"歡迎來到有趣的資料庫世界! 企業 ERP 系統、電子商務網站，幾乎所有大型資訊系統都會用到資料庫!!"

輕鬆學會使用資料庫 Java 程式設計

簡介

SQL 全名是結構化查詢語言（Structured Query Language），是用於資料庫中的標準數據查詢語言，IBM 公司最早使用在其開發的資料庫系統中。

資料庫(Data Base)將資料有秩序地放到檔案中，讓我們透過 SQL 指令，對資料進行資刪修改。

資料庫管理系統是資管系二或三年級最重要的專業課程之一。

資管系的學生在學生生涯、職場生涯幾乎都會間接、直接用到資料庫。

*資料庫基本名詞

資料庫以資料表構成，每張資料表則由許多筆記錄所組成，每筆記錄又以許多欄位組合而成，每個欄位則存放著一筆資料。

關聯式資料庫，是由資料表、紀錄、欄位以及資料所構成的。在關聯式資料庫中，有主鍵、次要鍵、外來鍵、超級關聯鍵等不同鍵值，提供資料庫索引或識別。

- Table 資料表

資料表顧名思義，就是存放資料的表單，是由多筆記錄匯集而成。每個資料表皆具有一個表單名稱，在同一個資料庫的資料表，它的表單名稱皆不得重

複。資料表的結構概念類似一般表格，具有行與列的特性，通常資料列的欄位數是固定的，並依照資料記錄多寡，而有不固定的行高。

在關聯式資料庫的定義中，每個資料表分別代表一個實體，例如員工這個實體，則可以將它轉換成員工資料表。

- Record 記錄

記錄是由一群有關聯性的欄位所集合而成。每一筆記錄，在資料表中代表著一列欄位，這些欄位存放的資料，在彼此之間都有一定的關聯。在關聯式資料庫中，每張資料表的資料列，則可視為該實體的屬性。例如員工資料表內，會記錄每位員工的資料，這些資料就是用來描述每位員工的屬性。而這些資料彼此間的關聯，在於每筆記錄的資料，皆具有一定的共通性。

- Field 欄位

資料庫中的欄位，就是存放資料的空間，類似微軟 Excel 中的儲存格。每一個欄位空間只能存放一筆資料。在設計資料表時，要先定義每個欄位的資料長度與型別為何，例如文字、數字、日期或是布林值。

部分的關聯式資料庫則將 Field 界定為與 Column 同義，因此欄位也可定義為資料表的縱向資料，而不同於上述的單一資料儲存格。

- Primary Key 主鍵

是用來識別資料表的唯一值。每個主鍵在資料表中，都是獨一無二的。資料庫管理系統可以藉由這些欄位，識別資料表內的每一筆記錄，並提供資料索引。主鍵可以直接使用一組不重複的資料，或是由系統自動產生，像員工資料表的員工編號，是用系統自動產生的流水號。由於主鍵是提供資料庫索引的重要欄位，故設計資料表時要慎選主鍵，避免造成資料庫系統產生資料錯誤等嚴重問題。

- Foreign Key 外來鍵 (外部鍵)

這個欄位會存放其他資料表的主鍵，主要用來確定資料的參考完整性，只有經過確認的資料才能輸入，避免資料在建立時，因為其他資料不完整而導致資料完整性有缺陷。外來鍵的資料來源也可以是自己本身的主鍵，例如員工資料表裡面的主管編號，這就是一個外來鍵，裡面的資料就是參考本身的員工編號。當輸入員工主管的編號時，會去尋找該主鍵是否存在，確保資料完整性。

資料來源：http://www.ithome.com.tw/node/46156

Lab0: 啟動與停止 資料庫伺服器 Server

*如何將資料庫服務開啟?

以系統管理員身分執行命令列

- 啟動 MariaDB 資料庫服務

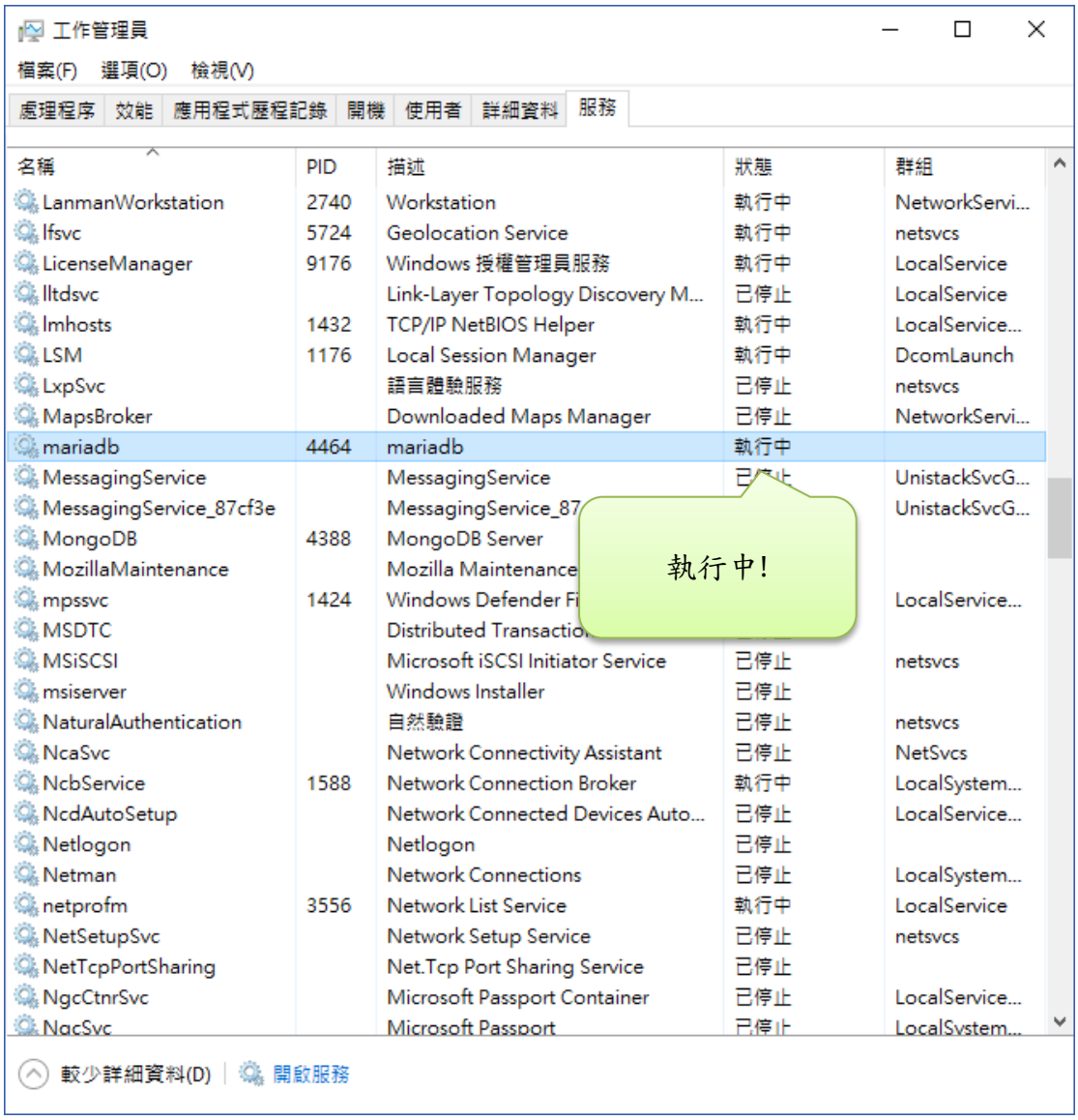
net start mysql

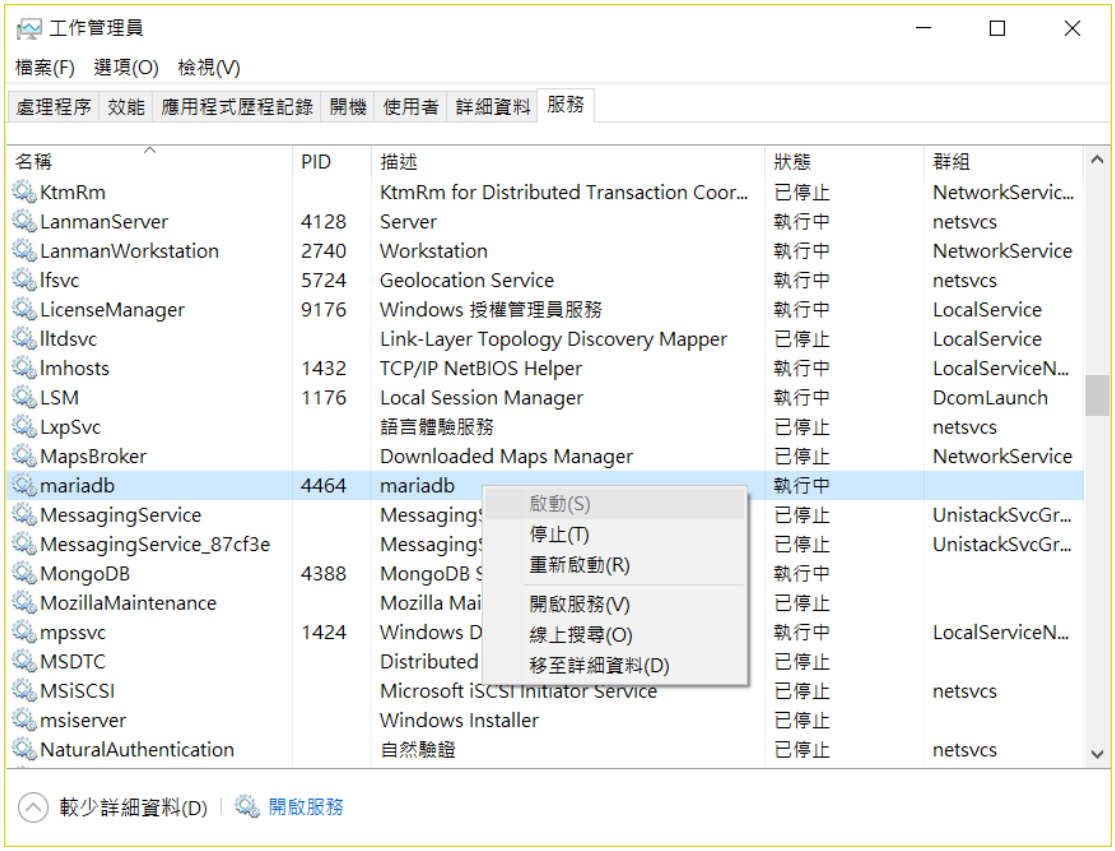
- 停止 MariaDB 資料庫服務 當你長期不用時，建議關閉資料庫服務!

net stop mysql

*注意:作業系統是否有將 mariadb 服務啟動，若無，則你須將其啟動。

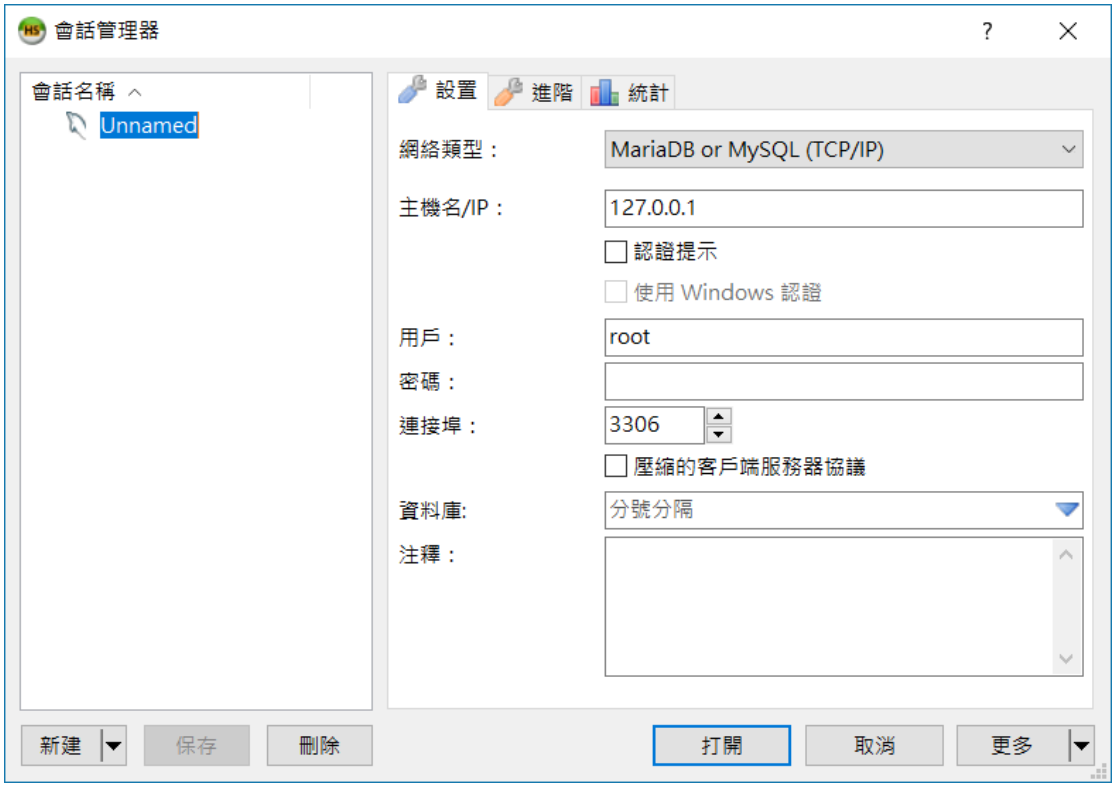
啟動方式如下:

