

第五堂 預存程序

聯成電腦

張明泰 mtchang.tw@gmail.com





SQL的製作

- 在執行程式時產生的SQL稱為**動態SQL**，通常有SQL inject的問題。

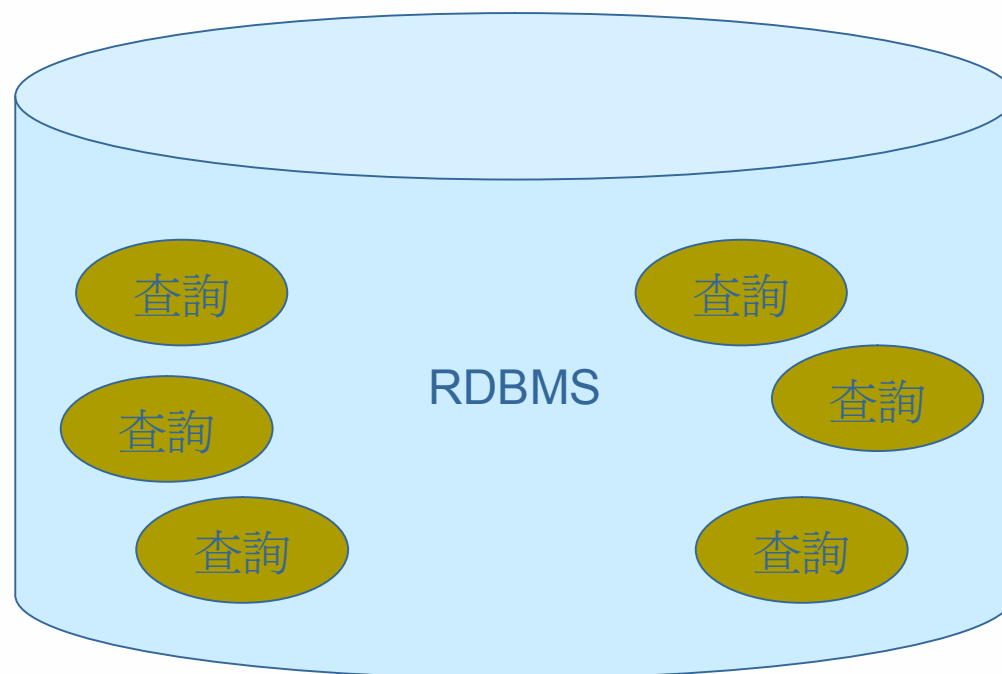
Select * from students where 姓名="`<?php ?>`";

- 不會變化的SQL稱為**靜態SQL**。

Select * from students where 姓名="張小明";

預存程序

- 預存程序：將相關的查詢放在一起，只要命令一次即可使用
- 優：減少網路流量、處理速度較快
- 缺：撰寫麻煩、應用不方便





擴充SQL

- 擴充SQL是使用SQL來進程式設計，擴充SQL是程式語言的一種，因此也可以進行引數的指定和流程控制。
- 擴充SQL也可以在預存程式中使用，只要把使用擴充SQL的製作出來的一連串程式當成預存程序儲存起來，就可以只用一個查詢指令完成很複雜的操作。
- 不同的RDBMS使用的擴充SQL就會不同。Oracle是PL/SQL。MS-SQL是使用Transact-SQL。



使用SQL的程式設計(mysql預存程序)

- 在使用mysql的Stored Procedure之前，請先確定你有下列三種權限。
- The **CREATE ROUTINE privilege** is needed to create stored routines.
- The **ALTER ROUTINE privilege** is needed to alter or drop stored routines. This privilege is granted automatically to the creator of a routine.
- The **EXECUTE privilege** is required to execute stored routines. However, this privilege is granted automatically to the creator of a routine. Also, the default SQL SECURITY characteristic for a routine is DEFINER, which allows users who have access to the database with which the routine is associated to execute the routine





補充：mysql權限表格-user,db

USE mysql;
SELECT * FROM USER ;

USE mysql;
SELECT * FROM db;

Host	User	Password	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_
localhost	root	*40CD9B7960EE57F256408140BA63D82E406E5E61	Y	Y	Y	Y	Y
%	mtchang	*A7A9FADD822492649C01057FB6ADBBC9D2FD02DE	N	N	N	N	N
localhost	a01	*01E0A6A2FBE8C3C166E726B1025F7F0888985637	N	N	N	N	N

Host	Db	User	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv	Grant_priv	References_priv	Index_
localhost	a01	a01	Y	Y	Y	Y	Y	Y	Y	Y	Y
%	mtchang	mtchang	Y	Y	Y	Y	Y	Y	Y	Y	Y

	使用者	主機	密碼	整體權限 	授權	
<input type="checkbox"/>	a01	localhost	是	USAGE	否	
<input type="checkbox"/>	mtchang	%	是	USAGE	否	
<input type="checkbox"/>	root	localhost	是	ALL PRIVILEGES	是	



補充：使用者帳號建立

```
# 建立資料庫  
CREATE DATABASE `a02` ;  
# 建立資料庫使用者a02密碼a02  
CREATE USER a02@'localhost' identified BY 'a02'  
# 給予a02@localhost的使用者，a02資料庫的全部使用權限  
GRANT ALL PRIVILEGES ON `a02` . * TO a02@'localhost' WITH GRANT  
OPTION ;
```




