

資料庫環境簡介



教育部新型態資安實務示範課程 發展計畫

屏東大學資料庫安全實務團隊

大綱

- Linux簡介
- 基礎資料庫簡介
 - Maria DB
 - SQL Server
 - HBase
- 進階資料庫簡介
 - Oracle Database
 - IBM DB2
 - Sybase

大綱

- MySQL 安裝、設定與登入
- 資料庫基本操作
- 資料庫進階操作



Linux簡介

Linux簡介 (1/5)

- 簡介：Linux是類Unix 的作業系統，核心是由Linus Torvalds在1991年開發，為遵循「大眾公有版權」(GPL, General Public License)的自由軟體。
- 特色：
 1. 自由開放：由於Linux是以GPL為基礎，所以任何人都可進行使用或修改。
 2. 擁有龐大的GNU軟體，和各地開發者的自行開發軟體。
 3. 易於安裝。
 4. 獨立作業：很多的軟體套件被Linux使用，且也在Linux上進行測試與開發，因此，Linux幾乎可獨立完成工作站或伺服器服務。

Linux簡介 (2/5)

■ 優點：

1. 作業系統穩定：繼承Unix架構的穩定性、高效率。
2. 漏洞修補快速：由於有支援者眾多，當出現漏洞時，可盡速修正。
3. 多工、多使用者：允許同時多人多工，主機可劃分不同階級的User，每位User登入系統的工作環境可不同，且不同User可同時登入主機，並使用主機資源。
4. 適合小核心的崙入式系統：由於Linux只需要幾百KB就能立一個OS，所以適合運用在家電、行動裝置、物聯網(IoT)上。

Linux簡介 (3/5)

■ 缺點：

1. 無特定支援廠商：因為Linux上的套件幾乎是自由軟體，其開發者不一定是公司或團體，若軟體發生問題，就得上網找答案。
2. 專業軟體支援不足：目前多數的專業繪圖軟體不支援Linux系統。
3. 不夠大眾化：由於市售電腦搭載的OS多為Windows，所以若非特殊需求，一般大眾並不會使用到Linux作業系統。

Linux介紹：http://neuron.csie.ntust.edu.tw/homework/94/os/homework/homework1/A9415061_2/linux.htm

Linux特色：<http://web.cjcu.edu.tw/~jchen/course/introcomputer/lecture/Unix&Linux.htm>

Linux優缺點：http://linux.vbird.org/linux_basic/0110whatislunix/0110whatislunix-fc4.php#linuxpt_adv

Linux簡介 (4/5)

- 目前Linux的發行版本有需多種類，舉凡Red Hat、Mandriva、Centos、Fedora、Debian、Gentoo、Ubuntu、OpenSUSE等。
- 這些版本都各有優缺點，以及目標客群，好比說Red Hat適合架設伺服器，但取得這項功能需要收費；而對於喜好嘗試新鮮事物的使用者，Fedora 就最具吸引力，因為它會將最新版，但尚未穩定的軟體包入發行。

Linux簡介 (5/5)

- Linux發行版本中，Ubuntu是較適合一般使用者初次使用的，因為其的優點有：
 1. 簡單、易上手。
 2. 有自己的Unity介面。
 3. 套件庫有分類、簡介、評價，讓User能快速找到需要的套件。
- 因此在本次課程中，我們將使用Ubuntu進行實作演練。



基礎資料庫簡介

MariaDB (1/3)

- **簡介：** 遵循GPL，並由MySQL分支的開源關聯式資料庫管理系統(RDBMS)，因此MySQL 5.5版以前可無痛轉移，5.6以後的版本，排除不相容的新功能，其餘也可輕易轉移。具有很強的可擴充套件特性，使資料處理快速，且其對於少量的資料管理快速、平穩，適合小型企業或個人專案。
- **特色：**
 1. API和協定與MySQL相容。
 2. 雖替換掉MySQL中幾項關鍵組件，但盡量保持其相容性。
 3. 使用Aria為儲存引擎，可作為事務式和非事務式引擎。

MariaDB (2/3)

■ 優點：

1. 可擴展性高。
2. 能即時訪問。
3. 具備MySQL核心功能。
4. MariaDB發展與升級速度遠勝MySQL，但對於安全性也有一定的重視。

MariaDB (3/3)

■ 缺點：

1. 由於IDX日誌檔在長期使用下，會使得檔案變大，以致拖累性能表現。
2. MariaDB的快取功能並沒有如想像中快速。
3. MariaDB與MySQL並非全部相容，轉移時還是需要搭配其他兼容工作。

MariaDB介紹：<https://twoss.gitbook.io/open-source-use-case/zi-liao-ku/mariadb>

MariaDB特色：<https://zh.wikipedia.org/wiki/MariaDB>

MariaDB優缺點：<https://tpu.thinkpower.com.tw/tpu/articleDetails/916>

SQL Server (1/2)

- **簡介：**由微軟公司與Sybase公司合作研發並推出的**關聯式資料庫系統(RDBMS)**，目前免費版本適用於個人網站或小型簡易網站使用，具商業性及進階網站，還是建議使用付費的商業性授權版本。
- **特色：**SQL Server最大的特色在於，它整合了微軟的各項產品及資源、強大的視覺化界面、高度整合的管理開發工具，且在快速建構商業智慧(BI)方面具有成就。

SQL Server介紹：https://zh.wikipedia.org/wiki/Microsoft_SQL_Server

SQL Server特色：<https://ithelp.ithome.com.tw/articles/10206222>

SQL Server優缺點：<https://www.webdesigns.com.tw/SQLServerAndMySQL.asp>

SQL Server (2/2)

■ 優點：

1. 與微軟相關開發工具整合性好。
2. 新手操作容易，並自帶資料庫介面操作工具。
3. 安全交易與控制嚴格。

■ 缺點：

1. 使用與開發環境封閉：屬於原始碼封閉的開發環境，且討論度不高，雖然有推出在Linux運行的版本，但較穩定的商務版依舊限定Windows為主。
2. 容易造成伺服器負擔：由於微軟對於DB技術不是主力，因此當稍微龐大、需要多資源的資料庫引擎時，會使得伺服器資源負載過大。

HBase (1/2)

