



# ● SQL的查詢語言

1

# SQL的基本語法

01	SELECT 欄位串列
02	FROM 資料表名稱
03	[WHERE <條件式>]
04	[GROUP BY <群組欄位> ]
05	[HAVING <群組條件>]
06	[ORDER BY <欄位> [ASC   DESC]]
07	[LIMIT 限制顯示筆數]

## 【說明】

- 行號01：**Select**後面要接所要列出的欄位名稱。
- 行號02：**From**後面接資料表名稱，它可以接一個以上的資料表。
- 行號03：**Where**後面要接條件式(它包括了各種運算子)
- 行號04：**Group By**欄位1,欄位2,...,欄位n，它可單獨存在，用來將數個欄位組合起來，以做為每次動作的依據。

## 說明

- 行號05：Having 條件式，是用來將數個欄位中加以有條件的組合。它不可以單獨存在，必須要搭配Group By。
- 行號06：Order By 欄位1,欄位2,...,欄位n [Asc|Desc]，它是依照某一個欄位來進行排序。例如：
  - (1) ORDER BY 成績 Asc ←可以省略(由小至大)
  - (2) ORDER BY成績 Desc ←不可以省略(由大至小)
- 行號07：LIMIT是指用來限制顯示筆數。

## 12-1-2 建立學生選課資料庫

- 在本單元中，為了方便撰寫SQL語法所須要的資料表，我們以「學生選課系統」的資料庫系統為例，利用phpMyAdmin資料庫管理工具，來建立三個資料表，分別為：學生表、課程表及選課表。

# 一、學生表

學生表 (5×5)				
學號	姓名	性別	電話	地址
S001	一心	男	1111111	前金區
S002	二聖	男	2222222	小港區
S003	三多	女	3333333	新興區
S004	四維	男	4444444	三民區
S005	五福	女	5555555	小港區

## 二、課程表

課程表 (3×8)		
課號	課名	學分數
C001	程式設計	4
C002	計算機概論	3
C003	資料結構	3
C004	系統分析	4
C005	資料庫系統	4
C006	數位學習	3
C007	手機程式	3
C008	機器人程式	3

### 三、選課表

選課表 (3×10)		
學號	課號	成績
S001	C001	56
S001	C005	73
S002	C002	92
S002	C005	63
S003	C004	92
S003	C005	70
S004	C003	75
S004	C004	88
S004	C005	68
S005	C005	95



## 12-2 使用SELECT子句

### ○【定義】

- **Select**是指在資料表中，選擇全部或部份欄位顯示出來，這就是所謂的「投影運算」。

### ○【格式】

Select	欄位串列
From	資料表名稱

# 查詢全部欄位

## ○【定義】

- 是指利用**SQL**語法來查詢資料表中的資料時，可以依照使用者的權限及需求來查詢所要看的資料。如果沒有指定欄位的話，我們可以直接利用星號「\*」代表所有的欄位名稱。

## ○【優點】不需輸入全部的欄位名稱。

## ○【缺點】

- 1.無法隱藏私人資料。
- 2.無法自行調整欄位順序。
- 3.無法個別指定欄位的別名。

# 實例

## 在「學生表」中顯示「所有學生基本資料」

解答▶▶

SQL指令

```
use ch12_db;  
  
SELECT *  
FROM 學生表
```

學生表 (5×5)

 學號	姓名	性別	 電話	地址
S001	一心	男	1111111	前金區
S002	二聖	男	2222222	小港區
S003	三多	女	3333333	新興區
S004	四維	男	4444444	三民區
S005	五福	女	5555555	小港區

## 查詢指定欄位(垂直篩選)

### ○【定義】

- 由於上一種方法，只能直接選擇全部的欄位資料，無法顧及隱藏私人資料及自行調整欄位順序的問題，因此，我們利用指定欄位來查詢資料。

### ○【優點】

- 1.顧及私人資料。
- 2.可自行調整欄位順序。
- 3.可以個別指定欄位的別名。

### ○【缺點】

- 如果確定要顯示所有欄位，則必須花較多時間輸入。

# 實例

在「學生表」中查詢所有學生的「姓名、性別及地址」 《參見12-1-2》

解答▶▶

SQL指令

```
use ch12_db;
```

```
SELECT 姓名,性別,地址
```

```
FROM 學生表
```

欄位與欄位名稱之間，必須要以逗號「，」隔開

學生表 (3×5)

