



第6章

SELECT敘述的基本查詢

(大海撈針)

SQL結構化查詢語言

- SQL的完整名稱是Structured Query Language簡稱SQL 中文全名是結構化查詢語言，SQL語言是「ANSI」(American National Standards Institute)制定的標準資料庫語言，其版本分為：
 - 1989年的ANSI-SQL 89
 - 1992年制定的ANSI-SQL 92，也稱為SQL 2，這是目前關聯式資料庫的標準語言
 - 最新版ANSI-SQL 99稱為SQL 3，適用在物件關聯或導向式資料庫的SQL語言。

SQL結構化查詢語言的種類

- SQL語言依指令功能，可以分成三類
 - 資料定義語言DDL(Data Definition Language)
 - 建立資料表、視界和索引等的SQL指令
 - 資料操作語言DML(Data Manipulation Language)
 - 資料表記錄插入、刪除、更新和查詢指令
 - 資料控制語言DCL(Data Control Language)
 - 資料庫安全管理的權限設定指令

本章大綱

- 6-1 SELECT查詢指令
- 6-2 SELECT子句
- 6-3 FROM子句
- 6-4 WHERE子句
- 6-5 聚合函數的摘要查詢
- 6-6 群組查詢GROUP BY子句
- 6-7 排序ORDER BY子句



6-1 SELECT查詢指令-語法

- SELECT指令是DML指令中語法最複雜的一個，其基本語法如下所示：

SELECT 欄位清單

FROM 資料表來源

[WHERE 搜尋條件]

[GROUP BY 欄位清單]

[HAVING 搜尋條件]

[ORDER BY 欄位清單]

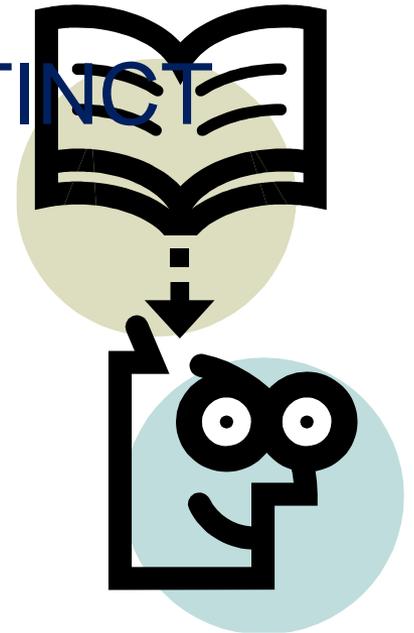
6-1 SELECT查詢指令-子句說明

- SELECT指令各子句的說明，如下表所示：

子句	說明
SELECT	指定查詢結果包含哪些欄位
FROM	指定查詢的資料來源是哪些資料表
WHERE	過濾查詢結果的條件，可以從資料表來源取得符合條件的查詢結果
GROUP BY	可以將相同欄位值的欄位群組在一起，以便執行群組查詢
HAVING	搭配 GROUP BY 子句來進一步過濾群組查詢的條件
ORDER BY	指定查詢結果的排序欄位

6-2 SELECT子句

- 6-2-1 資料表的欄位
- 6-2-2 欄位別名
- 6-2-3 計算值欄位
- 6-2-4 刪除重複記錄 - ALL與DISTINCT
- 6-2-5 前幾筆記錄 - TOP子句



6-2 SELECT子句-語法

- 在SELECT指令的SELECT子句是指定查詢結果包含哪些欄位，其語法如下所示：

SELECT [ALL | DISTINCT]

[TOP n | PERCENT] [WITH TIES]]

欄位規格 [[AS] 欄位別名] [, 欄位規格 [[AS] 欄位別名]]

- 上述ALL是預設值可以顯示所有記錄的欄位值，DISTINCT只顯示不重複欄位值的記錄，TOP關鍵字可以顯示查詢結果的前幾筆記錄或多少百分比。

6-2 SELECT子句-語法(M)

- 在SELECT指令的SELECT子句是指定查詢結果包含哪些欄位，其語法如下所示：

SELECT [ALL | DISTINCT]

[LIMIT n [OFFSET m] [WITH TIES]]

欄位規格 [[AS] 欄位別名] [, 欄位規格 [[AS] 欄位別名]]

- 上述ALL是預設值可以顯示所有記錄的欄位值，DISTINCT只顯示不重複欄位值的記錄，LIMIT關鍵字可以顯示查詢結果的前幾筆記錄加上 OFFSET 可跳過m筆資料。

6-2 SELECT子句-欄位規格

- 欄位規格(Column Specification)就是指查詢結果的欄位清單，可以使用AS關鍵字指定欄位別名。基本上，欄位規格可以是資料表欄位或計算值的運算式，其說明如下表所示：

欄位規格	說明
資料表的欄位 (Base Table Column)	即資料表的欄位名稱清單，或使用「*」符號代表所有欄位
計算值欄位 (Calculated Value Column)	算術運算子、字串或函數組成的運算式欄位

6-2-1 資料表的欄位-查詢資料表的部分欄位

- SELECT子句可以指明查詢結果所需的欄位清單，換句話說，我們可以只查詢資料表中所需的部分欄位。
- 查詢【學生】資料表的所有學生記錄，不過，只顯示學號、姓名和生日三個欄位，如下所示：

SELECT 學號, 姓名, 生日 FROM 學生

	學號	姓名	生日
1	S001	陳會安	1972-09-03 00:00:00.000
2	S002	江小魚	1978-02-02 00:00:00.000
3	S003	張無忌	1977-05-03 00:00:00.000
4	S004	陳小安	1976-06-13 00:00:00.000
5	S005	孫燕之	NULL
6	S006	周杰倫	1977-12-23 00:00:00.000
7	S007	蔡一零	1977-11-23 00:00:00.000
8	S008	劉得華	1977-02-23 00:00:00.000

