

Chapter

7

SQL語法介紹

技能指標

1. 了解MySQL運算子與常用函數
2. 了解DML – Select查詢語法
3. 了解DML – 資料異動語法

學習總結

完成本章後、您將了解如何使用SQL的DML語法，進行資料新增、修改、刪除、查詢，同時也了解如何相關條件進行處理。

了解MySQL運算子與常用函數

MySQL可用的運算，大致將它區分成為三個常的類型，分別有「算數運算子」、「關係運算子」、「條件運算子」等。與先前的單元我們所介紹的PHP的概念一樣，只是有些符號會有些微的不同，只要稍加注意就可以了。

當然常用的函數也不能錯過，因此筆者在此也分別介紹了一些常用的函數，讓您在DML函數當中可以搭配運算。

另外在MySQL中的運算結果「1」用來代表True、「0」用來代表False，至於練習的部份，可以在phpmyadmin的SQL中執行，如圖7-1所示。

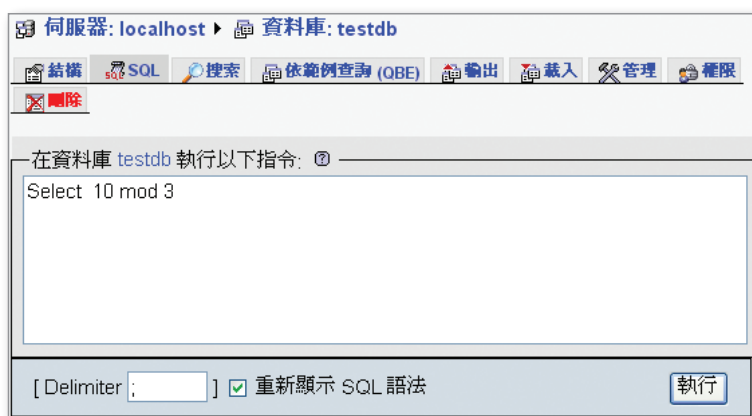


圖7-1 SQL語法撰寫

MySQL運算子介紹

所謂運算子就是使用於運算式中，使得運算子左右兩邊的運算元可以進行運算的符號，SQL可使用的運算子有許多，底下我們將介紹常用的三種運算子。

算數運算子

表7-1 算術運算子

運算子	範例	說明
+	Select 3+6;	相加
-	Select 6-3	相減
*	Select 6*3	相乘
/	Select 6/3	相除
%, MOD	Select 10%3	求餘數
DIV	Select 10 div 3	整除

關係運算子

表7-2 關係運算子

運算子	範例	說明
=	Select 3=3	相等
!=, <>	Select 3!=3	不相等
<	Select 3<4	小於
<=	Select 3<=3	小於等於
>	Select 3>4	大於
>=	Select 3>=4	大於等於
<=>	Select NULL <=> NULL	相等、只比較NULL值
IS NULL	Select NULL IS NULL	檢查欄位值是否為NULL
IS NOT NULL	Select 3 IS NOT NULL	檢查欄位值是否為非NULL
BETWEEN A AND B	Select 4 Between 1 AND 3	介於A與B之間範圍
IN(值1, 值2, …)	Select 3 IN(1,2,3,4,5);	檢查是否在列舉項目中
NOT IN(值1,值2, …)	Select 3 NOT IN(1,2,3,4,5);	檢查是否不在列舉項目中

條件運算子

表7-3 條件運算子

運算子	範例	說明
AND, &&	Select 1 && 1;	而且
OR,	Select 1 0;	或
NOT, !	Select !(1 && 1);	反轉
LIKE	Select 'gjun' Like '%j%';	找一某段與特定文字相同的資料
%	Select 'gjun' Like 'g%';	代表任意長度的任意字元
, ?	Select 'gjun' Like 'j';	單一長度任意字元

