



ST LECTUREs
小 孫 學 堂

Big Data MySQL關聯式資料庫

Created by 孫善堂 【小孫學堂】

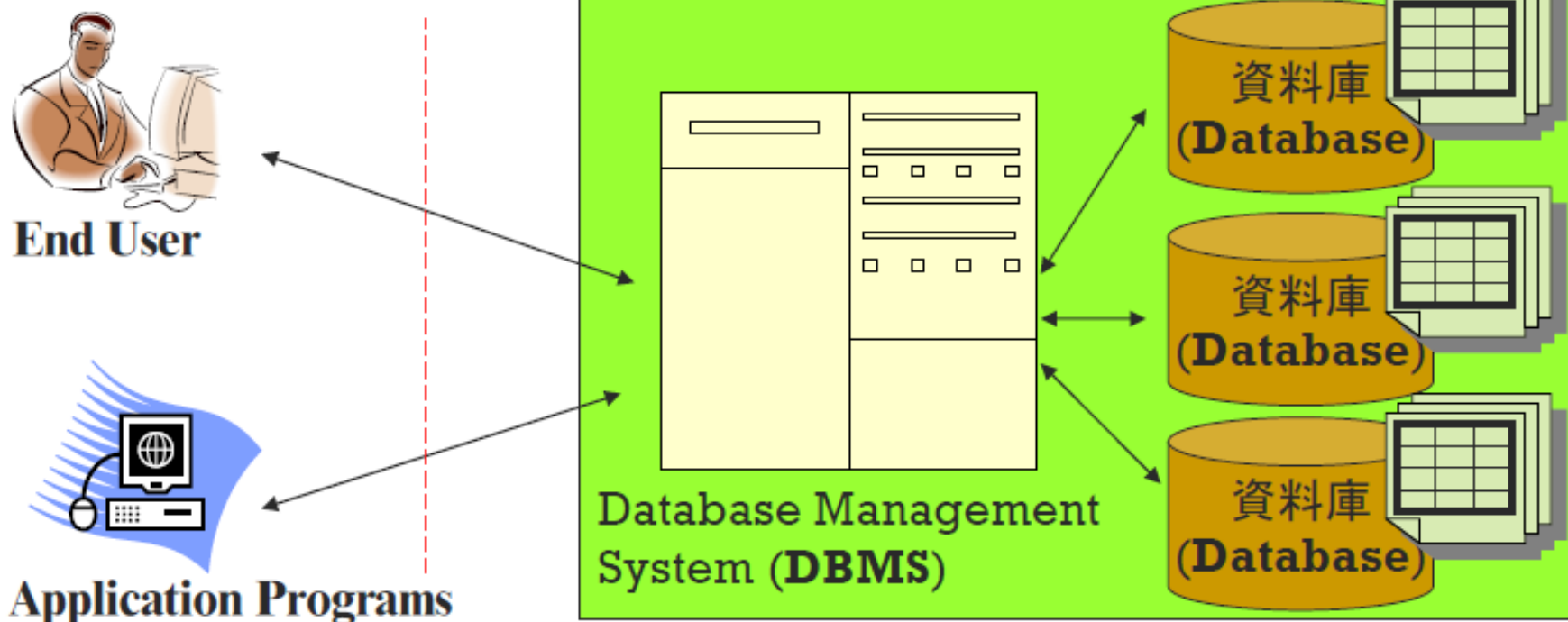
MySQL資料庫系統簡介

- MySQL資料庫系統是關聯式資料庫管理系統
- (relational database management system) RDBMS。
- 使用價格最便宜，甚至免費。在目前架設商業網站中，PHP配上MySQL資料庫是使用最為廣範的。
- 目前已被Oracle公司收購。
- (因擔心MySQL有朝一日走上收費之路，原作者另行開發一套資料庫系統軟體「MariaDB」，取名自作者女兒。)
- 新版MariaDB與NoSQL有良好的結合。

MySQL資料庫系統簡介

- 欲存放的資料在資料庫系統中組成**資料表格(Table，或稱資料表)**，再由數個相關的資料表格組成**資料庫**，透過資料庫管理系統來作管理。
- 下圖顯示了資料庫的階層。

- **SQL (結構化查詢語言)：**
User與AP皆透過SQL與DBMS溝通、存取DB data!



