



國立臺灣大學 National Taiwan University

13-關聯式資料庫

陳永樵





Ch13 關聯式資料庫

- 資料庫是常見資料儲存系統，若需要儲存大量資料，會以資料庫儲存於電腦系統中
- **關聯式資料庫(Relational database)** 提供結構化方式儲存資料，可以結合資料庫中多個資料表，資料庫提供在資料表中新增、刪除與更新資料，依據條件取出所需資料等功能





- 許多系統的資料儲存都是使用關連式資料庫
- 例如：校務行政系統、進出貨系統、學校網頁等，廣泛應用於各個領域。





13-1 關聯式資料庫概論

- 關聯式資料庫由事先定義好的多個資料表 (**table**) 組合而成，資料表中可以包含多筆資料，每一筆資料也稱作一筆記錄，佔用一行 (**row**)，每一行可以包含多個欄位 (**column**) 資料。



13-1 關聯式資料庫概論

- 一個資料庫可以有很多張資料表，以學生成績系統為例，至少需要有學生基本資料表與成績表
- 學生資料表：可能有學號、姓名、身分證字號與電話等欄位
- 成績表：可能有學號、學期、科目與成績等欄位，舉例如下。



13-1 關聯式資料庫概論

stu(學生資料表)

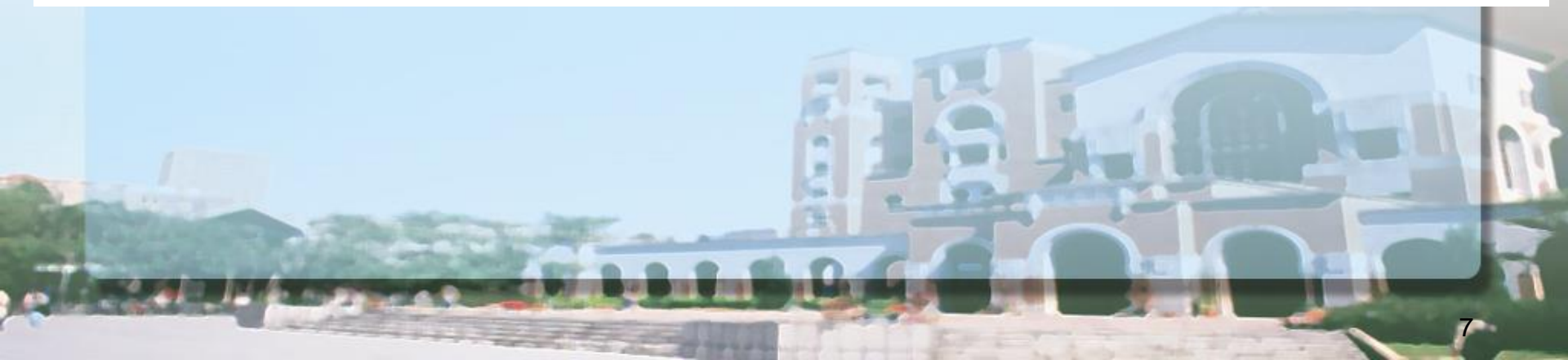
stuid(學號)	name(姓名)	pid(身分證字號)	phone(電話)
104001	Claire	B342222	245667
104002	John	J224122	222455
104003	Fiona	A152453	132435





score(成績表)

stuid(學號)	sem(學期)	sub(科目)	score(成績)
104001	1041	CH	95
104001	1041	EN	83
104002	1041	CH	65
104002	1041	EN	96
104003	1041	CH	85
104003	1041	EN	87





13-1 關聯式資料庫概論

- 可以在資料表設定某個欄位為主索引鍵(primary key)
- 主索引鍵需符合資料表中每一列資料的主索引鍵不能重複，且不可以是空值(null)，空值(null) 表示該儲存格沒有資料。



13-2 SQL 語法

- **SQL (Structural Query Language)** 是操作關聯式資料庫的語法，可以建立與刪除資料庫，建立與刪除資料表，插入、刪除與更新資料到資料表，結合多張資料表依據條件取出資料等操作都可以使用SQL 語法達成。