姓名	性別	地址
一心	男	前金區
二聖	男	小港區
三多	女	新興區
四維	男	三民區
五福	女	小港區

## 12-2-3 使用「別名」來顯示

### ○【定義】

- 使用**AS**運算子之後，可以使用不同名稱顯示原本的欄位名稱。

### ○【表示式】

- 原本的欄位名稱 **AS** 別名
- 《**AS**可省略不寫，只寫「別名」》

### ○【舉例】

- 課名 **AS** 課程名稱 或寫成→ 課名 課程名稱

## ○【注意】

- AS只是暫時性地變更列名，並不是真的會把原本的名稱覆蓋過去。

## ○【適用時機】

- 1.欲「合併」的資料表較多並且名稱較長時。
- 2.一個資料表扮演多種不同角色(自我合併)。
- 3.暫時性地取代某個欄位名稱(課名 AS 課程名稱)

# 替代欄位名稱字串

❖表12-1 SQL中的替代欄位名稱字串

替代字元	功能	語法
AS	設定別名	Select 課名 AS 課程名稱



# 範例 1

- 在「課程表」中將所有課程的「課名」設定別名為「課程名稱」之後，再顯示「課程名稱、學分數」

## 解答▶▶

### SQL指令

```
use ch12_db;  
SELECT 課名 AS 課程名稱,學分數  
FROM 課程表
```

利用AS來設定欄位的別名

課程表 (2×8)

課程名稱	學分數
程式設計	4
計算機概論	3
資料結構	3
系統分析	4
資料庫系統	4
數位學習	3
手機程式	3
機器人程式	3

## 範例 2


- 在「選課表」中將所有學生的「成績」各加5分，並且設定別名為「調整後成績」之後，再顯示「學號，課號，成績，調整後成績」。

## SQL指令

```
use ch12_db;
```

```
SELECT 學號, 課號, 成績, 成績+5 AS 調整後成績
```

```
FROM 選課表
```

 選課表 (4×10)

 學號	 課號	成績	調整後成績
S001	C001	56	61
S001	C005	73	78
S002	C002	92	97
S002	C005	63	68
S003	C004	92	97
S003	C005	70	75
S004	C003	75	80
S004	C004	88	93
S004	C005	68	73
S005	C005	95	100

## 使用「比較運算子條件」

- 如果我們所想要的資料是要符合某些條件，而不是全部的資料時，那就必須要在**Select**子句中再使用**Where**條件式即可。
- 並且也可以配合使用「比較運算子條件」來搜尋資料。若條件式成立的話則會傳回「**True(真)**」，若不成立的話則會傳回「**False(假)**」。

# 語法

SQL指令
Select 欄位集合
From 資料表名稱
<b>Where</b> 條件式

❖表12-2 比較運算子表

運算子	功能	例子	條件式說明
=（等於）	判斷A與B是否相等	A=B	成績=60
◇（不等於）	判斷A是否不等於B	A◇B	成績◇60
<（小於）	判斷A是否小於B	A<B	成績<60
<=（小於等於）	判斷A是否小於等於B	A<=B	成績<=60
>（大於）	判斷A是否大於B	A>B	成績>60
>=（大於等於）	判斷A是否大於等於B	A>=B	成績>=60

註：設A代表「成績欄位名稱」，B代表「字串或數值資料」。

## 查詢滿足條件的值組(水平篩選)

### ○【定義】

- 當我們所想要的資料是要符合某些條件，而不是全部的資料時，那就必須要在**Select**子句中再使用**Where**條件式即可。

### ○【優點】

- 1.可以依照使用者的需求來查詢。
- 2.資訊較為集中。



## 範例

在「選課表」中查詢課號為「C005」的學生的  
「學號及成績」

解答▶▶

SQL指令

```
use ch12_db;  
SELECT 學號, 成績  
FROM 選課表  
WHERE 課號='C005'
```

選課表 (2×5)	
學號	成績
S001	73
S002	63
S003	70
S004	68
S005	95

## 12-3-2 查詢比較大小的條件

### ○【定義】

- 當我們所想要的資料是要符合某些條件，例如：顯示出及格或不及格的學生名單等情況。此時，我們就必須要在Where 條件式中使用「比較運算子」來篩選。

## 【實例】

在「選課表」中查詢任何課程成績「不及格60」  
的學生的「學號、課號及成績」

解答>>

SQL指令

```
use ch12_db;  
SELECT 學號, 課號, 成績  
FROM 選課表  
WHERE 成績<60
```

選課表 (3×1)		
學號	課號	成績
S001	C001	56

## 12-4 使用「邏輯比較運算子條件」

- 在Where條件式中除了可以設定「比較運算子」之外，還可以設定「邏輯運算子」來將數個「比較運算子」條件組合起來，成為較複雜的條件式。

❖表12-3 邏輯運算子表

運算子	功能	條件式說明
And（且）	判斷A且B兩個條件式是否皆成立	成績 $\geq$ 60 And 課號='C005'
Or（或）	判斷A或B兩個條件式是否有一個成立	課號='C004' Or 課號='C005'
Not（反）	非A的條件式	Not 成績 $\geq$ 60

