

"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

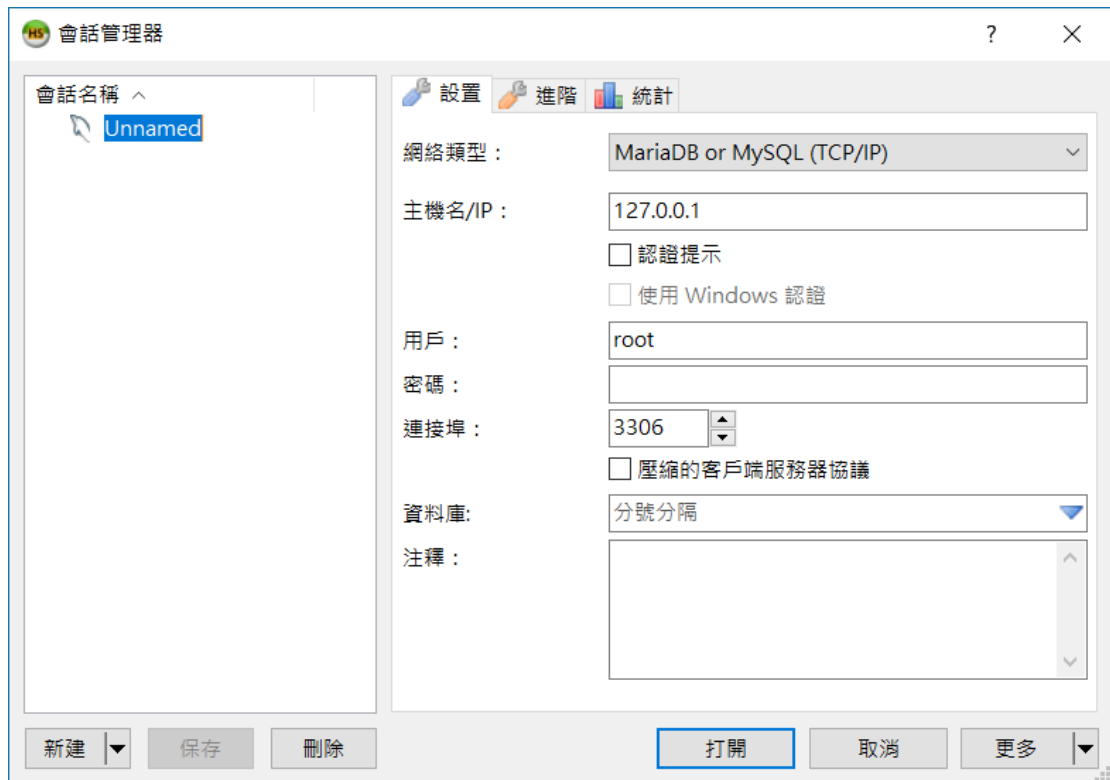
```
net start mysql
```

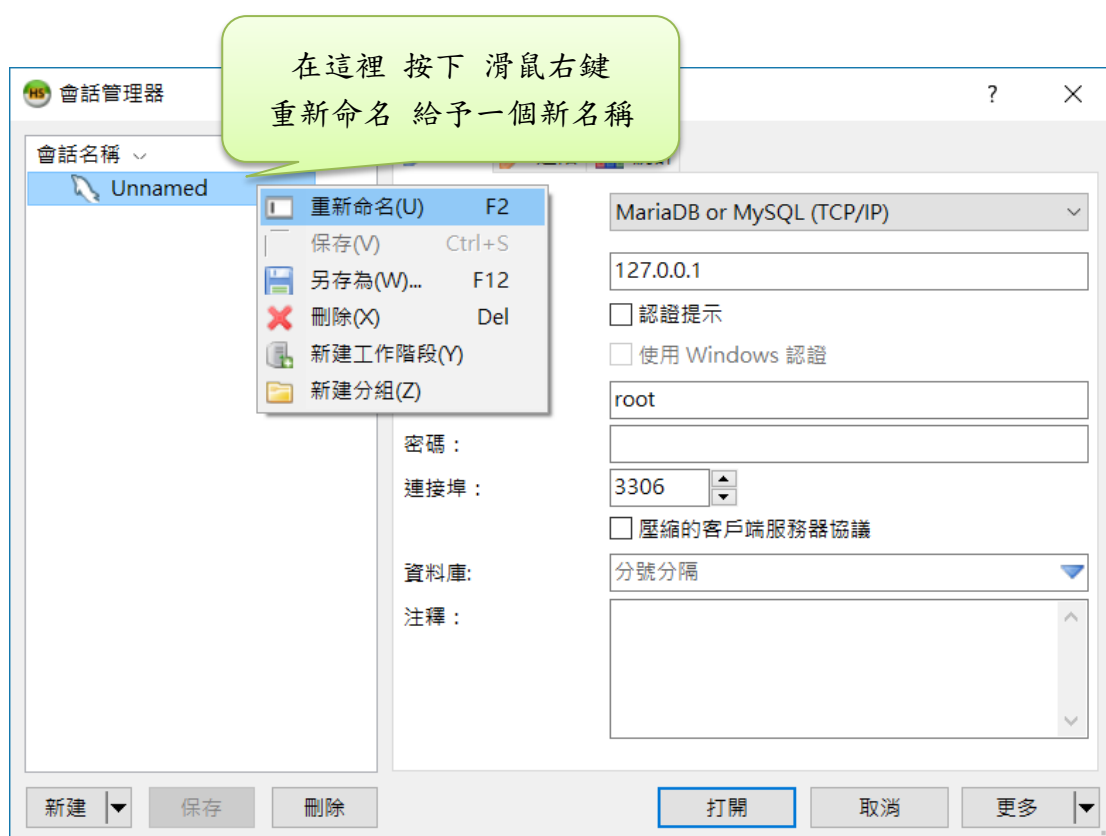
- 停止 MariaDB 資料庫服務 當你長期不用時，建議關閉資料庫服務!

```
net stop mysql
```