MySQL資料層級

MySQL 資料庫管理系統

DataBase資料庫

Table資料表

Table資料表

Table資料表

DataBase資料庫

Table資料表

Table資料表

Table資料表

DataBase資料庫

Table資料表

Table資料表

Table資料表

常見網頁管理架構

LOGO	定位	功能
	網頁伺服器軟體	將樹梅派從個人電腦變成網頁伺服器
	資料庫(=MySQL)	讓空間中的資料可以結構化的儲存
	資料庫管理工具	提供瀏覽器介面讓大家可以簡單操作，不用都要用終端機輸入指令操作，比較方便
	網站內容架設工具	架設一個網站，管理網站裡面的內容

phpMyAdmin概念簡介

- phpMyAdmin是用php所開發的一個免費的MySQL圖形化網頁管理介面。它直覺式的操作界面，設定方便，提供中文介面、SQL輔助編輯工具，是一個入門的好工具。
- phpMyAdmin是透過瀏覽器 來管理MySQL，它並不是MySQL的一部份，我們只是透過它來操作MySQL。
- phpMyAdmin是一套用PHP寫出來的程式，用來管理MySQL。也就是說，要有支援PHP 的網站伺服器才能用phpMyAdmin。
- 不管是哪一部電腦，只要經由網路，利用瀏覽器就可以操作phpMyAdmin來管理MySQL，相當方便！

啟動MySQL指令

定位	功能
<code>sudo mysql</code>	以linux管理員權限開啟MySQL
<code>Sudo mysql -u root -p</code>	以root權限操作MySQL (密碼預設為無密碼)
<code>Sudo mysql -u pi -p</code>	以使用者名稱pi操作MySQL， 並輸入密碼

MySQL常用指令表

顯示所有資料庫： `show databases;`

選擇某一資料庫： `use` 資料庫名稱;

列出資料庫底下的資料表： `show tables from` 資料庫名稱;

顯示目前正在使用的資料表： `show tables;`

顯示資料表欄位設定： `describe` 資料表名稱;

重新整理資料表： `optimize table` 資料表名稱;

顯示某個資料表的欄位設定： `show columns from` 資料表名稱;

顯示某個資料表的索引欄位設定： `show index from` 資料表名稱;

MySQL基本指令

資料定義語言 (DDL) :

建立資料庫或資料表 : `create`

變更資料庫或資料結構 : `alter`

刪除資料庫或資料表 : `drop`

MySQL基本指令

資料操作語言 (DML) :

查詢資料 : `select`

插入資料 : `insert`

更新資料 : `update`

刪除資料 : `delete`

MySQL基本指令

資料控制語言 (DCL):

確定資料的變更: `commit`

設定使用者對資料的操作權限: `grant`

取消使用者操作資料的權限: `revoke`

取消資料變更: `rollback`

MySQL指令演練

建立資料表：

create table 資料表名稱(欄位名稱1 資料型態,欄位名稱2 資料型態,...);

插入一列資料：

Insert into 資料表名稱 values(value1,value2,...);

插入多列資料：

**INSERT INTO table(column1,column2...) VALUES (value1,value2,...),
(value1,value2,...),...;**

MySQL資料類型

类型	大小	范围（有符号）	范围（无符号）	用途
TINYINT	1 byte	(-128 , 127)	(0 , 255)	小整数值
SMALLINT	2 bytes	(-32 768 , 32 767)	(0 , 65 535)	大整数值
MEDIUMINT	3 bytes	(-8 388 608 , 8 388 607)	(0 , 16 777 215)	大整数值
INT或 INTEGER	4 bytes	(-2 147 483 648 , 2 147 483 647)	(0 , 4 294 967 295)	大整数值
BIGINT	8 bytes			极大整数值
FLOAT	4 bytes			单精度 浮点数值
DOUBLE	8 bytes			双精度 浮点数值
DECIMAL				小数值

MySQL資料類型

类型	大小 (bytes)	范围	格式	用途
DATE	3	1000-01-01/9999-12-31	YYYY-MM-DD	日期值
TIME	3	'-838:59:59'/'838:59:59'	HH:MM:SS	时间值或持续时间
YEAR	1	1901/2155	YYYY	年份值
DATETIME	8	1000-01-01 00:00:00/9999-12-31 23:59:59	YYYY-MM-DD HH:MM:SS	混合日期和时间值
TIMESTAMP	4	1970-01-01 00:00:00/2038 结束时间是第 2147483647 秒，北京时间 2038-1-19 11:14:07 ，格林尼治时间 2038年1月19日 凌晨 03:14:07	YYYYMMDD HHMMSS	混合日期和时间值，时间戳