- SQL 分成DDL(Data Definition Language)
、DML(Data Manipulation Language) 與
DCL(Data Control Language) 三部分，學
會這些語法就可以管理關聯式資料庫
 - DDL 用於建立資料庫與定義資料表的資料結構
 - DML 用於選取、新增、刪除與更新資料表資料
 - DCL 用於資料庫的權限控管



DDL 指令

常用 DDL 指令，如下表。

指令	說明	語法與範例
CREATE	建立資料庫或資料表	CREATE DATABASE [資料庫名稱] CREATE TABLE [資料表名稱] ([資料表中每個欄位的定義])
		CREATE DATABASE school CREATE TABLE stu (stuid INTEGER PRIMARY KEY, name VARCHAR(50) not null, pid VARCHAR(20) not null, phone VARCHAR(20) not null)



DDL 指令

指令	說明	語法與範例
DROP	刪除資料庫或資料表	DROP [DATABASE 或 TABLE] [資料庫或資料表名稱]
		DROP DATABASE school DROP TABLE stu
USE	切換資料庫	USE [資料庫名稱]
		USE school
ALTER	修改資料表欄位	ALTER TABLE [資料表名稱] [動作]
		ALTER TABLE stu ADD addr VARCHAR(100) ALTER TABLE stu DROP COLUMN addr



DML 指令

常用 DML 指令，如下表。

指令	說明	語法與範例
SELECT	從資料表選取資料	SELECT 欄位名稱 FROM [資料表名稱] [WHERE 條件]
		SELECT * FROM stu WHERE stuid='104001'
INSERT	將資料插入資料表	INSERT INTO [資料表名稱] (欄位名稱 1 [, 欄位名稱 2, ...]) VALUES (數值 1 [, 數值 2,...])
		INSERT INTO stu (stuid,name,pid,phone) VALUES (104001,'Claire','B342222','1245667') 或 INSERT INTO stu VALUES (104001,'Claire','B342222','1245667')
UPDATE	更新資料表中資料	UPDATE [資料表名稱] SET 欄位名稱 1 = 數值 1 [, 欄位名稱 2 = 數值 2 ...] [WHERE 條件]
		UPDATE stu SET phone = '1245678' WHERE stuid=104001
DELETE	刪除資料表中資料	DELETE FROM [資料表名稱] [WHERE 條件];
		DELETE FROM stu WHERE stuid = 104003



DCL 指令

常用 DCL 指令，如下表。

指令	說明	語法與範例
GRANT	允許特定使用者存取指定資料庫的權限。	GRANT [權限] ON [資料庫名稱].[資料表名稱] TO [資料庫使用者]
		GRANT SELECT, UPDATE ON school.* TO john@localhost
REVOKE	取消特定使用者存取指定資料庫的權限。	REVOKE [權限] ON [資料庫名稱].[資料表名稱] TO [資料庫使用者]
		REVOKE SELECT, UPDATE ON school.* TO john@localhost



13-3 存取SQLite

- **SQLite** 為檔案型的關聯式資料庫，使用一個檔案當成資料庫，不用安裝資料庫伺服器，就可以練習SQL 語法操作資料庫，因為SQLite 資料庫的資料存取速度不是很好，當所需儲存的資料不多，就可以使用 SQLite 資料庫。
- Python 提供模組sqlite3 實作SQLite 資料庫，以下介紹模組sqlite3 的重要函式。



模組sqlite3 的重要函式

重要函式與說明	程式碼與執行結果
<p><code>sqlite3.connect(file)</code> 連線檔案 file 為 SQLite 資料庫，回傳物件 <code>sqlite3.Connection</code>。</p>	<pre>import sqlite3 con=sqlite3.connect('school.db')</pre>
	建立 sqlite3 資料庫檔案 school.db
<p><code>sqlite3.Connection.cursor()</code> 將物件 <code>sqlite3.Connection</code> 經由函式 <code>cursor</code>，回傳物件 <code>sqlite3.Cursor</code>。</p>	<pre>import sqlite3 con=sqlite3.connect('school.db') cur=con.cursor()</pre>
	產生物件 <code>sqlite3.Cursor</code> 。