- **簡介：**以 Hadoop 分散式檔案系統 (Hadoop Distributed File System, HDFS) 為檔案儲存系統，Hadoop MapReduce 處理大量資料，利用 Zookeeper 為分散式協同服務管理，用來儲存非結構和半結構式的分散資料。
- **特色：**擁有高吞吐量、低延遲的特點，並能即時提供隨機讀寫 (random read) 的功能，目前有 Facebook、Twitter、趨勢科技 (Trend Micro) 等企業與網站使用。

HBase (2/2)

■ 優點：

1. 儲存容量大。
2. 列可動態增加資料，並當列為Null就不儲存資料，以節省儲存空間。
3. 自動分割資料，使資料儲存具水平可擴展性。

■ 缺點：

1. 資料分析較弱。
2. 不適合大範圍掃描查詢。
3. 不直接支援SQL語法查詢。

HBase介紹：<http://www.htsjk.com/hbase/35256.html>
<https://www.itread01.com/content/1546602691.html>

HBase特色：<https://www.twblogs.net/a/5c362b2fbd9eee35b3a57cf7>

HBase優缺點：<https://kknews.cc/code/mexljm6.html>



進階資料庫簡介

Oracle Database (1/3)

- **簡介：**Oracle Database 是甲骨文公司推出的資料庫系統，基本上，是純粹針對公司行號等大型公司推出的資料庫。
- **特色：**歷史悠久、有提供雲端系統架構的產品、廣泛被大型企業使用。

Oracle Database (2/3)

■ 優點：

1. 極高的安全性：由於針對大型公司設計的緣故，在安全性機制的設定上，具有最高認證級別的**ISO**標準認證。
2. 效能出色：在從事各種大型運算，或查詢上，是經過 **TPC** 組織(專門執行商用負載)的認證，效能可以說是贏過全主流資料庫系統。
3. 功能齊全、強大：具有多種資料儲存格式可以選擇，還有各式能夠自行制定的功能，例如同義詞、套件等等不同的模塊。

Oracle Database (3/3)

■ 缺點：

1. 使用成本昂貴：基本上免費版是不太能用的，因此大部分都是在公司使用付費版，而付費版的價格較為高昂，相對應的硬體設施也有一定的要求。
2. 學習成本高：Oracle 是一個比較有技術含量的資料庫系統，對於新手來說，要快速學會核心的應用比較吃力些，需要較為Senior的技術開發者協助。

IBM DB2 (1/3)

- **簡介：**由IBM公司推出，是一種**混合式資料管理系統**，設計用於管理內部部署私有雲和公有雲環境中的結構化及非結構化資料。**DB2** 是專門提供擴充性和靈活性而設計的智慧型一般 **SQL**引擎。
- **特色：**近期發展利用機器學習評分的**SQL**查詢來獲得可信賴的查詢結果，強調由成本優化工具、機器學習優化工具及調適性工作負載管理來推動高效能。綜合企業安全、簡化安裝和部署、更佳的可用性和採用、流暢的升級程序、大型資料庫的加強功能。

IBM DB2 (2/3)

■ 優點：

1. 發展AI導入產品技術：透過開發AI技術來讓客戶利用優化後的工具提升整體資料庫效能。
2. 直欄組織表格的高階功能：引進 **BLU Acceleration** 核心技術的重大加強功能。這些加強功能包括「巢狀迴圈結合(NLJN)」支援、進階解除相關性方法、更快速的 **SQL MERGE**、記憶體管理的加強功能、進一步 **SIMD** 高階功能、業界領先平行排序，以及提升的 **SQL** 平行化。

IBM DB2 (3/3)

■ 缺點：

1. 需要的效能較多：內存鎖的使用在提升效率的同時也對系統的優化要求提到了最高。
2. 使用者的學習：DB2中的高可用性對於普通用戶來說可能比較復雜，但是IBM有一些經典的技術文檔可以用來參照幫助用戶自己設置。

IBM DB2簡介：<https://www.ibm.com/tw-zh/analytics/db2>

IBM DB2特色：https://www.ibm.com/support/knowledgecenter/zh-tw/SSEPGG_11.1.0/com.ibm.db2.luw.wn.doc/doc/c0060311.html

IBM DB2優缺點：<http://www.aspphp.online/shujuku/db2sjk/db2rm/201701/225470.html>

Sybase (1/3)

- **簡介：** Sybase隸屬SAP公司，一直擁有各行業中領導全球經濟的廣大客戶群，並特別專注於金融服務、電信與媒體、醫療產業及政府機構等市場。
- **特色：** 領先的企業門戶產品以及移動與無線解決方案，還致力於整合各種應用平臺、資料庫和應用軟體，並作為全球最大的獨立軟體廠商之一。

Sybase (2/3)

■ 優點：

1. 客戶/伺服器體繫結構的資料庫：它支持共用資源且在多台設備間平衡負載，允許容納多個主機的環境，充分利用了企業已有的各種系統。
2. 真正開放的資料庫：單地提供了預編譯，且公開了應用程式介面**DB-LIB**，鼓勵第三方編寫**DB-LIB**介面。
3. 高性能的資料庫：不讓操作系統來管理進程，把與資料庫的連接當作自己的一部分來管理。其引擎還代替操作系統來管理一部分硬體資源，如埠、記憶體、硬碟，繞過了操作系統這一環節，提高了性能。

Sybase (3/3)

■ 缺點：

1. Sybase技術支援並不完善、使用者頁面讓操作者不好使用、學習進入不易，教學較少。

Sybase簡介、特色：https://www.mpinfo.com.tw/products_1_detail2.php?BrandID=4&pType=4

Sybase優缺點：<https://bbs.csdn.net/topics/30322145>



MySQL 安裝、設定與登入

MySQL安裝、設定與登入 (1/7)