● MySQL常用函數介紹

一般算數運算子我們可以搭配數學函數，那麼字串函更就可以用來搭配字元或字串的應用。

數學常用函數

► 表7-4 數學常用函數

函數名稱	說明
ABS(N)	求N的絕對值
SIGN(N)	N為正數時回傳1，為負數時回傳-1、為0時回傳0
MOD(N,M)	求N/M的餘數
FLOOR(N)	求不大於N的最大整數
CEILING(N)	求不小於X 的最小整數值
ROUND(N)	四捨五入到個位數
ROUND(N,D)	四捨五入到小數第D位數
LOG(N)	求N的自然對數
LOG10(N)	求N的基數為10的對數
POW(X,Y) , POWER(X,Y)	求X 的Y乘方的結果值
SQRT	求非負數X 的二次方根
PI()	求圓周率
SIN(N), COS(N), TAN(N)	求N的三角函數
ASIN(N), ACOS(N), ATAN(N)	求 N的反三角函數
ATAN(Y,X) , ATAN2(Y,X)	求X/Y兩個變數X 及Y的反正切
COT(N)	求N的餘切
RAND(N)	產生一個隨機浮點值v，範圍在0到1 之間(其範圍為 $0 \leq v \leq 1.0$)
LEAST(N1,N2,N3,...)	當有2或多個參數時，返回值為最小(最小值的)參數
GREATEST(N1,N2,N3,...)	當有2或多個參數時，返回值為最大(最大值的)參數
DEGREES(N)	將徑度轉成角度
RADIANS(N)	將角度轉成徑度
TRUNCATE(N,D)	將N的小數至第D位後無條件捨去

字串函數

表7-5 字串長用函數

函數名稱	說明
ASCII(str)	傳回值為字串str的最左字元的數值。假如str為空字串，則返回值為 0。假如str為 NULL，則返回值為 NULL
CONV(N, from_base,to_base)	不同進制轉換數字，從from_base到to_base
BIN(N)	N的二進制值的字串資料表示
OCT(N)	N的八進制值的字串資料表示
HEX(N)	N的十六進制值的字串資料表示
CHAR(N,...[USING charset])	將N轉換為對應的字元
CONCAT(S1,S2,...)	連接參數產生的字串
LENGTH(str)	傳回字串str 的長度
LOCATE(substr,str)	傳回字串substr在str第一次出現的位置是第幾個字
LPAD(str,len,padstr)	傳回字串str,其左邊由字串padstr填補到len 字元長度
LEFT(str,len)	傳回從字串str 開始的len 最左字元
RIGHT(str,len)	傳回字串str 開始，返回最右len 字元
SUBSTRING(str,pos,len)	傳回從字串str的第pos位置開始，取出len 長度的字串。
SUBSTRING_INDEX(str,delim,count)	將字串str用delim切割成數個字串，接著取出count長度的子串傳回。當count為正數時從字串開頭開始，當count為負數時，從字串結尾開始。
LTRIM(str)	傳回已移除字串str開頭的空白字串
RTRIM(str)	傳回已移除字串str結尾的空白字串
TRIM([{both leading trailing} [remstr] From]str)	移除str字串當中與remstr相同的字串，both 是開始與結尾都移除，leading是僅移除開頭，trailing是僅移除結尾，預設為both
SPACE(N)	產生N個空格
REPLACE(str,from_str,to_str)	Str字串當中有from_str出現的地方to_str來代替
REPEAT(str,count)	產生count個str字串

函數名稱	說明
REVERSE(str)	將字串反序輸出
INSERT(str,pos,len,newstr)	將str字串的第pos個字元之後，插入一個長度為len的newstr字串
ELT(N,str1,str2,...)	傳回依據N 選擇str1,str2當中的第N個字串
FIELD(str,str1,str2,...)	將str與後的字串進行比較，找到相同具傳回第幾個，不則回傳0
FIND_IN_SET(str,strlist)	strlist是字串集合，元素之間使用逗號做為分隔符號，找出str是strlist中的第幾個元素
LCASE(str) , LOWER(str)	將str中的字串轉成小寫
UCASE(str) , UPPER(str)	將str中的字串轉成大寫

了解DML – Select查詢語法

前面的單元我們已經將資料匯入資料表當中，因此接下來我們主要介紹查詢Select語法的應用。

在底下的語法中我們將搭配前面所建立的資料表「student」及「score」，您可以跟著一起執行所有範例，這樣會讓您的印象更為深刻，底下讓我們再來回覆一下它們的結構。

「student」資料表結構：

表7-6 student資料表結構

欄位名稱	型別	欄位說明
studno	Int	學號
name	char	姓名
phone	char	電話
address	varchar	地址

「score」資料表結構：

表7-7 score資料表結構

欄位名稱	型別	欄位說明
studno	int	學號
examno	tinyint	月考代號
chi	smallint	國文
eng	smallint	英文
math	smallint	數學