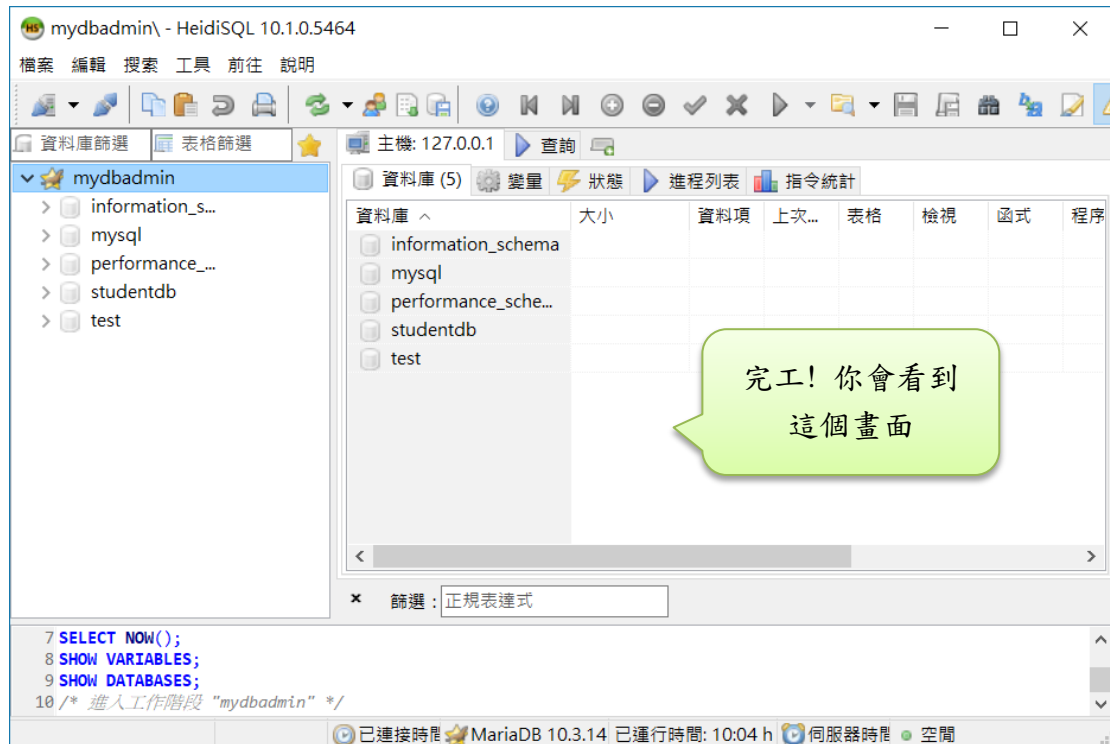


Lab0.1:新增一個資料庫連線管理





都寫好之後，按下
打開



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 x History +

```
1  -- 學生資料表
2  -- DROP Table student;
3  CREATE TABLE student`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)

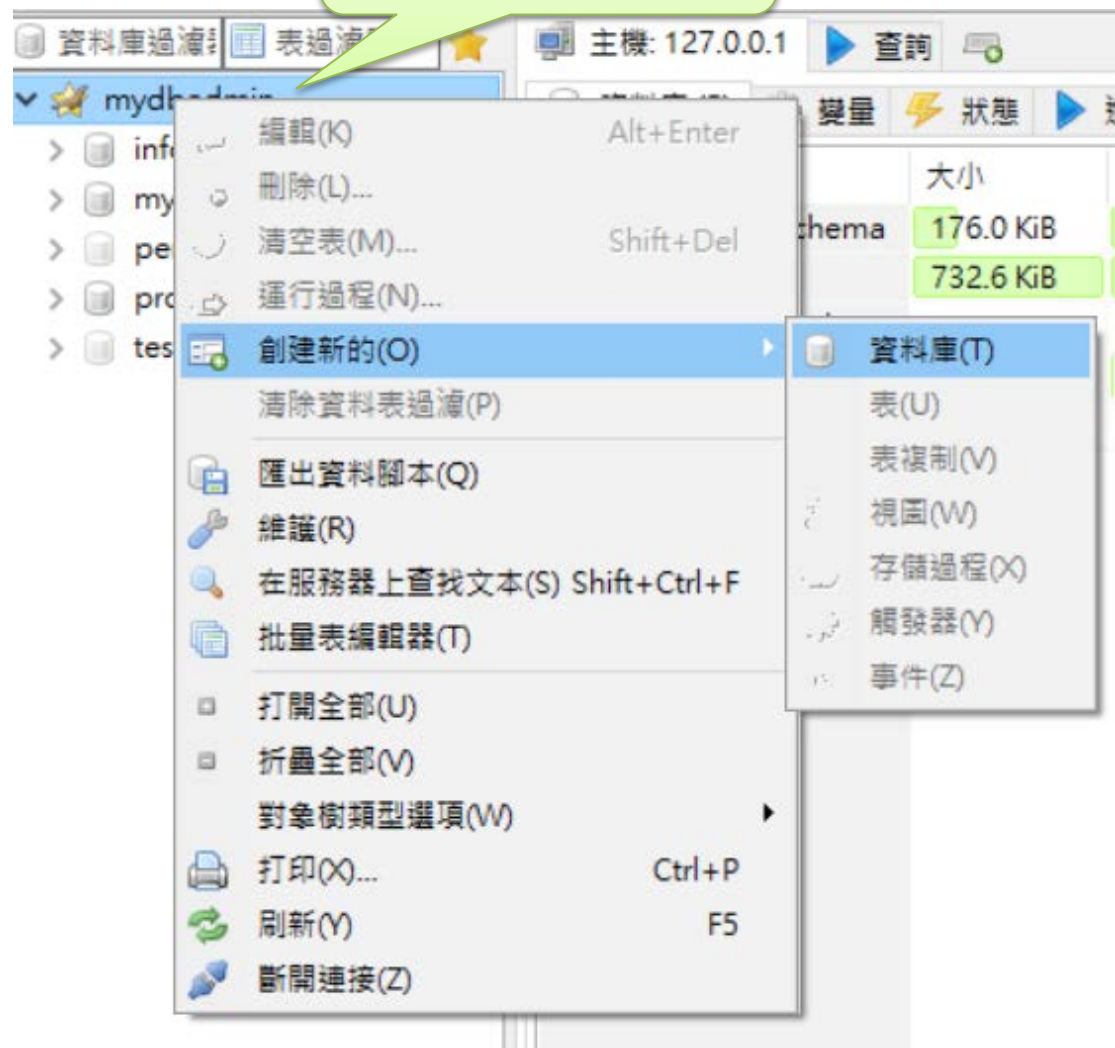
Database: studentdb Table: student

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

在這裡 按下 滑鼠右鍵
創建新資料庫



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
```

給予一個資料庫名稱
studentdb

mydbadmin\ - HeidiSQL 9.4.0.5125

文件 編輯 搜索 工具 Go to 幫助

主機: 127.0.0.1 查詢

資料庫過濾器 表過濾器

mydbadmin

- bookdb 80.0 KiB
- information_schema 176.0 KiB
- mysql
- performance_schema
- studentdb
- test

資料庫 (6)

