



網頁程式設計

PHP程式設計

Instructor: 馬豪尚



MySQL關聯式資料庫

關聯式資料庫

- › MySQL (Maria) DataBase 就是一個關聯式資料庫
 - 由一個到多個資料表所組成
 - 多個資料表之間使用欄位資料值來建立關係
 - 可以透過關聯性來存取其他資料表的相關資料
- › 資料表由欄位和紀錄(資料)組成

欄位				
紀錄	學號	姓名	性別	生日
	S1001	Allen	男	2000/1/3
	S1002	Kevin	男	2000/3/5
	S1003	Amy	女	2000/4/12
	S1004	Alice	女	2000/5/6

關聯式資料庫

› 不同資料表之間透過欄位資料值來建立關係

資料表1: 基本資料

學號	姓名	性別	生日
S1001	Allen	男	2000/1/3
S1002	Kevin	男	2000/3/5
S1003	Amy	女	2000/4/12
S1004	Alice	女	2000/5/6

資料表2: 修課紀錄

學號	課程名稱	開課系所	學分
S1003	資料庫資統	資工系	3
S1003	程式設計	資工系	3
S1004	英文(二)	資管系	3
S1004	離散數學	資管系	3

關聯式資料庫--資料表的鍵

- › 索引(index)代表的是資料庫的物理性結構，是輔助查詢用的
- › 鍵(Key)則包含了兩個功能
 - 約束 (constraint)
 - 索引 (index), 也就是 key 功能上包含了 index.
- › Primary key
 - 用來約束一個主鍵和具有唯一性，且一個資料表只有一個
 - 同時也在此key上建立了一個主鍵索引
 - 不能為空值
- › Foreign key
 - 指向另一個資料表的主鍵欄位，目的是確定兩個資料表的關聯性

MySQL 建立資料庫

- › 建立資料庫
 - 輸入新資料庫名稱

資料庫

SQL

狀態

使用者帳號

匯出

匯入

設定

備援

變數

資料庫

建立新資料庫

資料庫名稱

utf8mb4_general_ci

▼

建立

☐ 全選

刪除

資料庫	編碼與排序	動作
<input type="checkbox"/> information_schema	utf8_general_ci	檢查權限
<input type="checkbox"/> mysql	utf8mb4_general_ci	檢查權限
<input type="checkbox"/> performance_schema	utf8_general_ci	檢查權限
<input type="checkbox"/> phpmyadmin	utf8_bin	檢查權限
<input type="checkbox"/> test	latin1_swedish_ci	檢查權限

總計：5

MySQL 刪除資料庫

- › 勾選要刪除的資料庫
— 執行刪除指令

建立新資料庫 ?

資料庫名稱

utf8mb4_general_ci

建立

☐ 全選

☒ 刪除

	資料庫	編碼與排序	動作
<input type="checkbox"/>	information_schema	utf8_general_ci	檢查權限
<input type="checkbox"/>	mysql	utf8mb4_general_ci	檢查權限
<input type="checkbox"/>	performance_schema	utf8_general_ci	檢查權限
<input type="checkbox"/>	phpmyadmin	utf8_bin	檢查權限
<input type="checkbox"/>	test	latin1_swedish_ci	檢查權限
<input checked="" type="checkbox"/>	webprogram	utf8mb4_general_ci	檢查權限

總計：6

MySQL 建立資料表

› 建立資料表

- 輸入資料表名稱
- 輸入欄位數
- 設定資料表內欄位的資訊
 - › 欄位名稱、欄位資料型態、欄位編碼、設定索引(主鍵、一般索引) ...

結構 ⓘ

名稱	類型 ⓘ	長度 / 值 ⓘ	預設值 ⓘ	編碼與排序	屬性	空值 (Null)	索引
<input type="text" value="studentID"/> <small>從中央欄位挑選</small>	INT ▼	<input type="text"/>	無 ▼	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	PRIMARY ▼ PRIMARY
<input type="text" value="Name"/> <small>從中央欄位挑選</small>	TEXT ▼	<input type="text"/>	無 ▼	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	--- ▼
<input type="text" value="Gender"/> <small>從中央欄位挑選</small>	TEXT ▼	<input type="text"/>	無 ▼	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	--- ▼
<input type="text" value="Birth"/> <small>從中央欄位挑選</small>	TEXT ▼	<input type="text"/>	無 ▼	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	--- ▼

MySQL 在資料表內新增資料

› 輸入每一個欄位的值

- 若沒有設定該欄位值可為空值，就一定要指定值
- 執行新增指令

欄位	類型	函式	空值(Null)	值
studentID	varchar(10)	<input type="text"/>		<input type="text" value="S1002"/>
Name	varchar(10)	<input type="text"/>		<input type="text" value="Kevin"/>
Gender	varchar(5)	<input type="text"/>		<input type="text" value="男"/>
Birth	varchar(20)	<input type="text"/>		<input type="text" value="2000/3/5"/>

