MYSQL 預存程序

朱克剛

STORE PROCEDURE

建立預存程序

```
Create procedure pro_name()
Begin

End;
```

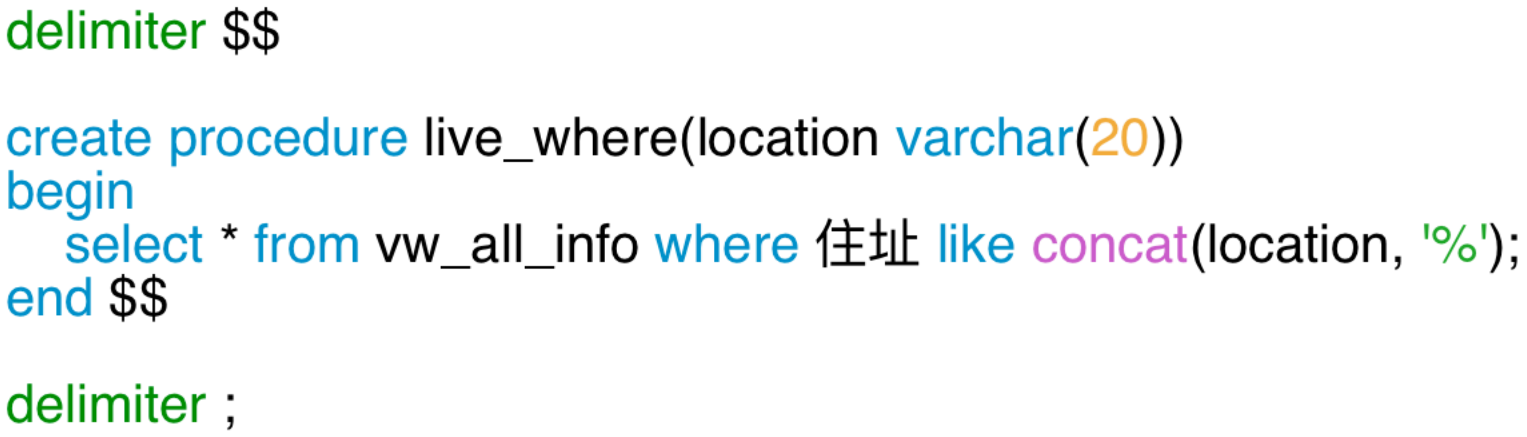
執行方式

- Call pro_name()

帶參數

帶參數的 store procedure

- Create procedure pro_name(p1 type, p2 type)

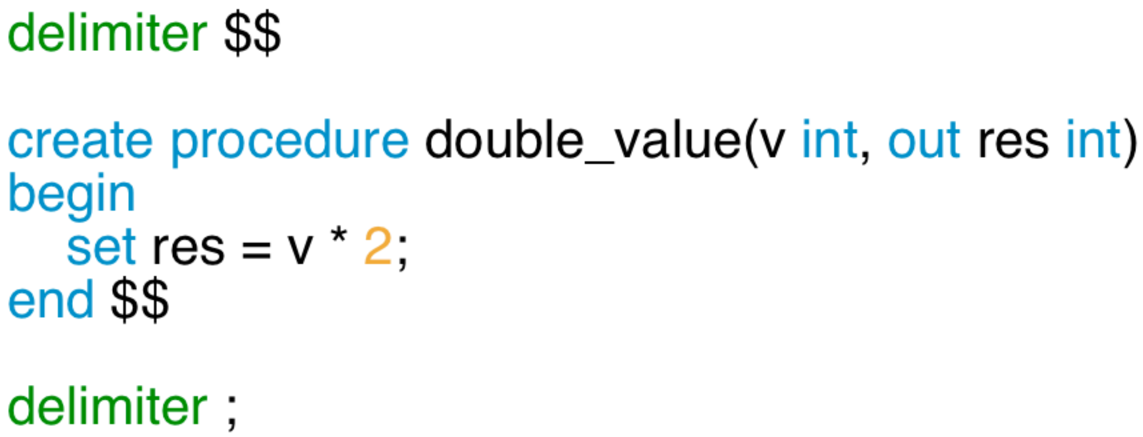


```
delimiter $$

create procedure live_where(location varchar(20))
begin
    select * from vw_all_info where 住址 like concat(location, '%');
end $$

delimiter ;
```