註：設A代表「左邊條件式」，B代表「右邊條件式」。

## AND(且)

- 【定義】 判斷A且B兩個條件式是否皆成立。

【實例】在「選課表」中查詢修課號為「C005」且成績是「及格60分」的學生的「學號及成績」

解答▶▶

SQL指令

```
use ch12_db;  
SELECT 學號,成績  
FROM 選課表  
WHERE 成績>=60 And 課號='C005'
```

選課表 (2×5)	
學號	成績
S001	73
S002	63
S003	70
S004	68
S005	95

## OR(或)

- 【定義】判斷A或B兩個條件式是否至少有一個成立。



## 【實例】

- 在「選課表」中查詢學生任選一科「課號為C004 或 課號為C005」的學生的「學號、課號及成績」

## 解答▶▶

### SQL指令

```
use ch12_db;  
SELECT 學號,課號,成績  
FROM 選課表  
WHERE 課號='C004' Or 課號='C005'
```

 選課表 (3×7)

 學號	 課號	成績
S001	C005	73
S002	C005	63
S003	C004	92
S003	C005	70
S004	C004	88
S004	C005	68
S005	C005	95

## Not(反)

- 【定義】當判斷結果成立時，則變成不成立。  
而判斷結果不成立時，則變成成立。

## 【實例】在「選課表」中，查詢有修課號為C001且成績不及格的學生的「學號及成績」

解答>>

SQL指令

```
use ch12_db;  
SELECT 學號, 成績  
FROM 選課表  
WHERE 課號='C001' And Not 成績>=60
```

選課表 (2×1)	
學號	成績
S001	56

## 12-4-4 IS NULL(空值)

- 【定義】NULL值是表示沒有任何的值(空值)，在一般的資料表中有些欄位中並沒有輸入任何的值。例如：學生月考缺考，使用該科目成績是空值。

# 【實例1】在「選課表」中查詢那些學生「缺考」的「學號、課號及成績」。

解答>>>

SQL指令

```
use ch12_db;  
SELECT 學號, 課號, 成績  
FROM 選課表  
WHERE 成績 IS NULL
```

設定IS NULL條件，其回傳的值True或False

**注意** 這裡的「IS」不能用等號（=）代替它。

執行結果>>>

選課表 (3×0)		
學號	課號	成績

## 【實例2】在「選課表」中查詢那些學生「沒有缺考」的「學號、課號及成績」

解答▶▶

SQL指令

```
use ch12_db;  
SELECT 學號, 課號, 成績  
FROM 選課表
```

```
WHERE 成績 IS NOT NULL
```

設定IS NOT NULL條件

選課表 (3×10)		
學號	課號	成績
S001	C001	56
S001	C005	73
S002	C002	92
S002	C005	63
S003	C004	92
S003	C005	70
S004	C003	75
S004	C004	88
S004	C005	68
S005	C005	95

## 使用「模糊條件與範圍」

### ○【定義】

- 在Where條件式中除了可以設定「比較運算子」與「邏輯運算子」之外，還可以設定「模糊或範圍條件」來查詢。

### ○【例如】

- 奇摩的搜尋網站，使用者只要輸入某些關鍵字，就可以即時查詢出相關的資料。



# 【模糊或範圍運算子表】

❖表12-4 模糊或範圍運算子表

運算子	功能	條件式說明
Like	模糊相似條件	Where 系所 LIKE '資管%'
IN	集合條件	Where 課號 IN('C001','C002')
Between……And	範圍條件	Where 成績 Between 60 And 80

## LIKE模糊相似條件

- 【定義】LIKE運算子利用萬用字元( % 及 \_ )來比較相同的內容值。
  - (1)萬用字元( % )百分比符號代表零個或一個以上的任意字元；
  - (2)萬用字元( \_ )底線符號代表單一個數的任意字元。