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

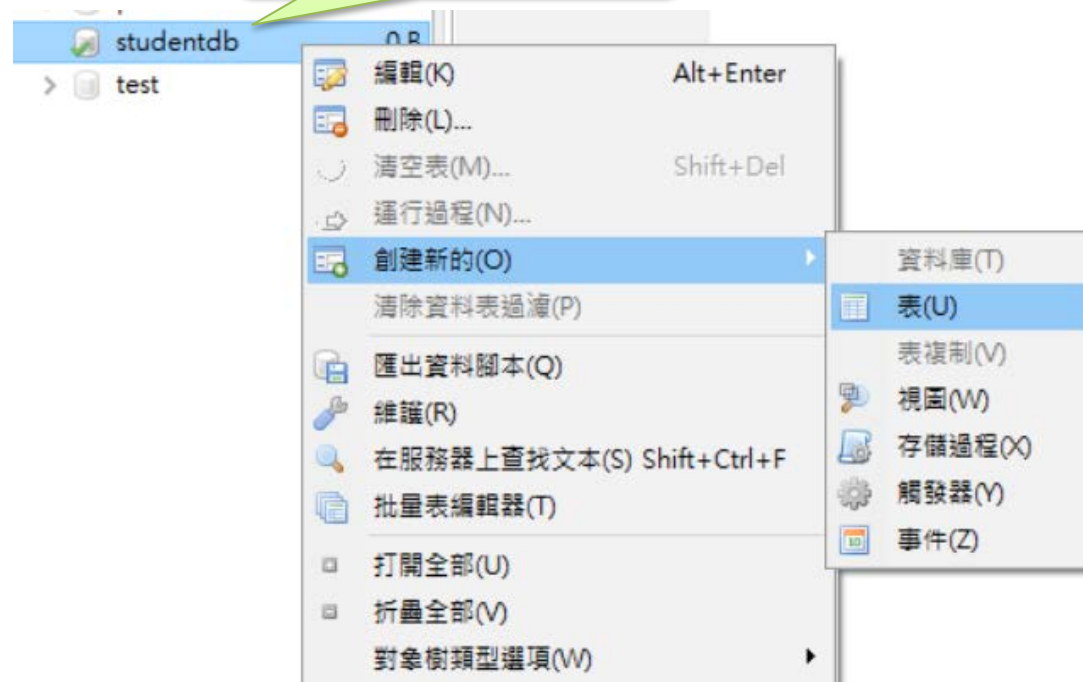
```
25 SHOW CREATE TABLE `bookdb`.`titles`;
26 SHOW COLLATION;
27 SHOW ENGINES;
28 SELECT * FROM `bookdb`.`titles` LIMIT 1000;
29 SHOW CREATE TABLE `bookdb`.`titles`;
```

MariaDB 10.2.14 運行時間: 01:07 h UTC: 2018-04-15 上午 06:1 空間

Lab1:學生資料表新增實作(手動操作)