6-2-1 資料表的欄位-查詢資料表的所有欄位

- 查詢結果如果需要顯示資料表的所有欄位，**SELECT**指令可以直接使用「*」符號代表資料表的所有欄位，而不用一一列出欄位清單。
- 查詢【課程】資料表的所有課程記錄並且顯示所有欄位，如下所示：

SELECT * FROM 課程

	課程編號	名稱	學分
1	CS101	計算機概論	4
2	CS111	線性代數	4
3	CS121	離散數學	4
4	CS203	程式語言	3
5	CS205	網頁程式設計	3
6	CS213	物件導向程式設計	2
7	CS222	資料庫管理系統	3
8	CS349	物件導向分析	3

6-2-2 欄位別名

- **SELECT**指令預設使用資料表定義的欄位名稱來顯示查詢結果，基於需要，我們可以使用**AS**關鍵字指定欄位別名，其中**AS**關鍵字本身可有可無。
- 查詢【學生】資料表的學號、姓名和生日資料，為了方便閱讀，顯示欄位名稱為【學生學號】、【學生姓名】和【學生生日】的中文欄位別名，如下所示：

```
SELECT 學號 AS 學生學號, 姓名 AS 學生姓名,  
       生日 AS 學生生日  
FROM 學生
```

	學生學號	學生姓名	學生生日
1	S001	陳會安	1972-09-03 00:00:00.000
2	S002	江小魚	1978-02-02 00:00:00.000
3	S003	張無忌	1977-05-03 00:00:00.000
4	S004	陳小安	1976-06-13 00:00:00.000
5	S005	孫燕之	NULL
6	S006	周杰倫	1977-12-23 00:00:00.000
7	S007	蔡一零	1977-11-23 00:00:00.000
8	S008	劉得華	1977-02-23 00:00:00.000

6-2-3 計算值欄位-說明

- 在SELECT子句的欄位規格如果是計算值欄位，我們可以使用算術運算子、字串或函數來組成運算式欄位。
- 因為計算值欄位並沒有欄位名稱，所以可以使用AS關鍵字來指定計算值欄位的別名。

6-2-3 計算值欄位-算術運算式(說明)

- 在SELECT子句的計算值欄位支援使用算術運算子(Arithmetic Operators)建立的運算式，可用的運算子如下表所示：

運算子	說明
+	加法
-	減法
*	乘法
/	除法
%	餘數

6-2-3 計算值欄位-算術運算式(範例)

- 因為【員工】資料表的薪水需要扣除稅金才是實拿的薪水，我們可以使用算術運算式來查詢【員工】資料表的薪水資料，顯示每位員工的薪水淨額，如下所示：

```
SELECT 身份證字號, 姓名,  
       薪水-扣稅 AS 薪水淨額  
FROM 員工
```

	身份證字號	姓名	薪水淨額
1	A123456789	陳慶新	78000.00
2	A221304680	郭富城	34200.00
3	A222222222	楊金樺	78000.00
4	D333300333	王心零	49000.00
5	D444403333	劉得華	24500.00
6	E444006666	小龍女	24500.00
7	F213456780	陳小安	49000.00
8	F332213046	張無忌	49000.00
9	H098765432	李鴻章	58500.00

6-2-3 計算值欄位-字串運算式(說明)

- 計算值欄位如果是字串運算式，它可以包含一至多個字串型別的欄位，和一些字串常數(Char String Constants)，這是使用單引號或雙引號括起的一序列字元，如下所示：
 - 'Abcdefg'
 - '5678'
 - 'SQL Server 2005資料庫設計'
- 上述字串常數可以使用字串連接運算函數子「CONCAT」號來連接欄位值和字串常數。

6-2-3 計算值欄位-字串運算式(範例)

- 在【員工】資料表的地址資料是由兩個欄位所組成，我們可以使用字串運算函數來顯示員工的地址資料，如下所示：

```
SELECT CONCAT(身份證字號, 姓名, 城市+'市'+街道)  
AS 地址 FROM 員工
```

	身份證字號	姓名	地址
1	A123456789	陳慶新	台北市信義路
2	A221304680	郭富城	台北市忠孝東路
3	A222222222	楊金樺	桃園市中正路
4	D333300333	王心零	桃園市經國路
5	D444403333	劉得華	板橋市文心路
6	E444006666	小龍女	板橋市中正路
7	F213456780	陳小安	新店市四維路
8	F332213046	張無忌	台北市仁愛路
9	H098765432	李鴻章	基隆市信四路

MariaDB SQL Functions

- **Aggregate Functions**

- 對多行資料執行計算，傳回單一值。包括 SUM、AVG、COUNT、MIN 和 MAX 等標準 SQL 函數，通常與 GROUP BY 一起使用。

- **Date & Time Functions**

- 介紹日期和時間函數。包含操作、格式化和計算日期和時間值的 SQL 函數，以應用於各種應用程式。

- **Numeric Functions**

- 介紹 MariaDB 伺服器中的數值函數。介紹用於執行數學計算、捨去和在查詢中操作數值的 SQL 函數。

- **String Functions**

- 說明 MariaDB 字串函數。介紹用於操作、搜尋和格式化文字字串的 SQL 函數，這些函數對於資料清洗和展示至關重要。

- 參考 <https://mariadb.com/docs/server/reference/sql-functions>

Aggregate Functions

- **AVG**
 - 計算平均值。此函數計算數值表達式的算術平均值，忽略 NULL 值。
- **COUNT**
 - 統計行數或值數。此函數傳回符合條件的行數或表達式中非 NULL 值的數量。
- **MAX**
 - 查找最大值。此函數傳回一組值中的最大值，適用於數字、字串和日期。
- **MIN**
 - 查找最小值。此函數傳回一組值中的最小值，適用於數字、字串和日期。
- **SUM**
 - 計算總和。此函數傳回數值表達式中所有值的總和，忽略 NULL 值。
- 參考 <https://mariadb.com/docs/server/reference/sql-functions/aggregate-functions>