補充：使用者帳號刪除於更改密碼

```
# 刪除 'a02'@'localhost' ...  
DROP USER a02@'localhost';  
# 重新讀取權限 ...  
FLUSH PRIVILEGES ;  
# 刪除資料庫  
DROP DATABASE a02;
```

```
# 使用者a02修改密碼為aaa  
SET PASSWORD FOR a02@'localhost' = PASSWORD( 'aaa' )
```


補充：使用者對於mysql環境全體權限

整體權限 (全選 / 全部取消)

注意: MySQL 權限名稱會以英語顯示

資料

- ☐ SELECT
- ☐ INSERT
- ☐ UPDATE
- ☐ DELETE
- ☐ FILE

結構

- ☐ CREATE
- ☐ ALTER
- ☐ INDEX
- ☐ DROP
- ☐ CREATE TEMPORARY TABLES
- ☐ CREATE VIEW
- ☐ SHOW VIEW
- ☐ CREATE ROUTINE
- ☐ ALTER ROUTINE
- ☐ EXECUTE

系統管理

- ☐ GRANT
- ☐ SUPER
- ☐ PROCESS
- ☐ RELOAD
- ☐ SHUTDOWN
- ☐ SHOW DATABASES
- ☐ LOCK TABLES
- ☐ REFERENCES
- ☐ REPLICATION CLIENT
- ☐ REPLICATION SLAVE
- ☐ CREATE USER

資源限制

註: 設定這些選項為 0 (零) 可解除限制。

MAX QUERIES PER HOUR
MAX UPDATES PER HOUR
MAX CONNECTIONS PER HOUR
MAX USER_CONNECTIONS

補充：使用者對於資料庫存取權限

指定資料庫權限 (全選 / 全部取消)

注意: MySQL 權限名稱會以英語顯示

資料

- ☒ SELECT
- ☒ INSERT
- ☒ UPDATE
- ☒ DELETE

結構

- ☒ CREATE
- ☒ ALTER
- ☒ INDEX
- ☒ DROP
- ☒ CREATE TEMPORARY TABLES
- ☒ CREATE VIEW
- ☒ SHOW VIEW
- ☒ CREATE ROUTINE
- ☒ ALTER ROUTINE
- ☒ EXECUTE

系統管理

- ☒ GRANT
- ☒ LOCK TABLES
- ☒ REFERENCES



補充：資料庫使用權限修改

從權限資料庫移除mtchang@%使用者對於資料庫mtchang的所有權限
REVOKE ALL PRIVILEGES ON `mtchang` . * FROM mtchang@'%';

給予mtchang@%使用者對於資料庫mtchang的權限(全部指定)
GRANT ALL PRIVILEGES ON `mtchang` . * TO mtchang@'% ' WITH
GRANT OPTION ;



補充：資料庫使用權限賦予單獨指定

```
# 給予mtchang@%使用者對於資料庫mtchang的權限(單獨指定)  
GRANT  
SELECT,INSERT ,UPDATE ,DELETE ,CREATE ,DROP ,REFERENCES ,INDE  
X ,ALTER ,  
CREATE TEMPORARY TABLES ,  
LOCK TABLES ,  
CREATE VIEW ,  
SHOW VIEW ,  
CREATE ROUTINE,  
ALTER ROUTINE,  
EXECUTE ON `mtchang` . * TO mtchang@'%' WITH GRANT OPTION ;
```

你的第一個StoreProcedure

建立procedure

```
CREATE PROCEDURE simpleproc()  
begin  
    SELECT * FROM students;  
end
```

Create procedure 為建立預存程序
Simpleproc() 為程序名稱，「()」內可宣告參數
Begin及end 為程式的區段範圍
call 為呼叫預存程序執行

執行storeProcedure

```
Call simpleproc();
```

執行結果

	學號	班級座號	姓名	出生年月日	身分證號碼	住址
▶	911001	10101	王子穎	1984/3/5	C100000012	基隆市安樂區安和一街
	911002	10102	王慧如	1980/9/8	F200000026	台北縣瑞芳鎮一坑路426
	911003	10103	王琇榆	1984/8/28	F200000035	基隆市暖暖區源遠路29
	911004	10104	朱勝真	1984/8/28	F100000042	台北縣瑞芳鎮逢甲路33
	911005	10105	江玉婷	1983/12/10	C200000058	基隆市安樂區安和一街
	911006	10106	余家男	1983/12/16	C100000067	基隆市七堵區百六街53
	911007	10107	余鳳如	1984/1/6	F200000071	台北縣瑞芳鎮三爪子坑
	911008	10108	呂啟升	1984/4/4	C100000085	基隆市瑞芳路111巷20號

2006/2/12

帶有參數的StoreProcedure

建立procedure

```
CREATE PROCEDURE simpleproc2( param1 varchar(9))  
begin  
SELECT * FROM view_records WHERE n1 LIKE param1;  
end
```

