

# Python MySQL資料庫

蘇柏原(teaching@bo-yuan.net)

# 資料庫概念

# 資料庫概念

- 資料庫，顧名思義就是存放資料的地方
- 在電腦中，它可以透過**資料庫管理系統**(Database Management System, DBMS)來操作其中的資料

# 資料庫概念

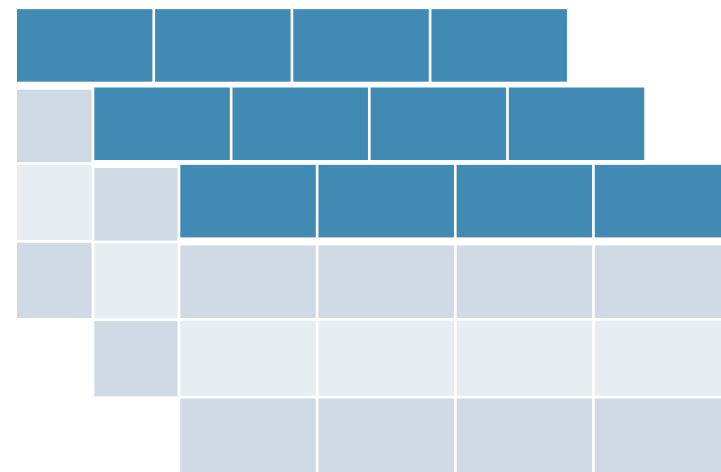
- 常見的資料庫管理系統有：
  - MySQL(大)
  - SQL Server(大)
  - Access(小)
  - SQLite(小)

# 資料庫概念

- 大型資料庫管理系統，通常會將多個資料庫合再一起管理
- 小型資料庫管理系統，則通常將一個檔案作為一個資料庫，分別管理

# 資料庫概念

- 資料庫是由多個像表格一樣的東西構成
- 這些東西稱為資料表



# 資料庫概念

- 每個資料表都有所謂的欄和列
- 其中每一欄代表一種資料
- 在最初的情況下，資料表的列會是 0
- 每新增一筆資料，列就會加 1

姓氏	名子	電話	住址
A	X1	001	
B	X2	002	
C	X3	003	
⋮			
Z	X26	026	

# 資料庫概念

- 有些資料庫的資料欄位有所謂的型態
- 常見的型態有文字跟數字兩種
  - 不同的資料庫會有更多不同的型態



# 資料庫概念

- 通常資料庫可以對欄位建立索引，以加速搜尋：
  - 主鍵(primary key)：一個資料表中只能有一個欄位被賦予當主鍵，作為整個資料表裡的主要索引，裡面的各筆資料皆不可以重複
  - 唯一鍵(unique key)：限制這個欄位的資料不可以重複
  - 一般索引(index)：單純設置索引

# 資料庫規劃

# 資料庫規劃

- 我們一般在電腦上所遇到的資料是非常凌亂的
- 但電腦只能處理有規律性的資料而已
- 所以我們必須學會將凌亂的資料分類和整理

# 資料庫規劃

- 由於資料庫中的資料表特性，所以我們要學會將資料製作成一個有欄位標題的表格

# 資料庫規劃

- 譬如，一個活動行程：

活動時間：

第一梯 101年07月05日至07月08日

第二梯 101年07月22日至07月25日

第三梯 101年08月07日至08月10日

第四梯 101年09月05日至09月07日

第五梯 101年09月15日至09月20日

第六梯 101年10月10日至10月15日

# 資料庫規劃

- 我們可以用一個活動行程資料表格來儲存：

梯次	開始時間	結束時間
1	101-07-05	101-07-08
2	101-07-22	101-07-25
3	101-08-07	101-08-10
4	101-09-05	101-09-07
5	101-09-15	101-09-20
6	101-10-10	101-10-15

# 作業

- 作業：
  - 將Google的網頁搜尋結果製作成表格來呈現。
  - 表格使用Excel製作，完成後繳交檔案。

# 資料庫操作軟體



# SQL語法

# SQL語法

- 控制資料庫的指令叫做SQL
- 他可以對資料表或裡面的資料做新增、刪除、修改和搜尋的操作

# SQL語法

- 新增資料：

insert into 資料表名稱(欄位名1,欄位名2,欄位名3...) values(資料1,資料2,資料3...)