● Select –查詢記錄

Select的語法是在所有SQL語法當中最常被使用到的部份，因此對於該語法必須好好掌握了解，請看底下的語法介紹與範例說明。

CHAPTER
01

CHAPTER
02

CHAPTER
03

CHAPTER
04

CHAPTER
05

CHAPTER
06

CHAPTER
07

CHAPTER
08

CHAPTER
09

CHAPTER
10

CHAPTER
11

CHAPTER
12

CHAPTER
13

CHAPTER
14

CHAPTER
15

語法：（常用非完整語法）

```
Select [DISTINCT] { * | 欄位1, 欄位2, ... }
FROM 資料表名稱1, 資料表名稱2, ...
Where 條件
GROUP By 群組1, 群組2, ...
Having 條件
Order By 排序1 [ASC|DESC]
```

Select範例1：查詢student資料表，所有欄位與記錄。

語法：

```
Select * From student;
```

說明：

「*」可以用來代表全部欄位，「;」用來表示敘述結尾符號，以便執行敘述。

Select 範例2：查詢student中部份欄位及所有記錄。

語法：

```
Select name, phone From student;
```

Select 範例3：查詢student中部份欄位及開頭前5筆記錄。

語法：

```
Select name, phone, address From student Limit 5;
```

Select 範例4：查詢student中所有欄位及學號為850315。

語法：

```
Select * From student Where studno=850315;
```

Select 範例5：查詢student中所有欄位及學號在850310至850313範圍。

語法：

```
Select * From student Where studno Between 850310 AND 850313;
```

Select 範例6：查詢student中所有欄位及學號分別為850301,850312,850323。

語法：

```
Select * From student Where studno IN(850301,850312,850323);
```

Select 範例7：查詢student中所有欄位及學號不是850312,850327。

語法：

```
Select * From student Where studno NOT IN(850312,850327);
```


Select 範例8：查詢student中所有欄位以及姓「王」的同學。

語法：

```
Select * From student Where name LIKE '王%';
```

Select 範例9：查詢student中所有欄位以及名字當中有「一」的同學。

語法：

```
Select * From student Where name LIKE '%一%';
```

Select 範例10：查詢student中所有欄位以及名字最後一個字是「花」的同學。

語法：

```
Select * From student Where name LIKE '%花';
```

Select 範例11：查詢score資料表中，每一位同學每次月考總分。

語法：

```
Select studno,examno,chi,eng,math,(chi+eng+math) as '總分' From score;
```

說明：

真實欄位名稱 AS 別名；別名可套用在欄位名稱、資料表名稱上。

Select 範例12：查詢score資料表中有多少筆記錄。

語法：

```
Select COUNT(studno) From score;
```

說明：

COUNT(?)為彙總函數，用來計算資料筆數，「?」可以是單一欄位名稱或改「*」代表所有欄位，請參考表7-8所示。

Select 範例13：查詢score資料表中第一次月考國文最高分。

語法：

```
Select MAX(chi) From score Where examno=1;
```

Select 範例14：查詢score資料表中第一次月考成績，依國文由大到小排序。

語法：

```
Select * From Score Where examno=1 ORDER BY chi DESC;
```

說明：

ASC 為遞增排序，是預設的排序方式，DESC 為遞減排序，必須明確表達。

Select 範例15：查詢第一次月考成績最差的前5名，分別顯示姓名、電話、總分。

語法：

```
Select name,phone,(chi+eng+math) as '總分' From student,score
Where examn o=1 && student.studno=score.studno ORDER BY 3 Limit 5;
```

說明：

關聯資料查詢必須讓資料表與資料表之間有相關欄位做為關聯依據。一般可以採用INNER JOIN的語法來替代 WHERE的語法。

Select 範例15：查詢第一次月考成績最差的6～10名，分別顯示姓名、電話、總分。

語法：

```
Select name,phone,(chi+eng+math) as '總分' From student,score Where
examn o=1 && student.studno=score.studno ORDER BY 3 Limit 5,5;
```

說明：

Limit 起始索引（起始索引為0）,讀取筆數。這是MySQL獨特的用法，它相當於一般Select TOP(N)的用法。

Select 範例17：查詢2次月考成績的總合。

語法：

```
Select studno,SUM(chi+eng+math) From score Group By studno;
```

Select 範例18：查詢2次月考成績總合超過500分的同學。

語法：

```
Select studno,SUM(chi+eng+math) From score Group By studno Having
SUM(chi+eng+math) > 500;
```

說明：

Having子句必須搭配Group By子句才能夠使用，也就先分完群組之後，再進行條件過濾，最後將符合條件的同學找出來。

彙總函數

表7-8 彙總函數

函數名稱	說明
COUNT(expr)	COUNT主要用來計算記錄筆數
AVG(expr)	AVG主要用來計算群組平均值
SUM(expr)	SUM主要用來計算群組總合
MIN(expr)	MIN主要用來找出群組中最小值
MAX(expr)	MAX主要用來找出群組中最大值
STD(expr)	STD主要用來計算群組中的標準差