執行

MySQL 匯出資料庫

› 進入要匯出的資料庫選擇匯出標籤



正在匯出「webprogram」資料庫的資料表

匯出模板：

新模板：

模板名稱

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：

SQL

MySQL 匯入資料庫

› 選擇要匯入的資料庫檔案

匯入到目前伺服器

要匯入的檔案：

可使用壓縮 (gzip, bzip2, zip) 或無壓縮的檔案。

壓縮檔案名稱必須以 **.[格式].[壓縮方式]** 結尾。如：**.sql.zip**

由電腦上傳：(上限：40 MB)

選擇檔案

也可以在任何頁面拖放檔案。

檔案字元編碼：

MySQL 管理使用者權限

› 在使用者帳號的標籤下可以看到目前有權限的使用者

伺服器: 127.0.0.1

資料庫 SQL 狀態 使用者帳號 匯出 匯入 設定 備援 變數 字元集

使用者帳號一覽 使用者群組

使用者帳號一覽

	使用者名稱	主機名稱	密碼	全域權限	使用者群組	允許授權(Grant)	動作
<input type="checkbox"/>	任何	%	否	USAGE		否	 編輯權限  匯出  Lock
<input type="checkbox"/>	pma	localhost	否	USAGE		否	 編輯權限  匯出  Lock
<input type="checkbox"/>	root	127.0.0.1	否	ALL PRIVILEGES		是	 編輯權限  匯出  Lock
<input type="checkbox"/>	root	::1	否	ALL PRIVILEGES		是	 編輯權限  匯出  Lock
<input type="checkbox"/>	root	localhost	否	ALL PRIVILEGES		是	 編輯權限  匯出  Lock