MySQL資料類型

类型	大小	用途
CHAR	0-255 bytes	定长字符串
VARCHAR	0-65535 bytes	变长字符串
TINYBLOB	0-255 bytes	不超过 255 个字符的二进制字符串
TINYTEXT	0-255 bytes	短文本字符串
BLOB	0-65 535 bytes	二进制形式的长文本数据
TEXT	0-65 535 bytes	长文本数据
MEDIUMBLOB	0-16 777 215 bytes	二进制形式的中等长度文本数据
MEDIUMTEXT	0-16 777 215 bytes	中等长度文本数据
LOBLOB	0-4 294 967 295 bytes	二进制形式的极大文本数据
LOBTEXT	0-4 294 967 295 bytes	极大文本数据

MySQL範例資料表架構

Student
Sid
Sname
Sage
Ssex



SC
Sid
Cid
score



Course
Cid
Cname
Tid



Teacher
Tid
Tname

MySQL指令演練

Select * from 資料表名稱;

查詢資料表中的所有欄位

```
MariaDB [score]> select * from SC;
```

S	C	score
01	01	80.0
01	02	90.0
01	03	99.0
02	01	70.0
02	02	60.0
02	03	80.0
03	01	80.0
03	02	80.0
03	03	80.0
04	01	50.0
04	02	30.0
04	03	20.0
05	01	76.0
05	02	87.0
06	01	31.0
06	03	34.0
07	02	89.0
07	03	98.0

18 rows in set (0.01 sec)

MySQL指令演練

Select 欄位名稱 from 資料表名稱;

查詢資料表中的特定欄位

```
MariaDB [score]> select T from Teacher;  
+-----+  
| T      |  
+-----+  
| 01     |  
| 02     |  
| 03     |  
+-----+  
3 rows in set (0.00 sec)
```

MySQL指令演練

Where 判斷式;

查詢符合特定條件的欄位

查詢2號同學的所有科目成績：

```
MariaDB [score]> select * from SC where S = '02';
```

S	C	score
02	01	70.0
02	02	60.0
02	03	80.0

```
3 rows in set (0.00 sec)
```

```
MariaDB [score]> 
```

MySQL指令演練

GROUP BY

根据一个或多个列对结果集进行分组。

在分组的列上我们可以使用 COUNT, SUM, AVG, 等函数。

GROUP BY 语法

```
SELECT column_name, function(column_name)
FROM table_name WHERE column_name operator value
GROUP BY column_name;
```


MySQL指令演練

跨工作表查詢欄位

查詢2號同學姓名的所有科目成績及老師名稱：

```
MariaDB [score]> select Sname, Cname, Tname, score from Student, Course, Teacher,  
SC where Student.Sid=SC.Sid and SC.Cid=Course.Cid and Course.Tid=Teacher.Tid an  
d SC.Sid='02' group by SC.Cid;
```

Sname	Cname	Tname	score
錢電	語文	李四	70.0
錢電	數學	張三	60.0
錢電	英語	王五	80.0

3 rows in set (0.00 sec)

MySQL指令演練

計算並列出所有同學的考試次數：

```
+-----+-----+-----+
| Sid  | Sname | COUNT(score) |
+-----+-----+-----+
| 01   | 趙雷  | 3             |
| 02   | 錢電  | 3             |
| 03   | 孫風  | 3             |
| 04   | 李雲  | 3             |
| 05   | 周梅  | 2             |
| 06   | 吳蘭  | 2             |
| 07   | 鄭竹  | 2             |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

MySQL指令演練

計算並列出所有同學的考試次數：

```
select SC.Sid, ST.Sname,  
COUNT(score)  
from SC, Student as ST  
where SC.Sid=ST.Sid  
group by SC.Sid;
```


MySQL指令演練

Having 判斷式;

針對group by的函數進行判斷

取考試次數超過2的同學：

```
MariaDB [score]> select SC.Sid, ST.Sname, COUNT(score)
-> from SC, Student as ST
-> where SC.Sid=ST.Sid
-> group by SC.Sid
-> having COUNT(score)>2;
```

Sid	Sname	COUNT(score)
01	趙雷	3
02	錢電	3
03	孫風	3
04	李雲	3

4 rows in set (0.00 sec)

MySQL指令演練：排序