String Functions

- **CONCAT**

- 連接字串。此函數將兩個或多個字串參數連接成一個字串。如果任何參數為 NULL，則傳回 NULL。

- **CONVERT**

- 將值轉換為特定類型或字元集。此函數使用 USING 關鍵字在類型或字元集之間轉換資料。

- **LEFT**

- 傳回最左側的字元。此函數傳回從字串開頭（左側）開始的指定數量的字元。

- **SUBSTRING**

- 傳回一個子字串。此函數從字串的特定位置開始提取指定長度的部分。

- 參考 <https://mariadb.com/docs/server/reference/sql-functions/string-functions>

Date & Time Functions

- **CURDATE, CURDATE(), CURRENT_DATE, CURRENT_DATE()**
 - 回傳目前日期。此函數根據上下文，以 “YYYY-MM-DD” 或 “YYYYMMDD” 格式輸出今天的日期。
- **CURTIME**
 - 回傳當前時間。此函數以 「HH:MM:SS」 或 「HHMMSS」 格式輸出目前時間。
- **DAY**
 - DAYOFMONTH() 的同義字。傳回給定日期的日期 (1-31)。
- **MONTH**
 - 提取月份。此函數傳回日期中的月份部分，數值範圍為 1 (1 月) 到 12 (12 月)。
- **DATEDIFF**
 - 計算兩個日期之間的差異。此函數傳回兩個日期值之間的天數，忽略時間部分。
 - 例: `DATEDIFF(expr1,expr2)` returns `(expr1 - expr2)`
- 參考 <https://mariadb.com/docs/server/reference/sql-functions/date-time-functions>

Numeric Functions

- **ABS**
 - 計算絕對值。此函數傳回一個數的非負值，並移除任何負號。
- **CEIL**
 - CEILING() 的同義字。將數字向上取整到最接近的整數。
- **FLOOR**
 - 向下取整到最接近的整數。此函數傳回小於或等於參數的最大整數值。
- **ROUND**
 - 將數字四捨五入。此函數將數字四捨五入到指定的小數位數。
- **TO_NUMBER**
 - 將字串轉換為數字。此與 Oracle 相容的函數使用指定的格式解析字串並傳回數值。
- 參考 <https://mariadb.com/docs/server/reference/sql-functions/numeric-functions>

6-2-3 計算值欄位-字串函數(說明)

- 在計算值欄位的運算式可以支援的數學、字串或日期/時間函數。例如：LEFT(str, len)、SUBSTRING(str, pos, [len]) 函數等。
- 關於進一步字串處理函數的說明，請參閱 <https://mariadb.com/docs/server/mariadb-quickstart-guides/mariadb-string-functions-guide>。

6-2-3 計算值欄位-日期函數(範例)

- 在【學生】資料表只有學生的生日資料，可以搭配日期函數來計算學生的年齡，如下所示：

```
SELECT 學號, 姓名,  
       CURDATE() AS 今天,  
       DATEDIFF(CURDATE(), 生日) AS 年齡  
FROM 學生
```

	學號	姓名	今天	年齡
1	S001	陳會安	2007-06-15 17:24:43.333	35
2	S002	江小魚	2007-06-15 17:24:43.333	29
3	S003	張無忌	2007-06-15 17:24:43.333	30
4	S004	陳小安	2007-06-15 17:24:43.333	31
5	S005	孫燕之	2007-06-15 17:24:43.333	NULL
6	S006	周杰倫	2007-06-15 17:24:43.333	30
7	S007	蔡一零	2007-06-15 17:24:43.333	30
8	S008	劉得華	2007-06-15 17:24:43.333	30

6-2-4 刪除重複記錄 - ALL與 DISTINCT(說明)

- 如果資料表記錄的欄位值擁有重複值，**SELECT**子句的預設值**ALL**是顯示所有欄位值，使用**DISTINCT**關鍵字可以刪除重複欄位值，一旦欄位擁有重複值，就只會顯示其中一筆記錄。

6-2-4 刪除重複記錄 - ALL與 DISTINCT(範例)

- 查詢【課程】資料表的課程資料擁有幾種不同的學分數，如下所示：

```
SELECT DISTINCT 學分 FROM 課程
```

	學分
1	2
2	3
3	4

6-2-5 前幾筆記錄 - LIMIT子句

- 在SELECT指令可以使用LIMIT子句取得查詢結果的前幾筆記錄，或前多少百分比的記錄資料。
- LIMIT n可以取得資料來源的前n筆記錄，若加上OFFSET m關鍵字就會先跳過前m筆記錄。如果使用ORDER BY子句進行排序，就可以顯示排序後的前幾筆記錄。
- 例: 忽略前 10 行並傳回以下 20 行

```
LIMIT 10, 20;
```

```
-- or
```

```
LIMIT 20 OFFSET 10;
```

6-2-5 前幾筆記錄 - LIMIT子句(範例 1)