了解DML – 資料異動語法

前面已經介紹了DML – **Select**的查詢語法，接下來我們要來介紹資料異動的相關語法，這三大語法也是不可或缺的重要角色，因為它讓我們可以對資料表當中的記錄，進行新增、修改、刪除等工作。

● INSERT – 新增記錄

Insert Into主要的目的是可以將記錄加入至已事先建立好的資料表當中，請看底下的語法介紹與範例說明。

語法：（常用非完整語法）

```
INSERT [LOW_PRIORITY ]  
      [INTO] tbl_name [(col_name,...)]  
      VALUES ((expression | DEFAULT),...), (...),...
```

說明：

LOW_PRIORITY：若有其它用戶端連線到MySQL伺服器上，會等待所有的連線終止後，才會執行Insert的動作。

另外在MySQL當中新增資料有一些常見的資料型態概念要說明，請看底下的介紹：

字串型態：

MySQL的字串可以使用「**’**」單引號、「**”**」雙引號表示。

數字型態：

在宣告可以存放小數點的型態中填入整數資料，MySQL會自動將整數轉換為小數點的資料型態。

日期時間：

在宣告可以存放日期(**Date**)或時間(**Time**)的型態中填入資料時，請使用字串搭配固定格式來使用。

日期格式：「**’**年-月-日**’**」或「**”**年-月-日**”**」

例如：「**’98.12.31’**」、「**’98-12-31’**」、「**’98/12/31’**」和「**’98@12@31’**」皆可。

時間格式：「**’**時:分:秒**’**」或「**”**時:分:秒**”**」且使用**24**小時制。

例如：「**’8:25:13’**」、「**’08:25:13’**」、「**’082513’**」皆可。

CHAPTER
01

CHAPTER
02

CHAPTER
03

CHAPTER
04

CHAPTER
05

CHAPTER
06

CHAPTER
07

CHAPTER
08

CHAPTER
09

CHAPTER
10

CHAPTER
11

CHAPTER
12

CHAPTER
13

CHAPTER
14

CHAPTER
15

範例1：部份欄位單筆記錄新增

語法：

```
INSERT INTO score (studno, examno) VALUES(850301,3);
```

說明：

(欄位清單) VALUES(值清單)，所以欄位清單的順序必須與值清單的順序搭配使用。

範例2：全部欄位單筆記錄新增

```
INSERT INTO score VALUES(850302,3,85,97,68);
```

說明：

因為新增所有欄位的內容，所以可以省略掉欄位清單，但是值清單的順序必須與當初建立資料表結構時的順序是相同的。

範例3：全部欄位連續多筆記錄新增

```
INSERT INTO score VALUES(850303,3,79,68,93),  
(850304,3,98,76,82),(850305,3,87,77,69);
```

說明：

一次新增3筆記錄。連續多筆記錄新增可以搭配單一INSERT敘述，每一筆記錄中間使用逗號進行分隔。

與INSERT敘述有一個類似的語法「REPLACE」，但不一樣的地方在於使用INSERT新增一筆記錄時，若欄位的值重複了，則INSERT動作會被忽略不執行；但是如果使用REPLACE語法時，則欄位的內容會被取代掉。

語法：（常用非完整語法）

```
REPLACE [LOW_PRIORITY]  
        [INTO] tbl_name [(col_name,...)]  
        VALUES ({expr | DEFAULT},...),(...),...
```

說明：

LOW_PRIORITY：若是有其它用戶端連線到MySQL伺服器上，會等待所有的連線終止後，才會執行Insert的動作。

● UPDATE –修改記錄

UPDATE可以修改資料表中已經存在的記錄，請看底下的語法介紹與範例說明。

語法：（常用非完整語法）

```
UPDATE [LOW_PRIORITY] tbl_name
      SET col_name1=expr1 [, col_name2=expr2 ...]
      [WHERE where_definition]
      [ORDER BY ...]
      [LIMIT row_count]
```

說明：

LOW_PRIORITY：若是有其它用戶端連線到MySQL伺服器上，會等待所有的連線終止後，才會執行Insert的動作。

範例1：修改學號850301同學的第3次月考成績。

語法：

```
UPDATE score SET chi=86, eng=93, math=79 WHERE studno=850301 &&
examno=3;
```