- 進行mariadb-server的安裝，指令完成後若出現下圖畫面即為安裝成功。

sudo apt install -y mariadb-server mariadb-client

```
boling@ling-VirtualBox:~$ sudo apt install -y mariadb-server mariadb-client
[sudo] password for boling:
正在讀取套件清單... 完成
正在重建相依關係
正在讀取狀態資料... 完成
下列【新】套件將會被安裝：
  mariadb-client mariadb-server
升級 0 個，新安裝 2 個，移除 0 個，有 180 個未被升級。
需要下載 25.7 kB 的套件檔。
此操作完成之後，會多佔用 131 kB 的磁碟空間。
下載:1 http://tw.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 mariadb-client all 1:10
.1.38-0ubuntu0.18.04.1 [12.8 kB]
下載:2 http://tw.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 mariadb-server all 1:10
.1.38-0ubuntu0.18.04.1 [12.9 kB]
取得 25.7 kB 用了 0秒 (169 kB/s)
選取了原先未選的套件 mariadb-client。
(讀取資料庫 ... 目前共安裝了 162142 個檔案和目錄。)
準備解開 .../mariadb-client_1%3a10.1.38-0ubuntu0.18.04.1_all.deb ...
解開 mariadb-client (1:10.1.38-0ubuntu0.18.04.1) 中...
選取了原先未選的套件 mariadb-server。
準備解開 .../mariadb-server_1%3a10.1.38-0ubuntu0.18.04.1_all.deb ...
解開 mariadb-server (1:10.1.38-0ubuntu0.18.04.1) 中...
設定 mariadb-client (1:10.1.38-0ubuntu0.18.04.1) ...
設定 mariadb-server (1:10.1.38-0ubuntu0.18.04.1) ...
boling@ling-VirtualBox:~$
```

Ps.若在安裝時出現資源暫時無法取得的狀況，直接將其刪除(rm)即可

```
user@pc:~$ sudo apt install -y mariadb-server mariadb-client
E: 無法將 /var/lib/dpkg/lock 鎖定 - open (11: 資源暫時無法取得)
E: Unable to lock the administration directory (/var/lib/dpkg/), is another proc
ess using it?
user@pc:~$ sudo rm /var/lib/dpkg/lock
user@pc:~$ sudo apt install -y mariadb-server mariadb-client
```

MySQL安裝、設定與登入 (2/7)

- 安裝及設定環境完成後，先將關閉mariadb，再重新開啟。

```
sudo systemctl stop mariadb.service
```

```
sudo systemctl start mariadb.service
```

- 設定mariadb在每次開機後即自動啟動。

```
sudo systemctl enable mariadb.service
```

- 確認mariadb是否有正常啟動與運作，若正常運作即會出現如圖中Active: active(running)。

```
sudo systemctl status mariadb.service
```

```
boling@ling-VirtualBox:~$ sudo systemctl stop mariadb.service
boling@ling-VirtualBox:~$ sudo systemctl start mariadb.service
boling@ling-VirtualBox:~$ sudo systemctl enable mariadb.service
● mariadb.service - MariaDB 10.1.38 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2019-05-01 21:56:02 CST; 8s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 10417 (mysqld)
    Status: "Taking your SQL requests now..."
     Tasks: 27 (limit: 4915)
   CGroup: /system.slice/mariadb.service
           └─10417 /usr/sbin/mysqld
```

MySQL安裝、設定與登入 (3/7)

- 為使Mysql安全性提高因此進行安全設定。
`sudo mysql_secure_installation`
- 第一次設定按Enter即可，接著完成密碼設定。
Enter current password for root (enter for none):
Set root password?[Y/n]: Y
New password:
Re-enter new password:

```
boling@ling-VirtualBox:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

MySQL安裝、設定與登入 (4/7)

- 移除匿名用戶。

Remove anonymous users? [Y/n] Y

- 關閉root遠端登入。

Disallow root login remotely? [Y/n] Y

- 不移除資料表。

Remove test database and access to it? [Y/n] n

MySQL安裝、設定與登入 (5/7)

- 重新載入資料表的權限。
Reload privilege tables now? [Y/n] Y
- 完成以上步驟後重啟MariaDB服務。

```
Remove anonymous users? [Y/n] Y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] n
... skipping.

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
boling@ling-VirtualBox:~$ sudo systemctl restart mariadb.service
```

MySQL安裝、設定與登入 (6/7)

- -u表示以使用者登入，root代表使用者名稱，-p表示密碼的參數如果不想密碼被看到可以直接-p後按Enter將會在下一行要求使用者輸入密碼，並有遮蔽密碼的效果。

`sudo mysql -u root -p`

```
boling@ling-VirtualBox:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.1.38-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

MySQL安裝、設定與登入 (7/7)

- 另外，MariaDB的設定檔位置在 /etc/mysql/my.cnf 下，此設定檔中我們可以更改目錄資料與語系、記憶體配置、斷線時間設定等，下圖為MariaDB的設定檔的內容。

cat /etc/mysql/my.cnf

```
# The MariaDB configuration file
#
# The MariaDB/MySQL tools read configuration files in the following order:
# 1. "/etc/mysql/mariadb.cnf" (this file) to set global defaults,
# 2. "/etc/mysql/conf.d/*.cnf" to set global options.
# 3. "/etc/mysql/mariadb.conf.d/*.cnf" to set MariaDB-only options.
# 4. "~/.my.cnf" to set user-specific options.
#
# If the same option is defined multiple times, the last one will apply.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# This group is read both both by the client and the server
# use it for options that affect everything
#
[client-server]

# Import all .cnf files from configuration directory
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mariadb.conf.d/
```



資料庫與資料表基本操作

資料庫與資料表基本操作 (1/16)

- 建立資料庫：為了在後續介紹資料表的應用，因此我們須先建立一個資料庫，新增一個名叫 securitytest 的資料庫。

`create database securitytest;`

- 顯示所有資料庫資訊，確認資料庫建立成功

○

`show databases;`

```
MariaDB [(none)]> create database securitytest;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| securitytest |
+-----+
4 rows in set (0.00 sec)
```

資料庫與資料表基本操作 (2/16)

- 使用securitytest，意指之後的指令都在這個資料庫執行，其他不會動到，出現Database changed代表已經成功指定在securitytest資料庫下動作。

use securitytest;

```
MariaDB [(none)]> use securitytest;  
Database changed  
MariaDB [securitytest]>
```

資料庫與資料表基本操作 (3/16)

- **建立資料表 (CREATE TABLE) :** 使用securitytest資料庫，資料表才能建立在此當中。

`use securitytest;`

- 建立資料表並且設置當中所需要的欄位型態。

```
create table studata(  
    stu_Id varchar(5) not null default '00000',  
    stu_Name varchar(20) not null default '',  
    stu_Sex varchar(2) default 'M',  
    stu_Tel varchar(10),  
    stu_Mail varchar(50) default 'unknow',  
    primary key(stu_Id));
```

