**註解**

-- 註解

**Prepared Statements**

* 定義 : 類似VIEW但可以執行變數
* 限制 : 離線後即刪除
* 程序

1. 設定Prepared Statements

PREPARE PS名 FROM 'SELECT... WHERE ...= ?'

1. 設定變數

SET(SELECT) @變數 := 值

1. 執行Prepared Statements

EXECUTE PS名 USING @變數

* 刪除 : DROP PREPARE PS名

**Stored Procedure**

* 程序

1. 建立

DELIMITER $$ => 讓結尾的;改成$$

CREATE PROCEDURE PROCEDURE名 ( [IN/OUT/INOUT] 參數1型態 , ...)

BEGIN

… ; => 因為PROCEDURE裡的每個QUERY結尾一定要有;

… ;

END $$

DELIMITER ; => 讓結尾的$$改回;

1. 呼叫

CALL PROCEDURE名 (參數1 , ...)

* 刪除 : DROP PROCEDURE PROCEDURE名
* 查詢 : SHOW CREATE PROCEDURE PROCEDURE名

**Stored Function (函式)**

* 程序

1. 建立

DELIMITER $$

CREATE FUNCTION FUNCTION名 ( [IN/OUT/INOUT] 參數1型態 , ...)

RETURNS 函式要回傳的資料型態

BEGIN

… ;

RETURN … ;

END $$

DELIMITER ;

1. 使用

SELECT FUNCTION名 (參數1,...)

* 刪除 : DROP FUNCTION FUNCTION名
* 查詢 : SHOW CREATE FUNCTION FUNCTION名

**※　參數功能設定 [IN/OUT/INOUT]**

* **IN(預設)**：輸入給Procedure用,可給值或變數
* **OUT：**Procedure 執行完會被賦值,只能給變數
* **INOUT：**只能給變數

**Trigger (做新修刪前後做什麼事)**

* 建立

DELIMITER $$

CREATE TRIGGER TRIGGER名 BEFORE/AFTER DELETE/UPDATE/INSERT ON 表格名 FOR EACH ROW

BEGIN

QUERY ; => 可用: OLD.欄位1 (修改前)/NEW.欄位1 (修改後)

END $$

DELIMITER ;

* 刪除 : DROP TRIGGER TRIGGER名;

**宣告 (置頂在BEGIN前!!)**

**(排序1) 宣告區域變數**

DECLARE 變數1 變數2 型態 DEFAULT 0 ;

BEGIN

SELECT … INTO 變數1 ; / SET … := 變數2 ;

RETURN 變數2 ;

END

**(排序2) 宣告BUG的變數**

DECLARE Condition名 CONDITION FOR 錯誤訊息 ;

**※　錯誤訊息**

* SQLSTATE 'XXXXX(5碼)'
* ERROR編號 (XXXX)
* Condition名
* SQLWARNING
* NOT FOUND
* SQLEXCEPTION

**(排序3) 宣告CURSOR => 可把查詢結果一一寫進變數**

DECLARE Cursor名 CURSOR FOR 查詢條件 ;

DECLARE EXIT HANDLER FOR 1329(查不到下一筆的ERROR) BEGIN END ; => 處理CURSOR會遇到的BUG

BEGIN

OPEN Cursor名 ;

LOOP

FETCH Cursor名 INTO 變數名 ;

　 => 根據查詢條件把查到的資料(一次查一筆)放進變數

　 => 每FETCH一次就移到下一筆,所以到最後沒有下一筆就一定會報錯

END LOOP ;

CLOSE Cursor名 ;

END

**(排序4) 錯誤處理**

DECLARE CONTINUE(ERROR後繼續執行)/EXIT(ERROR後直接結束) HANDLER FOR 錯誤訊息

BEGIN

... ;

END ;

BEGIN

…

END

**條件式**

1. IF ... THEN ... ;

ELSEIF ... THEN ... ;

ELSE ... ;

END IF ;

1. CASE ...

WHEN ... THEN ... ;

ELSE ... ;

END CASE ;

**迴圈**

1. WHILE ... DO

... ;

EHD WHILE ;

1. REPEAT

... ;

UNTIL ...

END REPEAT ;

1. LOOP

... ;

END LOOP ;

**用”標籤”離開/停止迴圈**

**LABEL名:** BEGIN/REPEAT/LOOP/WHILE

IF ... THEN

**LEAVE LABEL名 ;** => 跳出這個LABEL包起來的地方

**ITERATE LABEL名 ;** => 跳回這個LABEL的頭

END REPEAT/LOOP/WHILE **LABEL名** ;

**匯入檔案**

* 方法一

LOAD DATA INFILE '檔名' (主鍵重複時)IGNORE/REPLACE INTO TABLE 表名

FIELDS TERMINATED BY ' , ' \

ENCLOSED BY ' “ ' － CSV格式

LINES TERMINATED BY ' \r ' /

IGNORE XX LINES => 欄位數不匹配的話要省略幾個欄位

(欄位1,欄位2,變數) => 匹配的欄位

SET 變數=.....

* 方法二

$ mysqlimport -u XXX -pXXX -h XXX [如下] db名 表格名

--ignore/replace

--fields-terminated/enclosed-by=字串

--lines-terminated-by=字串

**匯出檔案**

* 方法一

SELECT ... INTO OUTFILE '檔名'

FIELDS TERMINATED BY '...'

ENCLOSED BY '...'

LINES TERMINATED BY '...'

FROM ...

* 方法二

$ mysqldump -u XXX -pXXX -h XXX --result-file=匯出檔名 [如下] db名 表格名

--all-databases(匯出所有資料庫)

--fields-terminated/enclosed-by=字串

--lines-terminated-by=字串