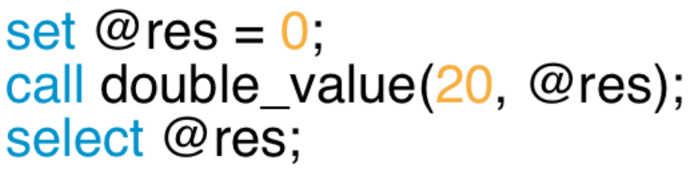
參數類型

參數可以設定成 IN | OUT | INOUT

分別表示輸入、輸出、輸入與輸出，預設值為 IN



```
delimiter $$

create procedure double_value(v int, out res int)
begin
    set res = v * 2;
end $$

delimiter ;
```



```
set @res = 0;
call double_value(20, @res);
select @res;
```

傳回RECORDSET

在store procedure中傳回recordset（或resultset）

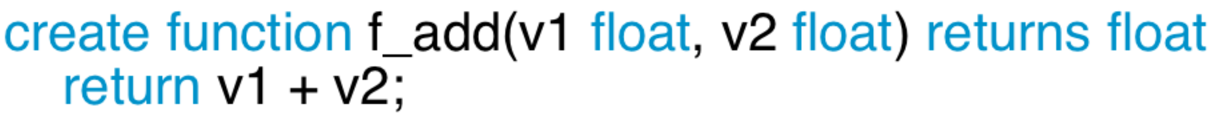
- 在 store procedure 中執行 select 指令即可（注意：不可使用於 function）

STORE FUNCTION

與 store procedure 的差異為：有傳回值

建立函數

- Create function fun_name() returns type
  Return value;

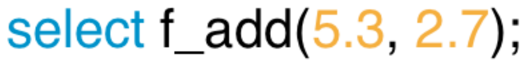


```
create function f_add(v1 float, v2 float) returns float
    return v1 + v2;
```

如果函數中只有一行程式碼，可以省略 BEGIN … END

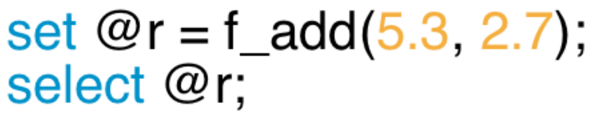
FUNCTION執行方式

使用 select



```
select f_add(5.3, 2.7);
```

使用 set



```
set @r = f_add(5.3, 2.7);
select @r;
```

程式語法

變數宣告(I)

宣告變數 n 並且指定初始值

- Set @n = 10

將查詢結果放到變數中(I)

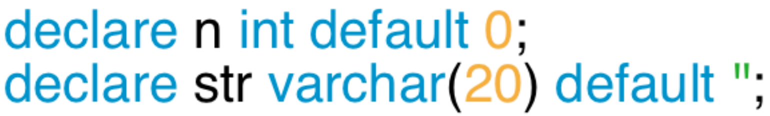
- Set @n = (select count(*) from userinfo)
- Select @n

將查詢結果放到變數中(II)

