

SQL 資料處理語言

SQL Data Manipulation Language (DML)





查詢資料庫資料的內容

- ◆ SQL 語法在應用上對於 DML (Data Manipulation Language)：查詢維護資料庫資料內容的語法在使用上是更重要的，無論是查詢資料庫或顯示資料庫的內容。
- ◆ 更新或刪除資料庫中的資料，都必須依靠這些指令。其中重要的功能關鍵字有：
 - INSERT：新增資料到資料表中。
 - UPDATE：更改資料表中的資料。
 - DELETE：刪除資料表中的資料。
 - SELECT：查詢選取資料庫中的資料 (DQL)。



新增資料 - INSERT INTO 敘述句

◆INSERT 指令：新增資料

- 可以使用 INSERT 語法為資料表新增資料，其基本語法如下：

```
INSERT [INTO] 資料表名稱 [(欄位名稱 1, 欄位名稱 2, …)]  
VALUES (值 1, 值 2, …);
```

INSERT INTO 資料表名稱 (欄位1, 欄位2, 欄位3...) Values (資料1, 資料2, 資料3...)

- 範例：

```
INSERT INTO grade (no, name, chinese, math, nature)  
VALUES ('A8608011', '小丸子', 88, 95, 92)
```



新增資料 - INSERT INTO 敘述句

- ◆ **INSERT 指令**：新增資料
- ◆ **INSERT INTO 語法 (SQL INSERT INTO Syntax)**

```
INSERT INTO table_name (column1, column2, column3...)  
VALUES (value1, value2, value3...);
```

- ◆ 或是可以簡寫成這樣：使用簡寫的語法每個欄位的值都必需要依序輸入。

```
INSERT INTO table_name  
VALUES (value1, value2, value3...);
```



新增資料 - INSERT INTO 敘述句

◆INSERT INTO 範例

- 假設我們想從下面的 customers 資料表中新增顧客資料：我們可以使用以下的 INSERT INTO 敘述句或簡寫。

```
INSERT INTO customers (C_Id, Name, City, Address, Phone)
```

```
VALUES (3, '李三', '高雄縣', 'ZZ路300號', '07-12345678');
```

```
INSERT INTO customers
```

```
VALUES (3, '李三', '高雄縣', 'ZZ路300號', '07-12345678');
```

- 查詢新增後的結果如下：

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678
3	李三	高雄縣	ZZ路300號	07-12345678



新增資料 - INSERT INTO 敘述句

◆INSERT INTO 範例

- 你也可以只輸入幾個特定的欄位值：查詢新增後的結果如下：

```
INSERT INTO customers (C_Id, Name, City)
```

```
VALUES (3, '李三', '高雄縣');
```

C_Id	Name	City	Address	Phone
C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678



新增資料 - INSERT INTO 敘述句

◆INSERT INTO 範例

- 一次新增多筆資料 (INSERT INTO SELECT) · 語法如下：

```
INSERT INTO table_name  
  
VALUES (value1_1, value2_2, value3_3,⋯),  
(value2_1, value2_2, value2_3,⋯),  
(value3_1, value3_2, value3_3,⋯), ⋯⋯⋯;
```



更新資料 - UPDATE 敘述句

◆UPDATE 指令：更新資料

- 可以使用 UPDATE 語法為資料表更新資料，其基本語法如下：

SET 欄位名稱 1 = 值 1, 欄位名稱 2 = 值 2, ...

WHERE 條件式；

UPDATE 資料表名稱 SET 欄位1 = 資料1, 欄位2 = 資料2... WHERE 條件

- UPDATE 更新資料的動作可以一次更動多筆資料的內容，所以 WHERE 後加上的條件式十分重要，只要符合條件的資料內容即會進行更新的動作，要特別注意。
- 範例：

UPDATE grade SET name = '張小毛', nature = 100 WHERE no = 'A8608011'



更新資料 - UPDATE 敘述句

◆UPDATE 指令：更新資料

- 如果我們要修改資料表中的資料我們就會需要用到 UPDATE 。
- UPDATE 語法 (SQL UPDATE Syntax)語法如下，記得要加WHERE 條件式，這樣才會只更新特定某(幾)筆資料，不然 "全部的" 資料都會更改 。

```
UPDATE table_name  
SET column1=value1, column2=value2, column3=value3…  
WHERE some_column=some_value;
```



更新資料 - UPDATE 敘述句

◆UPDATE 範例

- 假設我們想從下面的 `customers` 資料表中更改顧客王二的連絡電話：我們可以使用以下的 UPDATE 敘述句：

```
UPDATE customers SET Phone='03-8565431'
```

```
WHERE Name='王二';
```