格式：

```
CREATE PROCEDURE 程序名稱(參數名稱 資料型態)  
BEGIN  
    SQL敘述  
END
```

執行storeProcedure

```
call simpleproc2('10109');
```

執行結果

結果 1						
	n1	n2	s1	s2	s3	s4
▶	10109	900228	33	2	4	9
✱						

StoreProcedure的變數

建立procedure

```
CREATE PROCEDURE `simpleproc3` ( param2 VARCHAR(5))  
begin  
  DECLARE xname VARCHAR(5) DEFAULT 'bob';  
  DECLARE newname VARCHAR(5);  
  SET newname='test';  
  SELECT CONCAT(param2,xname,newname,'!');  
end
```

執行storeProcedure

```
call simpleproc3('var');
```

Declare 宣告要使用的變數名稱及資料型態
SET可設定變數的值
CONCAT()函數，將所有的字串傳回顯示。

結果 1

	CONCAT(param2,xname,newname,'!')
▶	varbobtest!
✱	

a function

建立procedure

```
CREATE FUNCTION hello (s CHAR(20)) RETURNS CHAR(50)
RETURN CONCAT('Hello, ',s,'!');
```

執行storeProcedure

```
SELECT hello('SQL');
```

執行結果

	hello('SQL')
▶	Hello, SQL!
✱	

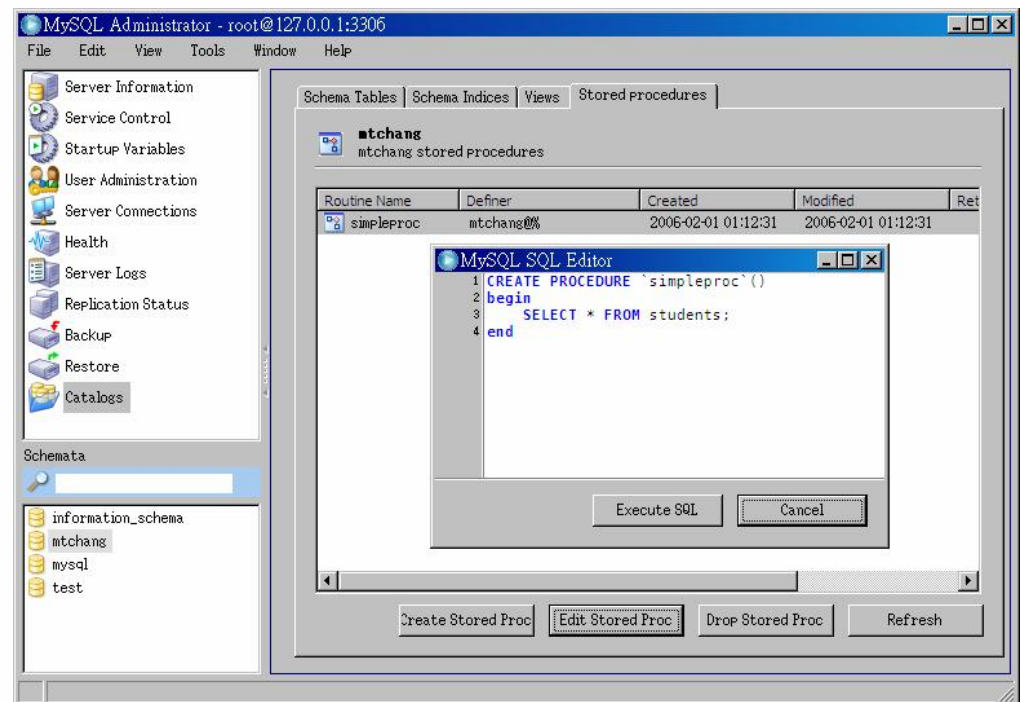
CONCAT函數會將內部的字串結合成為單一字串

刪除及修改預存程序

可以使用alter修改，drop刪除
ALTER PROCEDURE and ALTER FUNCTION
DROP PROCEDURE and DROP FUNCTION
使用方法請參考mysql 手冊

範例：**DROP PROCEDURE test;**
說明：刪除test此程序

也可用MySQL Administrator GUI介面修改





練習：

- 題目：請建立一個名為test的預存程序，此程序用途為輸入一個數值n就可以算出1,2,3....+n的總和。例如：使用者只需下call test(100);此指令，即會傳回5050的數值。
- Ps.請直接使用公式計算
$$\text{sum} = (\text{前} + \text{後}) * \text{總數} / 2$$



休息一下





StoreProcedure的流程控制

- The IF, CASE, LOOP, WHILE, ITERATE, and LEAVE constructs are fully implemented.



IF Statement

語法：

```
IF search_condition THEN statement_list  
  [ELSEIF search_condition THEN  
statement_list] ...  
  [ELSE statement_list]  
END IF
```

範例：

```
if a='10112' then  
    select a,b;  
end if;
```



CASE Statement

語法：

```
CASE case_value  
  WHEN when_value THEN statement_list  
  [WHEN when_value THEN statement_list] ...  
  [ELSE statement_list]  
END CASE
```

語法：

```
CASE  
  WHEN search_condition THEN statement_list  
  [WHEN search_condition THEN  
statement_list] ...  
  [ELSE statement_list]  
END CASE
```




LEAVE Statement

- This statement is used to exit any labeled flow control construct. It can be used with BEGIN ... END or loops



ITERATE Statement

- ITERATE can only appear within LOOP, REPEAT, and WHILE statements. ITERATE means “do the loop again.”