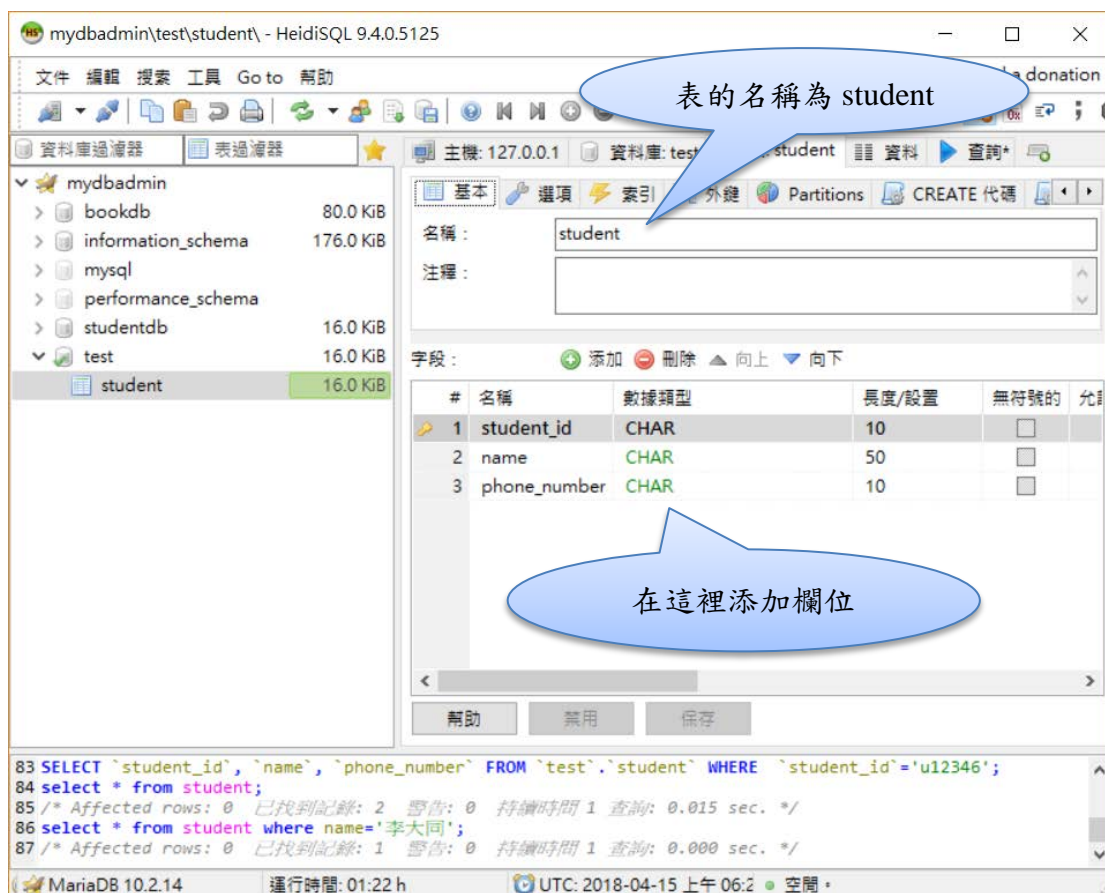
新增資料表

在 studentdb 按下 滑鼠右鍵
創建新資料表



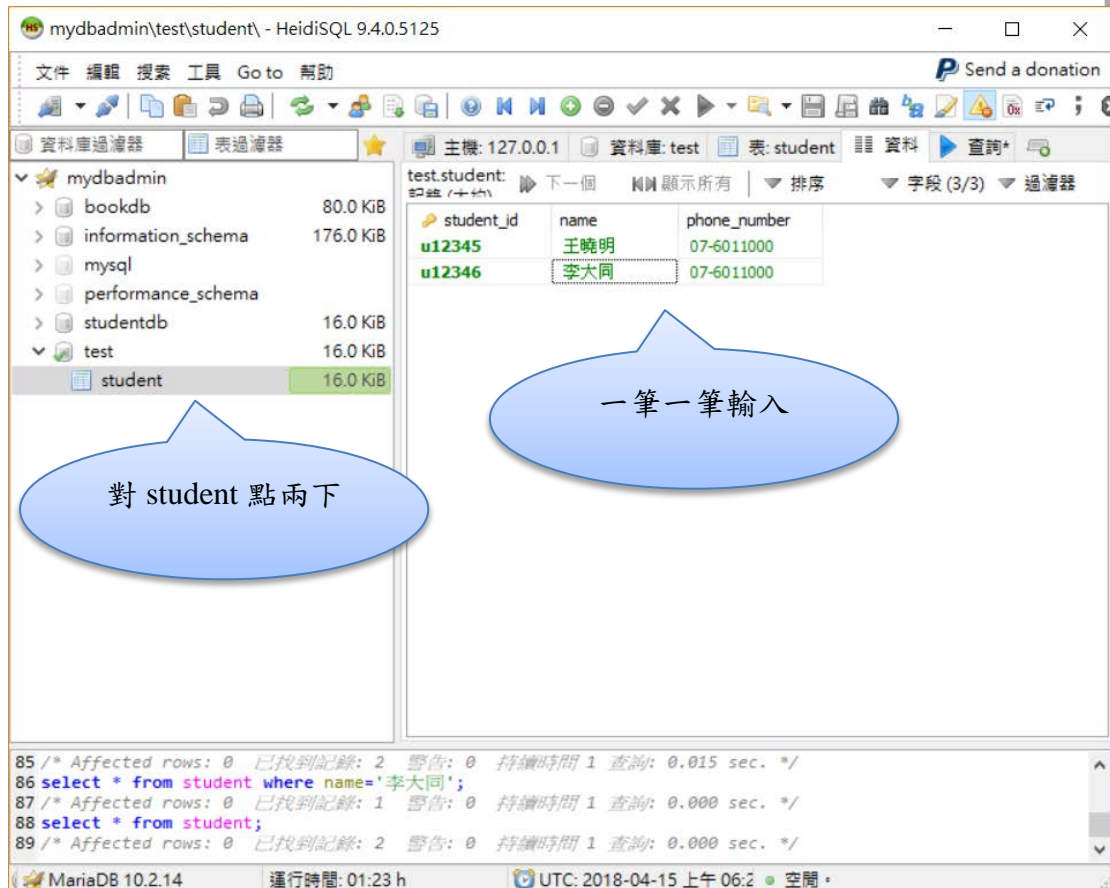
student table 欄位如下:

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



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



操作手冊:

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

Lab1.5: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:學生資料表新增實作(SQL 命令)