- 更新後的結果如下：

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678
3	李三	高雄縣	ZZ路300號	07-12345678

_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-87654321
3	李三	高雄縣	ZZ路300號	07-12345678



刪除資料 - DELETE FROM 敘述句

◆DELETE FROM 指令：刪除資料

- 可以使用 DELETE 語法為資料表刪除資料，其基本語法如下：

DELETE FROM 資料表名稱

WHERE 條件式；

- DELETE 刪除資料的動作可以一次刪除多筆資料的內容，所以 WHERE 後加上的條件式十分重要，只要符合條件的資料內容即會進行刪除的動作，要特別注意。
- 範例：

```
DELETE FROM grade WHERE nature < 85 AND math < 85
```



刪除資料 - DELETE 敘述句

◆ **DELETE FROM** 指令：刪除資料

- **DELETE FROM** 是用來刪除資料表中的資料。
- **DELETE FROM** 語法 (SQL **DELETE FROM** Syntax)語法如下，一樣要加**WHERE** 條件式，不然 "全部的" 資料都會刪除。

```
DELETE FROM table_name
```

```
WHERE column_name operator value;
```



刪除資料 - DELETE 敘述句

◆DELETE FROM 指令：刪除資料範例

- 假設我們想從下面的 `customers` 資料表中刪除顧客王二的資料：我們可以使用以下的 `DELETE FROM` 敘述句：

```
DELETE FROM customers WHERE Name='王二';
```

- 刪除後的資料表現在變成如下：

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678
3	李三	高雄縣	ZZ路300號	07-12345678

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
3	李三	高雄縣	ZZ路300號	07-12345678
C_Id	Name	City	Address	Phone



刪除資料 - DELETE 敘述句

◆ **DELETE FROM** 指令：刪除資料範例

- 一次刪除資料表中所有的資料
- 我們還可以用以下兩種方式一次刪除某資料表中所有的資料：

```
DELETE FROM table_name;
```

```
DELETE * FROM table_name;
```

Data Query Language (DQL)



資料查詢 - SELECT 敘述句

- ◆ 在SQL語法中，SELECT 很可能是最常用到的 SQL 語句，它是用來從資料庫取得資料，這個動作我們通常稱之為查詢 (query)，資料庫依SELECT 查詢的要求會返回一個結果資料表 (result table)，我們通常稱之為資料集 (result-set)。
- ◆ SELECT 敘述句主要由兩部分構成，第一部分是要 「拿什麼」 資料 (若有多項用逗號隔開)；第二部分則為 「從哪拿」 。

以下是基本的SELECT 語法格式。

```
SELECT table_column1, table_column2, table_column3...
FROM table_name;
```



資料查詢 - SELECT 敘述句

◆SELECT：查詢資料

- 資料的查詢應是資料庫系統最重要的工作了，所以 SELECT 可能是 SQL 語法中最重要的指令，因為所有查詢資料的動作都必須由這個指令開始。

◆SELECT 基本語法

- SELECT 指令應用於使用者要向資料庫系統查詢資料的時候，其基本的語法格式如下：

SELECT 欄位名稱

FROM 資料表名稱；

- 如果要顯示所有的欄位，可以使用「*」來代表，其語法與結果如下：

SELECT *

FROM `students`;



資料查詢 - SELECT 敘述句

◆SELECT：查詢資料

- SELECT 指令可以用來從資料表選取資料，其語法如下 (SQL關鍵字沒有英文字母大小寫之分，而且可以寫成多行或一行)：

```
SELECT 欄位名稱
```

```
FROM 資料表名稱
```

```
[WHERE 搜尋子句]
```

```
[ORDER BY 排序子句 {ASC|DESC}]
```

- 例如：

```
SELECT name, nature, chinese FROM grade
```

```
SELECT * FROM grade
```

```
SELECT name AS 姓名, nature AS 自然 FROM grade
```



資料查詢 - SELECT 敘述句

◆SELECT：查詢資料範例

◆ 假設我們想從下面的 customers 資料表中取得顧客的姓名 (Name) 及電話 (Phone)：我們可以下這個 SELECT 查詢敘述句：返回的結果 (資料集) 會像這樣：

```
SELECT Name, Phone  
FROM customers;
```

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678
3	李三	高雄縣	ZZ路300號	07-12345678