↑ ☐ 全選 已選擇項目:  匯出

MySQL 管理使用者權限

› 更改使用者密碼

全域 資料庫 Change password 登入資訊

編輯權限：使用者帳號 `'root'@'localhost'`

⚠ 注意：正在嘗試編輯您目前登入使用者的權限。

更改密碼

☐ 無密碼

☒ 密碼：

輸入：

強度：

重新輸入：

密碼雜湊(Hashing)方式：

原生 MySQL 認證 ▾

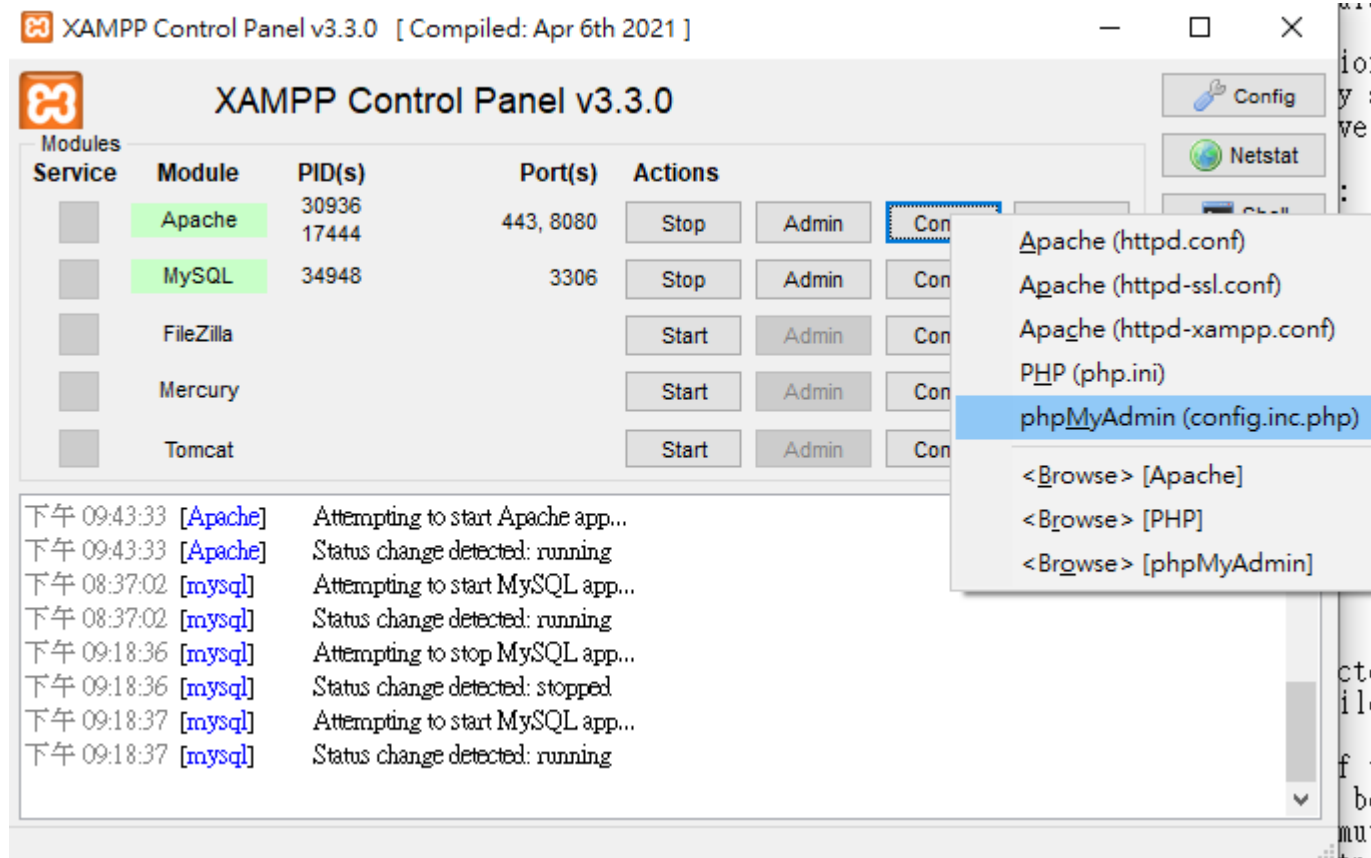
產生密碼

產生

執行

XAMPP更改管理者登入資訊

› 編輯xampp預設的管理者登入資訊



XAMPP更改管理者登入資訊

› 編輯config.inc裡

– \$cfg['Servers'][\$i]['password'] = '設定的密碼';

```
config.inc - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明
* cookie
*/
$config['blowfish_secret'] = 'xampp'; /* YOU SHOULD CHANGE THIS FOR A

/*
 * Servers configuration
 */
$i = 0;

/*
 * First server
 */
$i++;

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

/* Bind to the localhost ipv4 address and tcp */
$config['Servers'][$i]['host'] = '127.0.0.1';
$config['Servers'][$i]['connect_type'] = 'tcp';

/* User for advanced features */
$config['Servers'][$i]['controluser'] = 'pma';
$config['Servers'][$i]['controlpass'] = '';
```

PHP與MySQL

PHP開啟與MySQL伺服器的連結

- › 要存取資料庫前要先與伺服器建立連線
 - 用mysql_connect()函數來建立連結
 - 若連結成功回傳資源連接物件，失敗則回傳false

Example

```
$link = mysql_connect(  
    'localhost', // MySQL主機名稱  
    'root',      // 使用者名稱  
    'A12345678', // 密碼  
    'myschool'); // 預設使用的資料庫名稱
```

PHP開啟指定的資料庫

- › `mysqli_select_db($link, "course_db")`
 - `$link`為伺服器連結的物件
 - 第二個參數為指定的資料庫名稱
- › 使用完資料庫後，關閉連結以釋放記憶體空間
 - 用`mysqli_close($link)`來關閉
 - `$link`為開啟的伺服器連結物件

SQL的資料表查詢指令

› 基礎資料查詢指令

- SELECT 欄位 FROM 資料表名稱

- › 從資料表內查詢某個欄位的資料

› Example

- SELECT * FROM student

- 查詢student資料表內所有的欄位

PHP執行SQL查詢

- › 將SQL查詢指令包裝成一個字串
 - `$sql = "SELECT * FROM student"`
- › 用`mysqli_query($link, $sql)`函式來執行查詢指令
 - 指定伺服器連結`$link`
 - 傳入查詢指令`$sql`
 - 回傳結果的物件稱為**結果物件**

Example

```
$result = mysqli_query($link, $sql)
```

SQL的資料表查詢指令

› 有條件資料查詢指令

–SELECT 欄位 FROM 資料表名稱 WHERE 條件

- › 代表資料需要符合這個條件才會被查詢到
- › 條件子句可以是文字、數值或日期/時間
- › 條件支援邏輯運算子

Example

```
SELECT name FROM student WHERE sno='S001'
```

查詢student資料表內sno欄位值為S001的資料

只回傳name欄位的資料

運算子	說明
=	等於
<>	不等於
>	大於
>=	大於等於
<	小於
<=	小於等於
LIKE	包含子字串

PHP取得結果物件內容值

- › 將結果物件取出一筆資料儲存到關聯陣列
 - `mysqli_fetch_assoc($result)`
 - › `$result`為結果物件
 - › 回傳值為結果物件中一筆資料的關聯陣列
 - › 關聯陣列中的索引為結果物件中的欄位名稱