- 在【學生】資料表顯示前3筆學生記錄資料，如下所示：

```
SELECT * FROM 學生 LIMIT 3
```

	學號	姓名	性別	電話	生日
1	S001	陳會安	男	02-22222222	1972-09-03 00:00:00.000
2	S002	江小魚	女	03-33333333	1978-02-02 00:00:00.000
3	S003	張無忌	男	04-44444444	1977-05-03 00:00:00.000

6-2-5 前幾筆記錄 - TOP子句(WITH TIES與ORDER BY子句-範例)

- 在【課程】資料表取出前3筆課程記錄資料，如果有同學分的記錄也一併顯出來，如下所示：

```
SELECT TOP 3 WITH TIES * FROM 課程  
ORDER BY 學分
```

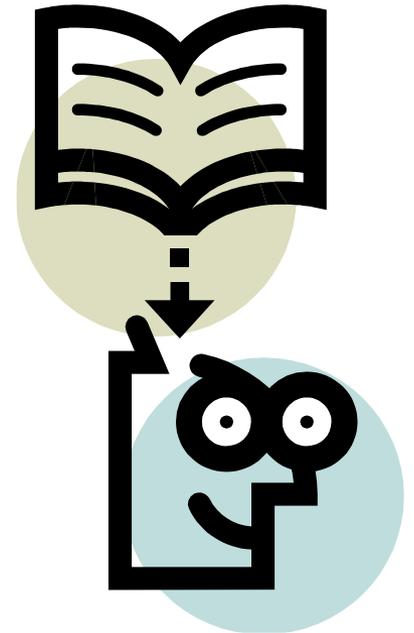
	課程編號	名稱	學分
1	CS213	物件導向程式設計	2
2	CS222	資料庫管理系統	3
3	CS349	物件導向分析	3
4	CS203	程式語言	3
5	CS205	網頁程式設計	3

6-3 FROM子句-說明

- **SELECT**指令是使用**FROM**子句指定查詢的來源資料表是哪些資料表，它可以是一個資料表或多個相關聯的資料表。
- 在本章的**SQL**指令碼檔都是從單一資料表取得查詢結果。

6-4 WHERE子句

- 6-4-1 比較運算子
- 6-4-2 邏輯運算子
- 6-4-3 算術運算子



6-4 WHERE子句

- SELECT指令和FROM字句是指出查詢哪個資料表的哪些欄位，WHERE子句的篩選條件才是真正的查詢條件，可以過濾記錄和找出符合所需條件的記錄資料，其基本語法如下所示：

WHERE 搜尋條件

- 上述搜尋條件就是使用比較和邏輯運算子建立的過濾條件，查詢結果是取回符合條件的記錄資料。

6-4-1 比較運算子-說明

- WHERE子句的搜尋條件可以是比較運算子建立的條件運算式，其運算元如果是欄位值，可以是文字、數值或日期/時間。支援的比較運算子(Comparison Operators)說明，如下表所示：

運算子	說明
=	相等
<>、!=	不相等
>	大於
>=	大於等於
<	小於
<=	小於等於
!<	不小於
!>	不大於

6-4-1 比較運算子-條件值為字串

- WHERE子句的條件運算式可以使用比較運算子來執行字串比較，欄位條件的字串需要使用單引號括起。
- 在【學生】資料表查詢學號為'S002'學生的詳細資料，如下所示：

```
SELECT * FROM 學生  
WHERE 學號='S002'
```

	學號	姓名	性別	電話	生日
1	S002	江小魚	女	03-33333333	1978-02-02 00:00:00.000

6-4-1 比較運算子-條件值為數值

- WHERE子句的條件運算式如果條件值是數值，數值欄位就不需使用單引號括起。
- 查詢【員工】資料表的薪水欄位小於50000元的員工記錄，如下所示：

```
SELECT * FROM 員工  
WHERE 薪水<50000
```

	身分證字號	姓名	城市	街道	電話	薪水	保險	扣稅
1	A221304680	郭富城	台北	忠孝東路	02-55555555	35000.00	1000.00	800.00
2	D444403333	劉得華	板橋	文心路	04-55555555	25000.00	500.00	500.00
3	E444006666	小龍女	板橋	中正路	04-55555555	25000.00	500.00	500.00

6-4-1 比較運算子-條件值為日期/時間

- WHERE子句的條件運算式如果是日期/時間的比較，如同字串，也需要使用單引號括起。
- 查詢【學生】資料表的學生生日是1978-02-02的學生記錄，如下所示：

```
SELECT * FROM 學生  
WHERE 生日='1978-02-02'
```

	學號	姓名	性別	電話	生日
1	S002	江小魚	女	03-33333333	1978-02-02 00:00:00.000

6-4-2 邏輯運算子-說明

- 在WHERE子句的搜尋條件可以使用邏輯運算子（Logical Operators）來連接條件運算式，以便建立更複雜的搜尋條件。常用的邏輯運算子說明，如下表所示：

運算子	說明
LIKE	包含，只需子字串即符合條件
BETWEEN/AND	在一個範圍之內
IN	屬於清單其中之一
NOT	非，可以否定運算式的結果
AND	且，需要連接的 2 個運算子都會真，才是真
OR	或，只需其中一個運算子為真，即為真

6-4-2 邏輯運算子- LIKE包含子字串運算子(說明)

- WHERE子句的條件欄位可以使用LIKE運算子進行比較，LIKE運算子是子字串查詢，只需是子字串就符合條件。我們還可以配合萬用字元來進行範本字串的比對，如下表所示：

