 16.

## 資料查詢

學習目標：

1. 認識 SQL 查詢語法
2. 了解查詢資料庫表格語法
3. 認識欄位別名

## 了解 SELECT 語法

- 查詢表格所有欄位內的值

```
SELECT * FROM table_name;
```

- 查詢表格指定欄位內的值 (可以計算)，可以給予欄位別名 (alias)

```
SELECT column1, column2 (AS) alias_name, ...  
FROM table_name;
```

- 查詢結果不會有資料重複情形

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

- 範例：M16 - SELECT

學習目標：

1. 了解排序語法
2. 認識 NULL 值

# Module 16.

## 資料查詢

了解 ORDER BY 語法



- 將查詢結果排序

SELECT column1, column2, ...

FROM table\_name

ORDER BY column1, column2, ... ASC|DESC;

➤ ASC或不寫都代表升冪排序；DESC代表降冪

- ORDER BY兩個以上欄位，可分別指定升冪或降冪

- **MySQL視NULL值小於non-NULL值**
  - 升冪排序：NULL排在non-NULL之前
  - 降冪排序：NULL排在non-NULL之後
  - 如果要自訂NULL排序先後，可搭配 ISNULL() 運算再做排序
  - ISNULL(NULL) 為1，其餘為0
- **範例：M16 - ORDER BY**

# Module 16.

## 資料查詢

學習目標：

1. 了解篩選條件
2. 認識 WHERE 語法  
可搭配之運算符號

## 了解 WHERE 語法

- 依據指定的條件查詢資料

```
SELECT column1, column2, ...
```

```
FROM table_name
```

```
WHERE condition;
```

- 可以搭配下列運算符號

= (等於)、<> (不等於)、> (大於)、>= (大於或等於)、< (小於)、

<= (小於或等於)、IS NULL (檢查是否為null)、LIKE (模糊比對)

- 注意：MySQL表格內的文字值不區別大小寫 (case-insensitive)
- 範例：M16 - WHERE

# Module 17.

## 邏輯運算

學習目標：

1. 認識AND, OR 語法特性
2. 認識 NOT 語法特性

了解 AND, OR,  
NOT 語法

- **AND, OR語法**

SELECT column1, column2, ...

FROM table\_name

WHERE condition1 **AND|OR** condition2 **AND|OR** condition3 ...;

- **NOT語法**

SELECT column1, column2, ...

FROM table\_name

WHERE **NOT** condition;

- **範例：M17 - AND, OR, NOT**

學習目標：

1. 認識 BETWEEN 語法特性

# Module 17.

## 邏輯運算

了解 BETWEEN 語法

- **BETWEEN語法 (類似AND語法)**  
SELECT column1, column2, ...  
FROM table\_name  
WHERE column **BETWEEN** low **AND** high ;
- **範例：M16 - BETWEEN**



學習目標：

1. 認識 IN 語法特性

# Module 17.

## 邏輯運算

了解 IN 語法

- **IN語法 (類似OR語法)**

SELECT column1, column2, ...

FROM table\_name

WHERE column **IN** (value1, value2, ...);

- **範例：M16 - IN**

學習目標：

1. 認識 MySQL 常用函式分類

# Module 18.

## MySQL 函式 1

認識 MySQL 函式種類

- MySQL提供許多實用的函式（function），這些函式可分成下列幾種類型
  - 字串函式
  - 數學函式
  - 日期時間函式
  - 資料類型轉換函式
  - 流程控制函式
  - 系統資訊函式