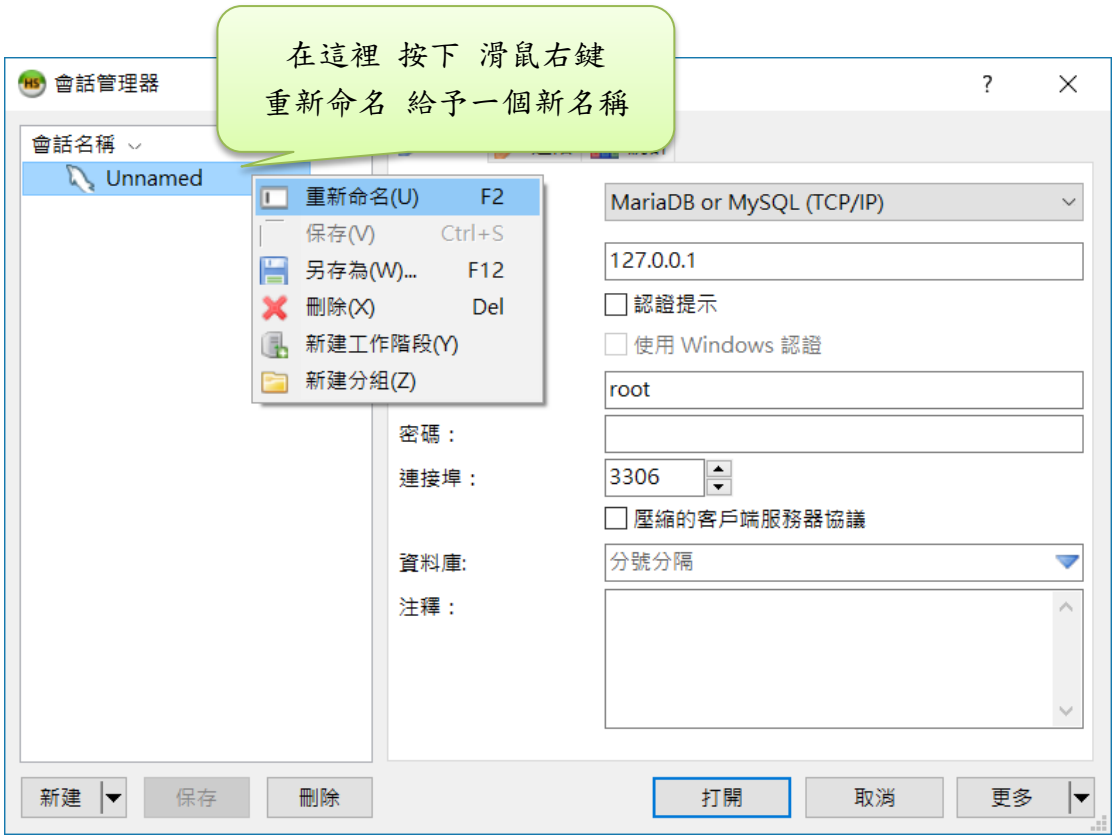

工作管理員
檔案(F) 選項(O) 檢視(V)
處理程序 效能 應用程式歷程記錄 開機 使用者 詳細資料 服務
名稱 PID 描述 狀態 群組
LanmanWorkstation 2740 Workstation 執行中 NetworkServi...
lfsvc 5724 Geolocation Service 執行中 netsvcs
LicenseManager 9176 Windows 授權管理員服務 執行中 LocalService
lltdsvc Link-Layer Topology Discovery M... 已停止 LocalService
lmhosts 1432 TCP/IP NetBIOS Helper 執行中 LocalService...
LSM 1176 Local Session Manager 執行中 DcomLaunch
LxpSvc 語言體驗服務 已停止 netsvcs
MapsBroker Downloaded Maps Manager 已停止 NetworkServi...
mariadb 4464 mariadb 執行中
MessagingService MessagingService 已停止 UnistackSvcG...
MessagingService_87cf3e MessagingService_87 UnistackSvcG...
MongoDB 4388 MongoDB Server
MozillaMaintenance Mozilla Maintenance
mpssvc 1424 Windows Defender Fi LocalService...
MSDTC Distributed Transactio
MSiSCSI Microsoft iSCSI Initiator Service 已停止 netsvcs
msiserver Windows Installer 已停止
NaturalAuthentication 自然驗證 已停止 netsvcs
NcaSvc Network Connectivity Assistant 已停止 NetSvcs
NcbService 1588 Network Connection Broker 執行中 LocalSystem...
NcdAutoSetup Network Connected Devices Auto... 已停止 LocalService...
Netlogon Netlogon 已停止
Netman Network Connections 已停止 LocalSystem...
netprofm 3556 Network List Service 執行中 LocalService
NetSetupSvc Network Setup Service 已停止 netsvcs
NetTcpPortSharing Net.Tcp Port Sharing Service 已停止
NgcCtnrSvc Microsoft Passport Container 已停止 LocalService...
NgcSvc Microsoft Passport 已停止 LocalSystem...
執行中!
較少詳細資料(D)
開啟服務



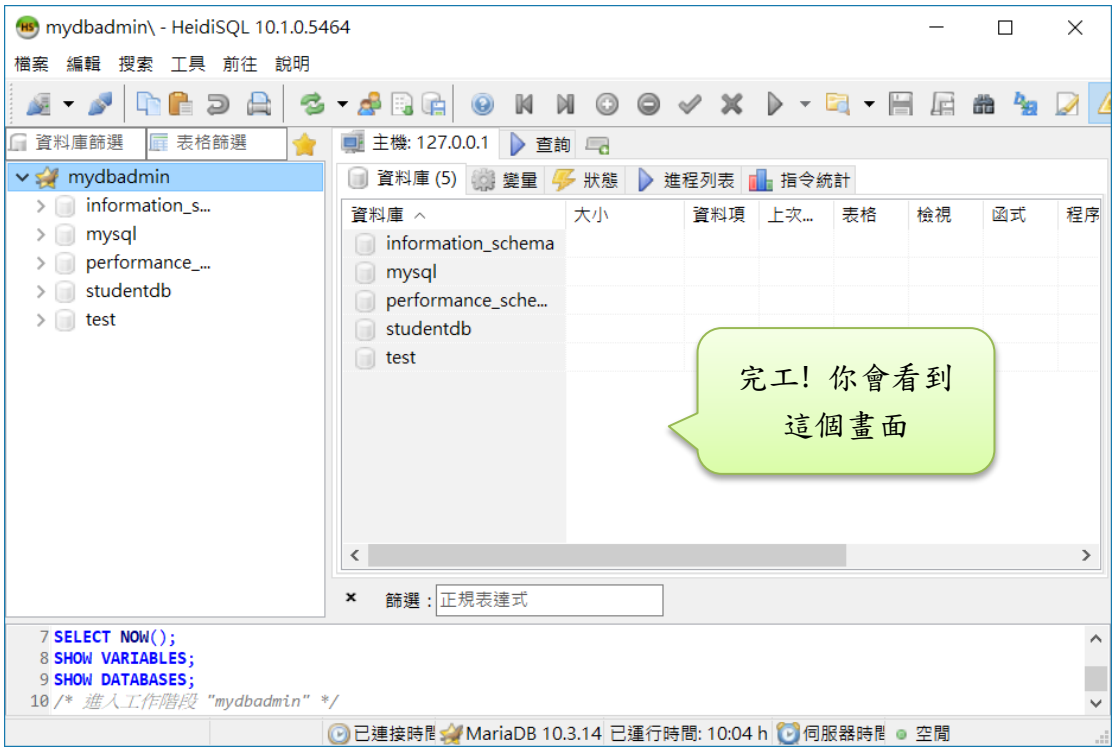
Lab0.1:新增一個資料庫連線管理


會話管理器
會話名稱
啟動
New here? In order to connect to a server, you have to create a so called "session" at first. Just click the "New" button on the bottom left to create your first session.Give it a friendly name (e.g. "Local DB server") so you'll recall it the next time you start HeidiSQL.
匯入設定檔案...
按下 新建 按鈕
新建
保存
刪除
打開
取消
更多

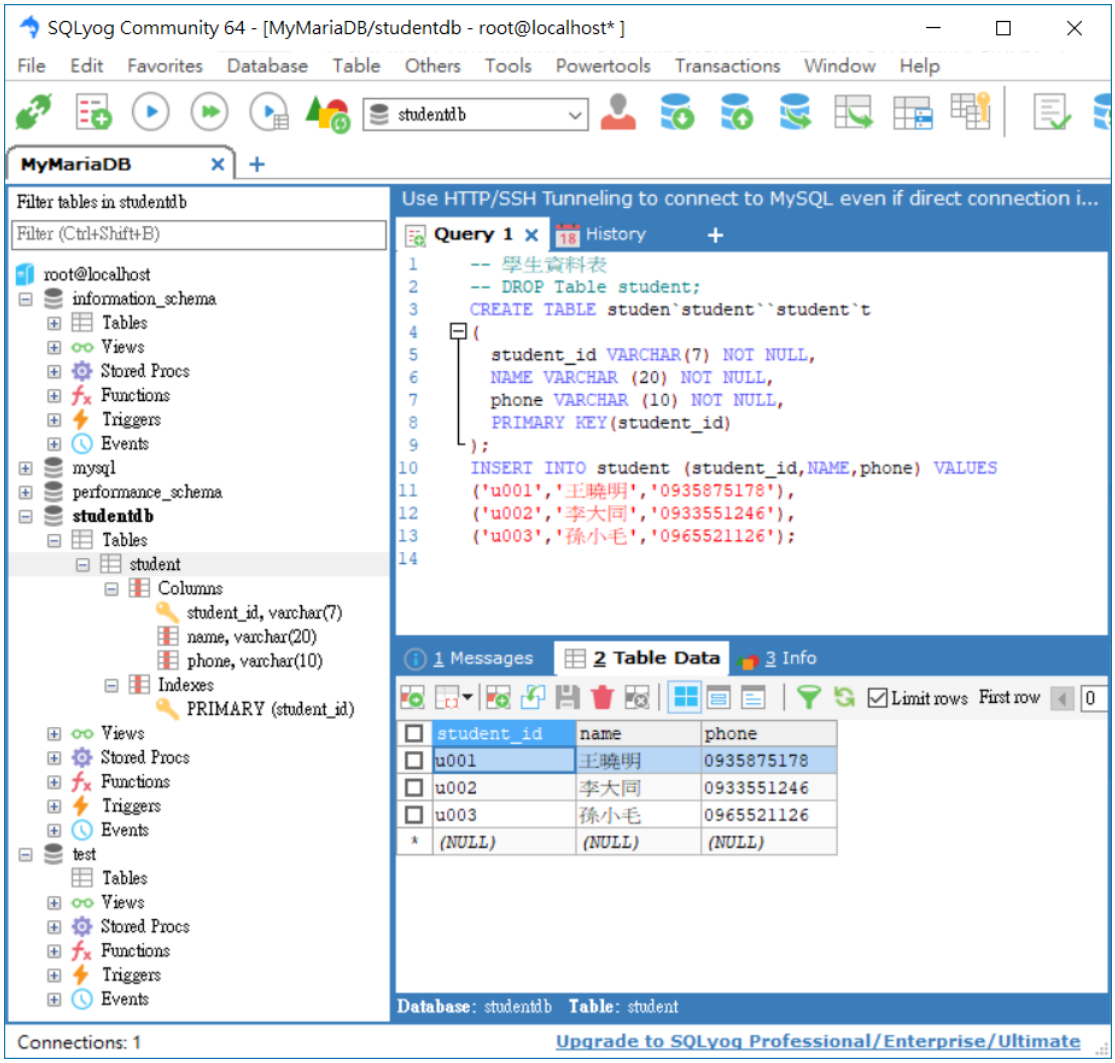

會話管理器
會話名稱
Unnamed
設置
進階
統計
網絡類型：
MariaDB or MySQL (TCP/IP)
主機名/IP：
127.0.0.1
認證提示
使用 Windows 認證
用戶：
root
密碼：
連接埠：
3306
壓縮的客戶端服務器協議
資料庫:
分號分隔
注釋：
新建
保存
刪除
打開
取消
更多