萬用字元	說明
%	代表 0 或更多字元任意長度的任何字串
_	代表一個字元長度的任何字元
[]	符合括號內字元清單的任何一個字元，例如：[EO]
[-]	符合括號內「-」字元範圍的任何一個字元，例如：[A-J]
[^]	符合不在括號內字元清單的字元，例如：[^K-Y]

6-4-2 邏輯運算子- LIKE包含子字串 運算子(範例1)

- 查詢【教授】資料表中，屬於資訊相關科系CS和CIS的教授記錄，如下所示：

```
SELECT * FROM 教授
```

```
WHERE 科系 LIKE '%S%'
```

	教授編號	職稱	科...	身份證字號
1	I001	教授	CS	A123456789
2	I002	教授	CS	A222222222
3	I003	副教授	CIS	H098765432

6-4-2 邏輯運算子- LIKE包含子字串 運算子(範例2)

- 查詢【班級】資料表中，上課教室是在二樓的課程資料，如下所示：

```
SELECT DISTINCT 課程編號, 上課時間, 教室  
FROM 班級  
WHERE 教室 LIKE '%2_-%'
```

	課程編號	上課時間	教室
1	CS111	1900-01-01 15:00:00.000	321-M
2	CS121	1900-01-01 08:00:00.000	221-S
3	CS203	1900-01-01 10:00:00.000	221-S
4	CS203	1900-01-01 14:00:00.000	327-S
5	CS213	1900-01-01 09:00:00.000	622-G

6-4-2 邏輯運算子- LIKE包含子字串 運算子(範例3)

- 查詢【員工】資料表中，身份證字號是A-D範圍字母開頭的員工資料，如下所示：

```
SELECT * FROM 員工
```

```
WHERE 身份證字號 LIKE '[A-D]%'
```

	身份證字號	姓名	城市	街道	電話	薪水	保險	扣稅
1	A123456789	陳慶新	台北	信義路	02-11111111	80000.00	5000.00	2000.00
2	A221304680	郭富城	台北	忠孝東路	02-55555555	35000.00	1000.00	800.00
3	A222222222	楊金樺	桃園	中正路	03-11111111	80000.00	4500.00	2000.00
4	D333300333	王心零	桃園	經國路	NULL	50000.00	2500.00	1000.00
5	D444403333	劉得華	板橋	文心路	04-55555555	25000.00	500.00	500.00

6-4-2 邏輯運算子-BETWEEN/AND 範圍運算子(說明)

- BETWEEN/AND運算子可以定義欄位值需要符合的範圍，其範圍值可以是文字、數值或和日期/時間資料。

6-4-2 邏輯運算子-BETWEEN/AND 範圍運算子(範例1)

- 查詢【學生】資料表生日欄位的範圍是1977年1月1日到1977年12月31日出生的學生記錄，如下所示：

```
SELECT * FROM 學生
```

```
WHERE 生日 BETWEEN '1977-1-1' AND '1977-12-31'
```

	學號	姓名	姓...	電話	生日
1	S003	張無忌	男	04-44444444	1977-05-03 00:00:00.000
2	S006	周杰倫	男	02-33333333	1977-12-23 00:00:00.000
3	S007	蔡一零	女	03-66666666	1977-11-23 00:00:00.000
4	S008	劉得華	男	02-11111122	1977-02-23 00:00:00.000

6-4-2 邏輯運算子-BETWEEN/AND 範圍運算子(範例2)

- 因為學生修課學分數還差了2~3個學分，我們可以查詢【課程】資料表看看還有哪些課可以選修，如下所示：

```
SELECT * FROM 課程
```

```
WHERE 學分 BETWEEN 2 AND 3
```

	課程編號	名稱	學分
1	CS203	程式語言	3
2	CS205	網頁程式設計	3
3	CS213	物件導向程式設計	2
4	CS222	資料庫管理系統	3
5	CS349	物件導向分析	3

6-4-2 邏輯運算子-IN運算子(說明)

- IN運算子只需是清單其中之一即可，我們需要列出一串文字或數值清單作為條件，欄位值只需是其中之一，就符合條件。

6-4-2 邏輯運算子-IN運算子(範例1)

- 學生已經準備修CS101、CS222、CS100和CS213四門課，我們準備查詢【課程】資料表關於這些課程的詳細資料，如下所示：

```
SELECT * FROM 課程
```

```
WHERE 課程編號 IN ('CS101', 'CS222', 'CS100',  
                    'CS213')
```

	課程編號	名稱	學分
1	CS101	計算機概論	4
2	CS213	物件導向程式設計	2
3	CS222	資料庫管理系統	3

6-4-2 邏輯運算子-IN運算子(範例2)

- 學生修課的學分數還差2或4個學分，請使用IN運算子查詢【課程】資料表看看還有哪些課可以修，如下所示：

```
SELECT * FROM 課程
```

```
WHERE 學分 IN (2, 4)
```

	課程編號	名稱	學分
1	CS101	計算機概論	4
2	CS111	線性代數	4
3	CS121	離散數學	4
4	CS213	物件導向程式設計	2

6-4-2 邏輯運算子- NOT運算子(說明)

- NOT運算子可以搭配邏輯運算子，取得與條件相反的查詢結果，如下表所示：

運算子	說明
NOT LIKE	否定 LIKE 運算式
NOT BETWEEN	否定 BETWEEN/AND 運算式
NOT IN	否定 IN 運算式

6-4-2 邏輯運算子- NOT運算子(範例)

- 學生已經修了CS101、CS222、CS100和CS213四門課，準備進一步查詢【課程】資料表，看看還有什麼課程可以修，如下所示：

```
SELECT * FROM 課程
```

```
WHERE 課程編號 NOT IN ('CS101', 'CS222',  
                        'CS100', 'CS213')
```