Name	Phone
張一	02-12345678
王二	03-12345678
李三	07-12345678



資料查詢 - SELECT 敘述句

◆ 指定資料表選取欄位

- 如果未來在設定資料表關聯與合併時，在 SELECT 語法中可能會由二個或多個資料表中選取欄位，此時若在二個資料表中有同名的欄位時，就會造成顯示的錯誤。
- 所以在選取欄位時要更明確指出該欄位是來自哪個資料表，其基本格式如下，也就是在欄位名稱前要加上資料表名稱，並以「.」符號加以串連，即可避免錯誤發生。

SELECT 資料表名稱 . 欄位名稱

FROM 資料表名稱；



資料查詢 - SELECT 敘述句

◆SELECT：查詢資料範例

- 如果我們想一次取得整張資料表裡所有的資料我們可以在 SELECT 語句裡用 * 這個特殊符號。這個 SELECT 查詢返回的資料集會是：

```
SELECT *
FROM customers;
```

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678
3	李三	高雄縣	ZZ路300號	07-12345678

- 若非必要，我們應該盡量避免使用 SELECT *，因為一次取得整張資料表會比較耗費系統資源。
- 記住一個原則，取得需要的資料就好，不多拿也不少拿。例如若只想知道客戶的電話不需要連他的住址也一起取得。



資料查詢 - SELECT 敘述句

◆SELECT ... AS：設定選取欄位別名

- 在選取要顯示的欄位時，可能因為該欄位名稱不易判讀或是套用函式而不易顯示，可以使用 AS 設定顯示時使用的別名，讓顯示時不僅美觀，也更加容易了解。
- 基本格式如下，只要在選取的欄位後加上 AS 並設定別名，在使用時即可用這個別名取代原欄位的名稱。

SELECT 欄位名稱 AS 欄位別名

FROM 資料表名稱；



資料查詢 - SELECT 敘述句

◆SELECT INTO：選擇欄位轉存資料表

- SELECT INTO 用來從某資料表查詢所得之資料集結果新增到另一個新建的資料表中。此一指令常用來複製備份資料表，或將資料表輸出至另一資料庫中。
- SELECT INTO 語法 (SQL SELECT INTO Syntax)如下，其中 new_table_name 為欲新建的資料表名稱，該資料表會自動建立，且不可與已經存在的資料表名稱相同；而 another_database 為至外部資料庫的路徑。

```
SELECT table_column1, table_column2, table_column3...
INTO new_table_name [IN another_database]
FROM table_name;
```

- MySQL 資料庫不支援 SELECT INTO，但你可以用 **CREATE TABLE table2 SELECT * FROM table1** 來達到同樣的功能。



資料查詢 - SELECT 敘述句

◆SELECT INTO 範例

- 我們現在想備份顧客資料表 customers，可執行如下的語法：

```
SELECT * INTO customers_backup FROM customers;
```

- 或者，我們想把它備份到 backup 資料庫中：

```
SELECT *  
INTO customers_backup IN 'backup.mdb'  
FROM customers;
```

- 我們也可以只複製資料表結構：

```
SELECT * INTO new_table_name FROM table_name WHERE 0=1;
```



資料查詢 - SELECT 敘述句

◆SELECT DISTINCT :去除重複資料顯示一筆

- 一個資料表的某欄位中可能會有多個紀錄都是相同值的情況，在SELECT查詢語句中我們可使用 DISTINCT 關鍵字過濾重複出現的紀錄值。
- 如果需要知道某個資料表欄位內有哪些不同的值，而每個值出現的次數並不重要時，可以使用 SELECT DISTINCT 的方式去達成，其基本的語法格式如下：

SELECT DISTINCT 欄位名稱

FROM 資料表名稱；

```
SELECT DISTINCT table_column1, table_column2...
FROM table_name;
```



資料查詢 - SELECT 敘述句

◆SELECT DISTINCT 範例

- 假設我們想從下面的 `customers` 資料表中查詢顧客分佈在哪幾個縣市：我們可以以下這個SELECT查詢敘述句：返回的結果如右：
SELECT DISTINCT City FROM customers;
- 原本資料表的 City 欄位中有兩個重複值台北市，可是我們只想知道有哪幾個縣市有顧客而已，故我們使用 DISTINCT 關鍵字來限制僅取出欄位中 "不相同" 的值。
- 若接著 `SELECT DISTINCT` 後面有指定兩個以上的欄位，則要符合所有欄位值皆同樣重複的情況下該筆資料才會被捨棄。若只有其中一個欄位值相同但其它欄位值並不同，則仍會取出該筆資料。