資料庫與資料表基本操作 (4/16)

```
MariaDB [securitytest]> create table studata(  
-> stu_Id varchar(5) not null default '00000',  
-> stu_Name varchar(20) not null default '',  
-> stu_Sex varchar(2) default 'M',  
-> stu_Tel varchar(10),  
-> stu_Mail varchar(50) default 'unknow',  
-> primary key(stu_Id));  
Query OK, 0 rows affected (0.04 sec)
```


資料庫與資料表基本操作 (5/16)

- 檢視資料表(**DESCRIBE TABLES**)：確認所建立的資料表是否成功，以及顯示剛剛所建立的studata資料表結構。

show tables;

describe studata;

```
MariaDB [securitytest]> show tables;
```

```
+-----+  
| Tables_in_securitytest |  
+-----+  
| studata                 |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
MariaDB [securitytest]> describe studata;
```

```
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| stu_Id     | varchar(5)    | NO   | PRI | 00000   |       |  
| stu_Name   | varchar(20)   | NO   |     |         |       |  
| stu_Sex    | varchar(2)    | YES  |     | M       |       |  
| stu_Tel    | varchar(10)   | YES  |     | NULL    |       |  
| stu_Mail   | varchar(50)   | YES  |     | unknow  |       |  
+-----+-----+-----+-----+-----+-----+
```

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

資料庫與資料表基本操作 (6/16)

- **刪除資料表 (DROP TABLES)** : 為了示範刪除資料表的語法，所以須先建立一個名為hellolinux的資料表。

```
create table hellolinux(user_ID int not null  
default '000');
```

```
MariaDB [securitytest]> create table hellolinux(user_ID int not null default '000');  
Query OK, 0 rows affected (0.06 sec)
```

資料庫與資料表基本操作 (7/16)

- 接下來，使用drop tables的語法做刪除資料表hellolinux，並確認是否成功刪除。

```
drop tables hellolinux;  
show tables;
```

```
MariaDB [securitytest]> drop tables hellolinux;  
Query OK, 0 rows affected (0.05 sec)  
  
MariaDB [securitytest]> show tables;  
+-----+  
| Tables_in_securitytest |  
+-----+  
| studata                 |  
+-----+  
1 row in set (0.00 sec)
```

資料庫與資料表基本操作 (8/16)

- 新增資料(INSERT)：新增三筆資料至 studata資料表中。

insert into studata values

(1,'Jack','M','0975730800','nptu123@mail.nptu.edu.tw'),

(2,'Cindy','F',null,'cindy900@mail.nptu.edu.tw'),

(3,'David',default,0978654334,'abc123@mail.nptu.edu.tw');

```
MariaDB [securitytest]> insert into studata values
->      (1,'Jack','M','0975730800','nptu123@mail.nptu.edu.tw'),
->      (2,'Cindy','F',null,'cindy900@mail.nptu.edu.tw'),
->      (3,'David',default,0978654334,'abc123@mail.nptu.edu.tw');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

資料庫與資料表基本操作 (9/16)

- 檢視資料(**SELECT**)：顯示出studata資料表中所有資料。

```
select * from studata;
```

```
MariaDB [securitytest]> select * from studata;
+-----+-----+-----+-----+-----+
| stu_Id | stu_Name | stu_Sex | stu_Tel   | stu_Mail                |
+-----+-----+-----+-----+-----+
| 1      | Jack     | M       | 0975730800 | nptu123@mail.nptu.edu.tw |
| 2      | Cindy    | F       | NULL       | cindy900@mail.nptu.edu.tw |
| 3      | David    | M       | 978654334  | abc123@mail.nptu.edu.tw  |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

資料庫與資料表基本操作 (10/16)

- 新增資料表欄位：新增updatetime 欄位作為紀錄資料被更新時間。

```
alter table studata add column updatetime  
timestamp default current_timestamp on update  
current_timestamp;
```

```
MariaDB [securitytest]> alter table studata add column updatetime timestamp default current_timestamp  
on update current_timestamp;  
Query OK, 0 rows affected (0.11 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

資料庫與資料表基本操作 (11/16)

- 確認是否有成功新增updatetime 欄位，並且查看資料。

```
select * from studata;
```

```
MariaDB [securitytest]> select * from studata;
```

stu_Id	stu_Name	stu_Sex	stu_Tel	stu_Mail	updatetime
1	Jack	M	0975730800	nptu123@mail.nptu.edu.tw	2019-11-09 02:34:44
2	Cindy	F	NULL	cindy900@mail.nptu.edu.tw	2019-11-09 02:34:44
3	David	M	978654334	abc123@mail.nptu.edu.tw	2019-11-09 02:34:44

```
3 rows in set (0.01 sec)
```

資料庫與資料表基本操作 (12/16)

- 由前頁圖可知，因為起初在設定studata資料表結構時將stu_Sex欄位設有預設值M，所以想使用預設值的話可以直接於insert資料中輸入default，如第三筆。再者，觀察第一筆與第三筆資料的stu_Tel之差異，可發現於insert時沒有將資料使用單引號(')，會使得電腦將資料誤判為數字而不是一段文字，所以將首字0去除，以下將會介紹使用修改的語法將此資料做修正。

資料庫與資料表基本操作 (13/16)

- **修改資料(UPDATE)**：修改studata資料表中的stu_Tel 欄位資料，當stu_Id欄位值為3時stu_Tel= 0978654334，修改後檢視是否成功。
update studata set stu_Tel ='0978654334'
where stu_Id=3;
select * from studata;

```
MariaDB [securitytest]> update studata
-> set stu_Tel ='0978654334' where stu_Id=3;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [securitytest]> select * from studata;
+-----+-----+-----+-----+-----+
| stu_Id | stu_Name | stu_Sex | stu_Tel   | stu_Mail                |
+-----+-----+-----+-----+-----+
| 1      | Jack    | M      | 0975730800 | nptu123@mail.nptu.edu.tw |
| 2      | Cindy   | F      | NULL      | cindy900@mail.nptu.edu.tw |
| 3      | David   | M      | 0978654334 | abc123@mail.nptu.edu.tw  |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

資料庫與資料表基本操作 (14/16)