學習目標：

1. 認識字串函式特性
2. 了解文字處理與計算

# Module 18.

## MySQL 函式 1

### 了解字串函式

- 字串函式專門用於文字的處理與計算
- 字串索引位置皆為1-based (第1個字元位置的索引為1)
- 範例：M18 - 字串函式

學習目標：

1. 了解如何使用數學函式進行數字處理與計算

# Module 18. MySQL 函式 1

了解數學函式

- 數學函式提供數學上所需要的處理與計算
- 範例：M18 - 數學函式



# Module 19.

## MySQL 函式 2

學習目標：

1. 認識日期時間的處理與計算
2. 認識格式化

了解日期時間函式

- 日期時間函式專門用於日期時間的處理、計算與格式化
- 範例：M19 - 日期時間函式

學習目標：

1. 認識不同型別資料間的轉換

# Module 19.

## MySQL 函式 2

了解資料類型轉換函式

- 資料類型轉換函式專門用於不同型別資料間的轉換
- 範例：M19 - 資料類型轉換函式

學習目標：

1. 了解如何使用條件式的成立與否決定流程走向

# Module 19.

## MySQL 函式 2

了解流程控制函式

- 流程控制函式利用條件式的成立與否決定流程走向
- 範例：M19 - 流程控制函式

學習目標：

1. 了解如何取得系統相關資訊

# Module 19.

## MySQL 函式 2

了解系統資訊函式

- 系統資訊函式可以取得系統相關資訊
- 範例：M19 - 系統資訊函式



學習目標：

1. 認識 MySQL 提供的聚集函式種類

# Module 20.

## 聚集函式

認識聚集函式

- 聚集函式 (aggregate functions) 是一種提供簡易統計功能的函式
  - SUM() : 加總
  - AVG() : 平均
  - COUNT() : 計數
  - MAX() : 最大值
  - MIN() : 最小值

# Module 20.

## 聚集函式

學習目標：

1. 了解 SUM()：加總
2. 了解 AVG()：平均
3. 了解 COUNT()：計數

了解 SUM(), AVG(),  
COUNT()

- SUM()：加總
- AVG()：平均
- COUNT()：計數
- 範例：M20 - SUM(), AVG(), COUNT()

# Module 20.

## 聚集函式

學習目標：

1. 了解 MAX()：最大值
2. 了解 MIN()：最小值

了解 MAX(), MIN()

- MAX()：最大值
- MIN()：最小值
- 聚集函式都可合併使用
- 範例：M20 - MAX(), MIN()

學習目標：

1. 了解 UNION 功能

# Module 21.

## UNION 語法

認識 UNION

- **UNION將不同欄位做垂直合併，不會增加欄位，但會增加資料筆數**

```
SELECT column_name(s) FROM table1
```

```
UNION (ALL)
```

```
SELECT column_name(s) FROM table2;
```



# Module 21. UNION 語法

學習目標：

1. 了解 UNION 合併資料須知
2. 學會使用 UNION

了解 UNION 注意事項

- **UNION合併資料必須注意**

- 要合併的2個SELECT子句，其欄位數要相同
- 要合併的2個SELECT子句，資料類型要相容
- 合併後會以第一個SELECT子句的欄位名稱為主
- GROUP BY和HAVING子句可用於SELECT子句中，但不可用於UNION之後的結果
- ORDER BY只能用於UNION之後的結果，不可用於SELECT子句中
- UNION後若有重複的結果，只會顯示一筆；加上ALL參數，則會全部顯示出來