LOOP Statement

範例：

```
CREATE PROCEDURE doiterate(p1 INT)
BEGIN
  #標籤label1 LOOP迴圈
  label1: LOOP
    # 設定p1=p1的值加1
    SET p1 = p1 + 1;
    # 如果p1<10就跳到label1
    IF p1 < 10 THEN ITERATE label1; END IF;
    # 直接離開迴圈
    LEAVE label1;
  # 迴圈到label1
  END LOOP label1;
  #設定p1直給外部變數@x，使用者可以下select @x取得。
  SET @x = p1;
END
```

語法：

```
[begin_label:] LOOP
  statement_list
END LOOP [end_label]x
```

執行1：

```
call doiterate(1);
```

```
SELECT @x;
```

回應1：

```
10
```

執行2：

```
call doiterate(100);
```

```
SELECT @x;
```

回應2：

```
101
```



REPEAT Statement

語法：

```
[begin_label:] REPEAT  
    statement_list  
UNTIL search_condition  
END REPEAT [end_label]
```

範例：

```
CREATE PROCEDURE dorepeat(p1 INT)  
BEGIN  
    SET @x = 0;  
    REPEAT  
    SET @x = @x + 1;  
    UNTIL @x > p1 END REPEAT;  
END
```

執行：

```
call dorepeat(100);  
SELECT @x
```

回應：

```
@x=101
```



WHILE Statement

語法：

```
[begin_label:] WHILE search_condition DO  
    statement_list  
END WHILE [end_label]
```

範例：(此範例不完整，但可參考)

```
CREATE PROCEDURE dowhile()  
BEGIN  
    DECLARE v1 INT DEFAULT 100;  
  
    WHILE v1 > 0 DO  
        ...<可加入敘述>...  
        SET v1 = v1 - 1;  
    END WHILE;  
END
```

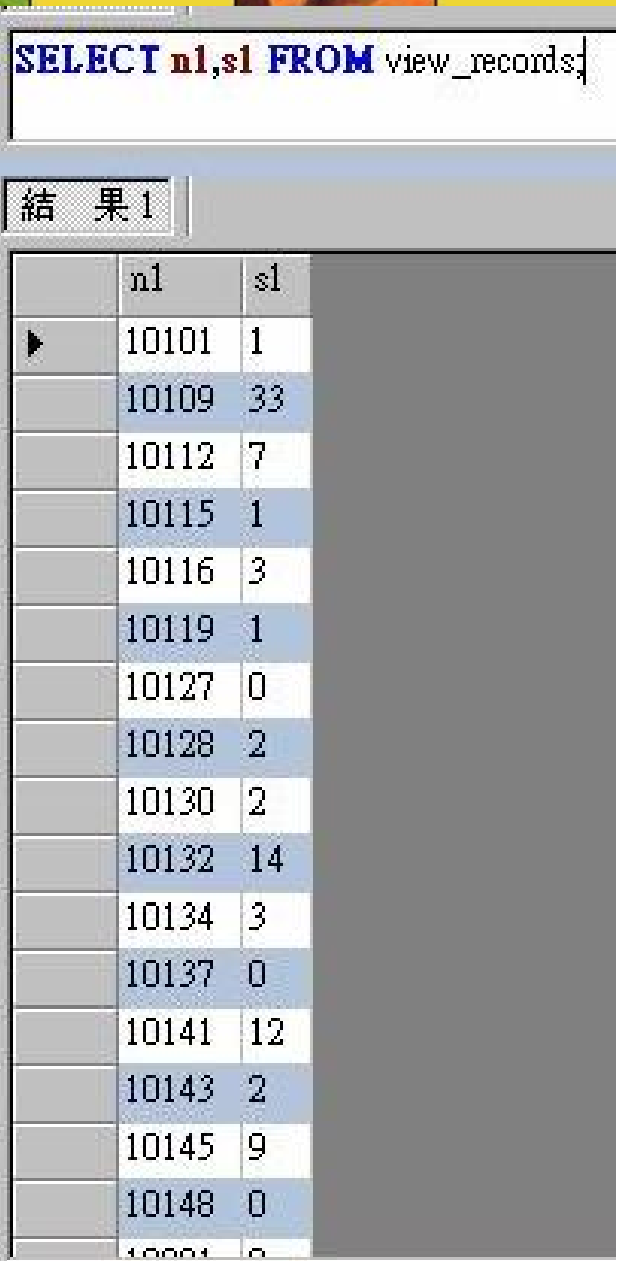
範例練習：

- 題目：請使用累積的算法，寫一預存程序，以迴圈方式計算 $1+2+3+\dots+100$ 的結果。
- 也就是說，只要執行`call dowhile()`;
- 就會回應 $s=5050$
- 請勿使用公式



結果集合result set

- 使用查詢取得的資料結合，稱之為結果集合。
- 結果集合通常是由行與列的組成。



```
SELECT n1,s1 FROM view_records;
```

	n1	s1
▶	10101	1
	10109	33
	10112	7
	10115	1
	10116	3
	10119	1
	10127	0
	10128	2
	10130	2
	10132	14
	10134	3
	10137	0
	10141	12
	10143	2
	10145	9
	10148	0
	10001	0