Example

```
$row=mysqli_fetch_assoc($result)  
echo $row['sno']
```

PHP取得結果物件內容值

- › 將結果物件取出一筆資料儲存到索引陣列
 - `mysqli_fetch_row($result)`
 - › `$result`為結果物件
 - › 回傳值為結果物件中一筆資料的索引陣列

Example

```
$row=mysqli_fetch_row($result)
```

```
echo $row[0] //表示印出第一個欄位值
```

PHP取得結果物件內容值

- › 將結果物件取出一筆資料儲存到陣列
 - `mysqli_fetch_array($result, type)`
 - › `$result`為結果物件
 - › `type`有兩種，根據不同的`type`設定有兩種回傳值
 - `MYSQLI_NUM` → 回傳值儲存成索引陣列
 - `MYSQLI_ASSOC` → 回傳值儲存成關聯陣列

Example

```
$row= mysqli_fetch_array($result, MYSQLI_NUM)
echo $row[0] //表示印出第一個欄位值
$row= mysqli_fetch_array($result, MYSQLI_ASSOC)
echo $row['sno'] //表示印出sno欄位值
```


PHP取得結果物件內容值

- › 移動記錄指標
 - `mysqli_data_seek($result, n)`
 - › `$result`為結果物件
 - › `n`為一個數值
 - › 成功則返回TRUE，如果失敗則返回FALSE
- › 找尋結果物件中第N筆資料
 - 使用`mysqli_data_seek($result, n)`將指標移動到第n筆
 - 搭配`mysqli_fetch_row($result)`取出

PHP取得結果物件內容值

- › 取得查詢結果物件的欄位數
 - `mysqli_num_fields($result)`
 - › `$result`為結果物件
 - › 回傳查詢結果物件中的欄位數
- › 取得查詢結果物件的資料筆數
 - `mysqli_num_rows($result)`
 - › `$result`為結果物件
 - › 回傳查詢結果物件中的資料筆數

PHP取得資料表和欄位資訊

- › 取得資料表中欄位資訊
 - `$meta = mysqli_fetch_field($result)`
 - › `$result`是結果物件
 - › 回傳為一個物件，物件成員包含多個欄位資訊
 - `$meta->name`：取得欄位名稱

成員變數	說明
<code>name</code>	欄位名稱
<code>orgname</code>	如果有使用別名，取得欄位真正的名稱
<code>table</code>	欄位所屬的資料表名稱
<code>orgtable</code>	如果有使用別名，取得資料表真正的名稱
<code>def</code>	欄位的預設值
<code>max_length</code>	欄位值的最大長度
<code>type</code>	欄位的資料類型

SQL的新增資料指令

- › INSERT INTO table(column1, column2) VALUES ('value1', 'value2')
 - 插入table資料表的column1欄位、column2欄位
 - 其中column1欄位的資料值為value1
 - 其中column2欄位的資料值為value2
 - 插入欄位的順序可以和資料表內不同，但是需要選擇資料表內有的欄位
 - 欄位名稱和值要對應到同一個順序

Example

```
INSERT INTO student(sno, name, address, birthday, username, password)
VALUES ('S007', '詹姆士', 'Taipei', '1999/4/1', 'James', '1234')
```

SQL的更新資料指令

- › UPDATE table SET column1 = 'value1'
 - 更新資料表table裡column1欄位的值為value1
 - 通常要搭配一個條件，否則會全部更新
- › UPDATE table SET column1 = 'value1', column2 = 'value2' WHERE 條件
 - 當條件符合時更新該筆資料的column1欄位值為value1，column2欄位值為value2

Example

```
UPDATE student SET address = '台中市北區' WHERE  
sno='S007'
```

SQL刪除資料指令

- › DELETE FROM table WHERE 條件
 - 刪除資料表table內符合條件的資料
 - 若沒有指定條件會將整個資料表所有資料都刪除

Example

```
DELETE FROM student WHERE sno='S007'
```

PHP執行SQL資料操作指令

› 執行SQL指令

–mysql_query(\$link, \$sql)

› \$link為伺服器連線物件, 需要指定資料庫

› \$sql為儲存成字串變數的sql操作指令(新增、更新、刪除等)

› 取得執行SQL指令之後的被影響資料數

–mysql_affected_rows(\$link)

› 取得前一個指令操作影響的資料數

› 回傳0是沒有資料被影響，回傳-1是有操作錯誤

› 可以用這個函式來檢查是否操作成功

實作範例 通訊錄

› 通訊錄應用的程式架構