	課程編號	名稱	學分
1	CS111	線性代數	4
2	CS121	離散數學	4
3	CS203	程式語言	3
4	CS205	網頁程式設計	3
5	CS349	物件導向分析	3

6-4-2 邏輯運算子- AND與OR運算子 (說明)

- AND運算子連接的前後運算式都必須同時為真，整個WHERE子句的條件才為真。
- OR運算子在WHERE子句連接的前後條件，只需任何一個條件為真，即為真。

6-4-2 邏輯運算子- AND與OR運算子 (範例1)

- 查詢【課程】資料表的課程編號欄位包含'1'子字串，而且課程名稱欄位有'程式'子字串，如下所示：

```
SELECT * FROM 課程
```

```
WHERE 課程編號 LIKE '%1%' AND 名稱 LIKE '%程式%'
```

	課程編號	名稱	學分
1	CS213	物件導向程式設計	2

6-4-2 邏輯運算子- AND與OR運算子 (範例2)

- 查詢【課程】資料表的課程編號欄位包含'1'子字串，或課程名稱欄位有'程式'子字串，如下所示：

```
SELECT * FROM 課程
```

```
WHERE 課程編號 LIKE '%1%' OR 名稱 LIKE '%  
程式%'
```

	課程編號	名稱	學分
1	CS101	計算機概論	4
2	CS111	線性代數	4
3	CS121	離散數學	4
4	CS203	程式語言	3
5	CS205	網頁程式設計	3
6	CS213	物件導向程式設計	2

6-4-2 邏輯運算子-連接多個條件與括號(說明)

- 在WHERE子句的條件可以使用AND和OR來連接多個不同條件。
- 其優先順序是位在括號中運算式優先，換句話說，我們可以使用括號來產生不同的查詢結果。

6-4-2 邏輯運算子-連接多個條件與括號(範例1)

- 查詢【課程】資料表的課程編號欄位包含'2'子字串，和課程名稱欄位有'程式'子字串，或學分大於等於4，如下所示：

```
SELECT * FROM 課程  
WHERE 課程編號 LIKE '%2%'  
AND 名稱 LIKE '%程式%'  
OR 學分 >= 4
```

	課程編號	名稱	學分
1	CS101	計算機概論	4
2	CS111	線性代數	4
3	CS121	離散數學	4
4	CS203	程式語言	3
5	CS205	網頁程式設計	3
6	CS213	物件導向程式設計	2

6-4-2 邏輯運算子-連接多個條件與括號(範例2)

- 查詢【課程】資料表的課程編號欄位包含'2'子字串，和課程名稱欄位有'程式'子字串，或學分大於等於4，後2個條件使用括號括起，如下所示：

```
SELECT * FROM 課程  
WHERE 課程編號 LIKE '%2%'  
AND (名稱 LIKE '%程式%'  
OR 學分 >= 4)
```

	課程編號	名稱	學分
1	CS121	離散數學	4
2	CS203	程式語言	3
3	CS205	網頁程式設計	3
4	CS213	物件導向程式設計	2

6-4-3 算術運算子-說明

- 在WHERE子句的運算式條件也支援算術運算子（Arithmetic Operators）的加、減、乘、除和餘數，我們可以在WHERE子句的條件加上算術運算子。

6-4-3 算術運算子-範例

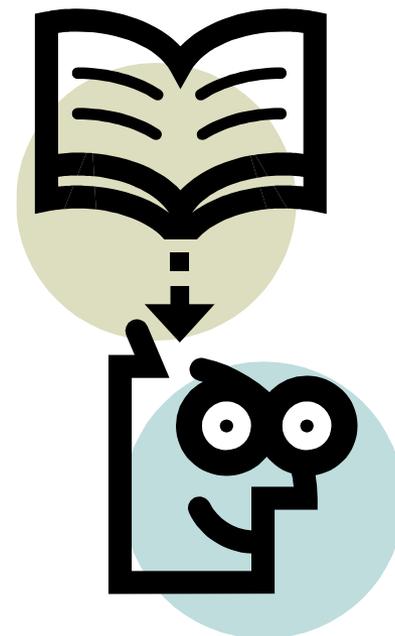
- 查詢【員工】資料表的薪水在扣稅和保險金額後的薪水淨額小於40000元的員工記錄，如下所示：

```
SELECT 身份證字號, 姓名, 電話 FROM 員工  
WHERE (薪水-扣稅-保險) < 40000
```

	身份證字號	姓名	電話
1	A221304680	郭富城	02-55555555
2	D444403333	劉得華	04-55555555
3	E444006666	小龍女	04-55555555

6-5 聚合函數的摘要查詢

- 6-5-1 COUNT() 函數
- 6-5-2 AVG() 函數
- 6-5-3 MAX() 函數
- 6-5-4 MIN() 函數
- 6-5-5 SUM() 函數



6-5 聚合函數的摘要查詢-說明

- 「聚合函數」 (Aggregate Functions) 也稱為「欄位函數」 (Column Functions)，可以進行選取記錄欄位值的筆數、平均、範圍和統計函數，以便提供進一步欄位資料的分析結果。
- 一般來說，如果SELECT指令敘述擁有聚合函數，稱為「摘要查詢」 (Summary Query)。

6-5 聚合函數的摘要查詢-常用的聚合函數

- 常用的聚合函數說明，如下表所示：

函數	說明
COUNT(運算式)	計算記錄筆數
AVG(運算式)	計算欄位平均值
MAX(運算式)	取得記錄欄位的最大值
MIN(運算式)	取得記錄欄位的最小值
SUM(運算式)	取得記錄欄位的總計