注意: 安裝在系上電腦的 mariadb 之 root 的密碼是 mis123

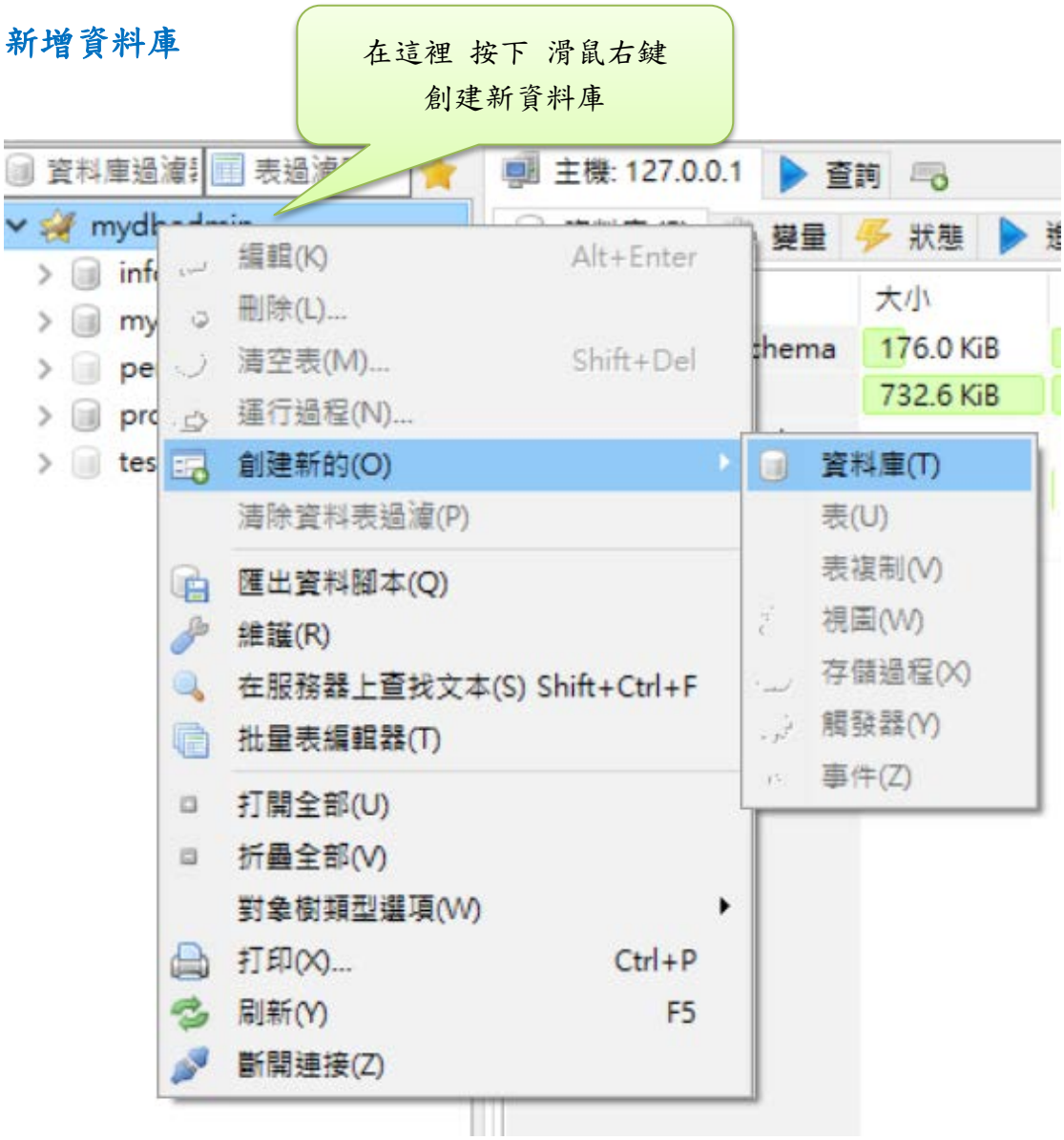


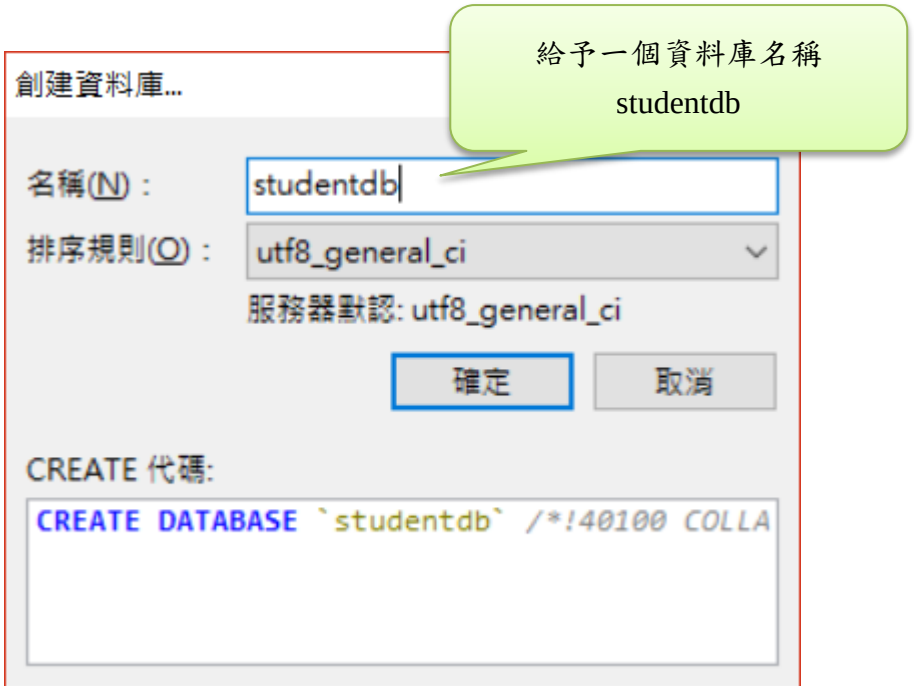
資料庫管理—SQLyog Community (給有興趣的人參考)

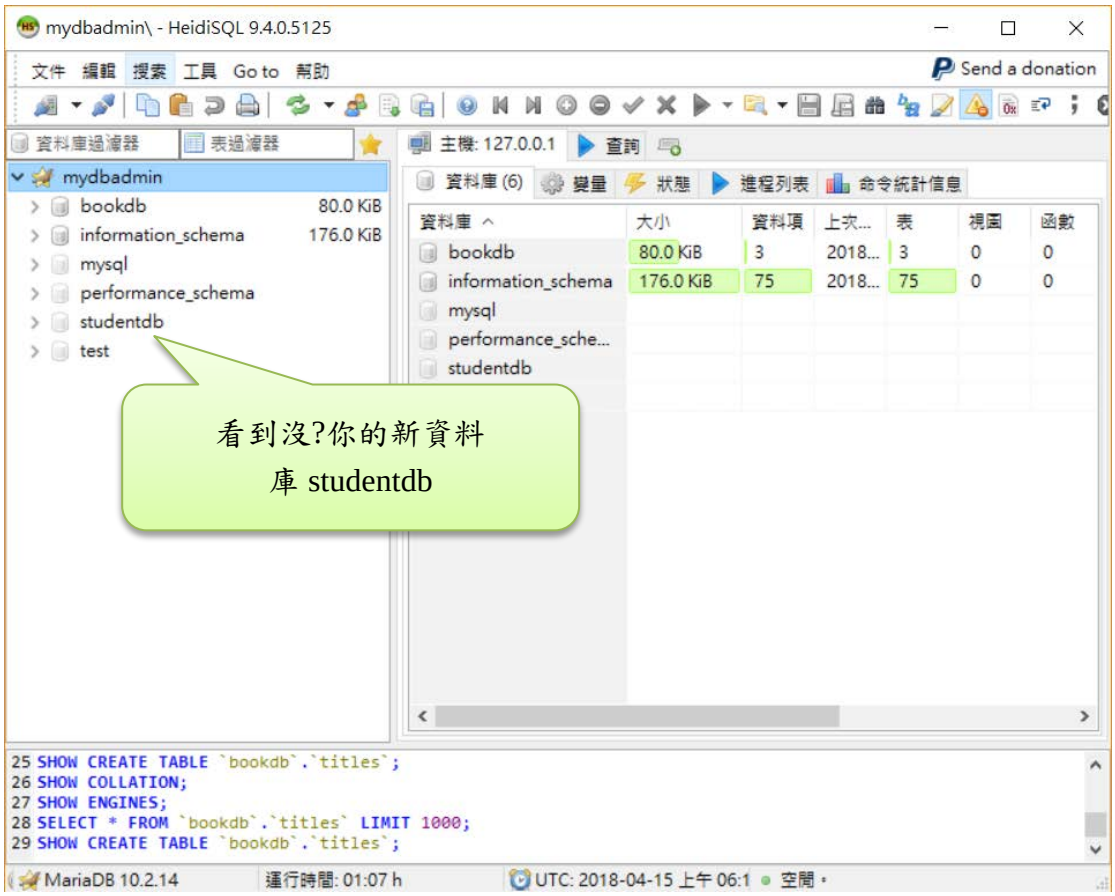

SQLyog Community 64 - [MyMariaDB/studentdb - root@localhost*]
File Edit Favorites Database Table Others Tools Powertools Transactions Window Help
studentdb
MyMariaDB
Filter tables in studentdb
Filter (Ctrl+Shift+B)
root@localhost
information_schema
Tables
Views
Stored Procs
Functions
Triggers
Events
mysql
performance_schema
studentdb
Tables
student
Columns
student_id, varchar(7)
name, varchar(20)
phone, varchar(10)
Indexes
PRIMARY (student_id)
Views
Stored Procs
Functions
Triggers
Events
test
Tables
Views
Stored Procs
Functions
Triggers
Events
Use HTTP/SSH Tunneling to connect to MySQL even if direct connection i...
Query 1
History
1 -- 學生資料表
2 -- DROP Table student;
3 CREATE TABLE studen`student``student`t
4 (
5 student_id VARCHAR(7) NOT NULL,
6 NAME VARCHAR (20) NOT NULL,
7 phone VARCHAR (10) NOT NULL,
8 PRIMARY KEY(student_id)
9);
10 INSERT INTO student (student_id,NAME,phone) VALUES
11 ('u001','王曉明','0935875178'),
12 ('u002','李大同','0933551246'),
13 ('u003','孫小毛','0965521126');
14
1 Messages
2 Table Data
3 Info
Limit rows
First row
0
student_id name phone
u001 王曉明 0935875178
u002 李大同 0933551246
u003 孫小毛 0965521126
* (NULL) (NULL) (NULL)
Database: studentdb Table: student
Connections: 1
Upgrade to SQLyog Professional/Enterprise/Ultimate

Lab0.2:新增一個學生資料庫 studentdb

新增資料庫




給予一個資料庫名稱
studentdb
創建資料庫...
名稱(N)：
studentdb
排序規則(O)：
utf8_general_ci
服務器默認: utf8_general_ci
確定
取消
CREATE 代碼:
CREATE DATABASE `studentdb` /*!40100 COLLA


mydbadmin\ - HeidiSQL 9.4.0.5125
文件 編輯 搜索 工具 Go to 幫助
Send a donation
資料庫過濾器
表過濾器
主機: 127.0.0.1
查詢
mydbadmin
bookdb 80.0 KiB
information_schema 176.0 KiB
mysql
performance_schema
studentdb
test
資料庫 (6)
變量
狀態
進程列表
命令統計信息
資料庫
大小
資料項
上次...
表
視圖
函數
bookdb 80.0 KiB 3 2018... 3 0 0
information_schema 176.0 KiB 75 2018... 75 0 0
mysql
performance_sche...
studentdb
看到沒?你的新資料庫 studentdb
25 SHOW CREATE TABLE `bookdb`.`titles`;
26 SHOW COLLATION;
27 SHOW ENGINES;
28 SELECT * FROM `bookdb`.`titles` LIMIT 1000;
29 SHOW CREATE TABLE `bookdb`.`titles`;
MariaDB 10.2.14
運行時間: 01:07 h
UTC: 2018-04-15 上午 06:1
空閒

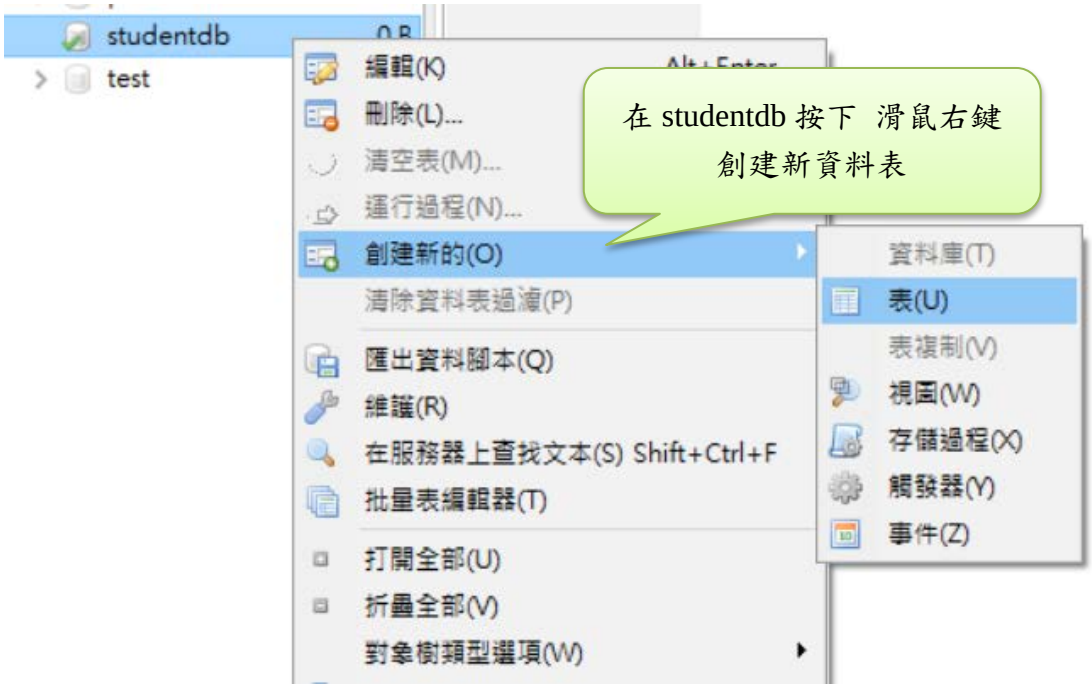
Lab0.5:學生資料表新增實作(手動操作)