模組sqlite3 的重要函式

重要函式與說明	程式碼與執行結果
<p><code>sqlite3.Cursor.execute(sql)</code></p> <p>經由物件 <code>sqlite3.Cursor</code> 執行 SQL 語法管理指定的 SQLite 資料庫。</p>	<pre>import sqlite3 con=sqlite3.connect('school.db') cur=con.cursor() cur.execute('DROP TABLE stu')</pre>
	刪除資料表 <code>stu</code> 。
<p><code>sqlite3.Cursor.fetchall()</code></p> <p>配合 <code>sqlite3.Cursor.execute(sql)</code> 中 <code>sql</code> 語法所選取的資料，使用函式 <code>fetchall</code>，取出所有資料儲存到串列。</p>	<pre>import sqlite3 con=sqlite3.connect('school.db') cur=con.cursor() cur.execute('SELECT * FROM stu') rows=cur.fetchall()</pre>
	從資料表 <code>stu</code> 取出所有資料。




重要函式與說明	程式碼與執行結果
<code>sqlite3.Cursor.close()</code> 關閉指定的物件 Cursor。	<pre>import sqlite3 con=sqlite3.connect('school.db') cur=con.cursor() cur.close()</pre>
	關閉指定的物件 Cursor。



13-3-1 使用模組sqlite3 管理SQLite 資料庫

- (ch13\13-3-1- 使用sqlite3 管理SQLite 資料庫.py)
- 使用模組sqlite3 新增資料庫school，新增資料表stu，用於儲存學生學號、姓名、身份證字號與電話，並練習新增一筆資料到資料表stu。



行號	範例 ( : ch13\13-3-1- 使用 sqlite3 管理 SQLite 資料庫 .py)
1	import sqlite3
2	con=sqlite3.connect('school.db')
3	cur=con.cursor()
4	#cur.execute('DROP TABLE stu')
5	cur.execute("""CREATE TABLE stu
6	(stuid INTEGER PRIMARY KEY,
7	name VARCHAR(50) not null,
8	pid VARCHAR(20) not null,
9	phone VARCHAR(20) not null)""")



```
10 cur.execute("INSERT INTO stu VALUES (104001,'Claire','B342222','1245667')")
11 cur.execute('SELECT * FROM stu')
12 rows=cur.fetchall()
13 print(rows)
14 cur.close()
```





程式解說

- 第1 行：匯入模組sqlite3。
- 第2 行：使用模組sqlite3 的函式connect 連線檔案school.db，將回傳的物件 sqlite3.Connection，指定物件名稱為con。



- 第3 行：使用物件con 的函式cursor 產生物件sqlite3.Cursor，指定給物件cur。
- 第4 行：使用物件cur 的函式execute 執行 SQL 語法「DROP TABLE stu」，刪除資料表stu，第一次執行時，此行程式需加上「#」表示此行不執行，第二次以後執行此程式時，因為檔案school.db 已經有資料表stu，須使用此行程式先刪除資料表stu，否則會出現資料表stu 已經存在的錯誤訊息。



- 第5 到9 行：使用物件cur 的函式execute 執行SQL 語法「CREATE TABLE stu(stuid INTEGER PRIMARY KEY, name VARCHAR(50) not null, pid VARCHAR(20) not null , phone VARCHAR(20) not null)」，建立資料表stu，每列資料有四個欄位，分別是stuid、name、pid 與phone。



- 第10 行：使用物件cur 的函式execute 執行SQL 語法「INSERT INTO stu VALUES (104001,'Claire','B342222','1245667)」，插入一筆資料到資料表stu。
- 第11 行：使用物件cur 的函式execute執行SQL 語法「SELECT * FROM stu」，從資料表stu 取出所有資料。



- 第12 行：使用物件cur 的函式fetchall取出第11 行的執行結果到物件rows。
- 第13 行：使用函式print 顯示物件rows 到螢幕上。
- 第14 行：使用物件cur 的函式close關閉資料庫連線。



13-3-1 使用模組sqlite3 管理SQLite 資料庫

執行結果

程式所在資料夾下多出檔案 school.db。

螢幕出現以下訊息，

```
[(104001, 'Claire', 'B342222', '1245667')]
```