- **範例：UNION**

# Module 22.

## JOIN 語法

學習目標：

1. 認識 JOIN 功能
2. 了解 JOIN 與 UNION 差異
3. 認識 JOIN 種類

## 認識 JOIN

- JOIN 語法利用 ON 設定關聯條件，將數個表格橫向連接起來；  
也就是將不同欄位做水平合併，會增加欄位數

- JOIN 種類

- [ INNER ] JOIN

- 預設 JOIN 方式；只顯示雙方表格都有的值，INNER 可以省略

- LEFT [ OUTER ] JOIN

- 只顯示左邊表格有的值，右邊表格無此值者會顯示 NULL，OUTER 可以省略

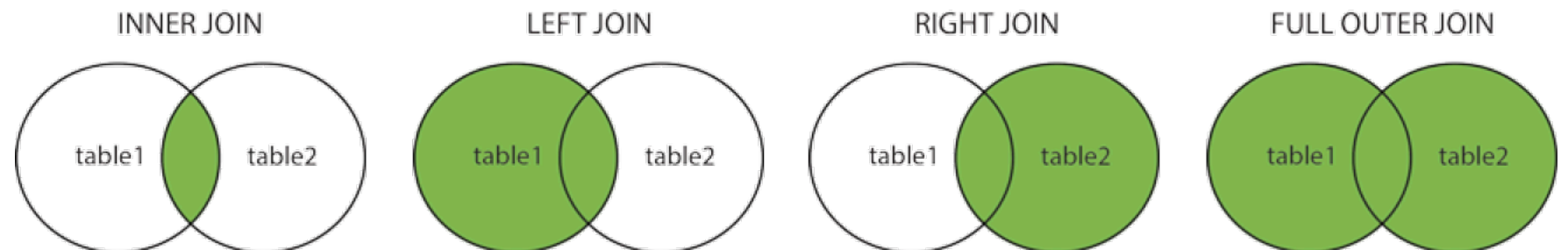
- RIGHT [ OUTER ] JOIN

- 只顯示右邊表格有的值，左邊表格無此值者會顯示 NULL，OUTER 可以省略

- FULL OUTER JOIN

- 顯示雙方表格都有的值，無值者會顯示 NULL，OUTER 不可省略；但是 MySQL 不支援

OUTER JOIN



# Module 22.

## JOIN 語法

學習目標：

1. 認識 INNER JOIN 觀念
2. 學會使用 INNER JOIN

## 了解 INNER JOIN

- **INNER JOIN**：表格都有相同值的資料列方會顯示，INNER 可省略，可使用表格別名

```
SELECT column_name(s) FROM table1 [ INNER ]  
JOIN table2 ON table1.column_name = table2.column_name  
JOIN table3 ON ...;
```

- **範例：M22 - INNER JOIN**

# Module 22.

## JOIN 語法

學習目標：

1. 了解 OUTER JOIN 與 INNER JOIN 差異
2. 認識 LEFT JOIN
3. 認識 RIGHT JOIN
4. 認識 FULL OUTER JOIN

## 了解 OUTER JOIN



- **LEFT JOIN**：只顯示左邊表格有的值，右邊表格無此值者會顯示 NULL

```
SELECT column_name(s) FROM table1
```

```
LEFT JOIN table2 ON table1.column_name = table2.column_name;
```

- **RIGHT JOIN**：只顯示右邊表格有的值，左邊表格無此值者會顯示 NULL

```
SELECT column_name(s) FROM table1
```

```
RIGHT JOIN table2 ON table1.column_name = table2.column_name;
```

- **範例：M22 - LEFT JOIN, RIGHT JOIN**

- FULL OUTER JOIN：顯示雙方表格都有的值，無值者會顯示 NULL。  
但是 MySQL 不支援 FULL OUTER JOIN，可將 LEFT JOIN 與 RIGHT JOIN 用 UNION 語法結合以達到目的
- 範例：M22 - FULL OUTER JOIN

# Module 22.

## JOIN 語法

學習目標：

1. 認識 SELF JOIN 功能
2. 學會使用 SELF JOIN

## 了解 SELF JOIN

- SELF JOIN 就是一個欄位參照到同表格的另一個欄位
- 同一個表格有 2 個欄位具有相同意義即可使用 SELF JOIN
- 範例：M22 - SELF JOIN

# Module 23. GROUP BY 語法

學習目標：

1. 認識 GROUP BY 語法
2. 學會使用 GROUP BY 語法進行資料的分組統計

了解 GROUP BY

- 資料要做分組統計，可以善用 GROUP BY 語法並搭配聚集函式

SELECT column\_name(s)

FROM table\_name

WHERE condition

GROUP BY column\_name(s)

- SELECT 語句列出的欄位中，除了聚集函式內的欄位外，其餘欄位應列在 GROUP BY 後面
- 範例：M23 - GROUP BY

# Module 23. GROUP BY 語法

學習目標：

1. 認識 HAVING 功能
2. 了解 HAVING 與 WHERE 差異
3. 學會使用 GROUP BY 語法搭配 HAVING 進行條件篩選

## 了解 HAVING

- **GROUP BY 語法可搭配 HAVING 子句將分組結果再加以篩選**

SELECT column\_name(s)

FROM table\_name

WHERE condition

GROUP BY column\_name(s)

HAVING condition;

- **範例：M23 - HAVING**



# Module 24.

## 子查詢

學習目標：

1. 認識子查詢功能
2. 了解子查詢規則
3. 學會使用子查詢

認識子查詢

- **SELECT 語句內可以再加上 SELECT 語句，內部的 SELECT 語句就是子查詢 (subquery)**
  - 子查詢需要放在小括弧內
  - 提供主要 SELECT 語句的資料來源或是條件判斷的依據
  - 在運算式內或是比較運算符號之後的子查詢，回傳的值只能1個
- **範例：M24 - Subquery**

# Module 24.

## 子查詢

學習目標：

1. 認識 IN 子查詢觀念
2. 認識 ANY 子查詢觀念
3. 認識 ALL 子查詢觀念
4. 了解 IN, ANY, ALL 三者差異

了解 IN, ANY, ALL 子查詢

- **IN**：用來判斷指定的值是否在子查詢中，子查詢的欄位只能有 1 個

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name IN (SELECT statement);
```

- **ANY/ALL**：用來判斷指定的值是否在子查詢中，ANY 類似 OR；

**ALL 類似 AND**

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name operator ANY|ALL (SELECT statement);
```