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678
3	李三	高雄縣	ZZ路300號	07-12345678
4	陳四	台北市	AA路400號	02-87654321

City
台北市
新竹縣
高雄縣



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... : 設定篩選條件

- 在查詢資料時，並不是每一次都要顯示所有的內容。我們可能會為顯示的資料設定一些條件，來篩選顯示的內容，這就是 WHERE 指令的功能。
- WHERE 的基本語法格式為：

SELECT 欄位名稱

FROM 資料表名稱

WHERE 條件敘述句；

```
SELECT table_column1, table_column2...
```

```
FROM table_name
```

```
WHERE column_name operator value;
```

資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... : 設定篩選條件

➤ 在條件式中會使用比較運算子，說明如下。

運算子	說明	運算結果
=	相等	當兩者相等時成立
!=	不等於	當兩者不等時成立
<>	不等於	當兩者不等時成立
<	小於	前者小於後者時成立
>	大於	當前者大於後者時成立
<=	小於或等於	當前者比後者小或兩者一樣時成立
>=	大於或等於	當前者比後者大或兩者一樣時成立
IS NULL	是否為 NULL	指定的內容是否為 NULL 值



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... : 設定篩選條件

- AND、OR、NOT：連接多個條件式
- 當在 SQL 語法中所要使用的條件式不只一個時，就要應用到比較運算子來結合不同的條件式。
- MySQL 使用 AND、OR、NOT 來連結多個條件式，其用法如下：

運算子	運算結果
AND	AND (且)，左右二方的條件都必須成立時。
OR	OR (或)，左右二方的條件有一成立時。
NOT	NOT(否)，條件不成立時。



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... : 設定篩選條件

- SELECT ... FROM ... 的篩選範圍涵蓋整個資料表的資料，但有時我們可能需要將篩選範圍限制在符合某些條件的資料，例如所有chinese分數大於90之資料的name和math兩個欄位，此時，我們可以加上WHERE子句設定篩選範圍，例如：

```
SELECT name, math FROM grade WHERE chinese > 90
```

```
SELECT name, chinese, math FROM grade WHERE chinese > 90 OR math > 90
```

```
SELECT * FROM grade WHERE chinese < 90 AND (math > 90 OR nature > 90)
```



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... : 設定篩選條件範例

- 假設我們想從下面的 `customers` 資料表中查詢 "王二" 的資料：我們可以下 "取出表中姓名欄位值是王二" 如此條件的
SELECT 敘述句：查詢的結果如下：

```
SELECT * FROM customers WHERE Name = '王二';
```

- 查詢條件中的 "字串值" 需要用單引號 " 包住，否則會出現錯誤；另一方面，對於數值型別則不可以加上引號，否則也會出現錯誤。被單引號圍住的字元即表示其資料型別為字串，但有些資料庫也接受用雙引號。

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678
3	李三	高雄縣	ZZ路300號	07-12345678
4	陳四	台北市	AA路400號	02-87654321

C_Id	Name	City	Address	Phone
2	王二	新竹縣	YY路200號	03-12345678



資料查詢 - SELECT 敘述句

◆ **SELECT ... FROM ... WHERE ...** : 設定篩選條件範例

◆ 若想要以查詢條件式進行查詢，可以使用 AND 或 OR 運算子同時下達多個條件：

- AND 運算子表示其左右兩條件皆需符合
- OR 運算子表示其左右兩條件至少需符合一個

```
SELECT table_column1, table_column2...
FROM table_name
WHERE column_name1 operator value1
AND column_name2 operator value2
[AND|OR]...;
```



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... : 設定篩選條件範例

◆ 舉個例子，我們想在下面這個 customers 資料表中查詢住在台北市而且薪水大於 5 萬的顧客姓名及連絡電話可以這樣查詢。

```
SELECT Name, Phone FROM table_name
```

```
WHERE City = '台北市' AND Salary >= 50000;
```

C_Id	Name	City	Address	Phone	Salary
1	張一	台北市	XX路100號	02-12345678	25000
2	王二	新竹縣	YY路200號	03-12345678	30000
3	李三	高雄縣	ZZ路300號	07-12345678	30000
4	陳四	台北市	AA路400號	02-87654321	50000

Name	Phone
陳四	02-87654321



資料查詢 - SELECT 敘述句

◆ **SELECT ... FROM ... WHERE ...** : 設定篩選條件範例

- ◆ 利用圓括號 () 可以來執行更複雜的查詢條件，圓括號內圍住的的條件式會優先判斷。例如：

WHERE 條件一 AND (條件二 OR 條件三)