請在資料庫管理工具執行以下 SQL 命令

按下 執行

作用在 studentdb 上

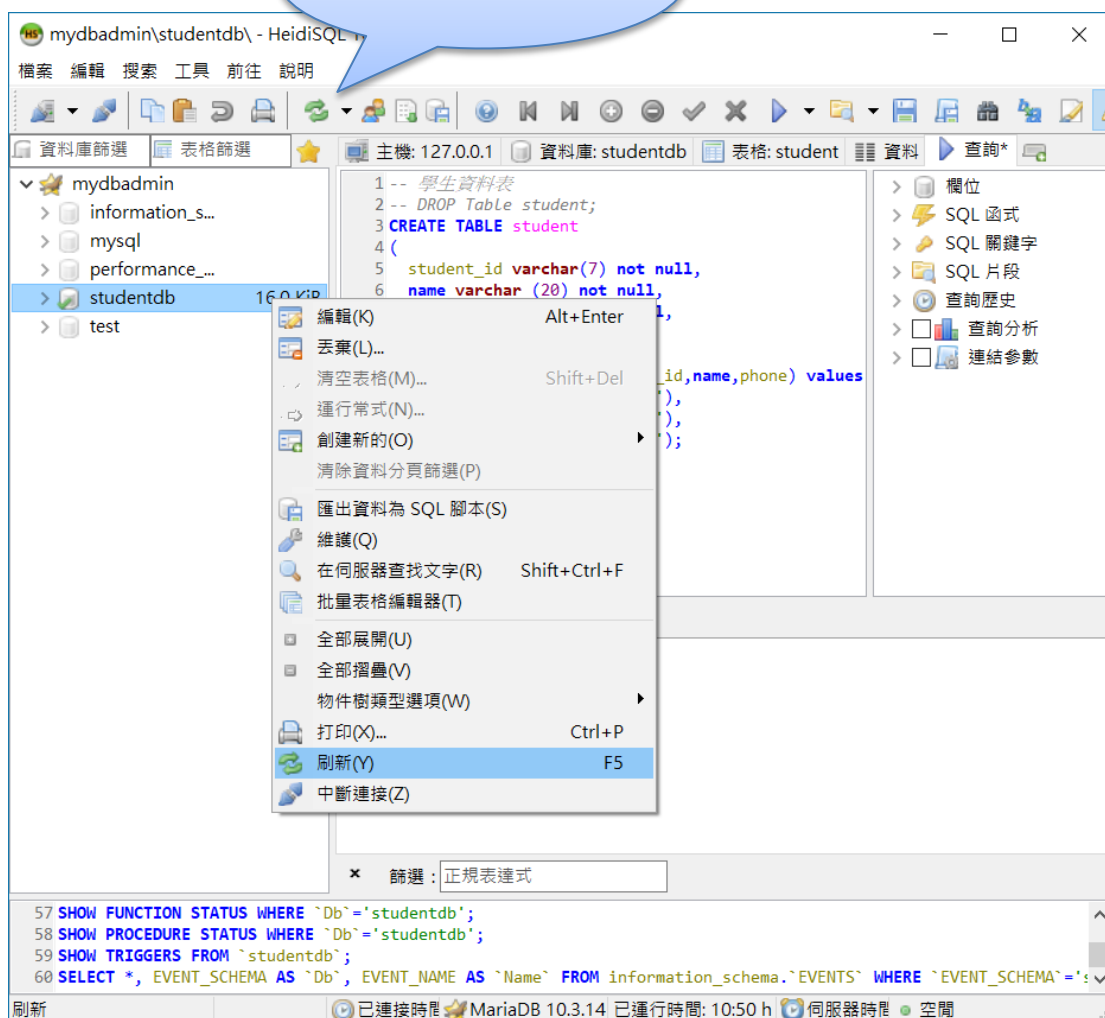
點選查詢，並貼上 SQL 程式碼

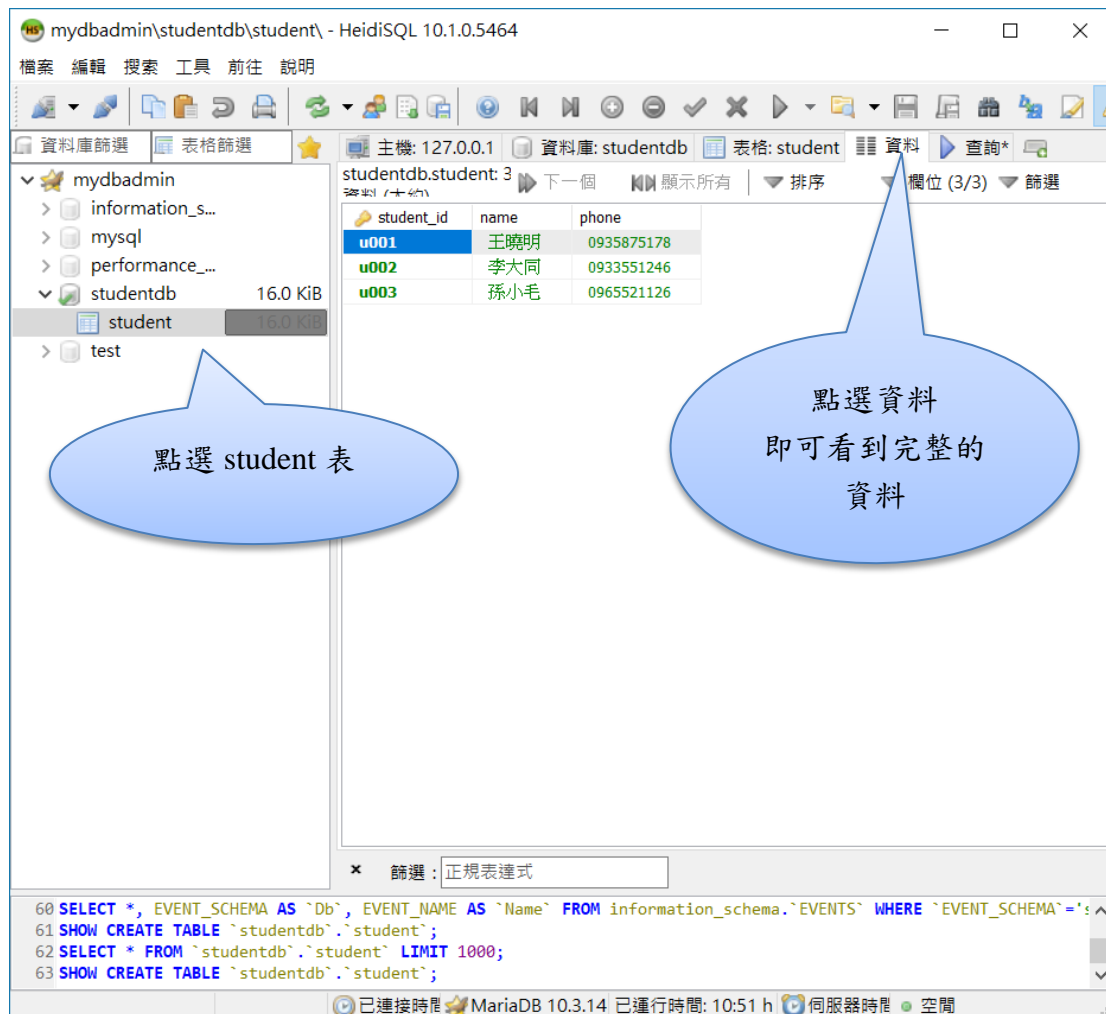
```
1 -- 學生資料表
2 -- DROP Table student;
3 CREATE TABLE student
4 (
5     student_id varchar(7) not null,
6     name varchar(20) not null,
7     phone varchar(10) not null,
8     Primary Key(student_id)
9 );
10 Insert Into student (student_id,name,phone) values
11 ('u001','王曉明','0935875178'),
12 ('u002','李大同','0933551246'),
13 ('u003','孫小毛','0965521126');
14
```

38 /* 未有在樹中發現表格節點「student」。 */
39 /* 進入工作階段 "mydbadmin" */
40 SHOW VARIABLES LIKE 'collation_server';
41 SHOW CREATE DATABASE `studentdb`;

14:1 (395 B) 已連接時間 MariaDB 10.3.14 已運行時間: 10:42 h 伺服器時間 空間

重新整理





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

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`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- 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’);
```