- Select @n := count(*) from userinfo
- Select @n

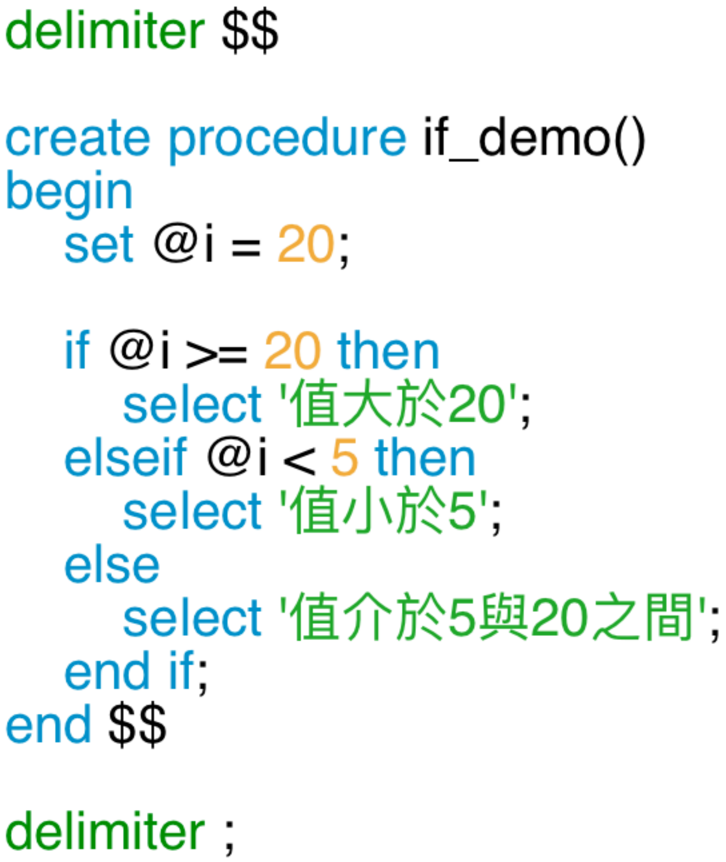
變數宣告(II)

使用 declare



```
declare n int default 0;
declare str varchar(20) default '';
```

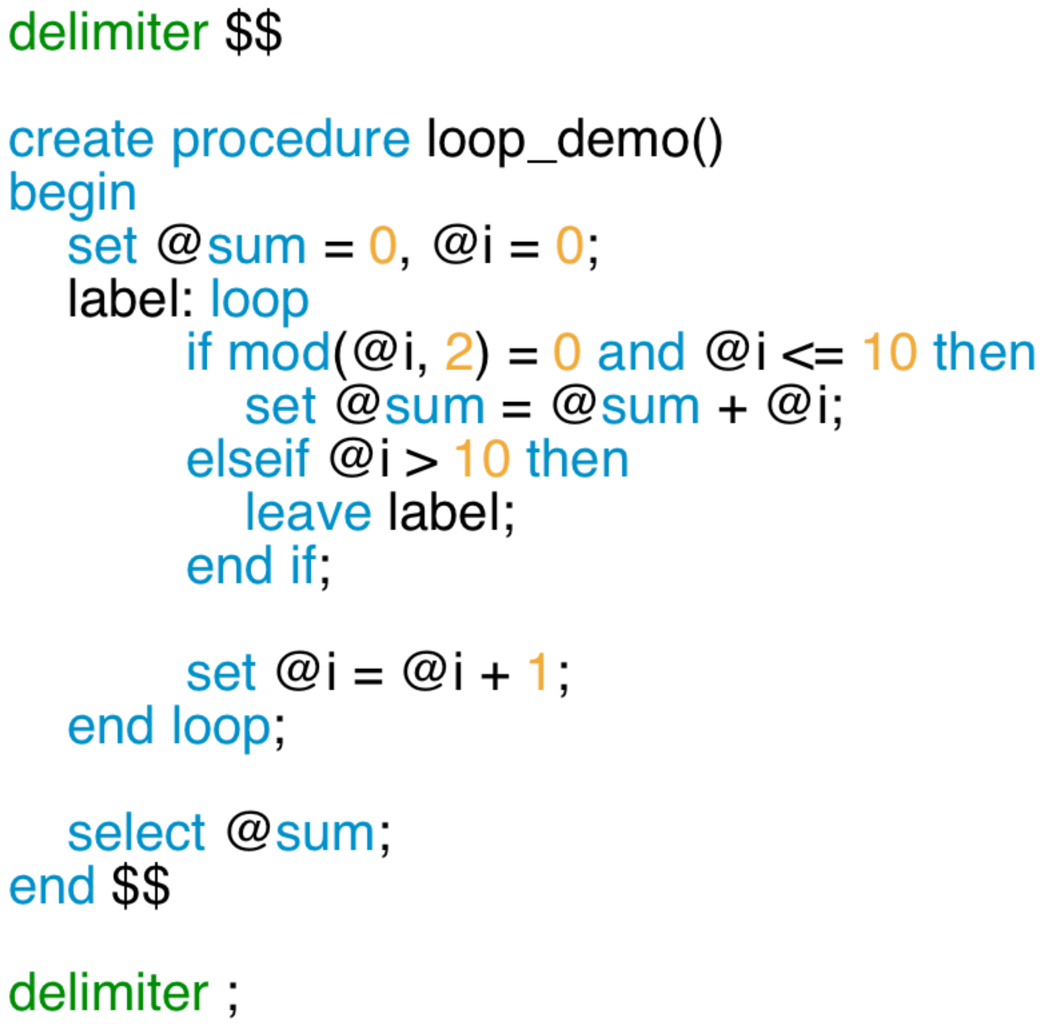
IF 判斷



```
delimiter $$

create procedure if_demo()
begin
    set @i = 20;

    if @i >= 20 then
        select '值大於20';
    elseif @i < 5 then
        select '值小於5';
    else
        select '值介於5與20之間';
    end if;
end $$

delimiter ;
```