The diagram illustrates the components of the SQL insert statement with color-coded text and arrows:

- 資料表名稱** (Table Name): Green text.
- 欄位名1, 欄位名2, 欄位名3...** (Column Names): Blue text.
- 資料1, 資料2, 資料3...** (Values): Purple text.

Arrows indicate the following relationships:

- A purple arrow points from the **欄位名1, 欄位名2, 欄位名3...** group to the **資料1, 資料2, 資料3...** group.
- A blue arrow points from the **資料1, 資料2, 資料3...** group to the **欄位名1, 欄位名2, 欄位名3...** group.
- A brown arrow points from the **資料1, 資料2, 資料3...** group to the **資料3...** part of the values list.

資料如果為文字需要加上單引號

# SQL語法

- 避免特殊字元的方法：
  - 資料庫名稱、資料表名稱或欄位名稱可以用「`」字元前後框起來，以避免特殊單字影響SQL指令運行。
- 「`」和「'」的使用容易被混淆，判斷上有一個技巧：
  - 凡是從資料庫裡面讀出來的資料用「`」。
  - 凡是從外面要存進去的資料用「'」。

# SQL語法

- 修改資料：

update 資料表名稱 set 欄位名1=資料, 欄位名2=資料...  
where 篩選條件

符合條件的就修改

要修改的欄位，與它要修改成的資料

# SQL語法

- 篩選條件：

判斷說明	判斷寫法	判斷說明	判斷寫法
等於	欄位名=值	在其中	欄位名 in (值1,值2,值3...)
不等於	欄位名<>值	不在其中	欄位名 not in (值1,值2,值3...)
大於	欄位名<值	相似於	欄位名 like '%值%'
小於	欄位名>值	不相似於	欄位名 not like '%值%'
大於等於	欄位名<=值	介於	欄位名 between 值1 and 值2
小於等於	欄位名>=值	不介於	欄位名 not between 值1 and 值2

判斷說明	判斷寫法
判斷是否為NULL	欄位名 is NULL

# SQL語法

- 修改資料(示例) :  
update test set value1='bb',value2='cc',value3='dd'  
where number>5

# SQL語法

- 刪除資料：  
delete from 資料表名稱 where 篩選條件



# SQL語法

- 刪除資料(示例) :  
delete from test where name='aa' and (month>5 or day<7)



符合這個條件的全部都刪除

# SQL語法

- 搜尋資料：  
select 欄位名1, 欄位名2... from 資料表名稱 where 篩選條件

可使用「\*」字號當作搜尋全部欄位

# SQL語法

- 搜尋資料(示例) :  
select value1,value2,value3 from test where value3>5  
select \* from test where value3>5

# SQL語法

- 搜尋的資料**排序**：  
select 欄位名1,欄位名2... from 資料表名稱 where 篩選條件  
**order by** 排序欄位 排序方式

# SQL語法

- 搜尋的資料**排序**(示例)：

select \* from test where number>5 order by value asc

Select \* from test where number>5 order by value desc

由小到大

由大到小

以t2欄位來排序

# SQL語法

- 統計函式：
  - 加總欄位裡的所有值：`sum(欄位名)`
  - 平均欄位裡的所有值：`avg(欄位名)`
  - 統計欄位個數：`count(欄位名)`
  - 取得欄位裡資料的最大值：`max(欄位名)`
  - 取得欄位裡資料的最小值：`min(欄位名)`

# SQL語法

- 函式使用：  
select **sum**(number) from test  
select **avg**(number) from test  
select **count**(number) from test  
select **max**(number) from test  
select **min**(number) from test

# SQL語法

- 別名：
  - 在下SQL指令時，可以對欄位取一個別名
    - select 欄位名 as 別名, 欄位名 as 別名... from 資料表名稱
- 別名的使用時機：
  - 使用函式時：  
select sum(欄位名) as 別名 from 資料表名稱