```
SELECT field1, field2,...fieldN FROM table_name1,  
table_name2...  
ORDER BY field1 ASC, field2... ASC
```

- 你可以使用任何字段来作为排序的条件，从而返回排序后的查询结果。
- 你可以设定多个字段来排序。
- 你可以使用 **ASC** 或 **DESC** 关键字来设置查询结果是按升序或降序排列。默认情况下，它是按升序排列。
- 你可以添加 **WHERE...LIKE** 子句来设置条件。

MySQL指令演練

```
MariaDB [score]> select * from SC  
-> order by score;
```

Sid	Cid	score
04	03	20.0
04	02	30.0
06	01	31.0
06	03	34.0
04	01	50.0
02	02	60.0
02	01	70.0
05	01	76.0
01	01	80.0
03	03	80.0
03	02	80.0
03	01	80.0

04	01	50.0
02	02	60.0
02	01	70.0
05	01	76.0
01	01	80.0
03	03	80.0
03	02	80.0
03	01	80.0
02	03	80.0
05	02	87.0
07	02	89.0
01	02	90.0
07	03	98.0
01	03	99.0

18 rows in set (0.00 sec)

MySQL指令演練

Update 資料表 set 欄位名='資料內容'; 更新特定欄位資料

將第一個老師名稱改成小孫：

```
MariaDB [score]> update Teacher set Tname = '小孫' where T = '01';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [score]> select * from Teacher;
+-----+-----+
| T      | Tname  |
+-----+-----+
| 01     | 小孫   |
| 02     | 李四   |
| 03     | 王五   |
+-----+-----+
3 rows in set (0.01 sec)
```

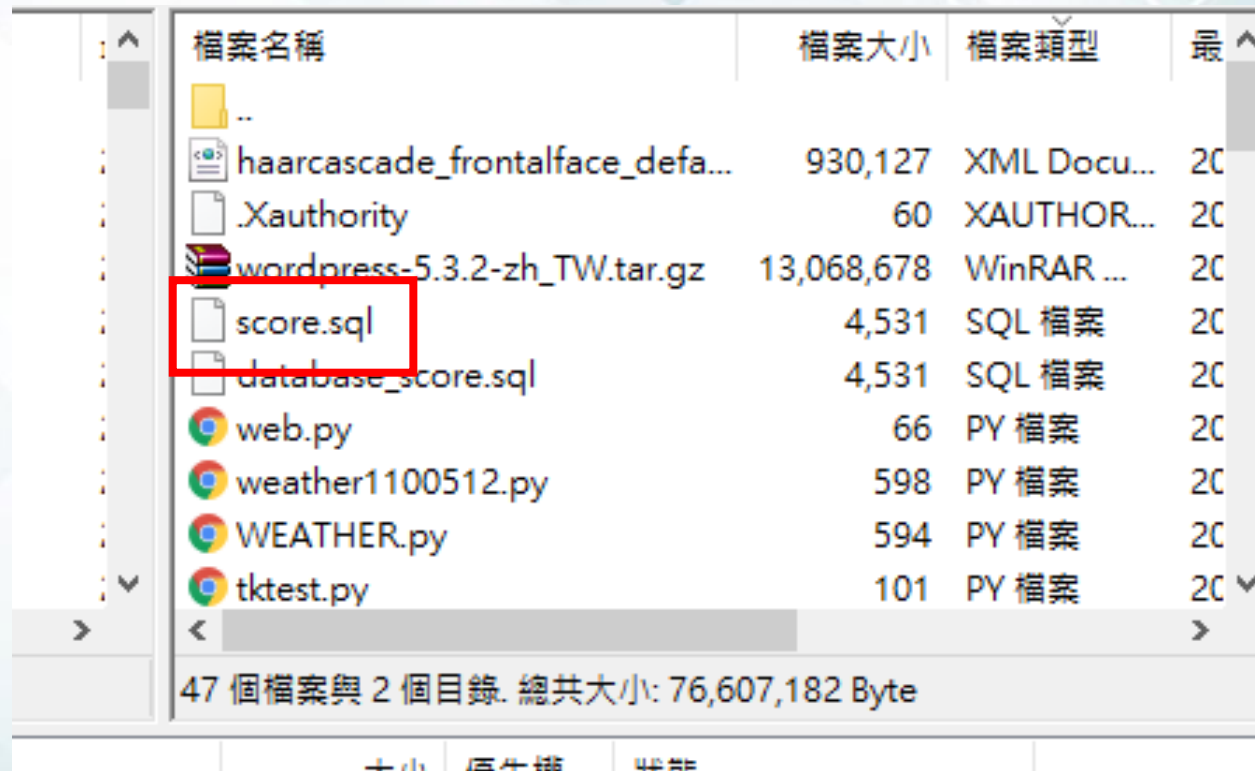

MySQL指令演練

`mysqldump -u root -p 資料庫名稱 > dbname.sql`

匯出資料庫

`mysql -p 資料庫名稱 < dbname.sql`

匯入資料庫



MySQL經典範例題庫

查詢平均成績大於等於 60 分的同學的學生編號
和學生姓名和平均成績