```
DROP Table student;
```

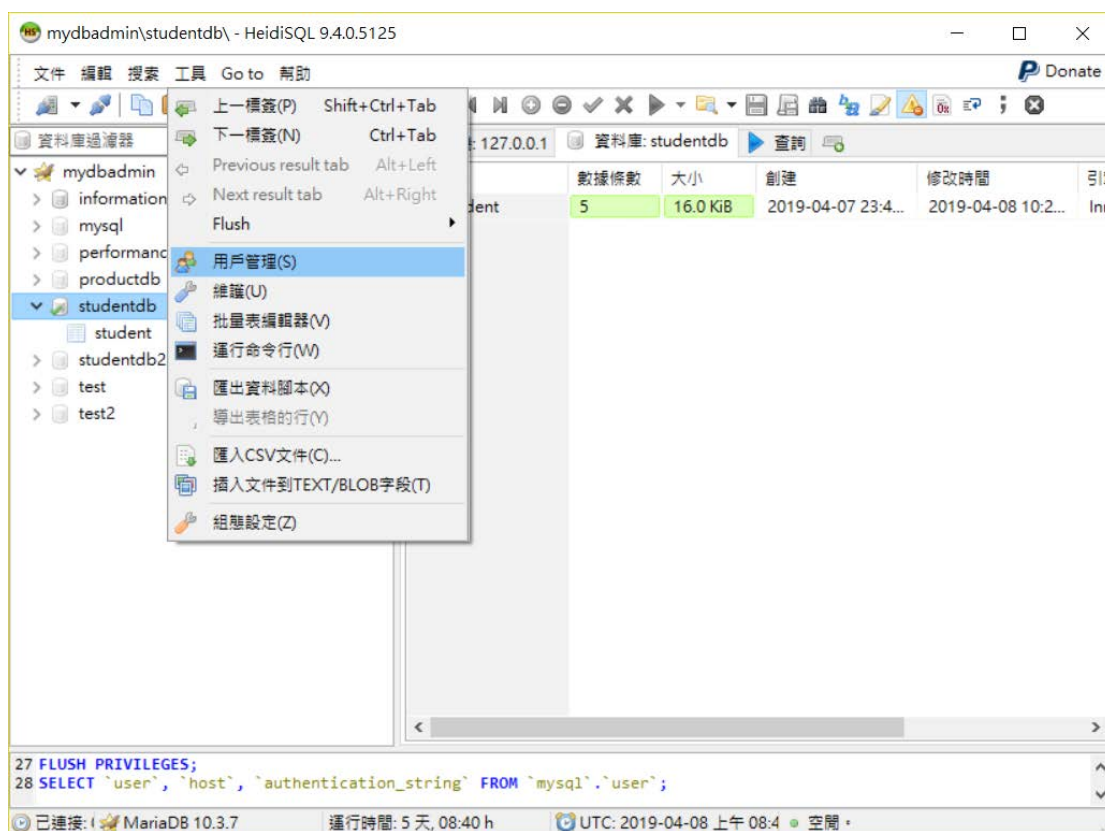
```
CREATE DATABASE IF NOT EXISTS `studentdb2`;
```

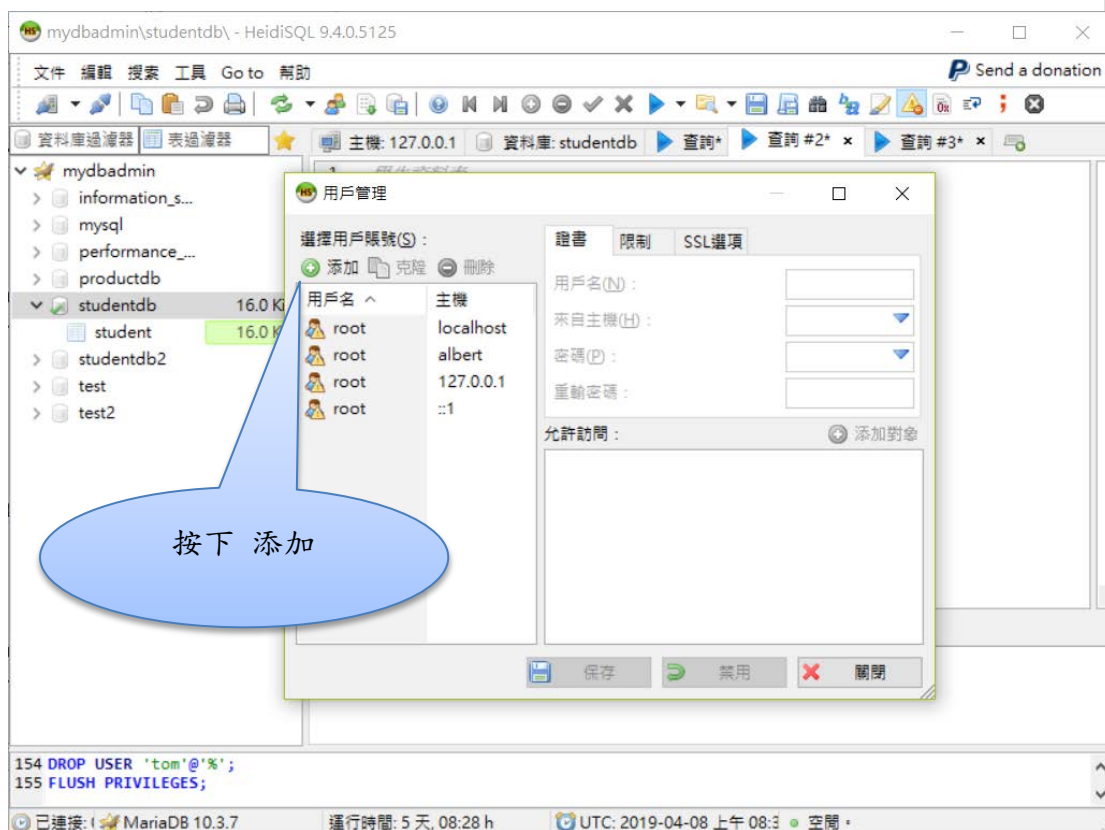
```
USE `studentdb2`;
```

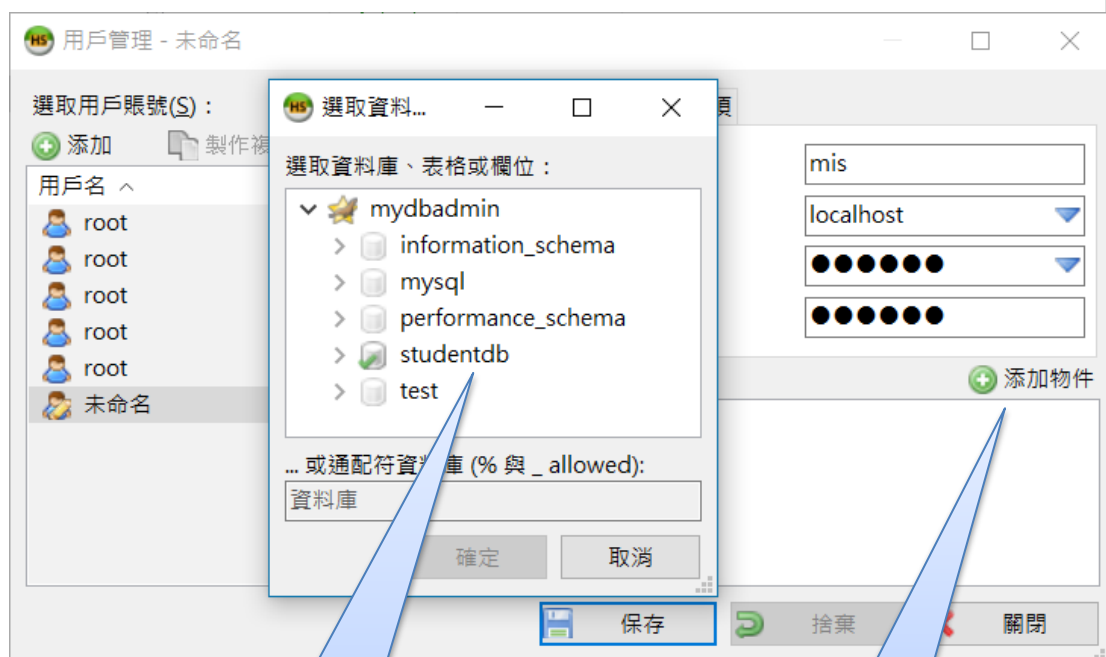

Lab2:新增使用者，並做權限管理

- 新增使用者:mis
- 密碼:mis123

工具 → 用戶管理

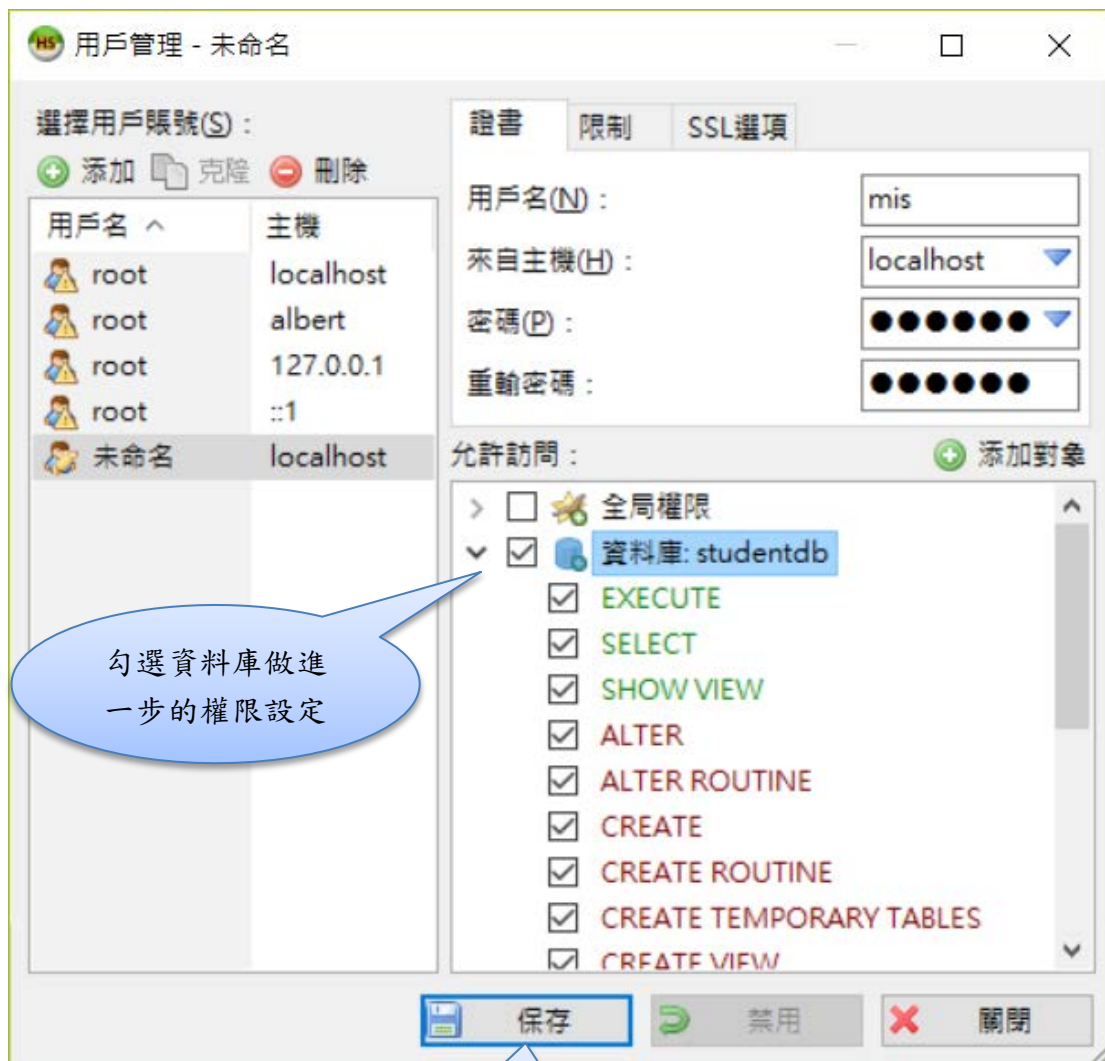






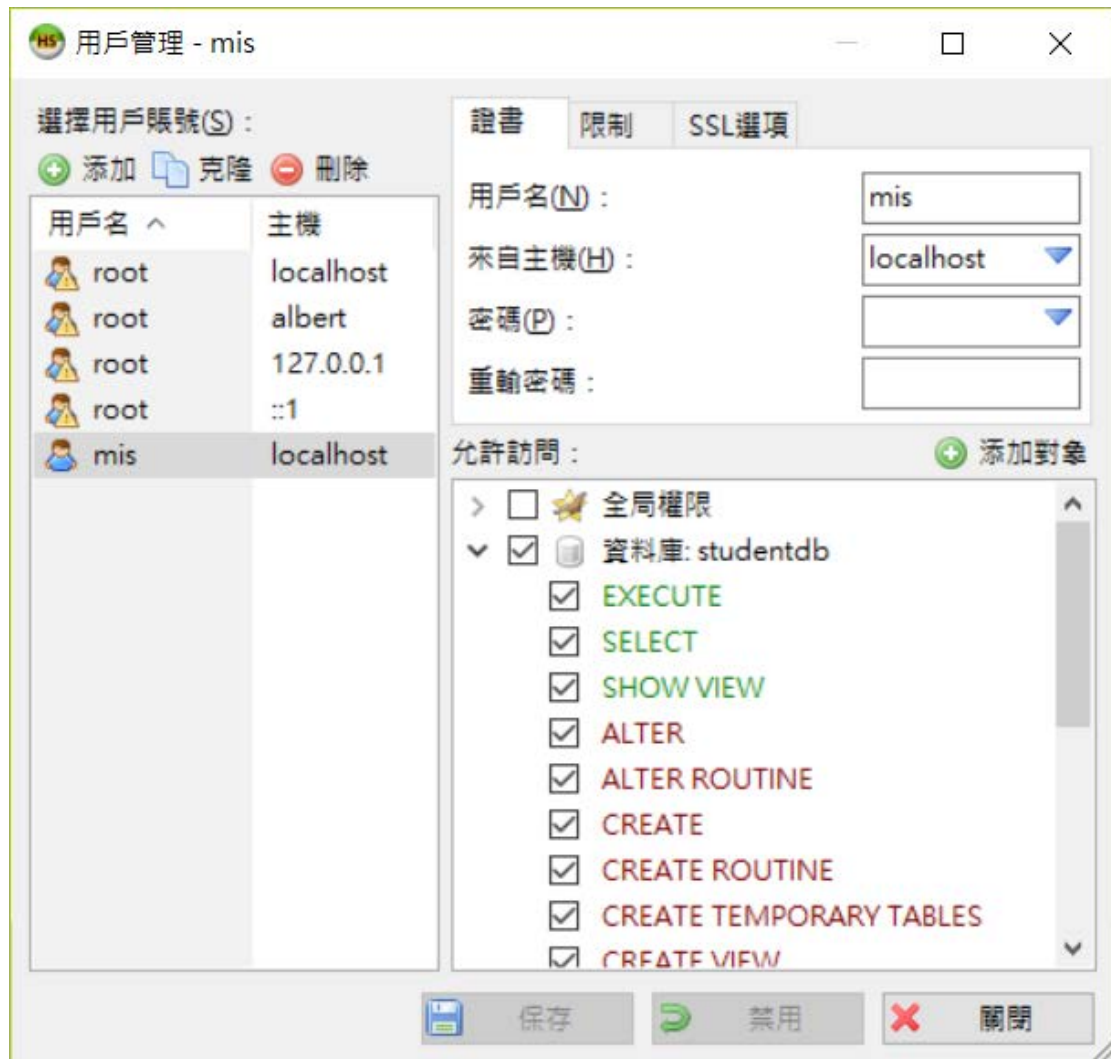
點選 studentdb 資料庫 或其他允許的資料庫

按下 添加物件 或添加對象



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。

Lab3:學生資料庫操作

基本的 SQL command:

--(1) select

select * from student;

select * from student where phone='0965521126';

select * from student where name like '李%';

--(2) insert

Insert Into students (student_id,name,phone) values ('123','孫大毛','0965521126');

--(3) delete

delete from student where student_id = '123';

--(4)update

update students set name = '李曉同' where student_id = '123';