- 刪除資料(TRUNCATE)：首先建立stu_score資料表並新增五筆資料作為後續示範使用。

```
create table stu_score(  
    stu_Id varchar(5) not null default '00000',  
    english int,  
    math int,  
    chinese int,  
    primary key(stu_Id));
```

```
MariaDB [securitytest]> describe stu_score;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type      | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| stu_Id | varchar(5) | NO   | PRI | 00000   |       |  
| english | int(11)    | YES  |     | NULL    |       |  
| math   | int(11)    | YES  |     | NULL    |       |  
| chinese | int(11)    | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

資料庫與資料表基本操作 (15/16)

insert into stu_score values
 (1,65,70,55),
 (2,94,49,74),
 (3,83,58,76),
 (4,65,92,46);

```
MariaDB [securitytest]> select * from stu_score;  
+-----+-----+-----+-----+  
| stu_Id | english | math | chinese |  
+-----+-----+-----+-----+  
| 1      | 65      | 70   | 55      |  
| 2      | 94      | 49   | 74      |  
| 3      | 83      | 58   | 76      |  
| 4      | 65      | 92   | 46      |  
+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

資料庫與資料表基本操作 (16/16)

- 先顯示資料表中內容，確定資料表內有資料，再將所有資料清空，最後確認資料表內為Empty。

```
select * from stu_score;  
truncate table stu_score;  
select * from stu_score;
```

```
MariaDB [securitytest]> select * from stu_score;  
+-----+-----+-----+-----+  
| stu_Id | english | math | chinese |  
+-----+-----+-----+-----+  
| 1      | 65      | 70   | 55      |  
| 2      | 94      | 49   | 74      |  
| 3      | 83      | 58   | 76      |  
| 4      | 65      | 92   | 46      |  
+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

```
MariaDB [securitytest]> truncate table stu_score;  
Query OK, 0 rows affected (0.11 sec)
```

```
MariaDB [securitytest]> select * from stu_score;  
Empty set (0.01 sec)
```



資料表進階操作

資料表進階操作 (1/26)

- 資料排序(**ORDER BY**)：將stu_Name 由ASCII碼做大到小排序，再將其由小排到大。
select * from studata order by stu_Name desc;
select * from studata order by stu_Name asc;

```
MariaDB [securitytest]> select * from studata order by stu_Name desc;
+-----+-----+-----+-----+-----+
| stu_Id | stu_Name | stu_Sex | stu_Tel   | stu_Mail                |
+-----+-----+-----+-----+-----+
| 1      | Jack    | M       | 0975730800 | nptu123@mail.nptu.edu.tw |
| 3      | David   | M       | 0978654334 | abc123@mail.nptu.edu.tw  |
| 2      | Cindy   | F       | NULL      | cindy900@mail.nptu.edu.tw |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
MariaDB [securitytest]> select * from studata order by stu_Name asc;
+-----+-----+-----+-----+-----+
| stu_Id | stu_Name | stu_Sex | stu_Tel   | stu_Mail                |
+-----+-----+-----+-----+-----+
| 2      | Cindy   | F       | NULL      | cindy900@mail.nptu.edu.tw |
| 3      | David   | M       | 0978654334 | abc123@mail.nptu.edu.tw  |
| 1      | Jack    | M       | 0975730800 | nptu123@mail.nptu.edu.tw  |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

資料表進階操作 (2/26)

- 聚合函數：為了以下示範先新增score欄位於studata資料表中做使用，並新增資料於學生1、2、3中。

```
alter table studata add column score int;  
update studata set score=46 where stu_Id=1;  
update studata set score=94 where stu_Id=2;  
update studata set score=81 where stu_Id=3;
```

```
MariaDB [securitytest]> alter table studata add column score int;  
Query OK, 0 rows affected (0.09 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
MariaDB [securitytest]> update studata set score=46 where stu_Id=1;  
Query OK, 1 row affected (0.02 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
MariaDB [securitytest]> update studata set score=94 where stu_Id=2;  
Query OK, 1 row affected (0.03 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
MariaDB [securitytest]> update studata set score=81 where stu_Id=3;  
Query OK, 1 row affected (0.00 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
MariaDB [securitytest]> select * from studata;  
+-----+-----+-----+-----+-----+-----+-----+  
| stu_Id | stu_Name | stu_Sex | stu_Tel | stu_Mail | updatetime | score |  
+-----+-----+-----+-----+-----+-----+-----+  
| 1 | Jack | M | 0975730800 | nptu123@mail.nptu.edu.tw | 2019-05-01 23:09:02 | 46 |  
| 2 | Cindy | F | NULL | cindy900@mail.nptu.edu.tw | 2019-05-01 23:09:15 | 94 |  
| 3 | David | M | 0978654334 | abc123@mail.nptu.edu.tw | 2019-05-01 23:09:24 | 81 |  
+-----+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

資料表進階操作 (3/26)

- `select [欲使用的函數(參考欄位)] as ['顯示的欄位名稱'] from [資料表名稱];`
`select count(*) as '筆數',`
`max(score) as '最高分數',`
`min(score) as '最低分數',`
`avg(score) as '平均分數',`
`sum(score) as '總分' from studata;`

資料表進階操作 (4/26)

- count → 計算資料數量，max → 該欄位資料最大值，min → 該欄位資料最小值，avg → 欄位資料平均值，sum → 欄位資料值的總和。

```
MariaDB [securitytest]> select count(*) as '筆數',  
-> max(score) as '最高分數',  
-> min(score) as '最低分數',  
-> avg(score) as '平均分數',  
-> sum(score) as '總分' from studata;
```

筆數	最高分數	最低分數	平均分數	總分
3	94	46	73.6667	221

1 row in set (0.00 sec)

資料表進階操作 (5/26)

- 合併顯示資料：首先，建立stu_class_score資料表及新增五筆資料作為後續示範使用。

```
create table stu_class_score(  
    stu_Id varchar(5) not null default '00000',  
    english int,  
    math int,  
    chinese int,  
    primary key(stu_Id));
```

```
MariaDB [securitytest]> describe stu_class_score;
```

Field	Type	Null	Key	Default	Extra
stu_Id	varchar(5)	NO	PRI	00000	
english	int(11)	YES		NULL	
math	int(11)	YES		NULL	
chinese	int(11)	YES		NULL	

```
4 rows in set (0.01 sec)
```

資料表進階操作 (6/26)

insert into stu_class_score values
 (1,65,70,55),
 (2,94,49,74),
 (3,83,58,76),
 (4,65,92,46);