6-5-1 COUNT()函數-範例1

- SQL指令可以配合COUNT()函數計算查詢的記錄數，「*」參數可以統計資料表的所有記錄數，或指定欄位來計算欄位不是空值的記錄數。
- 查詢【學生】資料表的學生總數，如下所示：

```
SELECT COUNT(*) AS 學生數 FROM 學生
```

	學生數
1	8

6-5-1 COUNT()函數-範例2

- 在【學生】資料表查詢有填入生日資料的學生總數，即生日欄位不是空值NULL的記錄數，如下所示：

```
SELECT COUNT(生日) AS 學生數 FROM 學生
```

	學生數
1	7

6-5-1 COUNT()函數-範例3

- 查詢【員工】資料表的員工薪水高過40000元的員工總數，如下所示：

```
SELECT COUNT(*) AS 員工數 FROM 員工  
WHERE 薪水 > 40000
```

	員工數
1	6

6-5-2 AVG()函數-範例1

- SQL指令只需配合AVG()函數，就可以計算指定欄位的平均值。
- 在【員工】資料表查詢員工薪水的平均值，如下所示：

```
SELECT AVG(薪水) AS 平均薪水 FROM 員工
```

	平均薪水
1	50555.5555

6-5-2 AVG()函數-範例2

- 在【課程】資料表查詢課程編號包含'1'子字的課程總數，和學分的平均值，如下所示：

SELECT COUNT(*) AS 課程總數,

AVG(學分) AS 學分平均值

FROM 課程 WHERE 課程編號 LIKE '%1%'

	課程總數	學分平均值
1	4	3

6-5-3 MAX()函數-範例1

- SQL指令配合MAX()函數，可以計算符合條件記錄的欄位最大值。
- 在【員工】資料表查詢保險金額第一名員工的金額，如下所示：

```
SELECT MAX(保險) AS 保險金額 FROM 員工
```

	保險金額
1	5000.00

6-5-3 MAX()函數-範例2

- 在【課程】資料表查詢課程編號包含'1'子字串的最大學分數，如下所示：

```
SELECT MAX(學分) AS 最大學分數 FROM 課程  
WHERE 課程編號 LIKE '%1%'
```

	最大學分數
1	4

6-5-4 MIN()函數-範例1

- SQL指令配合MIN()函數，可以計算出符合條件記錄的欄位最小值。
- 在【員工】資料表查詢保險金額最後一名員工的金額，如下所示：

```
SELECT MIN(保險) AS 保險金額 FROM 員工
```

	保險金額
1	500.00

6-5-4 MIN()函數-範例2

- 在【課程】資料表查詢課程編號包含'1'子字串的最少學分數，如下所示：

```
SELECT MIN(學分) AS 最少學分數 FROM 課程  
WHERE 課程編號 LIKE '%1%'
```

	最少學分數
1	2

6-5-5 SUM()函數-範例1

- SQL指令配合SUM()函數，可以計算出符合條件記錄的欄位總和。
- 在【員工】資料表計算員工的薪水總和和平均，如下所示：

```
SELECT SUM(薪水) AS 薪水總額,  
       SUM(薪水)/COUNT(*) AS 薪水平均  
FROM 員工
```

	薪水總額	薪水平均
1	455000.00	50555.5555

6-5-5 SUM()函數-範例2

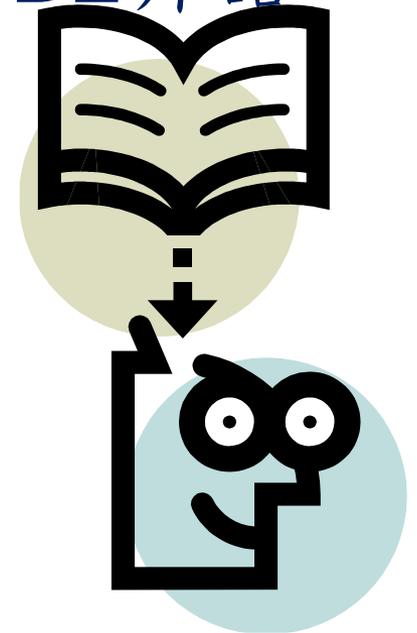
- 在【課程】資料表計算課程編號包含'1'子字的學分數總和，如下所示：

```
SELECT SUM(學分) AS 學分總和 FROM 課程  
WHERE 課程編號 LIKE '%1%'
```

	學分總和
1	14

6-6 群組查詢GROUP BY子句

- 6-6-1 GROUP BY子句
- 6-6-2 HAVING子句
- 6-6-3 WITHROLLUP和WITH CUBE片語



6-6-1 GROUP BY子句-說明

- 群組是以資料表的指定欄位來進行分類，分類方式是將欄位值中重複值結合起來歸成一類。例如：在【班級】資料表統計每一門課有多少位學生上課的學生數，【課程編號】欄位是建立群組的欄位，可以將修此課程的學生結合起來，如下圖所示：

教授編號	學號	課程編號	上課時間	教室
1001	S001	CS101	12:00pm	180-M
1002	S003	CS121	8:00am	221-S
1003	S001	CS203	10:00am	221-S
1003	S002	CS203	14:00pm	327-S
1002	S001	CS222	13:00pm	100-M
1002	S002	CS222	13:00pm	100-M
1002	S004	CS222	13:00pm	100-M
1001	S003	CS213	9:00am	622-G
1003	S001	CS213	12:00pm	500-K