# LIKE模糊相似條件的萬用字元比較表

❖表12-5 Like模糊相似條件的萬用字元比較表

撰寫SQL語法環境	MariaDB
比對一個字元	「_」
比對多個字元	「%」

## 【以MySQL的環境為例】

1. Select \*

意義：「\*」 代表在資料表中的所有欄位

2. WHERE 姓名 Like '李%'

意義：查詢姓名**開頭**為 '李' 的所有學生資料

3. WHERE 姓名 Like '%李'

意義：查詢姓名**結尾**為 '李' 的所有學生資料

4. WHERE 姓名 Like '%李%'

意義：查詢姓名**含有**為 '李' 的所有學生資料

5. WHERE 姓名 Like '李\_\_'

意義：查詢姓名中姓 '李' 且 3 個字的學生資料

# 【實例】在「學生表」中查詢「地址」開頭為「小」的學生基本資料

解答▶▶

SQL指令

```
use ch12_db;  
SELECT *  
FROM 學生表  
WHERE 地址 Like '小%'
```

學生表 (5×2)				
學號	姓名	性別	電話	地址
S002	二聖	男	2222222	小港區
S005	五福	女	5555555	小港區

# IN集合條件

## ○【定義】

- IN為集合運算子，只要符合集合之其中一個元素，將會被選取。

## ○【使用時機】

- 篩選的對象是兩個或兩個以上。

## 【實例 1】

- 在「選課表」中查詢學生任選一個「課號為C004 或 課號為C005」的學生的「學號、課號及成績」

## 解答▶▶

### SQL指令

```
use ch12_db;  
SELECT 學號, 課號, 成績  
FROM 選課表  
WHERE 課號 In ('C004','C005')
```

使用IN時可以在括號中設定好幾個值

註：以上的WHERE 課程代號 In ('C004','C005')亦可寫成如下：

```
WHERE 課號='C004' OR 課號='C005'
```



## 執行結果

選課表 (3×7)		
學號	課號	成績
S001	C005	73
S002	C005	63
S003	C004	92
S003	C005	70
S004	C004	88
S004	C005	68
S005	C005	95

## 範例 2

- 請在「學生表」中，列出學號為S001~S003的同學之「學號」、「姓名」。

## 解答>>

### SQL指令

```
use ch12_db;  
SELECT 學號,姓名  
FROM 學生表  
WHERE 學號 In ('S001', 'S002', 'S003')
```

## 執行結果>>

學生表 (2×3)	
學號	姓名
S001	一心
S002	二聖
S003	三多

## 12-5-3 BETWEEN / AND範圍條件

### ○【定義】

- Between/And是用來指定一個範圍，表示資料值必須在最小值(含)與最大值(含)之間的範圍資料。註：等同於「 $\geq$ 最小值 And 最大值 $\leq$ 」

# 【實例】在「選課表」中查詢成績60到90之間的學生的「學號、課號及成績」

解答▶▶

SQL指令

```
use ch12_db;
```

```
SELECT 學號, 課號, 成績
```

```
FROM 選課表
```

```
WHERE 成績 Between 60 And 90
```

等同於

成績 >= 60 And 成績 <= 90

選課表 (3×6)		
學號	課號	成績
S001	C005	73
S002	C005	63
S003	C005	70
S004	C003	75
S004	C004	88
S004	C005	68

## ●● 隨堂練習 ●●

**Q 1** 在「選課資料表」中查詢修課號為C004或C005的成績60到90之間的學生的「學號」、「課號」及「成績」（利用Between/And）。

**A**

### SQL指令

```
use ch12_db;  
SELECT 學號, 課號, 成績  
FROM 選課表  
WHERE 課號 In ('C004','C005') AND 成績 Between 60 And 90
```

執行結果：

選課表 (3×5)		
學號	課號	成績
S001	C005	73
S002	C005	63
S003	C005	70
S004	C004	88
S004	C005	68

**Q2** 在「選課表」中查詢修課號為C004或C005的成績60到90之間的學生的「學號」、「課號」及「成績」（利用比較運算式）。

**A**

SQL指令

```
use ch12_db;  
SELECT 學號, 課號, 成績  
FROM 選課表  
WHERE 課號 In ('C004','C005') AND 成績>=60 And 成績<=90;
```

## 使用「算術運算子」

### ○【定義】

- 在Where條件式中還提供算術運算的功能，讓使用者可以設定某些欄位的數值作四則運算。



# 【算術運算子表】

❖ 表12-6 算術運算子表

運算子	功能	例子	執行結果
+(加)	A與B兩數相加	14+28	42
-(減)	A與B兩數相減	28-14	14
*(乘)	A與B兩數相乘	5*8	40
/(除)	A與B兩數相除	10/3	3.33333333...
%(餘除)	A與B兩數相除後，取餘數	10%3	1