資料游標

- 資料游標是依種能讓使用者一航一行的去使用結果集合中所含的資料的機制。
- 使用資料游標之後，我們可以針對結果集合中的資料重複的進行同樣的處理。

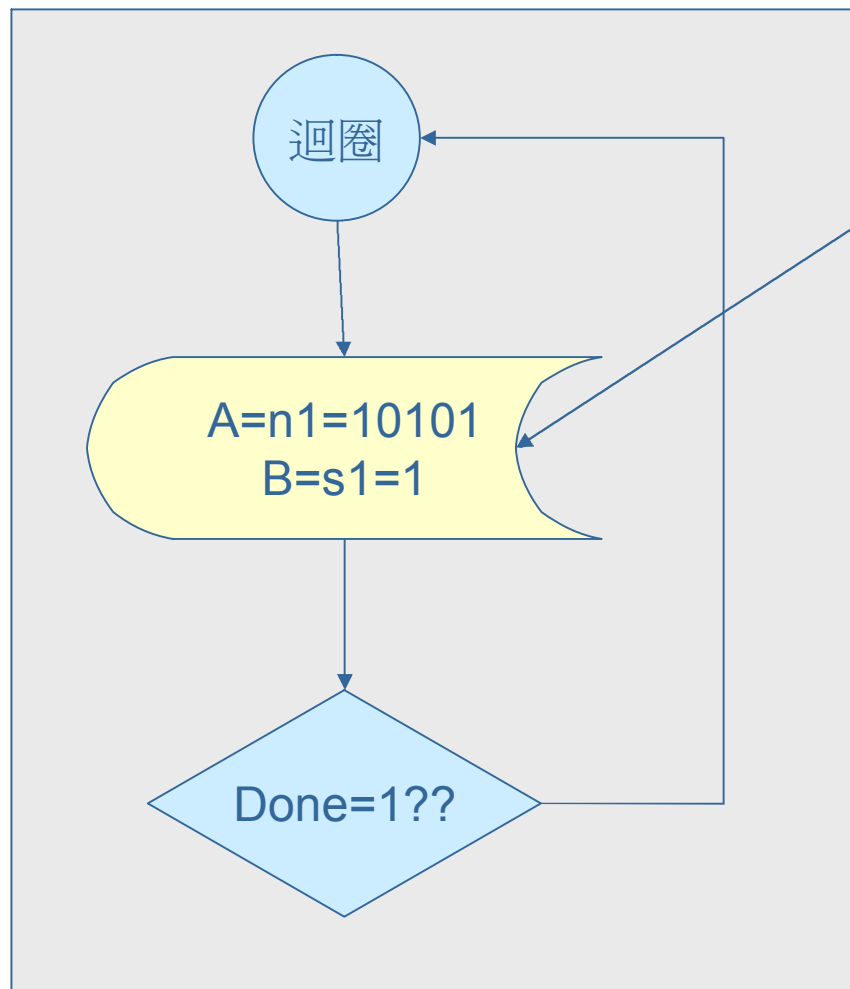
資料游標

```
SELECT n1,s1 FROM view_records;
```

結果1

n1	s1
10101	1
10109	33
10112	7
10115	1
10116	3
10119	1
10127	0
10128	2
10130	2
10132	14
10134	3
10137	0
10141	12
10143	2
10145	9
10148	0
10001	0

資料游標-在預存程序中的使用



資料游標

```
SELECT n1,s1 FROM view_records;
```

結果1

	n1	s1
	10101	1
	10109	33
	10112	7
	10115	1
	10116	3
	10119	1
	10127	0
	10128	2
	10130	2
	10132	14
	10134	3
	10137	0
	10141	12
	10143	2
	10145	9
	10148	0
	10001	0

資料游標-程式範例

```
CREATE PROCEDURE curdemo()
BEGIN
# 變數宣告
DECLARE done INT DEFAULT 0;
DECLARE a CHAR(16);
DECLARE b,c INT;
# 宣告資料游標對應的SQL敘述
DECLARE cur1 CURSOR FOR SELECT n1,s1 FROM view_records;
# 宣告當發生SQLSTATE '02000'的錯誤時將done變數設為1
DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = 1;
# 開啓資料游標變數
OPEN cur1;
# 迴圈開始
REPEAT
# 擷取資料游標的值放到a,b變數裡
  FETCH cur1 INTO a, b;
# 針對資料游標所取到的值作處理
  if a='10112' then
    select a,b;
  end if;
# 判斷done是否改變為1，是的話結束迴圈
UNTIL done END REPEAT;
CLOSE cur1;
END
```

執行
call curdemo();