新增資料表

- Table 資料表

資料表顧名思義，就是存放資料的表單，是由多筆記錄匯集而成。每個資料表皆具有一個表單名稱，在同一個資料庫的資料表，它的表單名稱皆不得重複。資料表的結構概念類似一般表格，具有行與列的特性，通常資料列的欄位數是固定的，並依照資料記錄多寡，而有不固定的行高。

在關聯式資料庫的定義中，每個資料表分別代表一個實體，例如員工這個實體，則可以將它轉換成員工資料表。



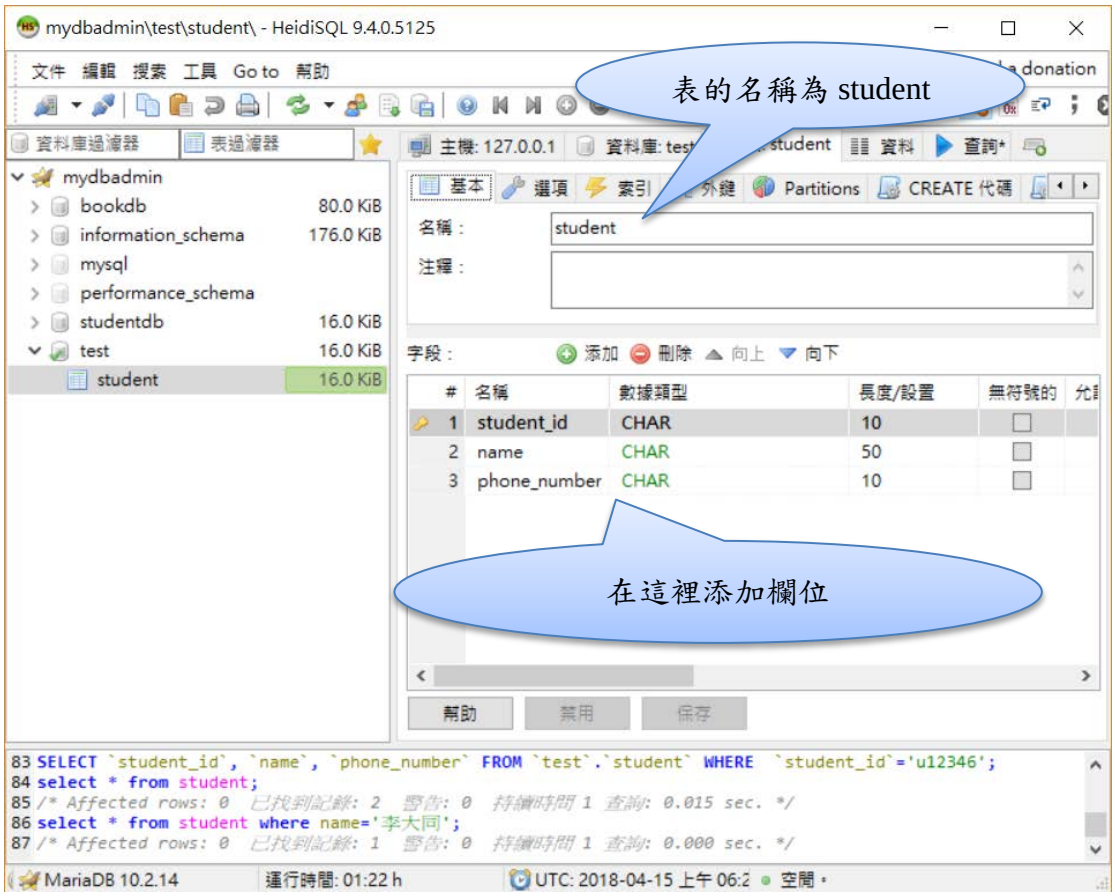
student table 欄位如下:

- Field 欄位

資料庫中的欄位，就是存放資料的空間，類似微軟 Excel 中的儲存格。每一個欄位空間只能存放一筆資料。在設計資料表時，要先定義每個欄位的資料長度與型別為何，例如文字、數字、日期或是布林值。

部分的關聯式資料庫則將 Field 界定為與 Column 同義，因此欄位也可定義為資料表的縱向資料，而不同於上述的單一資料儲存格。

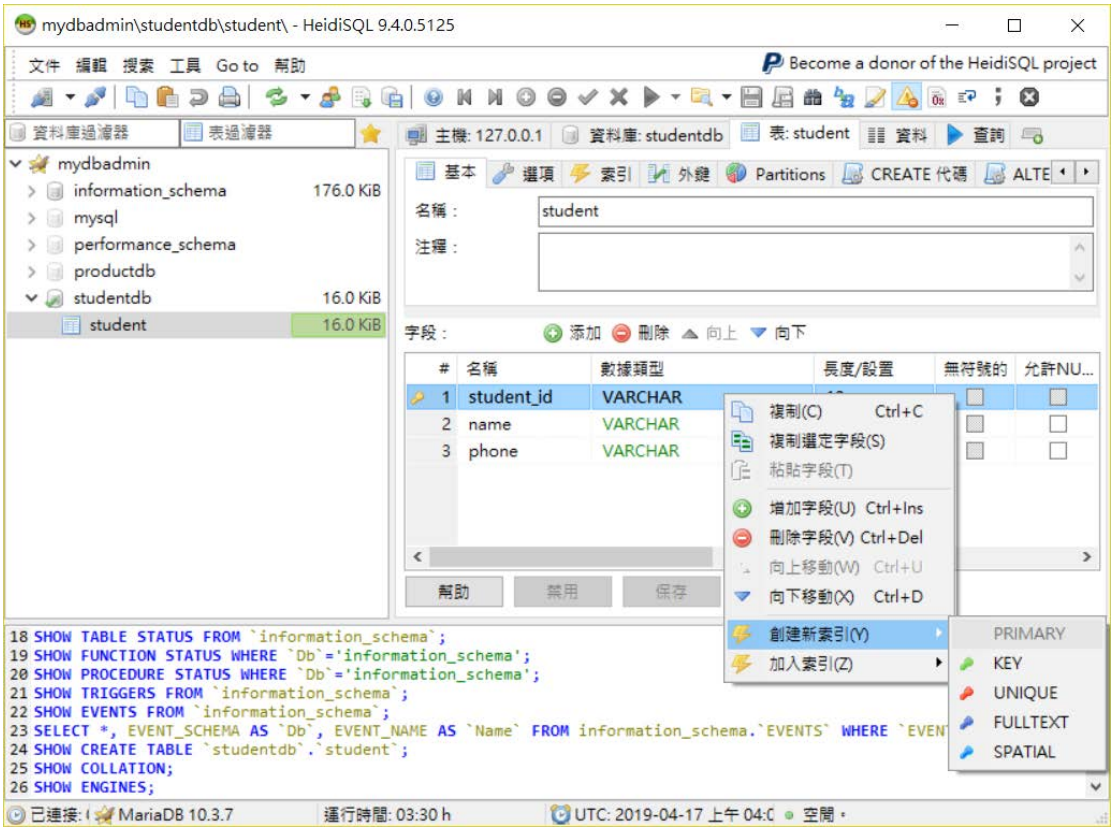
#	名稱	數據類型	長度/設置
1	student_id	VARCHAR	10
2	name	VARCHAR	20
3	phone	VARCHAR	10



索引鍵 Student_id 欄位

- Primary Key 主鍵

是用來識別資料表的唯一值。每個主鍵在資料表中，都是獨一無二的。資料庫管理系統可以藉由這些欄位，識別資料表內的每一筆記錄，並提供資料索引。主鍵可以直接使用一組不重複的資料，或是由系統自動產生，像員工資料表的員工編號，是用系統自動產生的流水號。由於主鍵是提供資料庫索引的重要欄位，故設計資料表時要慎選主鍵，避免造成資料庫系統產生資料錯誤等嚴重問題。



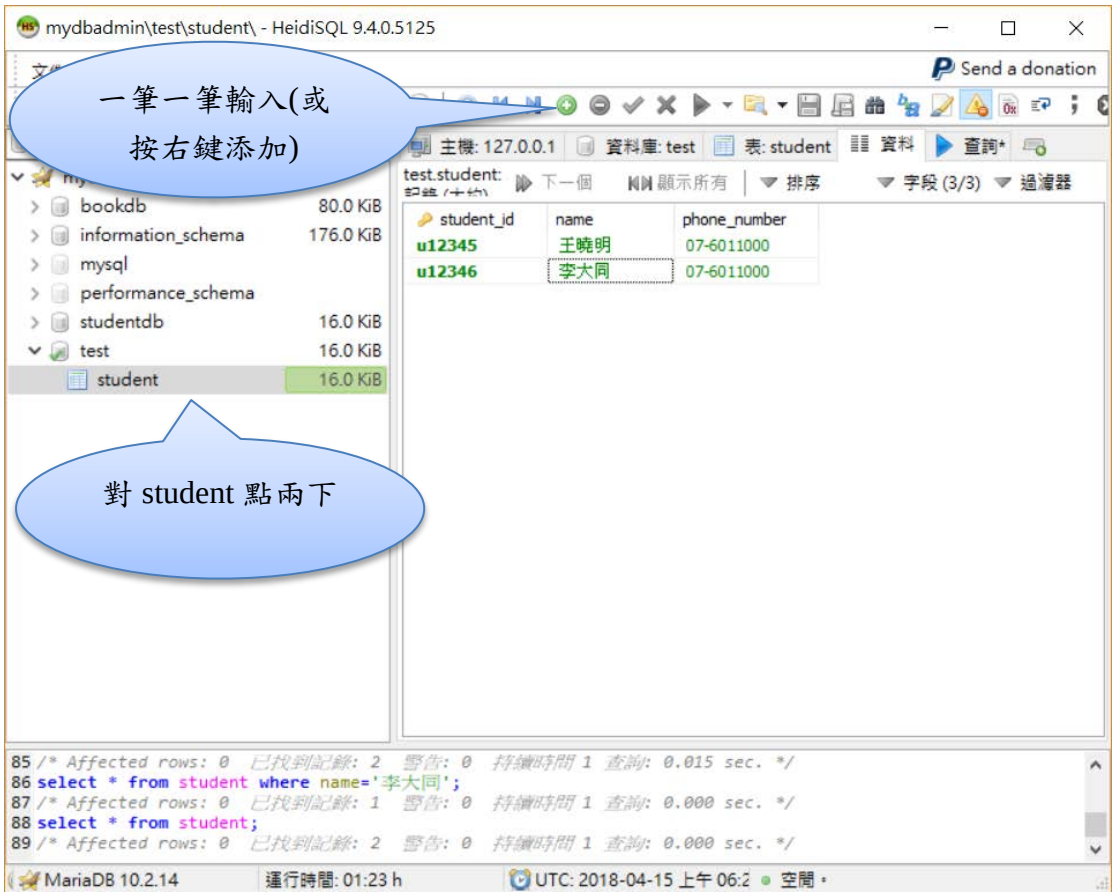
新增 student table 內容 rows 如下:

student_id	name	phone
u001	王曉明	0935875178
u002	李大同	0933551246
u003	孫小毛	0965521126

- Record 記錄

記錄是由一群有關聯性的欄位所集合而成。每一筆記錄，在資料表中代表著一列欄位，這些欄位存放的資料，在彼此之間都有一定的關聯。在關聯式資料庫中，每張資料表的資料列，則可視為該實體的屬性。例如員工資料表內，會記錄每位員工的資料，這些資料就是用來描述每位員工的屬性。而這些資料彼此間的關聯，在於每筆記錄的資料，皆具有一定的共通性。

- 學號索引鍵，只能唯一，而且不可空值



操作手冊:

https://www.heidisql.com/help.php?place=frmTableEditor#createtable

Lab2:學生資料庫操作

- 指令大小寫沒差別
- 欄位名稱可以用中文，不過不建議。
- 新增一筆紀錄record, 學號不能重覆，因為學號索引鍵，只能唯一。
- 分號; 很多行指令放一起，分號是必要的
- Text文字欄位的值必須用引號包起來: 單引號、雙引號皆可。
- 欄位、表、資料庫名稱 可以 用一個特殊的符號包起來: `student`
  在哪裡? 鍵盤左上角有一條蟲~

基本的 SQL command:

SQL的四種基本操作:

CRUD: Create, Retrieve, Update, Delete
SQL命令: Insert, Select, Update, Delete

```
--(1) select
select * from student;
select * from student where student_id ='u003';
select * from student where phone='0965521126';
select * from student where name like '李%';


select student_id, name from student;
select student_id as  學號, name as  姓名  from student;
```

若你在某些系統執行SQL有中文錯誤的問題，可以嘗試以下寫法(中文欄位名稱也用同樣方式包起來):

```
select student_id as `學號`, name as `姓名` from student;
```

```
--(2) insert
Insert Into student (student_id,name,phone) values ('u123','孫大毛','0965521126');
```

```
--(3) delete
delete from student where student_id = 'u123';
```