# SQL語法

- 取得搜尋結果中的指定幾筆資料：

編號	時間	開始	結束	課程
1	2014-01-05	09:00	10:00	1
2	2014-01-05	10:00	11:00	1
3	2014-01-05	11:00	12:00	2
4	2014-01-06	09:00	10:00	3
5	2014-01-06	10:00	11:00	3
6	2014-01-06	11:00	12:00	4
7	2014-01-07	09:00	10:00	5

在select搜尋指令的  
結尾加上 limit 2,4



# PyMySQL

# MySQLdb

- PyMySQL是Python中用於跟MySQL連結的函式庫
- 安裝方式：  
`pip install pymysql`

# MySQLdb

- 連結MySQL :  
連結變數=pymysql.connect(  
    host=主機名稱,  
    user=帳號,  
    passwd=密碼,  
    db=資料庫名稱,  
    charset=編碼  
)
- 取得指令操作變數 :  
指令操作變數=連結變數.cursor()
- 關閉MySQL連線 :  
連結變數.close()

# MySQLdb

如果要存資料進資料庫才要帶入這參數，為串列或字典格式，SQL中需設定對應的%s來帶入資料

- 傳送SQL指令：

指令操作變數.execute(SQL指令, 要帶入SQL中的變數)

- 如果是新增、刪除或修改指令需再執行：

連結變數.commit()

- 如果是查詢指令需再執行：

指令操作變數.fetchall()

tuple格式，每一個值代表一筆資料

指令操作變數.fetchone()

只取結果的第一筆資料

指令操作變數.rowcount

總共的資料筆數

# MySQLdb

- 取得最後新增的資料，其所自動產生的流水號ID：  
指令操作變數.lastrowid

# 作業

- 作業：
  - 用Python搭配MySQL製作一個會員管理系統，須具備新增、刪除和修改會員的功能。
  - 會員的資料欄位至少需要包含姓名、生日、地址等三個欄位。
  - 繳交程式碼檔案。

# SQL語法

多資料表聯合搜尋與處理



# SQL語法

多資料表聯合搜尋與處理

- 一個存放課表資料的資料表，假設各課程還有細項資料要儲存時：

編號	時間	開始	結束	課程
1	2014-01-05	09:00	10:00	國文
2	2014-01-05	10:00	11:00	國文
3	2014-01-05	11:00	12:00	英文
4	2014-01-06	09:00	10:00	數學
5	2014-01-06	10:00	11:00	數學

# SQL語法

多資料表聯合搜尋與處理

- 如果我們把課程細項都放在同一個資料表裡儲存，那麼會發生：

編號	時間	開始	結束	課程	教室	授課教師
1	2014-01-05	09:00	10:00	國文	TE-110	王大維
2	2014-01-05	10:00	11:00	國文	TE-110	王大維
3	2014-01-05	11:00	12:00	英文	TE-112	李大人
4	2014-01-06	09:00	10:00	數學	TE-113	陳大王
5	2014-01-06	10:00	11:00	數學	TE-113	陳大王

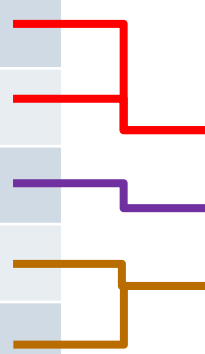
重複資料過多，佔用太多空間

# SQL語法

多資料表聯合搜尋與處理

- 資料表關聯：

編號	時間	開始	結束	課程	編號	名稱	教室	授課教師
1	2014-01-05	09:00	10:00	1	1	國文	TE-110	王大維
2	2014-01-05	10:00	11:00	1	2	英文	TE-112	李大人
3	2014-01-05	11:00	12:00	2	3	數學	TE-113	陳大王
4	2014-01-06	09:00	10:00	3				
5	2014-01-06	10:00	11:00	3				



# SQL語法

多資料表聯合搜尋與處理

- select \* from A inner join B on A.欄位名=B.欄位名

A和B分別為兩個資料表

這裡有個點

# SQL語法

- 別名：
  - 在下SQL指令時，也可以對資料表取一個別名
    - `select * from 資料表名稱 as 別名`
- 別名的使用時機：
  - 使用關聯時：  
`select * from A as 別名1 inner join B as 別名2  
on 別名1.欄位名 = 別名2.欄位名`