```
+-----+-----+-----+
| sid  | sname | avg_score |
+-----+-----+-----+
| 01   | 趙雷  | 89.66667  |
| 02   | 錢電  | 70.00000  |
| 03   | 孫風  | 80.00000  |
| 05   | 周梅  | 81.50000  |
| 07   | 鄭竹  | 93.50000  |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

MySQL經典範例題庫

查詢平均成績大於等於 60 分的同學的學生編號
和學生姓名和平均成績

```
select s.sid, sname, avg(score) as avg_score  
from Student as s, SC  
where s.sid = SC.sid  
group by s.sid  
having avg_score > 60;
```


MySQL經典範例題庫*50

<https://www.itread01.com/content/1545134226.html>

安裝MySQL範例資料庫

```
Sudo apt-get install git
```

```
git clone https://github.com/datacharmer/test\_db
```

```
cd test_db
```

```
mysql -p < employees.sql
```

安裝MySQL範例資料庫

```
pi@raspberrypi-ST: ~  
pi@raspberrypi-ST:~/test_db $ mysql -p < employees.sql  
Enter password:  
INFO  
CREATING DATABASE STRUCTURE  
INFO  
storage engine: InnoDB  
INFO  
LOADING departments  
INFO  
LOADING employees  
  
INFO  
LOADING dept_emp  
INFO  
LOADING dept_manager  
INFO  
LOADING titles  
INFO  
LOADING salaries  
data_load_time_diff  
NULL  
pi@raspberrypi-ST:~/test_db $
```

```
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| 123      |  
| employees|  
| information_schema |  
| mysql    |  
| performance_schema |  
| phpmyadmin|  
| score    |  
| wordpress|  
| worldpress|  
+-----+  
9 rows in set (0.01 sec)
```



PYMYSQL PYTHON+MYSQL

Created by 孫善堂 【小孫學堂】

接受遠端主機連線要求

/etc/mysql/mariadb.conf.d/50-server.cnf

```
#
user                = mysql
pid-file            = /var/run/mysqld/mysqld.pid
socket              = /var/run/mysqld/mysqld.sock
port                = 3306
basedir             = /usr
datadir             = /var/lib/mysql
tmpdir              = /tmp
lc-messages-dir     = /usr/share/mysql
skip-external-locking

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address        = 0.0.0.0

#
# * Fine Tuning
#
key_buffer_size     = 16M
max_allowed_packet  = 16M
thread_stack        = 192K
thread_cache_size   = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover       = BACKUP
```


開啟遠端使用者並授權

```
shell> mysql --user=root -p
```

輸入密碼

```
mysql> use mysql
```

```
mysql> GRANT ALL ON OVS.* TO 'wayne21205'@'140.115.%.%' identified  
by 'xxxxxx';
```

新增一個wayne21205使用者，密碼為xxxxxx，他可以存取OVS資料庫中的所有Table，

且限制IP為140.115開頭的IP Address

連接MySQL資料庫

```
import pymysql

setting={
    'host': '192.168.43.140',
    'port': 3306,
    'user': 'pc',
    'password': 'raspberry',
    'db': 'score',
    'charset': 'utf8'
}

db=pymysql.connect(**setting)
```

目標主機IP

轉譯 MySQL 資料 to Python

```
# prepare a cursor object using cursor() method
cursor = db.cursor()

# execute SQL query using execute() method.
cursor.execute("SELECT * from Student")

# Fetch a single row using fetchone() method.
data = cursor.fetchall()

for stu in data:
    print (stu)
```

```
In [21]: cursor.execute("insert into Student values('09',N'小孫','1991-05-23',N'男');")
```

```
Out[21]: 1
```