```
MariaDB [securitytest]> select * from stu_class_score;
```

stu_Id	english	math	chinese
1	65	70	55
2	94	49	74
3	83	58	76
4	65	92	46

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

資料表進階操作 (7/26)

- 卡氏積：又稱交叉乘積、交叉合併，以下範例為同時顯示出兩個資料表 (studata&stu_class_score) 的資料，顯示資料時的資料數量為資料表1的a筆資料*資料表2的b筆資料。
- `select * from 資料表1,資料表2;`
`select * from studata,stu_class_score ;`

```
MariaDB [securitytest]> select * from studata,stu_class_score;
```

stu_Id	stu_Name	stu_Sex	stu_Tel	stu_Mail	updatetime	score	stu_Id	english	math	chinese
1	Jack	M	0975730800	nptu123@mail.nptu.edu.tw	2019-11-09 02:42:37	46	1	65	70	55
2	Cindy	F	NULL	cindy900@mail.nptu.edu.tw	2019-11-09 02:43:07	94	1	65	70	55
3	David	M	0978654334	abc123@mail.nptu.edu.tw	2019-11-09 02:43:16	81	1	65	70	55
1	Jack	M	0975730800	nptu123@mail.nptu.edu.tw	2019-11-09 02:42:37	46	2	94	49	74
2	Cindy	F	NULL	cindy900@mail.nptu.edu.tw	2019-11-09 02:43:07	94	2	94	49	74
3	David	M	0978654334	abc123@mail.nptu.edu.tw	2019-11-09 02:43:16	81	2	94	49	74
1	Jack	M	0975730800	nptu123@mail.nptu.edu.tw	2019-11-09 02:42:37	46	3	83	58	76
2	Cindy	F	NULL	cindy900@mail.nptu.edu.tw	2019-11-09 02:43:07	94	3	83	58	76
3	David	M	0978654334	abc123@mail.nptu.edu.tw	2019-11-09 02:43:16	81	3	83	58	76
1	Jack	M	0975730800	nptu123@mail.nptu.edu.tw	2019-11-09 02:42:37	46	4	65	92	46
2	Cindy	F	NULL	cindy900@mail.nptu.edu.tw	2019-11-09 02:43:07	94	4	65	92	46
3	David	M	0978654334	abc123@mail.nptu.edu.tw	2019-11-09 02:43:16	81	4	65	92	46

資料表進階操作 (8/26)

- 對等合併 (Equi-Join) 。
- select * from 資料表1,資料表2 where 條件;
select * from studata,stu_class_score where
studata.stu_Id=stu_class_score.stu_Id;

```
MariaDB [securitytest]> select * from studata,stu_class_score where studata.stu_Id=stu_class_score.stu_Id;
```

stu_Id	stu_Name	stu_Sex	stu_Tel	stu_Mail	updatetime	score	stu_Id	english	math	chinese
1	Jack	M	0975730800	nptu123@mail.nptu.edu.tw	2019-11-09 02:42:37	46	1	65	70	55
2	Cindy	F	NULL	cindy900@mail.nptu.edu.tw	2019-11-09 02:43:07	94	2	94	49	74
3	David	M	0978654334	abc123@mail.nptu.edu.tw	2019-11-09 02:43:16	81	3	83	58	76

3 rows in set (0.11 sec)

資料表進階操作 (9/26)

- 左外部合併。
- `select * from 資料表1 left outer join 資料表2 on 資料表1.[欄位]=資料表2.[欄位];`
`select * from studata left outer join stu_class_score on studata.stu_Id= stu_class_score.stu_Id;`

```
MariaDB [securitytest]> select * from studata left outer join stu_class_score on studata.stu_Id=stu_class_score.stu_Id;
```

stu_Id	stu_Name	stu_Sex	stu_Tel	stu_Mail	updatetime	score	stu_Id	english	nath	chinese
1	Jack	M	0975730800	nptu123@mail.nptu.edu.tw	2019-11-09 02:42:37	46	1	65	70	55
2	Cindy	F	NULL	cindy900@mail.nptu.edu.tw	2019-11-09 02:43:07	94	2	94	49	74
3	David	M	0978654334	abc123@mail.nptu.edu.tw	2019-11-09 02:43:16	81	3	83	58	76

3 rows in set (0.10 sec)

資料表進階操作 (10/26)

- 右外部合併。
- `select * from 資料表1 right join 資料表2 on 資料表1.[欄位]=資料表2.[欄位];`
`select * from studata right join stu_class_score on studata.stu_Id= stu_class_score.stu_Id;`

```
MariaDB [securitytest]> select * from studata right join stu_class_score on studata.stu_Id= stu_class_score.stu_Id;
```

stu_Id	stu_Name	stu_Sex	stu_Tel	stu_Mail	updatetime	score	stu_Id	english	math	chinese
1	Jack	M	0975730800	nptu123@mail.nptu.edu.tw	2019-11-09 02:42:37	46	1	65	70	55
2	Cindy	F	NULL	cindy900@mail.nptu.edu.tw	2019-11-09 02:43:07	94	2	94	49	74
3	David	M	0978654334	abc123@mail.nptu.edu.tw	2019-11-09 02:43:16	81	3	83	58	76
NULL	NULL	NULL	NULL	NULL	NULL	NULL	4	65	92	46

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

資料表進階操作 (11/26)

- **MySQL加密**：新增一個名為Userinfo的資料表作以下MySQL加密示範使用。

```
create table Userinfo (  
    user_SN int(10) not null auto_increment,  
    user_ID varchar(20) not null,  
    user_PW varbinary(255),  
    user_Name varchar(20) not null,  
    time timestamp not null default  
current_timestamp,  
    primary key(user_SN),  
    unique key (user_ID));
```


資料表進階操作 (12/26)

```
MariaDB [securitytest]> describe Userinfo;
```

Field	Type	Null	Key	Default	Extra
user_SN	int(10)	NO	PRI	NULL	auto_increment
user_ID	varchar(20)	NO	UNI	NULL	
user_PW	varbinary(255)	YES		NULL	
user_Name	varchar(20)	NO		NULL	
time	timestamp	NO		CURRENT_TIMESTAMP	

5 rows in set (0.00 sec)

資料表進階操作 (13/26)

- 雙向加密-Encode : 先新增兩筆資料進入資料表中，如下圖。

- `encode('密碼','金鑰')`

`insert into`

`Userinfo(user_ID,user_PW,user_Name)`

`values('abc123','qwe456','周杰倫');`

`insert into`

`Userinfo(user_ID,user_PW,user_Name)`

`values('def654','qcv54w','楊陳林');`