課程編號	學生數
CS101	1
CS121	1
CS203	2
CS222	3
CS213	2

6-6-1 GROUP BY子句-語法

- 在SQL語言是使用GROUP BY子句來指定群組欄位，其語法如下所示：

GROUP BY 欄位清單

- 上述語法的欄位清單就是建立群組的欄位，如果不只一個，請使用「,」逗號分隔。

6-6-1 GROUP BY子句-條件

- 當使用GROUP BY進行查詢時，資料表需要滿足一些條件，如下所示：
 - 資料表的欄位擁有重複值，可以結合成群組。
 - 資料表擁有其他欄位可以配合聚合函數進行資料統計，如下表所示：

函數	進行的資料統計
AVG()函數	計算各群組的平均
SUM()函數	計算各群組的總和
COUNT()函數	計算各群組的記錄數

6-6-1 GROUP BY子句-範例1

- 在【班級】資料表查詢課程編號和計算每一門課程有多少位學生上課，如下所示：

```
SELECT 課程編號, COUNT(*) AS 學生數  
FROM 班級 GROUP BY 課程編號
```

	課程編號	學生數
1	CS101	3
2	CS111	3
3	CS121	2
4	CS203	4
5	CS213	4
6	CS222	3
7	CS349	2

6-6-1 GROUP BY子句-範例2

- 在【學生】資料表查詢統計性別男和女的學生數，如下所示：

```
SELECT 性別, COUNT(*) AS 學生數  
FROM 學生 GROUP BY 性別
```

	性別	學生數
1	女	3
2	男	5

6-6-2 HAVING子句-語法

- GROUP BY子句可以配合HAVING子句來指定搜尋條件，以便進一步縮小查詢範圍，其語法如下所示：
 HAVING 搜尋條件
- HAVING子句和WHERE子句的差異，如下所示：
 - HAVING子句可以使用聚合函數，但WHERE子句不可以。
 - 在HAVING子句條件所參考的欄位一定屬於SELECT子句的欄位清單；WHERE子句則可以參考FROM子句資料表來源的所有欄位。

6-6-2 HAVING子句-範例1

- 在【班級】資料表找出學生S002所上課的課程清單，如下所示：

```
SELECT 學號, 課程編號 FROM 班級  
GROUP BY 課程編號, 學號  
HAVING 學號 = 'S002'
```

	學號	課程編號
1	S002	CS111
2	S002	CS203
3	S002	CS222

6-6-2 HAVING子句-範例2

- 在【班級】資料表找出教授編號是'I003'，其教授課程有超過2位學生上課的課程清單，如下所示：

```
SELECT 課程編號, COUNT(*) AS 學生數  
FROM 班級  
WHERE 教授編號 = 'I003'  
GROUP BY 課程編號  
HAVING COUNT(*) >= 2
```

	課程編號	學生數
1	CS203	4
2	CS213	2

6-6-3 WITHROLLUP和WITH CUBE片語-說明

- GROUP BY子句可以使用ROLLUP和CUBE執行加總運算。
- WITH CUBE是針對GROUP BY子句的各群組欄位執行小計與加總；WITH ROLLUP則是針對第一個欄位執行加總運算。

6-6-3 WITHROLLUP和WITH CUBE 片語-範例1

- 在【班級】資料表找出教授I001和I003教授課程的學生數小計和加總，和各課程的學生總數，如下所示：

```
SELECT 教授編號, 課程編號,  
COUNT(學號) AS 總數  
FROM 班級  
WHERE 教授編號 IN ('I001', 'I003')  
GROUP BY 教授編號, 課程編號  
WITH CUBE
```

	教授編號	課程編...	總數
1	I001	CS101	3
2	I001	CS213	2
3	I001	CS349	2
4	I001	NULL	7
5	I003	CS203	4
6	I003	CS213	2
7	I003	NULL	6
8	NULL	NULL	13
9	NULL	CS101	3
10	NULL	CS203	4
11	NULL	CS213	4
12	NULL	CS349	2

6-6-3 WITHROLLUP和WITH CUBE 片語-範例2

- 在【班級】資料表找出教授I001和I003教授課程的學生數的小計和加總，如下所示：

```
SELECT 教授編號, 課程編號, COUNT(學號) AS 總數
FROM 班級
WHERE 教授編號 IN ('I001', 'I003')
GROUP BY 教授編號, 課程編號
WITH ROLLUP
```

	教授編號	課程編...	總數
1	I001	CS101	3
2	I001	CS213	2
3	I001	CS349	2
4	I001	NULL	7
5	I003	CS203	4
6	I003	CS213	2
7	I003	NULL	6
8	NULL	NULL	13

6-7 排序ORDER BY子句-語法

- SELECT指令可以使用ORDER BY子句依照欄位由小到大或由大到小進行排序，其語法如下所示：
`ORDER BY 運算式 [ASC | DESC] [, 運算式 [ASC | DESC]`
- 上述語法的排序方式預設是由小到大排序的ASC，如果希望由大至小，請使用DESC關鍵字。

6-7 排序ORDER BY子句-範例

- 在【員工】資料表查詢薪水大於35000元的員工記錄，並且使用薪水欄位進行由大至小排序，如下所示：

```
SELECT 姓名, 薪水, 電話 FROM 員工  
WHERE 薪水 > 35000  
ORDER BY 薪水 DESC
```

	姓名	薪水	電話
1	陳慶新	80000.00	02-11111111
2	楊金樺	80000.00	03-11111111
3	李鴻章	60000.00	02-33111111
4	王心零	50000.00	NULL
5	陳小安	50000.00	NULL
6	張無忌	50000.00	02-55555555