結果 1

	a	b
▶	10112	7
✱		

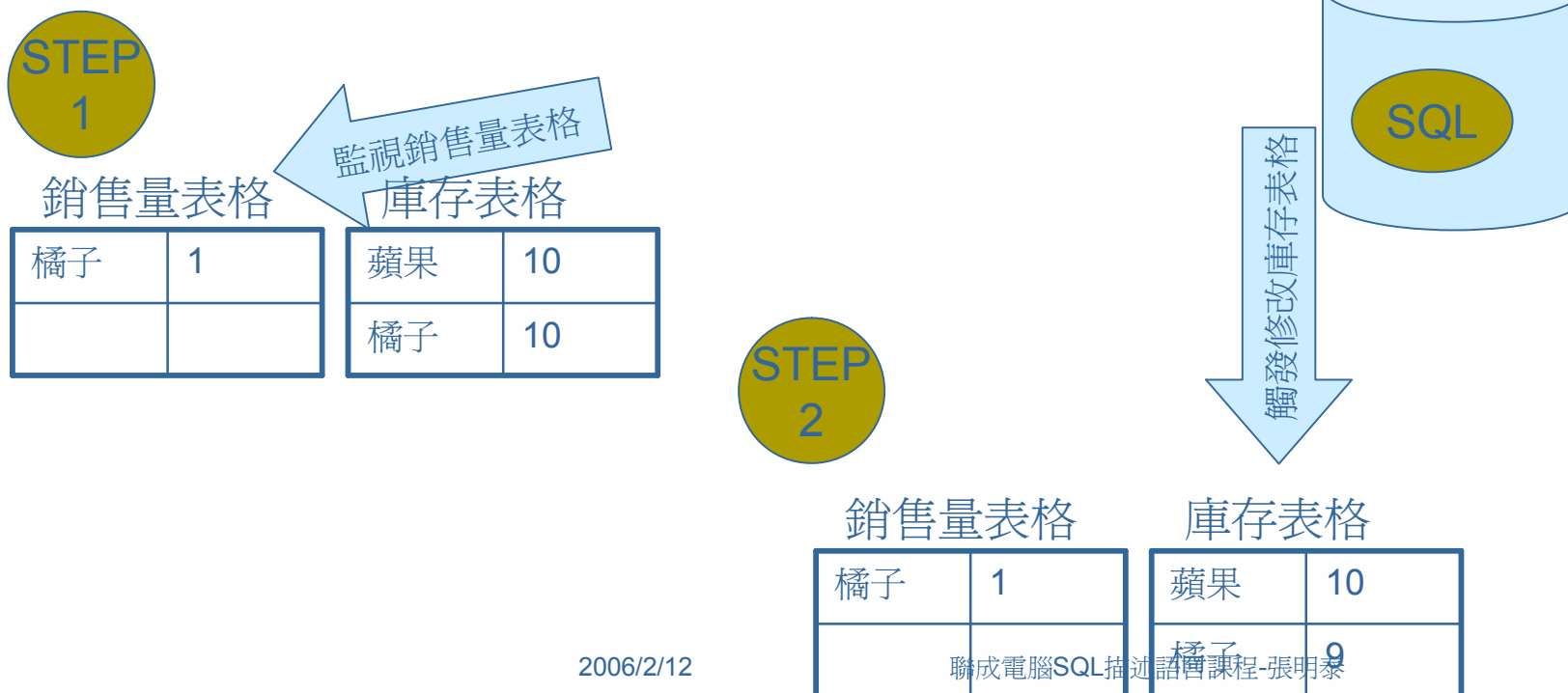


休息一下



觸發

- 指的是在表格被insert,update,delete等等作特殊操作時，自動執行的程序。





Triggers觸發

語法：

CREATE

[DEFINER = { user | CURRENT_USER }]

TRIGGER trigger_name trigger_time trigger_event

ON tbl_name FOR EACH ROW trigger_stmt

DEFINER 定義由那個使用者權限使用執行

trigger_time 觸發的時機. 他能再啟動的敘述執行前BEFORE或執行後AFTER啟動。

trigger_event 觸發的事件能夠是：INSERT，UPDATE，DELETE

目前止能以EACH ROW為單位監控觸發
使用觸發指令需具有MYSQL整體權限的SUPER權限

建立觸發

```
建立觸發： CREATE TRIGGER testref BEFORE INSERT ON test
FOR EACH ROW BEGIN
INSERT INTO students SET 學號 = NEW.學號;
END
```

(當test資料庫有insert異動時，插入一筆資料到student資料庫的學號欄位，並將學號值設定與插入test資料庫的學號值一樣)

引起觸發：

```
INSERT INTO test(學號,姓名,住址) VALUES ('11111111','張小明','高雄縣');
```

結果 1					
	學	班級座號	姓名	出生年月日	身
▶	1111111	(Null)	(Null)	(Null)	(Null)