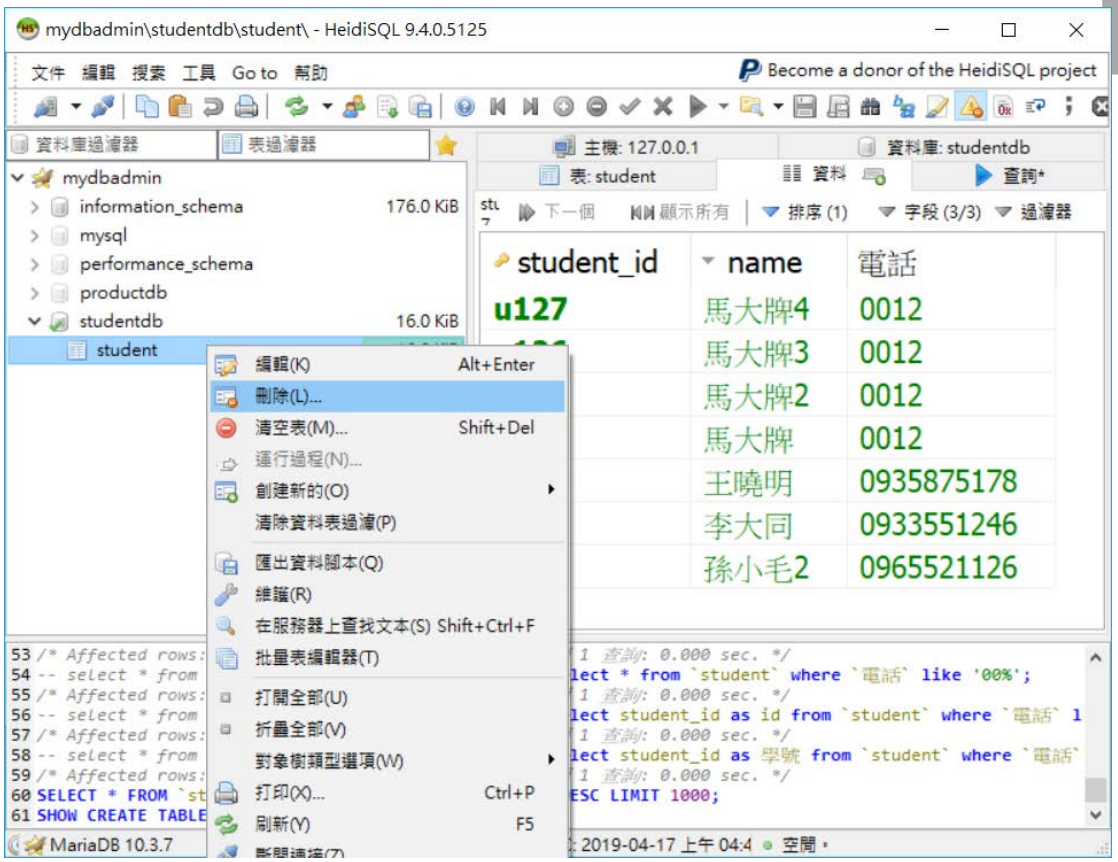
```
--(4)update
```

```
update student set name = '李曉同' where student_id = 'u123';
```

```
還有很多SQL指令:
DROP Table student;
CREATE DATABASE IF NOT EXISTS `studentdb2`;
USE `studentdb2`;
```

Lab2.5:刪除表格或資料庫(手動操作或使用 SQL 命令)

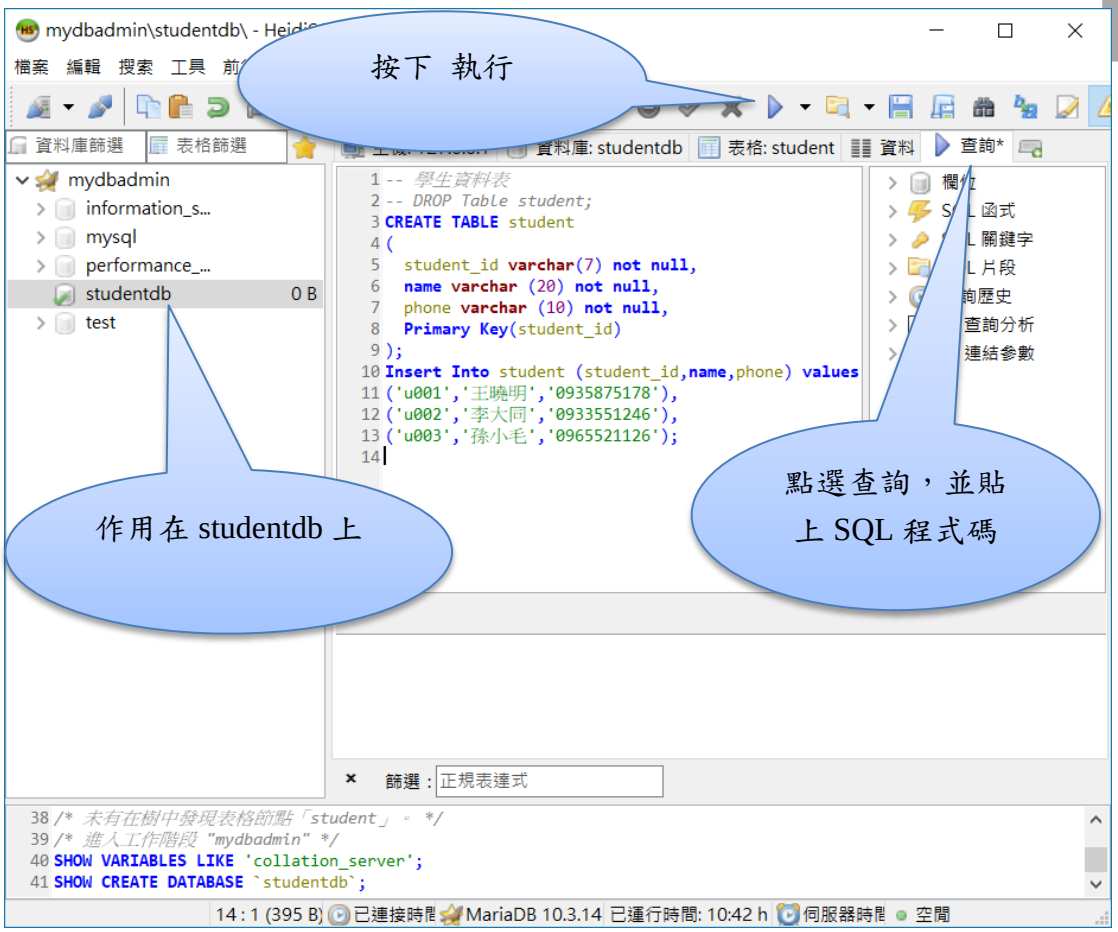
按下右鍵，選擇刪除即可。

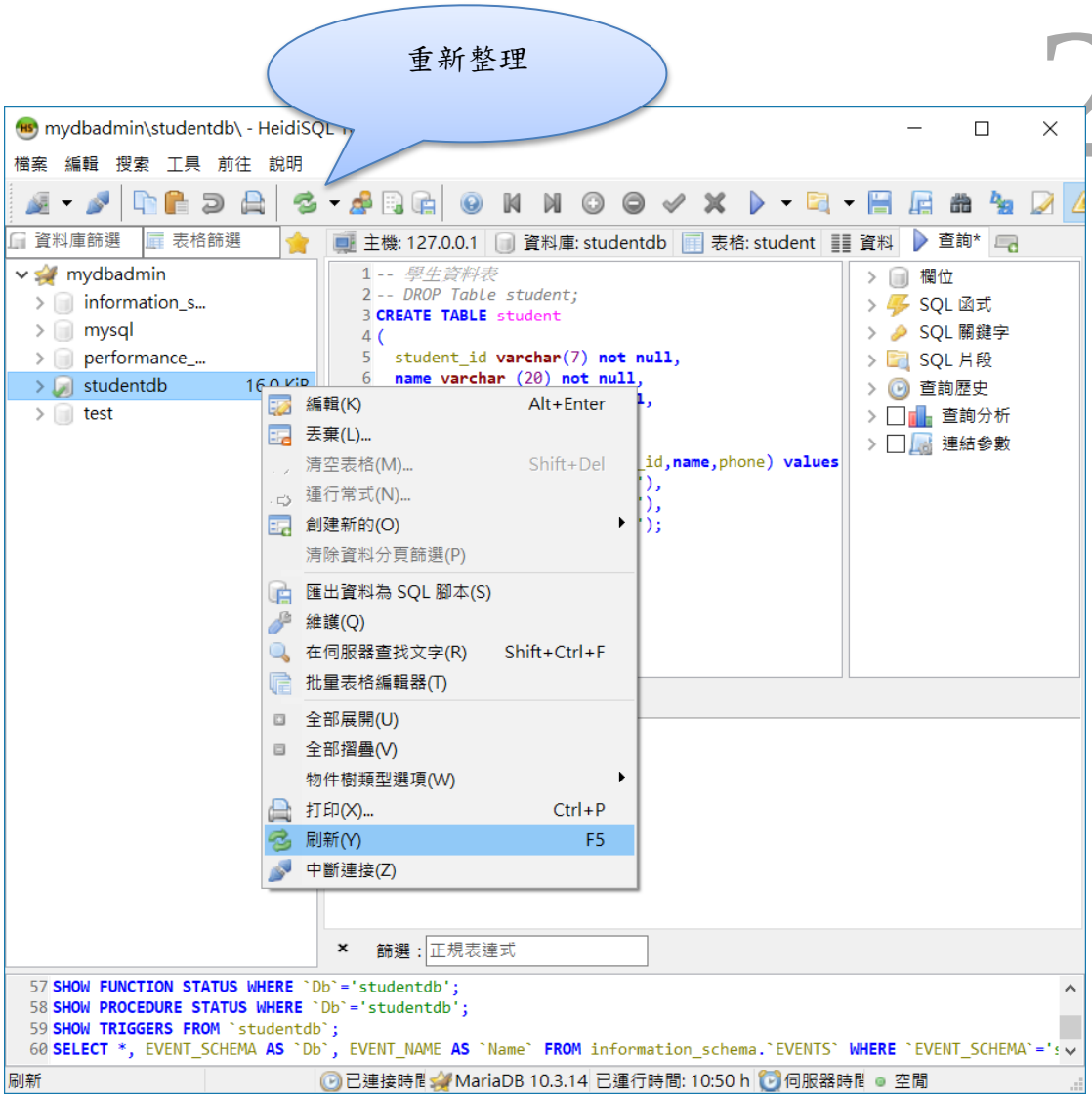


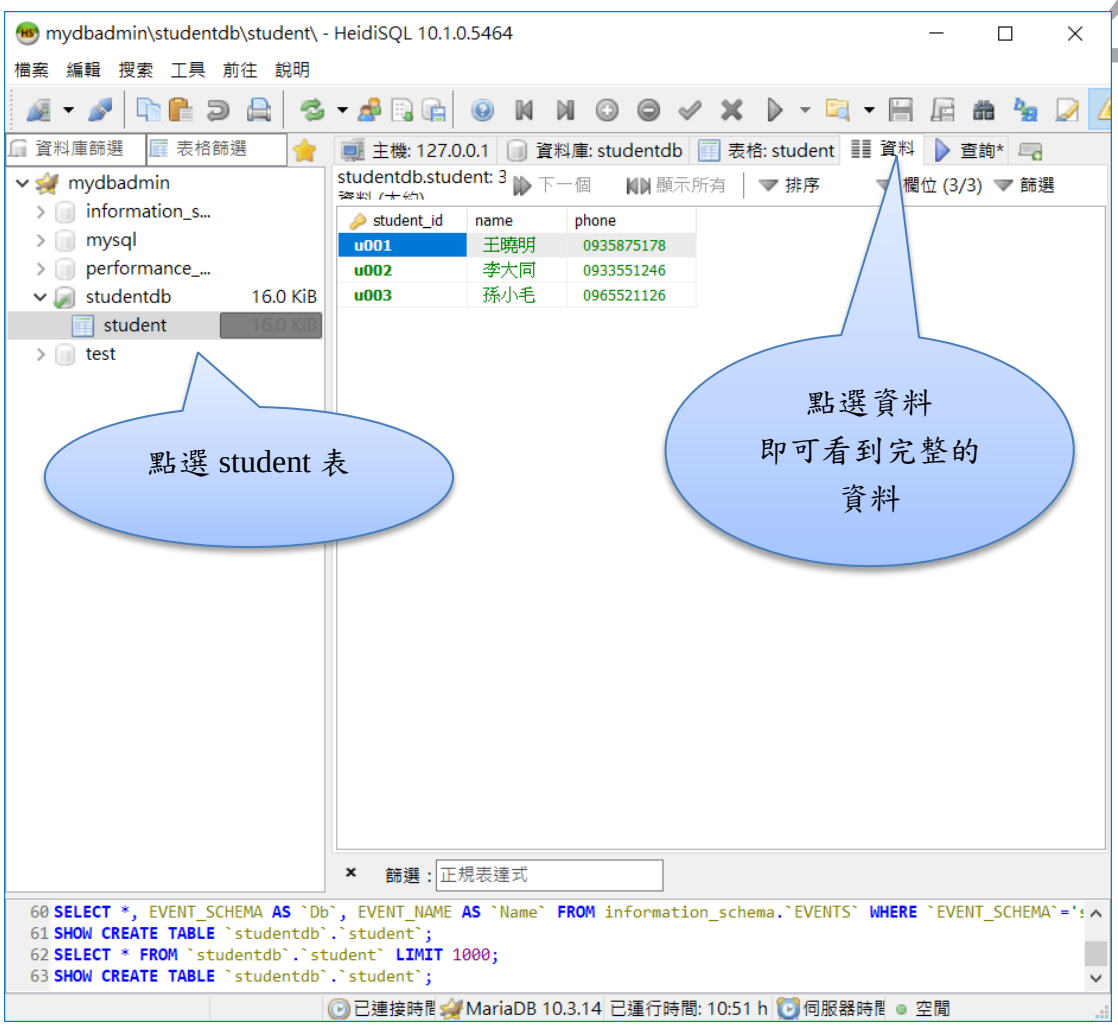
Lab2.3:學生資料表新增實作(SQL 命令)