Lab4: 入門介紹 Java 程式去執行 SQL 指令

1. 驅動程式下載(for Java)



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 held liable for any issues you may encounter.

MariaDB Connector/J 2.2.3 Stable

2018-03-14

[View all releases](#)

[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.com/connector-java/2.2.3/>

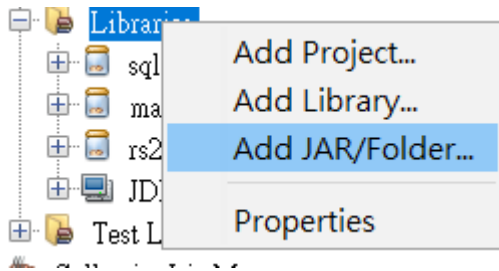
2. 載入驅動程式

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。

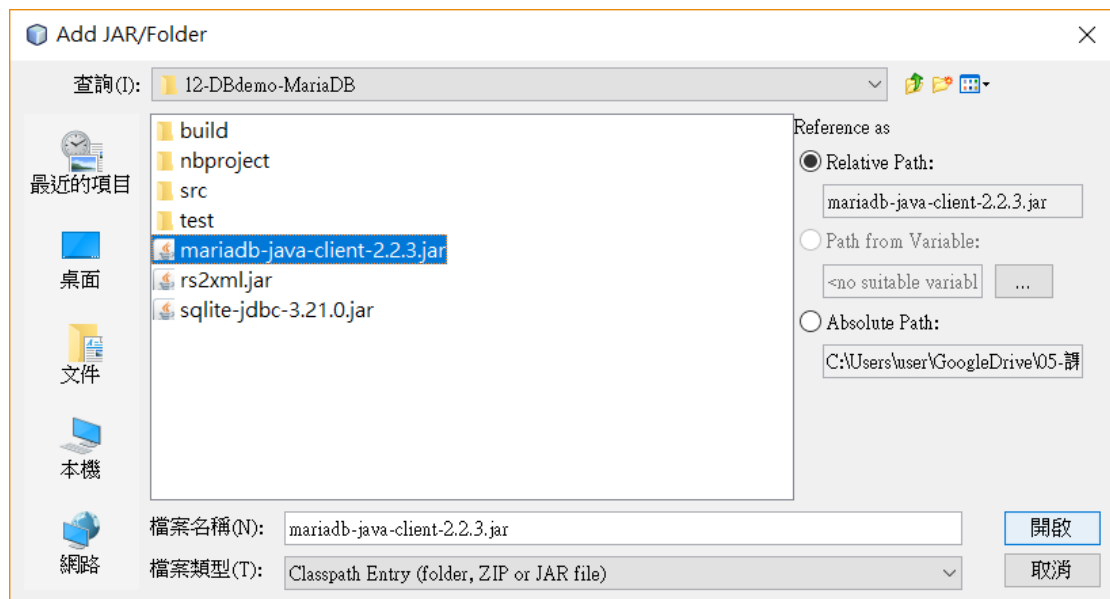
*必須將 MySQL, **MariaDB**, or JavaDB 驅動程式載入才能執行

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

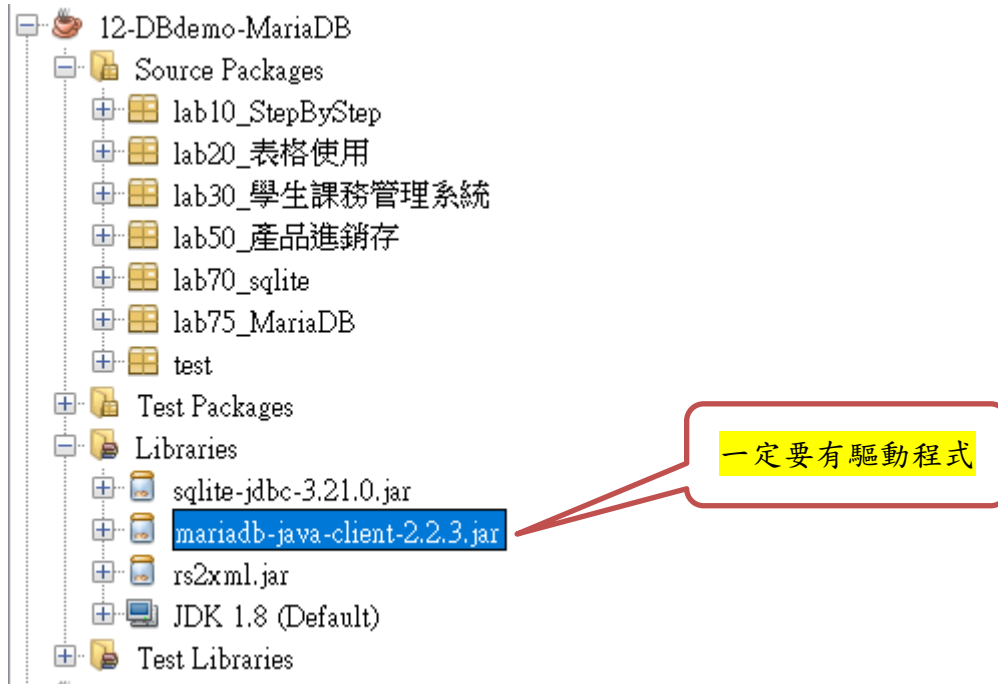
Libraries→按右鍵→加入 jar 檔



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
- **CachedRowSet**—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/jcrowsset.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();
```