```
MariaDB [securitytest]> select * from Userinfo;
```

user_SN	user_ID	user_PW	user_Name	time
1	abc123	qwe456	周杰倫	2019-05-01 23:30:17
2	def654	qcv54w	楊陳林	2019-05-01 23:30:41

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

資料表進階操作 (13/26)

- 利用update方法將user_PW使用encode()加密。
update Userinfo set
user_PW=encode('qwe456','test1') where
user_SN=1;
update Userinfo set
user_PW=encode('red4v63','test1') where
user_SN=2;

```
MariaDB [securitytest]> select * from Userinfo;
```

user_SN	user_ID	user_PW	user_Name	time
1	abc123	♦♦^.Gb	周杰倫	2019-05-01 23:30:17
2	def654	tqq>41♦	楊陳林	2019-05-01 23:30:41

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

資料表進階操作 (14/26)

- 利用insert方法新增user_PW時使用encode()加密。

insert into

Userinfo(user_ID,user_PW,user_Name) values
('tgh963da',encode('rtg964da','test1'),'蕭勁
藤');

```
MariaDB [securitytest]> select * from Userinfo;
```

user_SN	user_ID	user_PW	user_Name	time
1	abc123	^^^.Gb	周杰倫	2019-05-01 23:30:17
2	def654	tqq>41^	楊陳林	2019-05-01 23:30:41
3	tgh963da	t^^>~^^^	蕭勁藤	2019-05-01 23:34:37

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

資料表進階操作 (15/26)

- 雙向加密-Decode : 使用decode方法將加密過後的密碼解密。
 - decode(欄位名稱,'金鑰')
- ```
select * , decode(user_PW,'test1') as decode
from Userinfo;
```

```
MariaDB [securitytest]> select * , decode(user_PW,'test1') as decode from Userinfo;
+-----+-----+-----+-----+-----+-----+
| user_SN | user_ID | user_PW | user_Name | time | decode |
+-----+-----+-----+-----+-----+-----+
1	abc123	♦♦^..Gb	周杰倫	2019-05-01 23:30:17	qwe456
2	def654	tqg>41♦	楊陳林	2019-05-01 23:30:41	red4v63
3	tgh963da	t♦♦>~♦~♦	蕭勁藤	2019-05-01 23:34:37	rtg964da
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

## 資料表進階操作 (16/26)

- **雙向加密-AES\_ENCRYPT** :利用insert 資料於資料表時將密碼使用aes\_encrypt()加密。
- aes\_encrypt('密碼','金鑰')  
insert into Userinfo(user\_ID,user\_PW,user\_Name)  
values('wec5413',aes\_encrypt('okg9621','test2'),'林又家');  
insert into Userinfo(user\_ID,user\_PW,user\_Name)  
values('ked963',aes\_encrypt('lkm85daw61','test2'),'莫雯尉');

```
MariaDB [securitytest]> select * from Userinfo;
```

| user_SN | user_ID  | user_PW | user_Name | time                |
|---------|----------|---------|-----------|---------------------|
| 1       | abc123   | ^Gb     | 周杰倫       | 2019-05-01 23:30:17 |
| 2       | def654   | tqq>41  | 楊陳林       | 2019-05-01 23:30:41 |
| 3       | tgh963da | t~      | 蕭勁藤       | 2019-05-01 23:34:37 |
| 4       | wec5413  | Y}~0pc  | 林又家       | 2019-05-01 23:41:29 |
| 5       | ked963   | lw!Z-j7 | 莫雯尉       | 2019-05-01 23:41:44 |

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

## 資料表進階操作 (17/26)

- 雙向加密-AES\_ENCRYPT :將user\_PW使用 aes\_decrypt()且金鑰是test2的密碼解密。
  - aes\_decrypt(欄位,'金鑰')
- select \* , aes\_decrypt(user\_PW,'test2') as aes\_decrypt from Userinfo where user\_SN=4 OR user\_SN=5;

```
MariaDB [securitytest]> select * , aes_decrypt(user_PW,'test2') as aes_decrypt from Userinfo where user_SN=4 OR user_SN=5;
```

| user_SN | user_ID | user_PW | user_Name | time                | aes_decrypt |
|---------|---------|---------|-----------|---------------------|-------------|
| 4       | wec5413 | Y~0pc   | 林又家       | 2019-05-01 23:41:29 | okg9621     |
| 5       | ked963  | lw!Z_}7 | 莫雯尉       | 2019-05-01 23:41:44 | lkm85daw61  |

2 rows in set (0.00 sec)

## 資料表進階操作 (18/26)

- **雙向加密-DES\_ENCRYPT**：利用insert 資料於資料表時將密碼使用des\_encrypt()加密。
- `des_encrypt('密碼', '金鑰')`  
`insert into Userinfo(user_ID,user_PW,user_Name)`  
`values('96weq85',des_encrypt('63gwes84','test3')`  
`),'王大福');`  
`insert into Userinfo(user_ID,user_PW,user_Name)`  
`values('85fsw',des_encrypt('vbn3669','test3'),'張`  
`卿坊');`

```

MariaDB [securitytest]> select * from Userinfo;
+-----+-----+-----+-----+-----+
| user_SN | user_ID | user_PW | user_Name | time |
+-----+-----+-----+-----+-----+
1	abc123	^.Gb	周杰倫	2019-05-01 23:30:17
2	def654	tq>41	楊陳林	2019-05-01 23:30:41
3	tgh963da	>~>	蕭勁藤	2019-05-01 23:34:37
4	wec5413	Y}~>0pc	林又家	2019-05-01 23:41:29
5	ked963	lw!Z }>7	莫雯尉	2019-05-01 23:41:44
6	96weq85	>	王太福	2019-05-01 23:45:28
7	85fsw	z>t>	張卿坊	2019-05-01 23:45:37
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