在資料庫管理工具執行以下 SQL 命令

1. 新增一個查詢窗
2. 滑鼠作用在 studentdb 上
3. 點選或新增一個查詢窗格，
4. 貼上 SQL 程式碼
5. 按下 執行


按下 執行
作用在 studentdb 上
點選查詢，並貼
上 SQL 程式碼
mydbadmin
information_s...
mysql
performance_...
studentdb 0 B
test
-- 學生資料表
-- DROP Table student;
CREATE TABLE student
(
student_id varchar(7) not null,
name varchar (20) not null,
phone varchar (10) not null,
Primary Key(student_id)
);
Insert Into student (student_id,name,phone) values
('u001','王曉明','0935875178'),
('u002','李大同','0933551246'),
('u003','孫小毛','0965521126');
/* 未有在樹中發現表格節點「student」。 */
/* 進入工作階段 "mydbadmin" */
SHOW VARIABLES LIKE 'collation_server';
SHOW CREATE DATABASE `studentdb`;


重新整理
mydbadmin\studentdb\ - HeidiSQL
檔案 編輯 搜索 工具 前往 說明
資料庫篩選
表格篩選
主機: 127.0.0.1
資料庫: studentdb
表格: student
資料
查詢*
mydbadmin
information_s...
mysql
performance_...
studentdb
test
1 -- 學生資料表
2 -- DROP Table student;
3 CREATE TABLE student
4 (
5 student_id varchar(7) not null,
6 name varchar (20) not null,
欄位
SQL 函式
SQL 關鍵字
SQL 片段
查詢歷史
查詢分析
連結參數
編輯(K) Alt+Enter
丟棄(L)...
清空表格(M)... Shift+Del
運行常式(N)...
創建新的(O)
清除資料分頁篩選(P)
匯出資料為 SQL 腳本(S)
維護(Q)
在伺服器查找文字(R) Shift+Ctrl+F
批量表格編輯器(T)
全部展開(U)
全部摺疊(V)
物件樹類型選項(W)
打印(X)... Ctrl+P
刷新(Y) F5
中斷連接(Z)
篩選：正規表達式
57 SHOW FUNCTION STATUS WHERE `Db`='studentdb';
58 SHOW PROCEDURE STATUS WHERE `Db`='studentdb';
59 SHOW TRIGGERS FROM `studentdb`;
60 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='
刷新
已連接時間
MariaDB 10.3.14
已運行時間: 10:50 h
伺服器時間
空閒



新增學生資料表-SQL 命令

```
-- 學生資料表
CREATE TABLE student
(
   student_id varchar(7) not null,
   name varchar (20) not null,
   phone varchar (10) not null,
   Primary Key(student_id)
);
Insert Into student (student_id,name,phone) values
('u001','王曉明','0935875178'),
('u002','李大同','0933551246'),
('u003','孫小毛','0965521126');
```

varchar(20)表示甚麼意思？
Variable char
Acronym Definition
VARCHAR Various Characters
VARCHAR Variable Character

同時新增 studentdb 學生資料庫與 student 資料表較為複雜的寫法

```
-- 導出 studentdb 的資料庫結構
CREATE DATABASE IF NOT EXISTS `studentdb2`;
USE `studentdb2`;

-- 導出 表 studentdb.student 結構
CREATE TABLE IF NOT EXISTS `student` (
  `student_id` varchar(10) NOT NULL,
  `name` varchar(20) NOT NULL,
  `phone` varchar(10) NOT NULL,
  PRIMARY KEY (`student_id`)
);

-- 導出表 studentdb.student 的資料：~3 rows (大約)
INSERT INTO `student` (`student_id`, `name`, `phone`) VALUES
	('u001', '王曉明', '0935875178'),
	('u002', '李大同', '0933551246'),
	('u003', '孫小毛2', '0965521126');
```

Lab2.5:學生資料表新增實作(import csv 檔案)

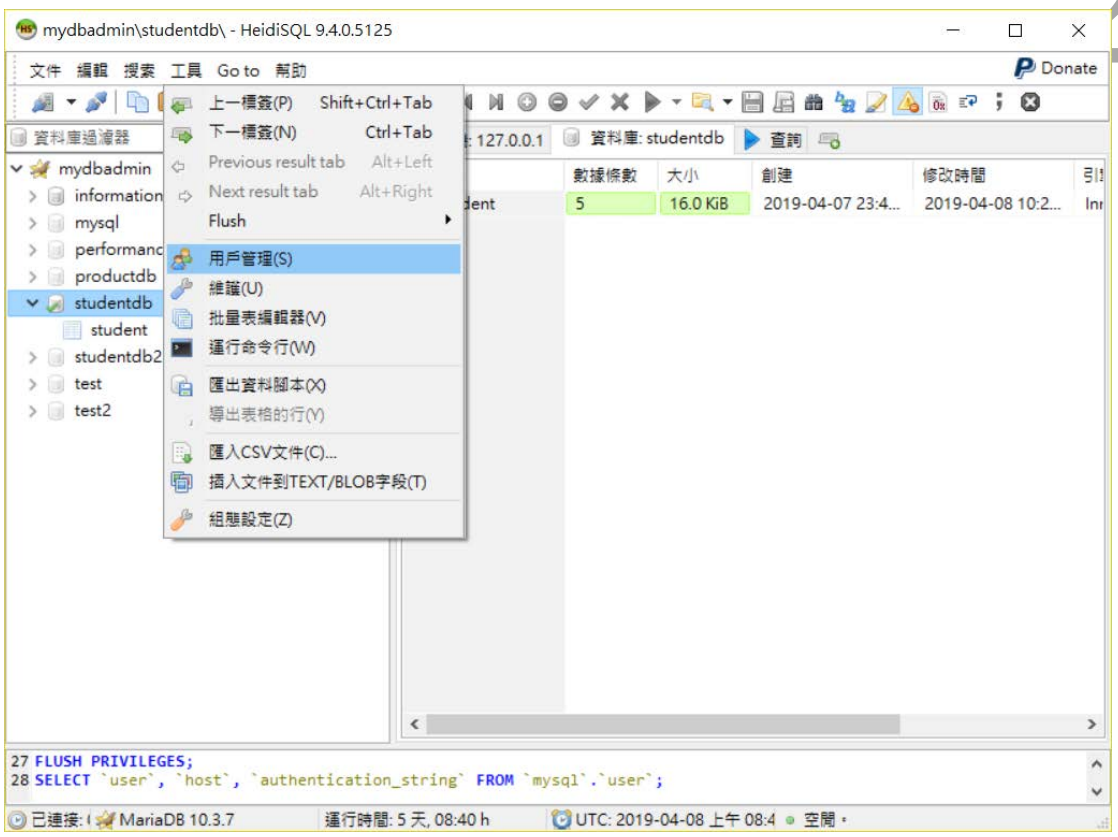
由 csv 格式檔案匯入 student table 內容 rows:

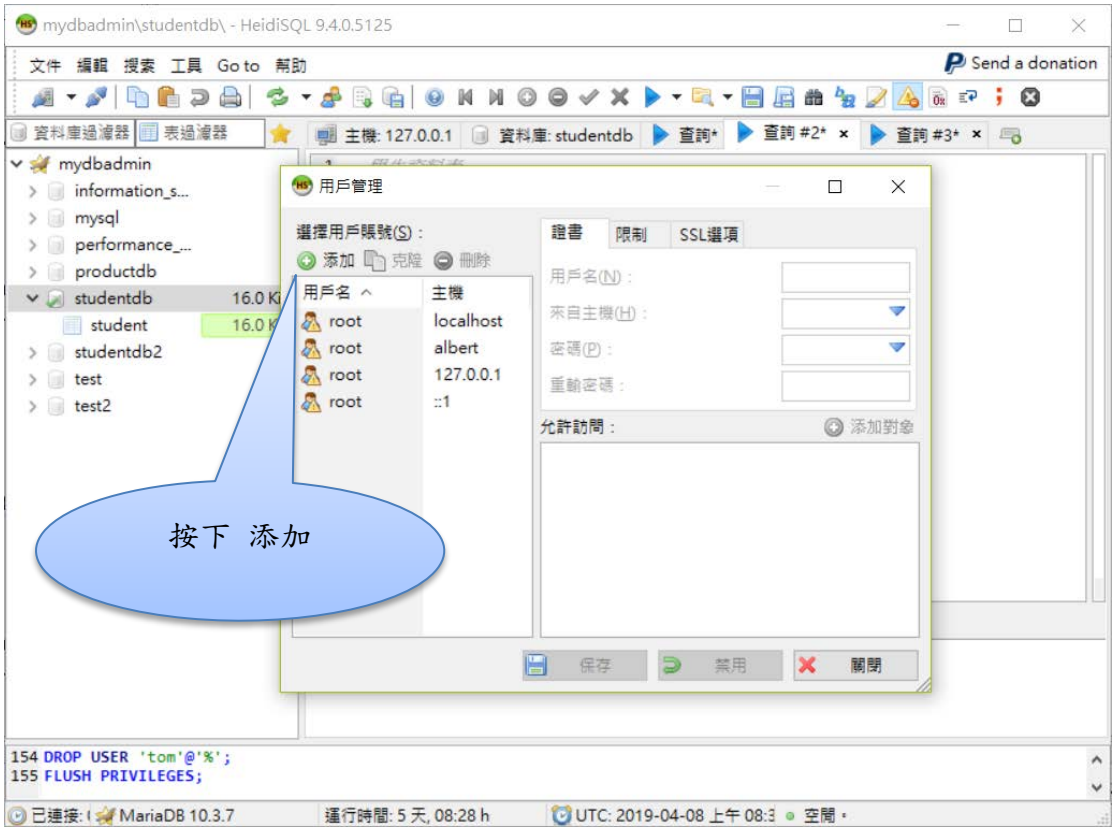
事先準備好 student.csv 檔案，利用資料庫管理工具匯入。
import from csv
按照指示，一步一步匯入即可。