Lab4.1: Throws SQLException 版本

<pre>public class StudentDB_15_ThrowException { public static void main(String[] args) throws SQLException { String DB_URL = "jdbc:mariadb://localhost:3306/studentdb"; Connection conn = null; Statement state = null; ResultSet result = null; //連線 getConnection 須給帳號與密碼 conn = DriverManager.getConnection(DB_URL, "mis", "mis123");</pre>
--

	<pre> state = con.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"); while (result.next()) { System.out.printf("學號:%s ", result.getString(1)); //取出字串型態欄位 System.out.printf("姓名:%s ", 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 System.out.printf("電話:%s ", result.getString("phone")); //取出欄位名稱為"phone"的字串 System.out.println(); } //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(); } } } </pre>
--	--

*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

	<pre> public class StudentDB_16_TryCatch { public static void main(String[] args) { String DB_URL = "jdbc:mariadb://localhost:3306/studentdb"; </pre>
--	---

```
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("資料庫操作出問題:" + ex.toString());

}

try {

    result = state.executeQuery("select * from student");

    while (result.next()) {

        System.out.printf("學號:%s ", result.getString(1)); //取出字串型態欄位之內

        System.out.printf("姓名:%s ", result.getObject(2)); //用物件，較通用，可以取 整數 字串

        //各種型態欄位之內容

        System.out.printf("電話:%s ", result.getString(3)); //取出字串型態欄位之內

        System.out.println();

    }

} catch (SQLException ex) {

    System.out.println("資料庫操作出問題:" + ex.toString());

}

}
```