- **範例：M24 - IN, ANY, ALL**

- **IN, ANY, ALL 差別**

- IN: 單純比較是否與子查詢結果值相同，所以無法使用  $<>$ ,  $<$ ,  $>$ ,  $<=$ ,  $>=$
- ANY: 除了值是否相同外，還可以跟子查詢結果的任一個值比大小，屬於 OR 觀念
- ALL: 跟子查詢結果的每一個值比大小，屬於 AND 觀念

# Module 24.

## 子查詢

學習目標：

1. 認識 EXISTS 子查詢
2. 了解 EXISTS 與 IN 子查詢差異
3. 學會使用 EXISTS 子查詢

## 了解 EXISTS 子查詢

- **EXISTS**：檢查資料是否存在於子查詢中，如果有回傳 true，否則false，與上述 IN 做值的比對不同，所以子查詢欄位使用一般常數即可

```
SELECT column_name(s)
```

```
FROM table_name
```

```
WHERE EXISTS
```

```
(SELECT 1 FROM table_name WHERE condition);
```

- **範例：M24 - EXISTS**

- **EXISTS與IN差別**

- EXISTS 判斷主查詢的欄位是否存在於子查詢，而回傳 true 或 false 以決定是否要剔除該筆資料，該欄位大都是 PK，會以索引比對，所以速度較快



# Module 25.

## 限定資料筆數

學習目標：

1. 認識 LIMIT 功能
2. 學會使用 LIMIT 進行分頁查詢

了解 LIMIT 分頁查詢

- 只將查詢結果的前幾筆資料列出

```
SELECT column1, column2, ...
```

```
FROM table_name
```

```
LIMIT n OFFSET m;
```

- 會先執行 m 偏移量，再取 n 筆資料
- 可用於取得各分頁資料，例如取得第一頁 (1 ~ 10)、第二頁 (11 ~ 20)
- 範例：M25 - LIMIT

# Module 25.

## 限定資料筆數

學習目標：

1. 認識 RANK() 功能
2. 學會使用 RANK() 進行排名查詢

了解 RANK() 排名查詢

- 如果相同值資料要並列，要使用 RANK() 排名次；欄位值相同，名次也會相同

```
RANK() OVER (  
    ORDER BY <expression> [ASC|DESC]  
)
```

- 範例：M25 - RANK()

# Module 26.

## 利用查詢建立表格 與資料

學習目標：

1. 認識如何利用查詢語句建立表格
2. 了解 CREATE TABLE AS SELECT 語法

### 利用查詢建立表格

- 利用查詢建立表格

```
CREATE TABLE table_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

- 範例：M26 - CREATE TABLE AS SELECT

# Module 26.

## 利用查詢建立表格 與資料

學習目標：

1. 認識如何利用查詢語句新增資料
2. 了解 INSERT INTO SELECT 語法

### 利用查詢新增資料

- 利用查詢新增資料

`INSERT INTO table2 (column1, column2, column3, ...)`

`SELECT column1, column2, ...`

`FROM table_name`

`WHERE condition;`

- 範例：M26 - INSERT INTO SELECT



學習目標：

1. 認識 View 觀念

# Module 27. View

認識 View

- View (視表) 可說是一個虛擬表格
- 使用 SELECT 語法來建立 view
- View 永遠都呈現最新內容
- 每次 user 查詢 view 時，資料庫系統都會依照該 SELECT 語法重新建立一次 View 內容

學習目標：

1. 學會 View 建立方式

# Module 27. View

## 建立 View

- **建立 View**

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

- **建立或取代 View**

```
CREATE OR REPLACE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

- **範例：M27 - 建立 View**

# Module 27. View

學習目標：

1. 學會查詢 View
2. 學會刪除 View

## 查詢與刪除 View

- **查詢 View**

`SELECT column1, column2, ... FROM view_name;`

- **刪除 View**

`DROP VIEW view_name;`

- **範例：M27 - 查詢 View、刪除 View**

# Module 28.

## 索引

學習目標：

1. 認識索引功能
2. 學會查詢表格的索引資訊
3. 認識有無索引的效能差異

## 認識索引

- 如果要找書的內容，一般會從目錄找起，然後再翻倒指定頁面。如果從書的第一頁開始找則曠日費時
- 資料庫的索引 (index) 類似書的目錄，沒有建立索引，就會從第一筆資料搜尋到最後一筆，執行效能會很差