Lab2.8:新增使用者，並做權限管理

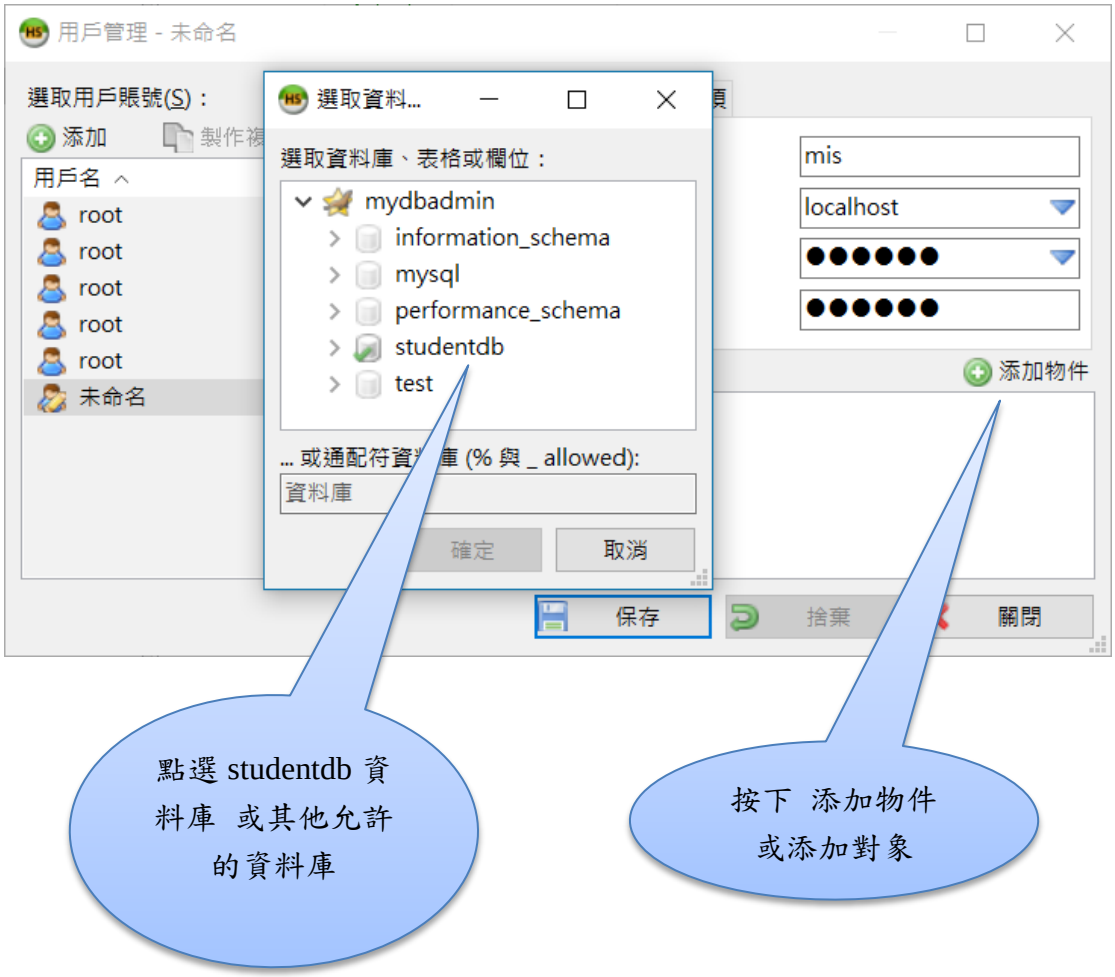
- 新增使用者:mis
- 密碼:mis123

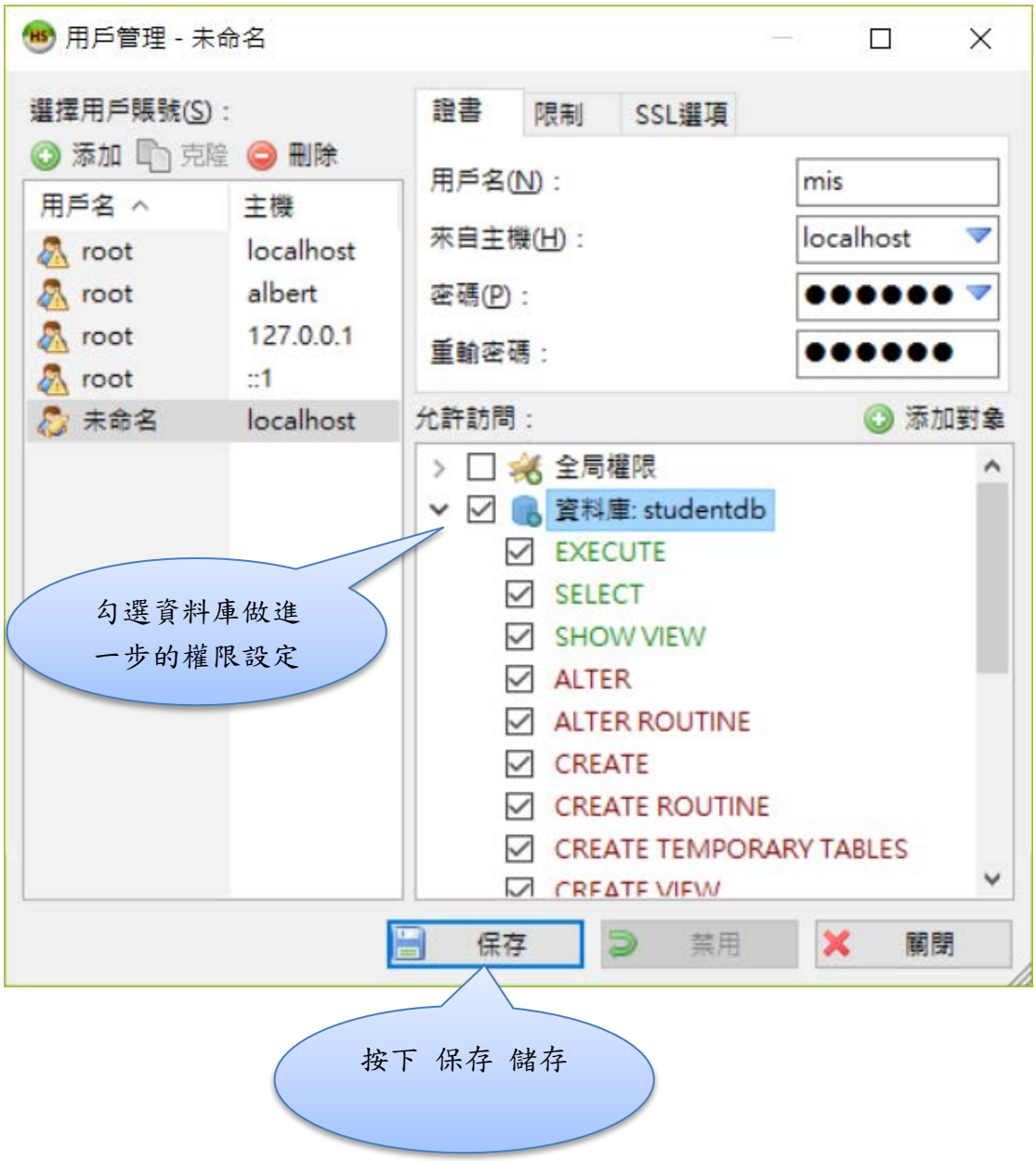
工具 → 用戶管理

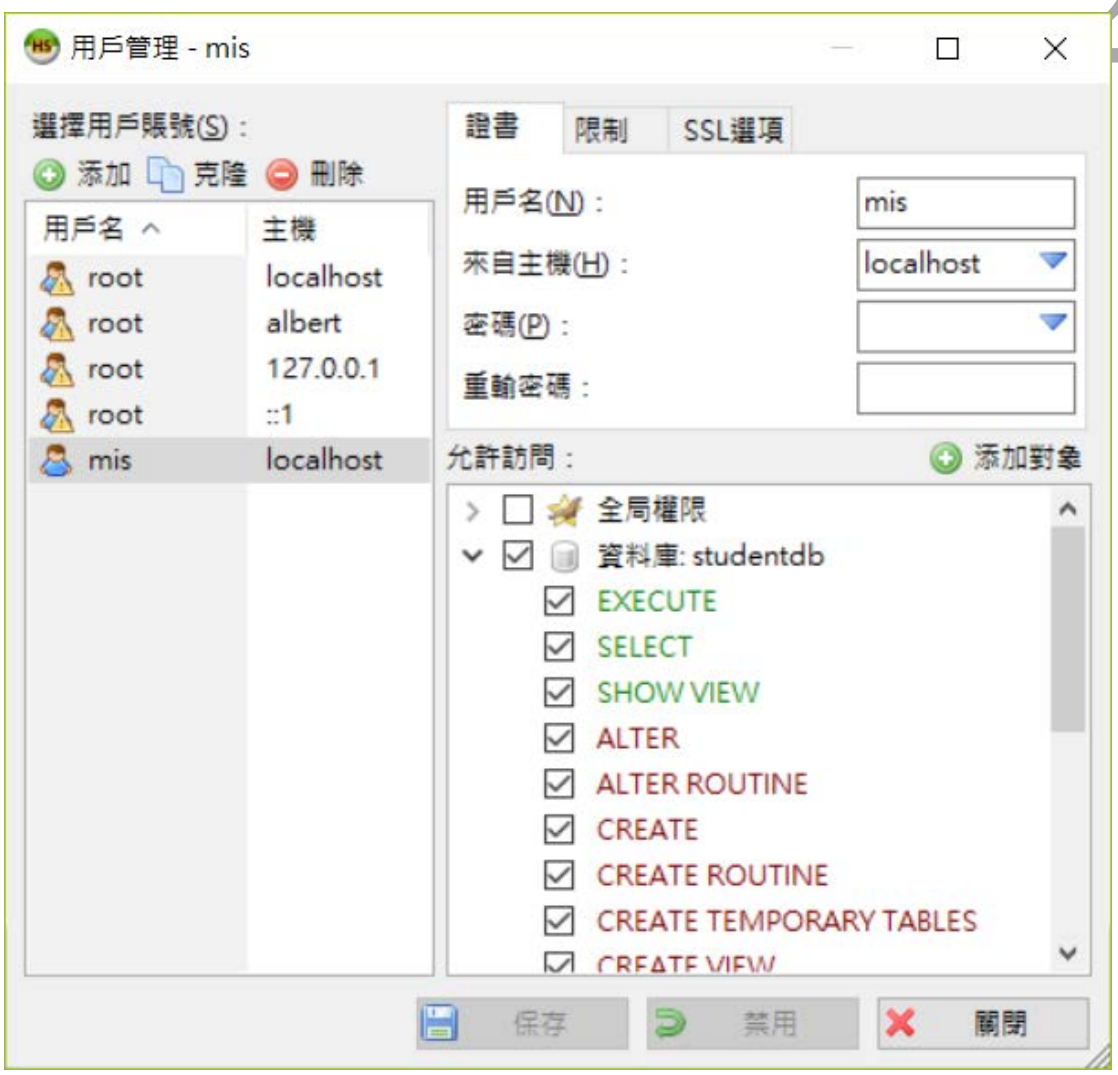

mydbadmin\studentdb\ - HeidiSQL 9.4.0.5125
文件 編輯 搜索 工具 Go to 幫助
Donate
上一標籤(P) Shift+Ctrl+Tab
下一標籤(N) Ctrl+Tab
Previous result tab Alt+Left
Next result tab Alt+Right
Flush
用戶管理(S)
維護(U)
批量表編輯器(V)
運行命令行(W)
匯出資料腳本(X)
導出表格的行(Y)
匯入CSV文件(C)...
插入文件到TEXT/BLOB字段(T)
組態設定(Z)
資料庫過濾器
mydbadmin
information
mysql
performanc
productdb
studentdb
student
studentdb2
test
test2
127.0.0.1
資料庫: studentdb
查詢
數據條數
大小
創建
修改時間
5
16.0 KiB
2019-04-07 23:4...
2019-04-08 10:2...
27 FLUSH PRIVILEGES;
28 SELECT `user`, `host`, `authentication_string` FROM `mysql`.`user`;
已連接:
MariaDB 10.3.7
運行時間: 5 天, 08:40 h
UTC: 2019-04-08 上午 08:4
空閒


mydbadmin\studentdb\ - HeidiSQL 9.4.0.5125
文件 編輯 搜索 工具 Go to 幫助
Send a donation
資料庫過濾器
表過濾器
主機: 127.0.0.1
資料庫: studentdb
查詢*
查詢 #2*
查詢 #3*
mydbadmin
information_s...
mysql
performance_...
productdb
studentdb
student
studentdb2
test
test2
用戶管理
選擇用戶賬號(S):
添加
克隆
刪除
用戶名
主機
root
localhost
root
albert
root
127.0.0.1
root
::1
證書
限制
SSL選項
用戶名(N):
來自主機(H):
密碼(P):
重輸密碼:
允許訪問:
添加對象
保存
禁用
關閉
按下 添加
154 DROP USER 'tom'@'%';
155 FLUSH PRIVILEGES;
已連接:
MariaDB 10.3.7
運行時間: 5 天, 08:28 h
UTC: 2019-04-08 上午 08:3
空閒


用戶管理 - 未命名
選取用戶賬號(S)：
添加
製作複本
刪除
用戶名
主機
root
localhost
root
mypc2
root
127.0.0...
root
::1
root
%
未命名
localhost
憑證
限制
SSL 選項
用戶名(N)：
mis
來自主機(H)：
localhost
密碼(P)：
重輸密碼：
允許存取：
添加物件
全域權限
輸入帳號密碼
保存
捨棄
關閉


用戶管理 - 未命名
選取用戶賬號(S)：
添加
用戶名
root
root
root
root
root
未命名
選取資料...
選取資料庫、表格或欄位：
mydbadmin
information_schema
mysql
performance_schema
studentdb
test
... 或通配符資料庫 (% 與 _ allowed):
資料庫
確定
取消
mis
localhost
添加物件
保存
捨棄
關閉
點選 studentdb 資料庫 或其他允許的資料庫
按下 添加物件 或添加對象


用戶管理 - 未命名
選擇用戶賬號(S)：
添加
克隆
刪除
用戶名
主機
root
localhost
root
albert
root
127.0.0.1
root
::1
未命名
localhost
證書
限制
SSL選項
用戶名(N)：
mis
來自主機(H)：
localhost
密碼(P)：
重輸密碼：
允許訪問：
添加對象
全局權限
資料庫: studentdb
EXECUTE
SELECT
SHOW VIEW
ALTER
ALTER ROUTINE
CREATE
CREATE ROUTINE
CREATE TEMPORARY TABLES
CREATE VIEW
保存
禁用
關閉
勾選資料庫做進
一步的權限設定
按下 保存 儲存



設定與資料輸入完畢之後，你的Heidi就可以關閉不用了。接下來就是寫程式了。

Lab4: 入門介紹 Java 程式去執行 SQL 指令

1. 驅動程式下載(for Java)

MariaDB FOUNDATION

Downloads Source, Binaries, and Packages

To show only the files you want, use the checkboxes in the sidebar. For other MariaDB Connector/J releases, click on "View All Releases". For faster downloads choose a mirror close to you.

MariaDB Connector/J is used to connect applications developed in Java to MariaDB and MySQL databases using the standard JDBC API. The client library is LGPL licensed.

See this article for more information.

MariaDB Connector/J .jar files are available at: https://downloads.mariadb.com/Connectors/java/

MariaDB is free and open source software

The MariaDB database server is published as free and open source software under the General Public License version 2. You can download and use it as much as you want free of charge. *All use of the binaries from mariadb.org is at your own risk as stated in the GPLv2.* While we do our best to make the world's best database software, the MariaDB Foundation does not provide any guarantees and cannot be hold liable for any issues you may encounter.