## 【範例】

- 在「選課表」中查詢學生成績乘1.2倍後還尚未達70分的學生顯示「學號、課號及成績」

解答>>

SQL指令

```
use ch12_db;  
SELECT 學號, 課號, 成績  
FROM 選課表  
WHERE 成績*1.2<70
```

執行結果>>

選課表 (3×1)		
學號	課號	成績
S001	C001	56

## 12-7 使用「聚合函數」

### ○【定義】

- 在SQL中提供聚合函數來讓使用者統計資料表中數值資料的最大值、最小值、平均值及合計值等等。其常用的聚合函數的種類如下表所示。

# 【聚合函數表】

❖表12-7 聚合函數表

聚合函數	說明
Count(*)	計算個數函數
Count(欄位名稱)	計算該欄位名稱之不具NULL值列的總數
Avg	計算平均函數
Sum	計算總和函數
Max	計算最大值函數
Min	計算最小值函數

## 記錄筆數(COUNT)

- 【定義】COUNT函數是用來計算橫列記錄的筆數。

# 【實例1】在「學生表」中查詢目前選修課程的全班人數

解答>>

SQL指令

```
use ch12_db;  
SELECT Count(*) AS 全班人數  
FROM 學生表
```

執行結果>>

結果 #1 (1×1)	
全班人數	
5	

## ● ● 隨堂練習 ● ●

**Q 1** 在「選課表」中查詢已經選課的「筆數」。

**A**

SQL指令

```
use ch12_db;  
SELECT Count(*) AS 全班人數  
FROM 選課表;
```

執行結果：

結果 #1 (1×1)	
全班人數	
10	

**Q2** 在「選課資料表」中查詢已經有「成績」紀錄的筆數。

**A**

#### SQL指令

```
use ch12_db;  
SELECT Count(成績) AS 有成績總筆數  
FROM 選課表;
```

執行結果：

結果 #1 (1×1)	
有成績總筆數	
10	

❖ 圖12-23

註：Count(欄位名稱)→計算該欄位名稱之不具NULL值列的總數。



## 平均數(AVG)

- 【定義】 AVG函數用來傳回一組記錄在某欄位內容值中的平均值。

## 【實例】在「選課表」中查詢有選修「課號為C005」的全班平均成績

解答▶▶

SQL指令

```
use ch12_db;  
SELECT AVG(成績) AS 資料庫平均成績  
FROM 選課表  
WHERE 課號='C005'
```

執行結果▶▶

結果 #1 (1×1)	
資料庫平均成績	
73.8000	

## ●● 隨堂練習 ●●

**Q2** 在「選課表」中計算每一位同學所修之科目的平均成績。

**A**

### SQL指令

```
use ch12_db;  
SELECT 學號, AVG(成績) AS 平均成績  
FROM 選課表  
GROUP BY 學號
```

執行結果：

選課表 (2×5)	
學號	平均成績
S001	64.5000
S002	77.5000
S003	81.0000
S004	77.0000
S005	95.0000

## 12-7-3 總和(SUM)

- 【定義】SUM函數是用來傳回一組記錄在某欄位內容值的總和。

## 【實例】在「選課表」中查詢有選修「課號為C005」的全班總成績

解答>>

SQL指令

```
use ch12_db;  
SELECT SUM(成績) AS 資料庫總成績  
FROM 選課表  
WHERE 課號='C005'
```

執行結果>>

結果 #1 (1×1)	
資料庫總成績	
	369

## ● ● 隨堂練習 ● ●

**Q 1** 在「選課表」中查詢「學號為S001」的各科總成績。

**A**

### SQL指令

```
use ch12_db;  
SELECT SUM(成績) AS 總成績  
FROM 選課表  
WHERE 學號='S001';
```

執行結果：

結果 #1 (1×1)	
總成績	
129	

# 最大值(MAX)

- 【定義】MAX函數用來傳回一組記錄在某欄位內容值中的最大值。

# 【實例】在「選課表」中查詢有選修「課號為C005」的全班成績最高分

解答▶▶

SQL指令

```
use ch12_db;  
SELECT MAX(成績) AS 資料庫最高分  
FROM 選課表  
WHERE 課號='C005'
```