說明：

SET 欄位1=新值, 欄位2=新值, …，可以將欄位重新指派新的值，並且應該要搭配WHERE條件過濾改變適當的記錄內容；但是如果沒有搭配WHERE條件則會導致所有記錄的內容被修改成為同樣的值。

範例2：修改學號850301同學的第1,2次月考國文成績為60分。

語法：

```
UPDATE score SET chi=60 WHERE studno=850301 LIMIT 2;
```

說明：

因為單一studno=850301的條件會過濾出不只2筆記錄，因此必搭配LIMIT？可以確定只修改前2筆記錄的值。

● DELETE –刪除記錄

DELETE可以用來刪除已經儲存在資料表當中的記錄，請看底下的語法介紹與範例說明。

語法：（常用非完整語法）

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name
      [WHERE where_definition]
      [ORDER BY ...]
      [LIMIT row_count]
```

說明：

LOW_PRIORITY：若是有其它用戶端連線到MySQL伺服器上，會等待所有的連線終止後，才會執行Insert的動作。

QUICK：在刪除過程中，儲存引擎不會合併索引端結點，這樣可以加快部分種類的刪除操作的速度。

IGNORE：使MySQL忽略所有的錯誤。

範例1：刪除學號850301同學的第3次月考成績。

語法：

```
DELETE FROM score WHERE studno= 850301 && examno=3;
```

說明：

刪除記錄時請務必搭配WHERE條件進行過濾，因為如果沒有加上WHERE子句則會導致整資料表的記錄被刪除，這是非常嚴重的一件事，請一定要特別小心僅甚使用才可以。

範例2：刪除有第3次月考成績同學的前2筆記錄。

語法：

```
DELETE FROM score WHERE examno=3 LIMIT 2;
```

說明：

如果搭配前面的INSERT範例，在使用WHERE examno=3的條件時，可以查詢共有4筆記錄，搭配LIMIT 2可將查詢出來的前2筆記錄刪除。

最後還是請同能夠反覆的重複多加練習，這樣一來就更能夠熟練的操控資料庫，更能快速的解決所有遇的困難，加油！

自我測驗 & 能力檢核

完成每一項測驗後，請在□中打勾、代表完成您已了解。

☆SQL語法介紹—請試著完成底下每道練習

☐我已了解

1. 請分別在前堂課中所建立的MyDB資料庫中，分別為「personal」、「friend」資料表增加6筆記錄。

personal資料表：

AA20090601	李大明	062139872	1977-01-29	M	175	69
AA20090602	王平洋	062937865	1973-11-16	M	179	73
AA20090603	陳居明	063359870	1975-09-23	M	177	70
AB20090604	楊美滿	062200930	1978-09-08	F	160	43
AB20090605	陳幸好	062316720	1979-10-12	F	157	45
AB20090606	劉美麗	062244990	1979-03-11	F	155	40

friend資料表：

AA20090601	1	柯藥中	0911221229	1977-05-27	zhong@yahoo.com	台南市中區中正路121號
AA20090602	2	許太平	0912618732	1974-06-09	taiping@hotmail.com	台南中區市成功路路158號
AA20090603	3	陶正直	0913937376	1975-12-27	zhengzh@gmail.com	台南市北區文賢路61弄11號
AA20090604	4	王明月	0973214603	1979-10-25	ming@gmail.com	台南市安平區觀月路一段7號
AA20090605	5	李曉娟	0988112211	1980-02-28	xaio_juan@yahoo.com	台南市東區裕農路一段62號
AA20090606	6	李綿	0972373615	1979-12-07	lee_mian@pchome.com.tw	台南市北區臨安路二段17號

2. 請找出personal資料表中，sex欄位為F的記錄。

答：

CHAPTER
01

CHAPTER
02

CHAPTER
03

CHAPTER
04

CHAPTER
05

CHAPTER
06

CHAPTER
07

CHAPTER
08

CHAPTER
09

CHAPTER
10

CHAPTER
11

CHAPTER
12

CHAPTER
13

CHAPTER
14

CHAPTER
15

3. 請找出personal資料表中，height欄位介於150 ~ 160之間的記錄。
答：
4. 請找出friend資料表中，所有住在「北區」的記錄。
答：
5. 請找出friend資料表中，年齡最大的前2筆記錄。
答：
6. 請將personal中ID分為AA20090601, AB20090605的friend資料表中對應的好朋友記錄列出來。
答：
7. 請將personal中的「楊美滿」修改為「楊幸福」。
答：
8. 請將friend資料表中，將年齡最大的前2筆記錄ID欄位值修改為AA20090601。
答：