- 效能評估
- 查詢表格的索引資訊  
SHOW INDEX FROM table\_name;
- 評估有無索引的效能差異  
EXPLAIN + SELECT語句
  - 查看執行結果的rows欄位資訊
- 範例：M28 - 查詢表格的索引資訊

# Module 28.

## 索引

學習目標：

1. 認識主索引應用
2. 認識唯一索引應用
3. 認識非唯一索引應用
4. 認識複合索引應用

認識索引種類與應用

- 主索引 ( primary index )
  - 建立主鍵時就會自動建立主索引
  - 該欄位值不可重複、不可為 NULL
- 唯一索引 (unique index)
  - 很類似主索引，唯一不同是值可以為 NULL

- 非唯一索引 (non-unique index)
  - 值可以重複，也可以為 NULL
  - 用在查詢條件欄位上，可以加快查詢速度
- 複合索引 (composite indexes)
  - 將多個欄位一起加上索引，可以套用在任何類型索引上
  - 查詢條件可能有多個欄位，如果都建立單一索引，MySQL 就會使用他認為最佳的索引來搜尋資料。如果條件欄位有多個而且索引有使用上的優先序，就可以建立複合索引，並且將優先序高的索引欄位列在左，低的列在右邊

# Module 28.

## 索引

學習目標：

1. 學會建立不同索引
2. 學會刪除索引

### 建立及刪除索引

- **主索引**

- 建立/移除 主鍵時就會自動 建立/移除 主索引
- 範例：M28 - 建立主索引

- **唯一索引**

- 建立/移除 唯一鍵時就會自動 建立/移除 唯一索引
- 範例：M28 - 建立唯一索引

- 非唯一索引

- 建立索引 (單一或複合)

- ```
CREATE INDEX index_name ON table_name (column_list);
```

- ```
ALTER TABLE table_name ADD INDEX index_name (column_list);
```

- 刪除索引

- ```
DROP INDEX index_name ON table_name;
```

- 範例：M28 - 建立非唯一索引

# Module 28.

## 索引

學習目標：

1. 認識如何評估是否建立索引
2. 了解建立索引須知

了解建立索引應注意事項



- 使用在 WHERE、ORDER BY 或 GROUP BY 的欄位可以考慮建立索引，搜尋比對時執行效能會大幅提升
  - 如果該欄位值的變化小，例如只有 0 與 1，可以不用建立索引，因為效果不好
- 索引的欄位儘量不為 NULL，以提升執行效能
- WHERE 後有 OR 就必須為每個 OR 欄位建立索引，否則 無效
- 如果將索引欄位放在函式內就會失去索引效果
- 建立索引會佔用空間，所以不要建立過多索引

# Module 29.

## 管理 User 帳號

學習目標：

1. 學會使用 Workbench  
列出 User 帳號
2. 學會使用 SQL 語法  
列出 User 帳號

列出所有 User 帳號

- **Workbench**
  - Administration > Users and Privileges
  - User為帳號，From Host為可以登入的主機
- **SQL**  
`SELECT user, host from mysql.user;`
- **範例：列出所有User帳號**

# Module 29.

## 管理 User 帳號

學習目標：

1. 學會使用 Workbench 新增 User 帳號
2. 學會使用 SQL 語法 新增 User 帳號

### 新增 User 帳號

- **Workbench**

- Administration > Users and Privileges > Add Account按鈕

- **SQL**

- 格式為「username@hostname」。「identified by」為設定密碼

- create user john@localhost identified by 'john';

- 新增帳號時可給予預設密碼，同時要求user一登入就要重設密碼

- create user john@localhost identified by '1234' password expire;

- 新增帳號時要求user一登入就要重設密碼（沒有預設密碼，所以user無需輸入舊密碼）

- create user john@localhost password expire;

- 沒有加上hostname則hostname為「%」，代表user可以從任何主機登入

`create user john identified by 'john';`

- username或hostname有空白或是符號要加上單引號

`create user john@'%' identified by 'john';`

- **範例：新增User帳號**

# Module 29.

## 管理 User 帳號

學習目標：

1. 學會使用 Workbench  
修改 User 資訊
2. 學會使用 SQL 語法  
修改 User 密碼

### 修改 User 帳號

- **Workbench**

- Administration > Users and Privileges > 選擇user帳號
- 可以直接修改密碼，或是點擊Expire Password按鈕要求重設密碼