```



## 資料表進階操作 (19/26)

- 雙向加密-**DES\_DENCRYPT** :將user\_PW使用des\_decrypt()且是金鑰test2的密碼解密。
  - des\_decrypt(欄位,'金鑰')
- ```
select * , des_decrypt(user_PW,'test3') as  
des_decrypt from Userinfo where user_SN=6  
OR user_SN=7;
```

```
MariaDB [securitytest]> select * , des_decrypt(user_PW,'test3') as des_decrypt from Userinfo where user_SN=6 OR user_SN=7;
```

user_SN	user_ID	user_PW	user_Name	time	des_decrypt
6	96weq85	王	王大福	2019-05-01 23:45:28	63gwes84
7	85fsw	張	張卿坊	2019-05-01 23:45:37	vbn3669

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

資料表進階操作 (20/26)

- **單向加密-MD5** :利用insert資料於資料表時將密碼使用MD5()加密，由於此方法為單向加密，因此無法將加密後的密碼還原，但仍可以用count去比對加密前後的密碼是否匹配。

- md5('密碼')

insert into

Userinfo(user_ID,user_PW,user_Name)

values('cef86314',md5('ty9621w'),'周興哲');

```
MariaDB [securitytest]> select * from Userinfo where user_SN=8;
```

user_SN	user_ID	user_PW	user_Name	time
8	cef86314	47d923619176583b13ae7c92d37cf553	周興哲	2019-05-01 23:49:48

```
1 row in set (0.00 sec)
```

資料表進階操作 (21/26)

select count(*) from Userinfo where user_SN=8
and user_PW=md5('ty9621w');

```
MariaDB [securitytest]> select count(*) from Userinfo where user_SN=8 and user_PW=md5('ty9621w');
+-----+
| count(*) |
+-----+
|         1 |
+-----+
1 row in set (0.00 sec)
```

資料表進階操作 (22/26)

- **單向加密-PASSWORD** : 利用insert資料於資料表時將密碼使用PASSWORD()加密，接著使用count比對加密後的密碼，顯示結果如圖。
- password('密碼')
insert into
Userinfo(user_ID,user_PW,user_Name)
values('asd567',password('qwdf234'),'茄子蛋');
select count(*) from Userinfo where user_SN=9
and user_PW=password('qwdf234');

資料表進階操作 (23/26)

```
MariaDB [securitytest]> select * from Userinfo where user_SN=9;
```

user_SN	user_ID	user_PW	user_Name	time
9	asd567	*5D2E4A16670726F1EE88E8345189BD9C712DA478	茄子蛋	2019-05-01 23:54:52

1 row in set (0.00 sec)

```
MariaDB [securitytest]> select count(*) from Userinfo where user_SN=9 and user_PW=password('qwdf234');
```

count(*)
1

1 row in set (0.00 sec)

資料表進階操作 (24/26)

- 單向加密-**ENCRYPT** :利用insert 資料於資料表時將密碼使用ENCRYPT()加密。
- encrypt('密碼', '金鑰')
insert into
Userinfo(user_ID,user_PW,user_Name)
values('olm84k',encrypt('yui8413','testencrypt'),'蔡一霖');

```
MariaDB [securitytest]> select * from Userinfo where user_SN=10;
+-----+-----+-----+-----+-----+
| user_SN | user_ID | user_PW          | user_Name | time                |
+-----+-----+-----+-----+-----+
|      10 | olm84k  | tenWCq/ayHJZc    | 蔡一霖    | 2019-05-01 23:58:16 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

資料表進階操作 (25/26)

- 單向加密-SHA、SHA1：SHA()&SHA1()方法是一樣的所以本範例採用SHA()加密，利用insert 資料於資料表時將密碼使用SHA()加密。
 - sha('密碼') or sha1('密碼')
- insert into
Userinfo(user_ID,user_PW,user_Name)
values('85osk9',sha('854ewwe'),'張匯妹');

```
MariaDB [securitytest]> select * from Userinfo where user_SN=11;
```

user_SN	user_ID	user_PW	user_Name	time
11	85osk9	fe344d892d812fdcdb5a8488087a7a79aca9a280	張匯妹	2019-05-01 23:59:53

1 row in set (0.00 sec)

資料表進階操作 (26/26)

- 使用count比對加密後的密碼，顯示結果如下圖。
select count(*) from Userinfo where user_SN=11
and user_PW=sha('854ewwe');

```
MariaDB [securitytest]> select count(*) from Userinfo where user_SN=11 and user_PW=sha('854ewwe');
+-----+
| count(*) |
+-----+
|         1 |
+-----+
1 row in set (0.00 sec)
```


參考資料

- Linux優缺點：
http://linux.vbird.org/linux_basic/0110whatislinux/0110whatislinux-fc4.php#linuxpt_adv
- Linux比較：
https://mropengate.blogspot.com/2015/08/linux-linux_11.html
- MariaDB介紹：
<https://twoss.gitbook.io/open-source-use-case/zi-liao-ku/mariadb>
- MariaDB特色：
<https://zh.wikipedia.org/wiki/MariaDB>

參考資料

- MariaDB優缺點：
<https://tpu.thinkpower.com.tw/tpu/articleDetails/916>
- SQL Server介紹：
https://zh.wikipedia.org/wiki/Microsoft_SQL_Server
- SQL Server特色：
<https://ithelp.ithome.com.tw/articles/10206222>
- SQL Server優缺點：
<https://www.webdesigns.com.tw/SQLServerAndMySQL.asp>

參考資料

- HBase介紹：

<http://www.htsjk.com/hbase/35256.html>

<https://www.itread01.com/content/1546602691.html>

- HBase特色：

<https://www.twblogs.net/a/5c362b2fbd9eee35b3a57cf7>

- HBase優缺點：

<https://kknews.cc/code/mexljm6.html>

- Oracle Database：

<https://ithelp.ithome.com.tw/articles/10206222>

參考資料

- IBM DB2簡介：
<https://www.ibm.com/tw-zh/analytics/db2>
- IBM DB2特色：
[h-
tw/SSEPGG_11.1.0/com.ibm.db2.luw.wn.doc/doc/c0060311.html](https://www.ibm.com/support/knowledgecenter/z<h-
tw/SSEPGG_11.1.0/com.ibm.db2.luw.wn.doc/doc/c0060311.html)
- IBM DB2優缺點：
<http://www.asp.php.online/shujuku/db2sjk/db2rm/201701/225470.html>

參考資料

- Sybase簡介、特色：

https://www.mpinfo.com.tw/products_1_detail2.php?BrandID=4&pType=4

- Sybase優缺點：

<https://bbs.csdn.net/topics/30322145>