刪除觸發事件：

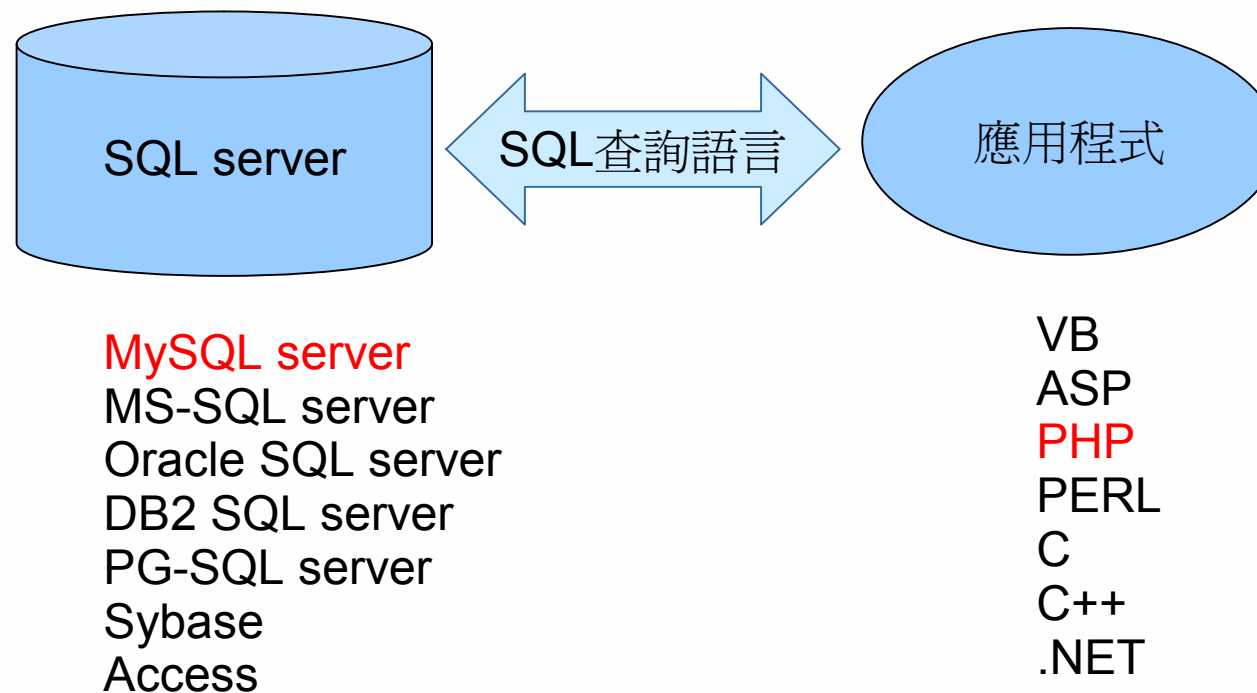
```
DROP trigger testref;
```




休息一下



應用程式與SQL server的溝通



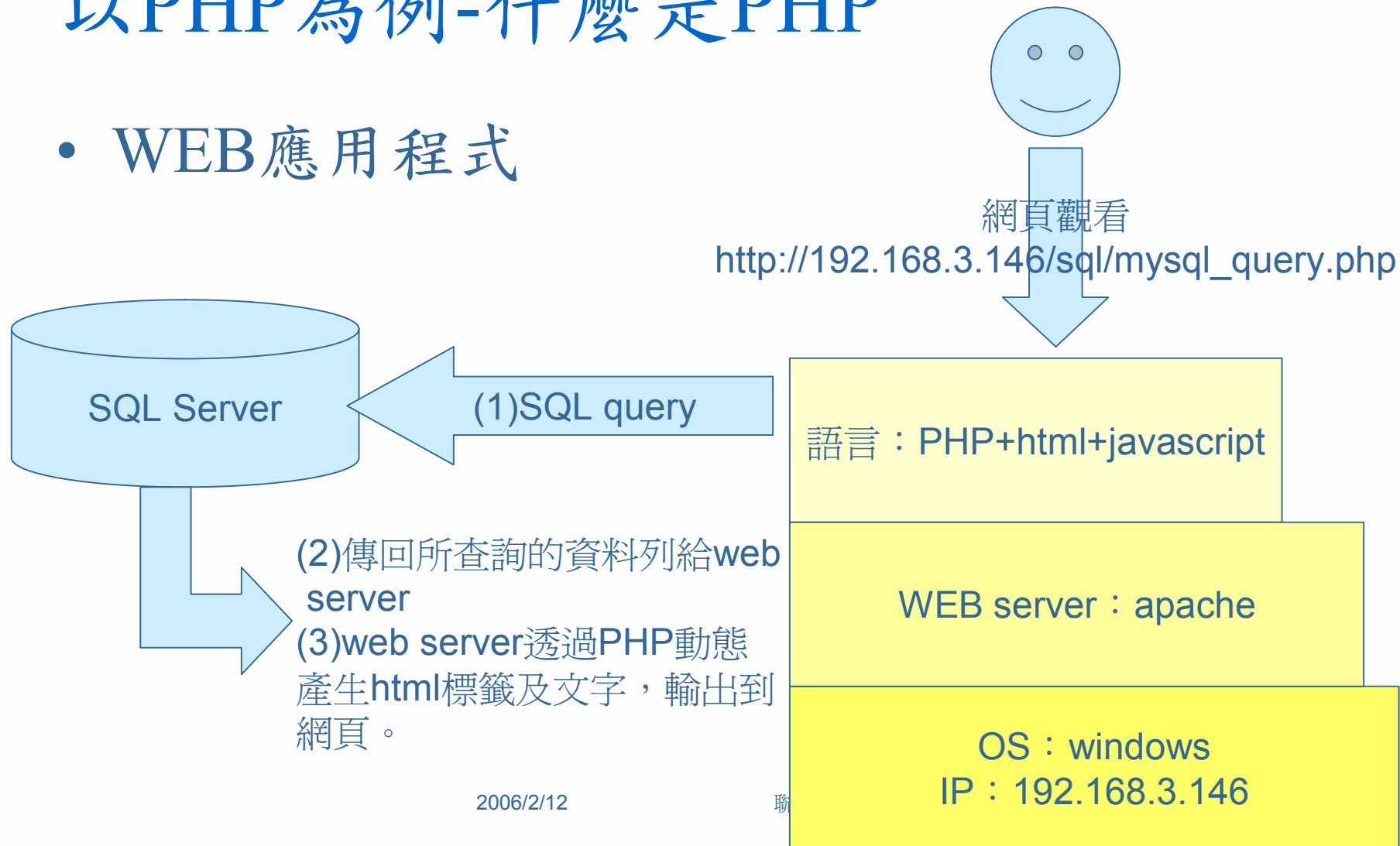
2006/2/12

聯成電腦SQL描述語言課程-張明泰

38

以PHP為例-什麼是PHP

- WEB應用程式





PHP及APACHE的安裝

- <http://httpd.apache.org/>
- <http://www.php.net/>



快速安裝apache+php+mysql+phpmyadmin

- 在windows上快速安裝
php+mysql+phpmyadmin+apache的整合套件包
- 官方網站：<http://www.appservnetwork.com/>
- SF網站：
<http://sourceforge.net/projects/appserv>
- 中文安裝說明：
http://eg-land.com/html_data/22/0511/1057.html



安裝完成-測試php

- 建立一個檔案phpinfo.php並放置於網頁目錄底下
- 檔案內容如下：
<? phpinfo(); ?>
- 使用瀏覽器執行：
<http://xxx.xxx.xxx.xxx/phpinfo.php>
- 此檔案內容主要顯示此web server及php的相關環境資訊，可用來做為我們確定Server環境的判斷。

Phpinfo 資訊

phpinfo() - Mozilla Firefox

檔案(F) 編輯(E) 檢視(V) 瀏覽(G) 書籤(B) 工具(T) 說明(H)


http://127.0.0.1/myweb/phpinfo.php

mtchang 常用文件網站 下載 最近焦點 快速搜尋 127.0.0.1 - /myweb/ Yahoo!奇摩 Google 電子化政府入口網

Google 免費空間 搜尋 PageRank ABC 檢查 選項 免費空間

Barkley Light RapidShare Webhosting + ... 127.0.0.1 / localhost / mys... phpinfo0 Gmail - 在 1 主題中的 1 ...

PHP Version 5.1.1



System	Windows NT MT-NB 5.1 build 2600
Build Date	Nov 27 2005 21:34:13
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	CGI/FastCGI