LOOP 迴圈

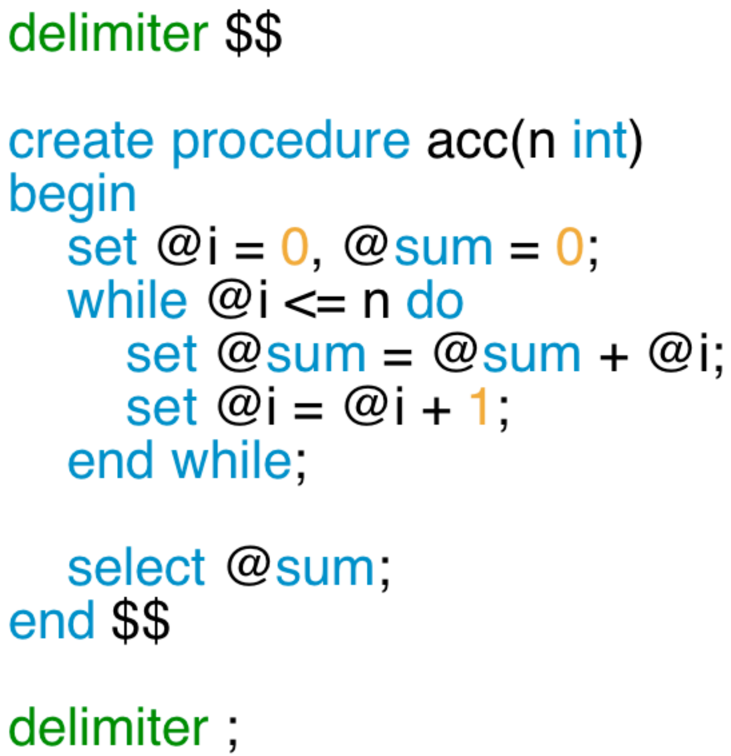
計算 0+2+4+6+8+10



```
delimiter $$

create procedure loop_demo()
begin
    set @sum = 0, @i = 0;
    label: loop
            if mod(@i, 2) = 0 and @i <= 10 then
                set @sum = @sum + @i;
            elseif @i > 10 then
                leave label;
            end if;

            set @i = @i + 1;
    end loop;

    select @sum;
end $$

delimiter ;
```