執行結果▶▶

結果 #1 (1×1)	
資料庫最高分	
95	



## ●● 隨堂練習 ●●

**Q1** 在「選課表」中查詢「成績介於60~80」中最高分為何？

**A**

### SQL指令

```
use ch12_db;  
SELECT MAX(成績) AS 資料庫成績介於至之最高分  
FROM 選課表  
WHERE 成績 Between 60 And 80;
```

執行結果：

結果 #1 (1×1)	
資料庫成績介於至之最高分	
	75

## 最小值(MIN)

- 【定義】MIN函數用來傳回一組記錄在某欄位內容值中的最小值。

## 【實例】在「選課表」中查詢有選修「課號為C005」的全班成績最低分

解答▶▶

SQL指令

```
use ch12_db;  
SELECT MIN(成績) AS 資料庫最低分  
FROM 選課表  
WHERE 課號='C005'
```

執行結果▶▶

結果 #1 (1×1)	
資料庫最低分	
63	

## ● ● 隨堂練習 ● ●

**Q** 在「選課表」中查詢「及格成績」中最低分為何？

**A**

### SQL指令

```
use ch12_db;  
SELECT MIN(成績) AS 「及格成績」中最低分  
FROM 選課表  
WHERE 成績 Between 60 And 100;
```

執行結果：

結果 #1 (1×1)	
「及格成績」中最低分	
	63

# 使用「排序及排名次」

## ○【定義】

- 雖然撰寫**SQL**指令來查詢所須的資料非常容易，但如果顯示的結果筆數非常龐大而沒有按照某一順序及規則來顯示，可能會顯得非常混亂。
- 還好**SQL**指令還有提供排序的功能。其常用的排序及排名次的子句種類如下表所示。

## 【排序及排名次函數表】

❖ 表12-8 排序及排名次函數表

排序及排名次指令	說明
ORDER BY成績 <u>Asc</u>	<u>Asc</u> ← 可以省略（由小至大）
ORDER BY成績 <u>Desc</u>	<u>Desc</u> ← 不可以省略（由大至小）

註：ASC：Ascending（遞增）

DESC：Descending（遞減）

# Asc遞增排序

- 【定義】資料記錄的排序方式是由小至大排列。

## 【實例】在「選課表」中查詢全班成績由低到高分排序

解答▶▶

SQL指令

```
use ch12_db;  
SELECT 學號, 課號, 成績  
FROM 選課表  
ORDER BY 成績 Asc
```



## 執行結果

選課表 (3×10)		
學號	課號	成績
S001	C001	56
S002	C005	63
S004	C005	68
S003	C005	70
S001	C005	73
S004	C003	75
S004	C004	88
S003	C004	92
S002	C002	92
S005	C005	95

## DESC遞減排序

- 【定義】資料記錄的排序方式是由大至小排列。

## 【實例】在「選課表」中查詢的全班成績由高到低分排序

解答▶▶

SQL指令

```
use ch12_db;  
SELECT 學號, 課號, 成績  
FROM 選課表  
ORDER BY 成績 DESC
```

執行結果▶▶

選課表 (3×10)		
學號	課號	成績
S005	C005	95
S002	C002	92
S003	C004	92
S004	C004	88
S004	C003	75
S001	C005	73
S003	C005	70
S004	C005	68
S002	C005	63
S001	C001	56

# 比較複雜的排序

- 【定義】指定一個欄位以上來做排序時，則先以第一個欄位優先排序，當資料相同時，則再進行第二個欄位進行排序，依此類堆。

【實例】在「選課表」中查詢結果按照學號昇冪排列之後，再依成績昇冪排列。

選課表 (3×10)		
學號	課號	成績
S001	C001	56
S001	C005	73
S002	C002	92
S002	C005	63
S003	C004	92
S003	C005	70
S004	C003	75
S004	C004	88
S004	C005	68
S005	C005	95

依學號  
(小→大)

依成績尚未排序

## 解答▶▶

### SQL指令

```
use ch12_db;  
SELECT *  
FROM 選課表  
ORDER BY 學號,成績
```

## 執行結果>>

選課表 (3×10)		
學號	課號	成績
S001	C001	56
S001	C005	73
S002	C005	63
S002	C002	92
S003	C005	70
S003	C004	92
S004	C005	68
S004	C003	75
S004	C004	88
S005	C005	95

依成績(低→高)

依學號  
(小→大)