- **SQL**

- 直接修改user密碼
  - alter user john@localhost identified by 'john11';
- 要求user重設密碼
  - alter user john@localhost password expire;

- **範例：修改 User 密碼**



# Module 29.

## 管理 User 帳號

學習目標：

1. 學會使用 Workbench 刪除 User 帳號
2. 學會使用 SQL 語法 刪除 User 帳號

## 刪除 User 帳號

- **Workbench**

- Administration > Users and Privileges > 選擇user帳號 > 點擊Delete按鈕

- **SQL**

- 刪除user帳號，沒有加上hostname則hostname為「%」

- ```
drop user john@localhost;
```

- **範例：刪除User帳號**

# Module 30. User 授權

學習目標：

1. 認識授權觀念
2. 了解查詢 User 權限方式
3. 認識 MySQL 支援授權種類

認識授權

- 新增完 user 帳號後往往需要對該帳號加以授權，授權完畢後亦可視情況撤銷該權限
- 查詢授權
- Workbench
  - Administration > Users and Privileges > 選擇user帳號 > 選取Schema Privileges頁籤
  - 無論查詢、建立授權等功能都不如SQL指令完整，不建議使用
- SQL

```
show grants for john@localhost;
```
- 範例：查詢User權限

- MySQL支援下列授權種類

- Global Privileges ( 全域級授權 )

- 針對MySQL server上的所有資料庫及其表格，使用「\*.」語法

- Database Privileges ( 資料庫級授權 )

- 針對指定資料庫內的所有表格，使用「db\_name.\*」語法

- Table Privileges ( 表格級授權 )

- 針對指定表格內的所有欄位，使用「db\_name.table\_name」語法

- Column Privileges ( 欄位級授權 )

- 精準指定要授權哪些欄位

# Module 30. User 授權

學習目標：

1. 認識全域級授權
2. 認識「\*.」語法
3. 學會 Global Privileges 應用

## 了解 Global Privileges

- Global Privileges授權針對MySQL server上的所有資料庫及其表格，使用「\*.\*」語法
  - 授權user可在MySQL Server上建立、移除資料庫或表格  
`grant create, drop on *.* to john@localhost;`
  - 撤銷上述授權  
`revoke create, drop on *.* from john@localhost;`
- 範例：Global Privileges

# Module 30. User 授權

學習目標：

1. 認識資料庫級授權
2. 認識「db\_name.\*」語法
3. 學會 Database Privileges 應用

## 了解 Database Privileges



- Database Privileges授權針對指定資料庫內的所有表格，使用「db\_name.\*」語法
  - 授權user可查詢bookshop資料庫內的所有表格  
`grant select on bookshop.* to john@localhost;`
  - 撤銷上述授權  
`revoke select on bookshop.* from john@localhost;`
- 範例：Database Privileges

# Module 30. User 授權

學習目標：

1. 認識表格級授權
2. 認識「db\_name.table\_name」語法
3. 學會 Table Privileges 應用
4. 學會建立唯讀 View

## 了解 Table Privileges

- Table Privileges授權針對指定表格內的所有欄位，使用「db\_name.table\_name」語法
  - 授權/撤銷 user可查詢bookshop資料庫內的book表格

```
grant select on bookshop.book to john@localhost;  
revoke select on bookshop.book from john@localhost;
```
  - 授權/撤銷 user可查詢bookshop資料庫內的publisher表格

```
grant select on bookshop.publisher to john@localhost;  
revoke select on bookshop.publisher from john@localhost;
```
- 範例：Table Privileges

- **建立唯讀View**
  - 可以提供其他系統讀取資料
  - Oracle支援唯讀View但MySQL不支援
  - MySQL可以改用授權方式達到建立唯讀View目的
  - 先建立view
  - 再授權user只能查詢該view
- **範例：Read-Only View**

# Module 30. User 授權

學習目標：

1. 認識欄位級授權
2. 學會 Column Privileges 應用

## 了解 Column Privileges

- **Column Privileges授權針對表格內的指定欄位**

- 授權book表格內可查詢/修改的欄位

- ```
grant select(isbn, book_name, price), update(price) on book  
to john@localhost;
```

- 撤銷上述授權

- ```
revoke select(isbn, book_name, price), update(price) on book  
from john@localhost;
```

- **範例：Column Privileges**