MariaDB Connector/J 2.2.3 Stable View all releases

2018-03-14

Release Notes Changelog

Affordable,enterprise class product support, professional services, and training for your MariaDB database is available from the MariaDB Foundation's release sponsor, MariaDB Corporation. To learn more about them and their services for MariaDB, visit their website, or email MariaDB Corporation at sales@mariadb.com.

File Name	Package Type	OS / CPU	Size	Meta
mariadb-java-client-2.2.3-sources.jar	java source jar	Source	600.8 kB	Checksum Instructions
MariaDB Connector/J .jar files	jar	Universal		Checksum Instructions

Want to learn more about MariaDB? Check out our whitepapers.

Mirror

連結:https://downloads.mariadb.org/connector-java/2.2.3/

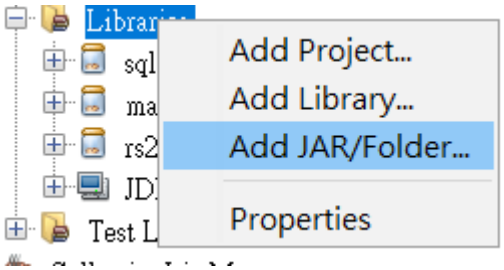
2. 載入驅動程式

資料庫系統一大堆，每一種都有不同的驅動程式。你先要告訴專案使用哪一種驅動程式。

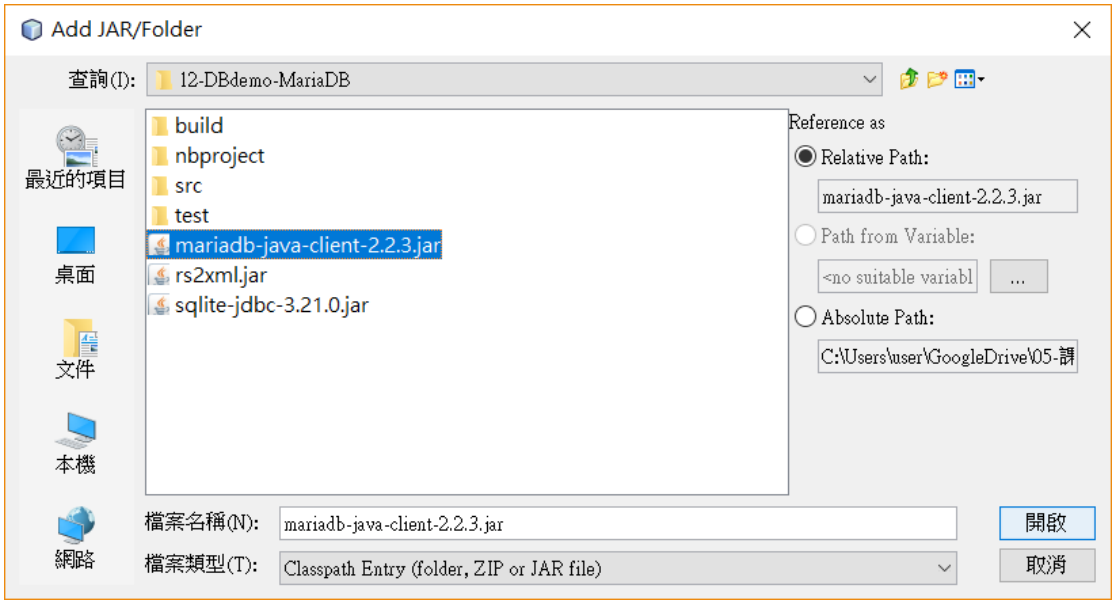
*必須將 MySql, MariaDB, or JavaDB 驅動程式載入才能執行

Libraries→按右鍵→加入 jar 檔→選擇你安裝的目錄之下的驅動程式.jar

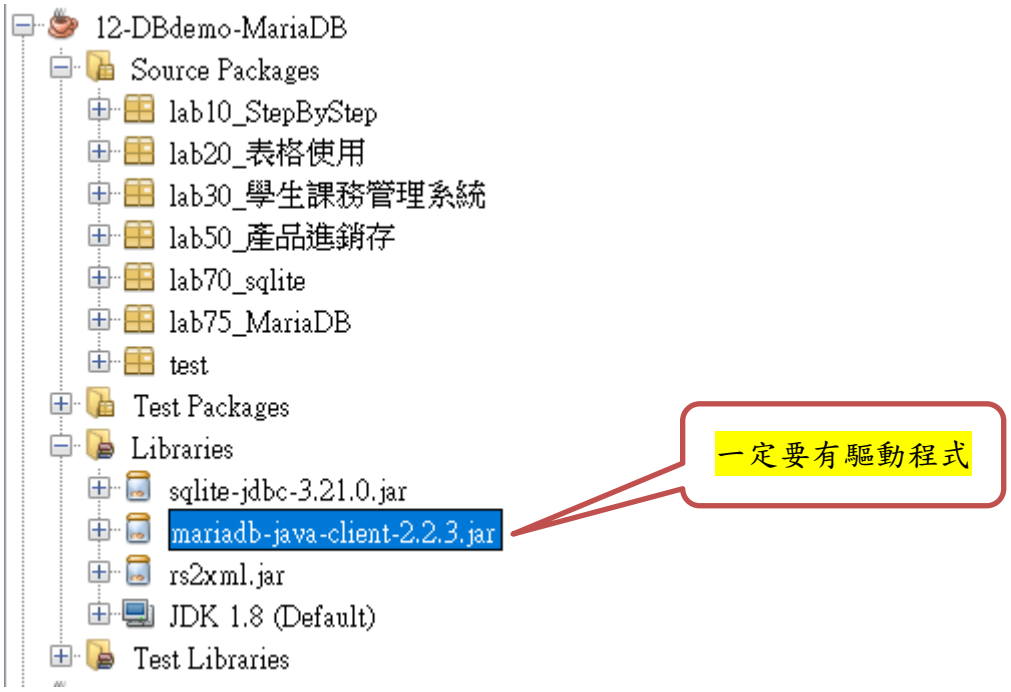
Libraries→按右鍵→加入 jar 檔



選擇你安裝的目錄之下的驅動程式.jar



你的專案下一定要有驅動程式



3. 開始寫 Java Console 程式，連線成功才能開始玩資料

```
String DB_URL = "jdbc:mariadb://localhost:3306/studentdb";
        Connection conn = null;
        //連線 getConnection 須給帳號與密碼
        conn = DriverManager.getConnection(DB_URL, "mis", "mis123");
```

JDBC 是哪幾個字的縮寫？ 答案是:Java Data Base Connectivity

4. 準備好 Statement SQL 敘述

Statement 敘述 描述 指的是 SQL 敘述

```
state = conn.createStatement();
state = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
```

PreparedStatement vs. Statement

兩種都可使用，只要能熟練其中一種即可。

```
String sql = "Select * from book";
PreparedStatement state = conn.prepareStatement(sql);
ResultSet rs = state.executeQuery();
```

5. 資料庫查詢結果放在哪裡?

查詢結果放在臨時的一張表 result
result = state.executeQuery("select * from student");

資料庫查詢與操作可以使用以下 3 種常用的物件，你任選一種去用。

- ResultSet—連線式 不可更新、不可以向前捲動，除非給以下參數:
  type: ResultSet.TYPE_SCROLL_INSENSITIVE (has a scrollable cursor)
  concurrency: ResultSet.CONCUR_UPDATABLE (can be updated)

- RowSet—與 ResultSet 類似，但是可以前後捲動 後一筆 next() 前一筆 previous()，可以更新

- CachedRowSett—與 RowSet 類似，但是存在記憶體中，是非連線式(disconnected)，也就是資料庫因網路斷開也沒關係，只要 rowset 物件需要更新時，可以再連得上資料庫就可以，不會有影響。ResultSet, RowSet 沒有這個特異功能(因為連線式的關係)。不過須確定你的資料量會不會太大，記憶體是否足夠。

A CachedRowSet object is special in that it can operate without being connected to its data source, that is, it is a disconnected RowSet object. It gets its name from the fact that it stores (caches) its data in memory so that it can operate on its own data rather than on the data stored in a database.

The CachedRowSet interface is the superinterface for all disconnected RowSet objects, so everything demonstrated here also applies to WebRowSet, JoinRowSet, and FilteredRowSet objects.

資料來源:https://docs.oracle.com/cd/B28359_01/java.111/b31224/jcrowset.htm

ResultSet:

```
state = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
//查詢結果放在臨時的一張表 result
result = state.executeQuery("select * from student");
```

RowSet:

```
RowSet rowset = new JdbcRowSetImpl();   (任選一種)
CachedRowSet rowset = new JdbcCachedRowSetImpl(); (任選一種)

rowset.setUrl(DB_URL);
rowset.setUsername(username);
rowset.setPassword(password);

rowset.setCommand("select * from student");
rowset.execute();

rowset.setCommand("INSERT INTO student (student_id,name,phone) VALUES
('u011','Paul','123');");
rowset.execute();
```

- 取出欄位內容:

```
result.getString(1); //取出字串型態欄位之內容
result.getObject(2);//用物件，較通用，可以取 整數 字串等各種型態欄位之內容
```

*使用 ResultSet 去操作資料庫，要對資料庫異動 updatable，讓資料 row 可以後退或往前一筆(scroll 滾動)，必須做以下的參數設定：

```
Statement sta =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
```

Lab4.2: TryCatch 版本

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class StudentDB_10 {

	public static void main(String[] args) {

		String DB_URL = "jdbc:mariadb://localhost:3306/studentdb";
		Connection conn = null;
		Statement state = null;
		ResultSet result = null;

		try {
			conn = DriverManager.getConnection(DB_URL, "mis", "mis123");
			state = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
		} catch (SQLException ex) {
			System.out.println("資料庫連線出問題:" + ex.toString());
		}

		// update
		String sql_update ="update student set name = '孫小毛' where student_id = 'u003' ";
		try{
		   state.execute( sql_update );
		}catch (SQLException ex) {
			System.out.println("資料庫 update 出問題:\n" + ex.toString());
		}


		// insert
		String sql_insert ="Insert Into student (student_id,name,phone) values ('u124','孫大毛','0965521126')";
		try{
```

```
            state.execute( sql_insert );
        }catch (SQLException ex) {
            System.out.println("資料庫 insert 出問題:\n" + ex.toString());
        }


       // delete
        String sql_delete ="delete from student where student_id = 'u123';";
        try{
            state.execute( sql_delete );
        }catch (SQLException ex) {
            System.out.println("資料庫 delete 出問題:\n" + ex.toString());
        }

        //    select
        String sql = "select * from student";

        try {
            result = state.executeQuery(sql);
            while (result.next()) {
                System.out.printf("學號:%s     姓名:%s    電話:%s\n",
                        result.getString(1),
                        result.getString(2),
                        result.getString(3));
            }
        } catch (SQLException ex) {
            System.out.println("資料庫 select 出問題:\n" + ex.toString());
        }
    }

}
```

Lab4.3: 資料庫操作獨立寫在另一個類別

```
package chapter32_database;
```

```
public class StudentDB_20 {

	public static void main(String[] args) {

		DBStudent.connect();
		DBStudent.selectAll();
		DBStudent.insert("u125","孫大毛","0965521126");
		//DBStudent.update("u124","孫大毛 2","0965521126");
		//DBStudent.delete("u125");
		DBStudent.selectAll();

	}
}
```

```
package chapter32_database;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class DBStudent {

	private static String DB_URL = "jdbc:mariadb://localhost:3306/studentdb";
	private static Connection conn = null;
	private static Statement state = null;
	private static ResultSet result = null;

	public static void connect() {

		try {
			conn = DriverManager.getConnection(DB_URL, "mis", "mis123");
			state = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
		} catch (SQLException ex) {
			System.out.println("資料庫連線出問題:" + ex.toString());
		}
```

```
    }

    public static void insert(String student_id, String name, String phone) {

       //String sql_insert ="Insert Into student (student_id,name,phone) values ('u124','孫大毛','0965521126')";

        String sql_insert = String.format("Insert Into student (student_id,name,phone) "
                + "values ('%s','%s','%s')",
                student_id, name, phone);
        try{
           state.execute( sql_insert );
        }catch (SQLException ex) {
            System.out.println("資料庫 insert 出問題:\n" + ex.toString());
        }

    }

    public static void selectAll() {

        // select
        String sql = "select * from student";

        try {
            result = state.executeQuery(sql);
            while (result.next()) {
                System.out.printf("學號:%s    姓名:%s   電話:%s\n",
                        result.getString(1),
                        result.getString(2),
                        result.getString(3));
            }
        } catch (SQLException ex) {
            System.out.println("資料庫 select 出問題:\n" + ex.toString());
        }
    }


}
```

Lab4.5 補充: 取得有多少個欄位(欄位數)

```
//另外一種方式列印內容--先取欄位數 再用迴圈取欄位內容
        ResultSetMetaData meta = result.getMetaData();
        int column = meta.getColumnCount(); //有幾個欄位
        System.out.println("有多少欄位:"+column);

        result.first();
        while (result.next()) {
            for (int i = 1; i <= column; i++) {
                System.out.printf("%s ", result.getObject(i)); //取出字串型態欄位之內
            }
            System.out.println();
        }
```