## ●● 隨堂練習 ●●

**Q 1** 在「選課表」中查詢結果按照「學號」升冪排列之後，再依「成績」降冪排列（亦即由高分到低分）。

**A**

SQL指令

```
use ch12_db;  
SELECT *  
FROM 選課表  
ORDER BY 學號 ASC, 成績 DESC;
```

## ●● 隨堂練習 ●●

執行結果：

學號	課號	成績
S001	C005	73
S001	C001	56
S002	C002	92
S002	C005	63
S003	C004	92
S003	C005	70
S004	C004	88
S004	C003	75
S004	C005	68
S005	C005	95

依成績(高→低)

依學號  
(小→大)

## 12-8-4 LIMIT 限制顯示筆數

- 【定義】資料記錄在排序之後，取排名前N名。
- 【使用時機】總筆數已知，例如：全班10人中取前三名

# 【實例】在「選課表」中查詢有選修「課號為C005」的5個同學中成績前二名的同學

解答▶▶

SQL指令

```
use ch12_db;  
SELECT *  
FROM 選課表  
WHERE 課號='C005'  
ORDER BY 成績 DESC  
LIMIT 2
```

執行結果▶▶

選課表 (3×2)		
學號	課號	成績
S005	C005	95
S001	C005	73

## 12-9 使用「群組化」

### ○【定義】

- 利用SQL語言，我們可以將某些特定欄位的值相同的記錄全部組合起來，以進行群組化，接著就可以在這個群組內求出各種統計分析。

### ○【語法】 Group By欄位1,欄位2,...,欄位n [Having 條件式]

- (1)Group By 可單獨存在，它是將數個欄位組合起來，以做為每次動作的依據。
- (2) [Having 條件式]是將數個欄位中以有條件的組合。它不可以單獨存在。
- (3) WHERE子句與HAVING子句之差別

# WHERE子句與HAVING子句之差別

❖ 表12-9 WHERE子句與HAVING子句之差別

	WHERE子句	HAVING子句
執行順序	GROUP BY之前	GROUP BY之後
聚合函數	不能使用聚合函數	可以使用

## (4) SQL的執行順序

SQL 的執行順序		
①	FROM	指定所需表格，如兩個表格以上(含) 先卡氏積，再 JOIN
	↓	
②	ON	資料表 JOIN 的條件
	↓	
③	[Inner   Left   Right]	Join 資料表
	↓	
④	WHERE	找出符合指定條件的所有列，一般不含聚合函數
	↓	
⑤	GROUP BY	根據指定欄位來分群
	↓	
⑥	HAVING	找出符合指定條件的所有群組，都是利用聚合函數
	↓	
⑦	SELECT	指定欄位並輸出結果
	↓	
⑧	DISTINCT	列出不重複的記錄
	↓	
⑨	ORDER BY	排序
	↓	
⑩	<b>LIMIT N</b>	列出前 N 筆記錄

# GROUP BY 欄位

- 【定義】 Group By 可單獨存在，它是將數個欄位組合起來，以作為每次動作的依據。
- 【語法】

Select 欄位 1，欄位 2，聚合函數運算

From 資料表

Where 過濾條件

Group By 欄位 1，欄位 2

- 【說明】在Select的非聚合函數內容一定要出現在Group By中，因為先群組化才能Select。



## 實例1

在「選課表」中，查詢每一位同學各選幾門科目。

解答▶▶

SQL指令

```
use ch12_db;
```

```
SELECT 學號, Count(*) AS 選科目數
```

```
FROM 選課表
```

```
GROUP BY 學號
```

註：在Select所篩選的非聚合函數。例如：學號，一定會在Group By後出現。

執行結果▶▶

選課表 (2×5)	
學號	選科目數
S001	2
S002	2
S003	2
S004	3
S005	1

## 實例2

在「選課表」中計算每一位同學所修之科目的平均成績

解答>>

SQL指令

```
use ch12_db;  
  
SELECT 學號, AVG(成績) AS 平均成績  
FROM 選課表  
GROUP BY 學號
```

執行結果>>

選課表 (2×5)	
學號	平均成績
S001	64.5000
S002	77.5000
S003	81.0000
S004	77.0000
S005	95.0000