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS\php.ini
PHP API	20041225
PHP Extension	20050922
Zend Extension	220051025
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled

觀看phpinfo-注意要點

Mbstring 多字元語系的支援

Mysql 指php對於mysql存取的支援

mbstring

Multibyte Support	enabled
Multibyte string engine	libmbfl
Multibyte (japanese) regex support	enabled
Multibyte regex (oniguruma) version	3.7.1

mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	4.1.7

Directive	Local Value	Master Value
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	no value	no value
mysql.default_password	no value	no value
mysql.default_port	no value	no value
mysql.default_socket	no value	no value
mysql.default_user	no value	no value
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited

Hello php and html

- 你的第一支php程式

- Hello.php

```
<?php
```

```
Echo 'hello php';
```

```
?>
```

```
<br />
```

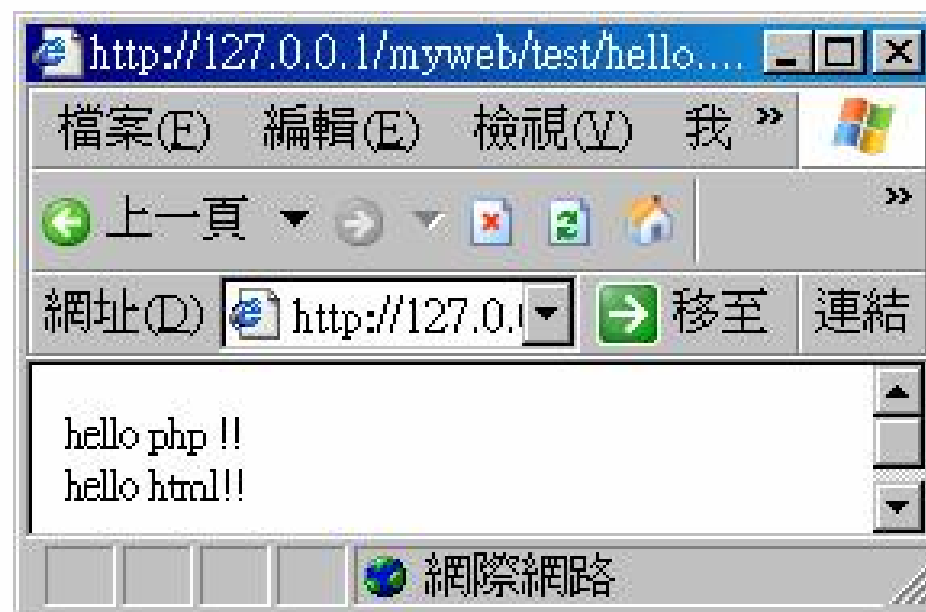
```
hello html!!
```

在DOS模式下以CLI方式執行PHP

```
D:\mtchang\web_site\test>c:\php\php.exe hello.php
```

```
hello php !!
```

```
D:\mtchang\web_site\test>
```





參考

- Mysql manu. Chapter 20. Stored Procedures and Functions
- <http://www.php.net/>
- <http://httpd.apache.org/>