WHILE 迴圈

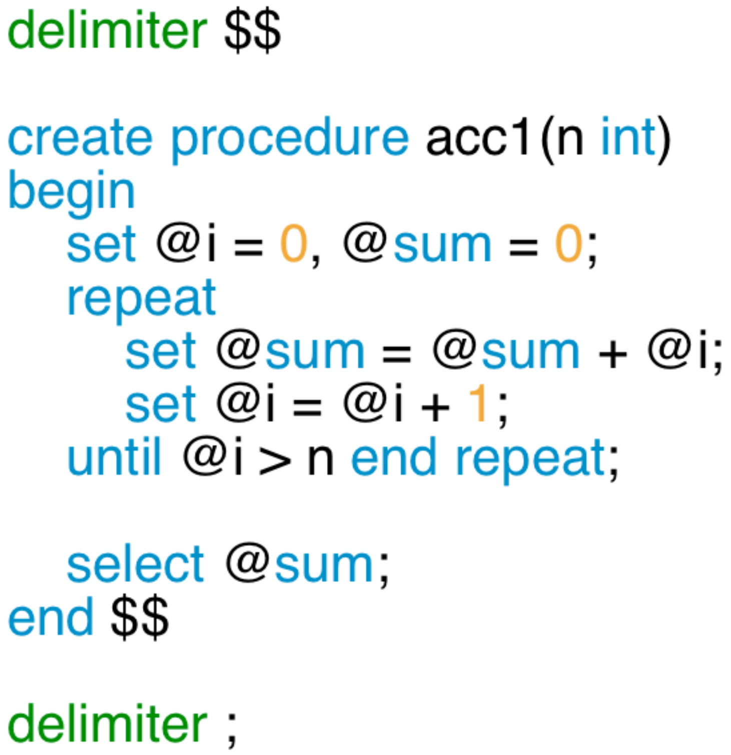
計算 1+2+...+10



```
delimiter $$

create procedure acc(n int)
begin
    set @i = 0, @sum = 0;
    while @i <= n do
        set @sum = @sum + @i;
        set @i = @i + 1;
    end while;

    select @sum;
end $$

delimiter ;
```

REPEAT 迴圈

與 while 迴圈的差異在於repeat 至少會執行一次



```
delimiter $$

create procedure acc1(n int)
begin
    set @i = 0, @sum = 0;
    repeat
        set @sum = @sum + @i;
        set @i = @i + 1;
    until @i > n end repeat;

    select @sum;
end $$

delimiter ;
```

錯誤處理

錯誤發生就離開

```
delimiter $$
create procedure pro_name()
begin
    declare exit handler for sqlexception select 'ERROR';
    insert into userinfo values ('A01', null);
end $$
delimiter ;
```

PK 重複

錯誤發生要繼續

```
delimiter $$
create procedure pro_name()
begin
    declare _rollback bool default false;
    declare continue handler for sqlexception set _rollback = true;
    start transaction;
        insert into userinfo values ('A01', null);
        if _rollback then
            select "FAIL: rollback"
            rollback;
        else
            select "SUCCESS: commit"
            commit;
        end if;
end $$
delimiter ;
```

PK 重複

CURSOR

CURSOR

對每一筆資料作最細微的控制

- 可以單獨處理每一筆資料
- 例如：將阿拉伯數字轉成大寫國字
  - 1 -> 壹元
  - 203 -> 貳佰零叁元

建立、開啟與關閉 CURSOR

建立 ⇨

```
DECLARE curs CURSOR FOR
      select fee from bill
```

開啟 ⇨

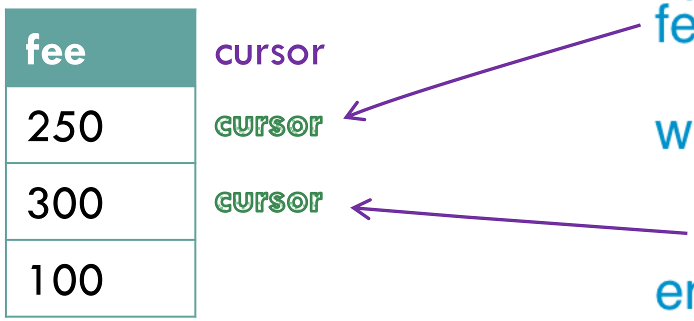
```
OPEN curs
…
```

關閉 ⇨

```
CLOSE curs
```

FETCH

讓資料處理可以一筆一筆進行



fee	cursor
250	cursor
300	cursor
100	

```
delimiter $$

create procedure pro_test()
begin
    declare done int default false;
    declare tmp_fee int;
    declare total int default 0;
    declare curs cursor for select fee from bill;
    declare continue handler for not found set done = true;

    open curs;
    fetch curs into tmp_fee;

    while not done do
        set total = total + tmp_fee;
        fetch curs into tmp_fee;
    end while;

    close curs;
    select total;
end  $$

delimiter ;
```