## 實例3

在「選課表」中，將每個課程的選修人數印出來，  
印出之結果並按「課號」由大到小排序

解答▶▶

### SQL指令

```
use ch12_db;  
SELECT 課號, Count(*) AS 選課學生人數  
FROM 選課表  
GROUP BY 課號  
ORDER BY 課號 DESC
```

執行結果▶▶

選課表 (2×5)	
課號	選課學生人數
C005	5
C004	2
C003	1
C002	1
C001	1

## 實例4

- 在「選課表」中，將每個課程的選修人數及該科最高分數印出來，印出之結果並按「課號」由小到大排序

## 解答▶▶

### SQL指令

```
use ch12_db;  
SELECT 課號, Count(*) AS 選課學生人數, MAX(成績) AS 最高分成績  
FROM 選課表  
GROUP BY 課號  
ORDER BY 課號
```

## 執行結果▶▶

選課表 (3×5)		
課號	選課學生人數	最高分成績
C001	1	56
C002	1	92
C003	1	75
C004	2	92
C005	5	95

## 實例5

- 在「選課表」中，將每個課程的選修人數及該科平均分數印出來，印出之結果並按「課號」由小到大排序

## SQL指令

```
use ch12_db;
SELECT 課號, Count(*) AS 選課學生人數, AVG(成績) AS 平均成績
FROM 選課表
GROUP BY 課號
ORDER BY 課號
```

## 執行結果▶▶

選課表 (3×5)		
課號	選課學生人數	平均成績
C001	1	56.0000
C002	1	92.0000
C003	1	75.0000
C004	2	90.0000
C005	5	73.8000

# HAVING 條件式

- 【定義】Having 條件式是將數個欄位中以有條件的組合。它不可以單獨存在。

選課表 (2×4)	
學號	平均成績
S002	77.5000
S003	81.0000
S004	77.0000
S005	95.0000



# 【實例1】在「選課表」中，計算所修之科目的平均成績，大於等於70者顯示出來。

解答>>

SQL指令

```
use ch12_db;  
SELECT 學號, AVG(成績) AS 平均成績  
FROM 選課表  
GROUP BY 學號  
HAVING AVG(成績)>=70
```

執行結果>>

選課表 (2×4)	
學號	平均成績
S002	77.5000
S003	81.0000
S004	77.0000
S005	95.0000

## 【實例2】

在「選課表」中，將選修課程在二科及二科以上的學生學號資料列出來。

解答>>

SQL指令

```
use ch12_db;  
SELECT 學號, Count(*) AS 選修數目  
FROM 選課表  
GROUP BY 學號  
HAVING COUNT(*)>=2
```

執行結果>>

選課表 (2×4)	
學號	選修數目
S001	2
S002	2
S003	2
S004	3

## WHERE子句與HAVING子句之差異

- 1. Where子句是針對尚未群組化的欄位來進行篩選。
- 2. HAVING子句則是針對已經群組化的欄位來取出符合條件的列。

## 12-10 使用「刪除重覆」

- 【定義】
- 利用**Distinct**指令來將所得結果有重覆者，去除重覆。若有一學生選了3門課程，其學號只能出現一次。

## 12-10-1 ALL(預設)使查詢結果的記錄可能重複

- 【定義】沒有利用Distinct指令
- 【實例】在「選課表」中，將有選修課程的學生之學號、課號印出來。
  - 註：沒有利用Distinct指令時，產生重覆出現的現象。

## 解答▶▶

### SQL指令

```
use ch12_db;  
SELECT 學號, 課號  
FROM 選課表
```

## 執行結果▶▶

選課表 (2×10)	
學號	課號
S001	C001
S001	C005
S002	C002
S002	C005
S003	C004
S003	C005
S004	C003
S004	C004
S004	C005
S005	C005

# DISTINCT使查詢結果的記錄不重複出現

## ○【定義】

- 如果使用DISTINCT子句，則可以將所指定欄位中重複的資料去除掉之後再顯示。
- 指定欄位的時候，可以指定一個以上的欄位，但是必須使用「,(逗點)」來區隔欄位名稱。

## ○【DISTINCT的注意事項】

- ①不允許配合COUNT(\*)使用
- ②允許配合COUNT(屬性)使用
- ③對於MIN()與MAX()是沒有作用的

## 範例

- 在「選課表」中，將有選修課程的學生之「學號」印出來。



## 解答▶▶

SQL指令	
<pre>use ch12_db; SELECT DISTINCT 學號 FROM 選課表</pre>	<div>相同</div> <pre>use ch12_db; SELECT 學號 FROM 選課表 GROUP BY 學號;</pre>

## 執行結果▶▶

選課表 (1×5)	
學號	
S001	
S002	
S003	
S004	
S005	

❖ 圖 12-48

註：利用Distinct指令時，刪除重複的現象。如果沒有指定Distinct指令時，則預設值為ALL，其查詢結果會重複。