# SQL語法

多資料表聯合搜尋與處理

- `join inner` 可以從兩個資料表中的指定欄位找尋對應的值，並把他列出來，所以下面的範例會有兩筆資料沒被列出來：

編號	時間	開始	結束	課程
1	2014-01-05	09:00	10:00	1
2	2014-01-05	10:00	11:00	1
3	2014-01-05	11:00	12:00	2
4	2014-01-06	09:00	10:00	3
5	2014-01-06	10:00	11:00	3
6	2014-01-06	11:00	12:00	4
7	2014-01-07	09:00	10:00	5

編號	名稱	教室	授課教師
1	國文	TE-110	王大維
2	英文	TE-112	李大人
3	數學	TE-113	陳大王

找不到課程編號為4和5的課程

# SQL語法

多資料表聯合搜尋與處理

- 如果我們想把其中一個資料表欄位沒對應到的資料也列出來，可以使用：

- select \* from A left join B on A.欄位名=B.欄位名



顯示所有左邊資料表(A資料表)的內容，不論欄位是否有對應

- select \* from A right join B on A.欄位名=B.欄位名



顯示所有右邊資料表(B資料表)的內容，不論欄位是否有對應

# SQL語法

多資料表聯合搜尋與處理

- 如果想確保關聯資料的一致性，可以使用外來鍵功能：
  - 它可以將兩個資料表的指定欄位連結在一起。
  - 兩邊的欄位皆必須設為索引。
  - MySQL的InnoDB引擎才支援。其他的DB類型不行，ex: myISAM
  - 如果資料表中已有資料，則兩個欄位的內容必須要能互相對應。



# SQL語法

多資料表聯合搜尋與處理

- CASCADE
  - 會將有所關聯的紀錄行也會進行刪除或修改(被設定的一方會被牽動)。
- SET NULL
  - 會將有所關聯的紀錄行設定成 NULL。
- NO ACTION
  - 有存在的關聯紀錄行時，會禁止父資料表的刪除或修改動作。
- RESTRICT
  - 與 NO ACTION 相同。

# SQL語法

- SQL中可以使用運算的地方：

- 搜尋時：

select 欄位名+5, 欄位名\*2, 欄位名... from 資料表名

搜尋後顯示出來的結果將會是欄位資料運算後的結果

- 更新時：

update 資料表名 set 欄位名=資料+1, 欄位名=資料...

會先做完運算後才將資料更新進資料表中

# SQL語法

- SQL中可以使用運算的地方：

- 新增時：

insert into 資料表名(欄位名,欄位名,欄位名...) values(資料+1,資料\*2,資料...)

會先做完運算後才將資料新增至資料表中


- 所有的篩選條件：

where 欄位名\*2=資料 and 欄位名>資料+1

會以欄位運算後的結果來跟資料做比對

會以資料運算後的結果來跟欄位做比對

# SQL語法

- SQL可以使用字串串接的地方：
  - 搜尋時：  
select concat(欄位名,欄位名,欄位名...),欄位名... from 資料表名  
  
搜尋後顯示出來的結果將會是欄位資料串接後的結果
  - 更新時：  
update 資料表名 set 欄位名=concat(資料,資料,資料...),欄位名=資料...  
  
會先將資料串接後才更新進資料表中

# SQL語法

- SQL可以使用字串串接的地方：
  - 新增時：  
insert into 資料表名(欄位名,欄位名...)  
values(concat(資料,資料,資料...),資料...)

會先將資料串接後才新增至資料表中

- 所有的篩選條件：

會以欄位串接後的結果來跟資料做比對

- **where** concat(欄位名,欄位名,欄位名...)=資料  
and 欄位名=concat(資料,資料,資料...)

會以資料串接後的結果來跟欄位做比對

# 作業

- 作業：
  - 為上一個作業擴充，建置每個會員的電話清單，每個會員可以有 multiple 電話資料，電話資料要可以查看、新增跟刪除，需使用到「多資料表聯合搜尋」功能。
  - 繳交程式碼檔案。