- ◆ 此 SQL 陳述式會優先對條件二與條件三做 OR 邏輯運算，所得結果再接著和條件一做 AND 邏輯運算。



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... IN：指定多個篩選值

- IN 指定多個篩選值是加在 WHERE 之後，它的基本語法格式為：

SELECT 欄位名稱

FROM 資料表名稱

WHERE 欄位名稱 IN (欄位值 1, 欄位值 2, ...);

- IN 搭配 WHERE 子句可以用來限定必需符合某些欄位值為條件來搜尋資料表中的特定資料。若是設定篩選值時，想要直接由某個欄位指定幾個值來顯示，可以使用 IN 進行值的指定。指定的值必須放置在「()」左右括號中，每個值之間以「,」逗號區隔。

```
SELECT table_column1, table_column2, table_column3...
```

```
FROM table_name
```

```
WHERE column_name
```

```
IN (value1, value2, value3...);
```



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... IN 範例

➤ 我們想從下面的 customers 資料表中取得顧客張一或李三的資料，我們可以這樣作 SQL 查詢：

```
SELECT *
FROM customers
WHERE Name IN ('張一', '李三');
```

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678
3	李三	高雄縣	ZZ路300號	07-12345678

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
3	李三	高雄縣	ZZ路300號	07-12345678



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... IN 範例

➤ 相反的，NOT IN 就是不包含在條件裡的的資料我通通要了，如上例多加上 NOT ，我們可以這樣作 SQL 查詢：

```
SELECT *  
FROM customers  
WHERE Name NOT IN ('張一', '李三');
```

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678
3	李三	高雄縣	ZZ路300號	07-12345678

C_Id	Name	City	Address	Phone
2	王二	新竹縣	YY路200號	03-12345678



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... BETWEEN ... AND : 設定篩選範圍

- BETWEEN ... AND 設定篩選範圍是加在 WHERE 之後，功能是設定篩選出某個欄位的值在起始值與結束值之間的資料。
- 它的基本語法格式如下，不同於 IN 運算子是依不連續的值作為 WHERE 查詢子句的篩選條件，BETWEEN 則是用來限定依某範圍內連續的值作為條件來搜尋資料表中的特定資料。BETWEEN 的頭和尾的值都會被包含進去。

SELECT 欄位名稱

FROM 資料表名稱

WHERE 欄位名稱

BETWEEN 起始值 AND 結束值

SELECT table_column1, table_column2, table_column3...

FROM table_name

WHERE column_name

BETWEEN value1 **AND** value2;



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... BETWEEN … AND : 設定篩選範圍

- 作為查詢範圍條件的欄位型態可為數值、日期或字串，其中字串是依照字母排列順序來界定範圍。
- 設定數值篩選範圍：若是要找出某個範圍之內的資料，可以使用 **BETWEEN** 開始敘述式，用 **AND** 來連結二個起始與終止的範圍值。
- 設定日期時間範圍：除了可以用來篩選數值型態資料的範圍外，也可以用來篩選日期時間型態資料的範圍。



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... BETWEEN ... AND 範例

➤ 假設我們想從下面的 customers 資料表中查詢訂單數量介於 1000~4000 的顧客資料，我們可以這樣作 SQL 查詢：

```
SELECT *  
FROM customers  
WHERE Quantity  
BETWEEN 1000 AND 4000;
```

C_Id	Name	City	Address	Phone	Quantity
1	張一	台北市	XX路100號	02-12345678	1000
2	王二	新竹縣	YY路200號	03-12345678	3000
3	李三	高雄縣	ZZ路300號	07-12345678	5000

C_Id	Name	City	Address	Phone	Quantity
1	張一	台北市	XX路100號	02-12345678	1000
2	王二	新竹縣	YY路200號	03-12345678	3000



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... BETWEEN ... AND 範例

➤ 相反的，NOT BETWEEN 就是不包含在這範圍裡的的資料我通通要了，如上例多加上 NOT：設我們想從下面的 customers 資料表中查詢訂單數量沒有介於 1000~4000 的顧客資料，我們可以這樣作 SQL 查詢：

```
SELECT *  
FROM customers  
WHERE Quantity  
NOT BETWEEN 1000 AND 4000;
```

C_Id	Name	City	Address	Phone	Quantity
1	張一	台北市	XX路100號	02-12345678	1000
2	王二	新竹縣	YY路200號	03-12345678	3000
3	李三	高雄縣	ZZ路300號	07-12345678	5000

C_Id	Name	City	Address	Phone	Quantity
3	李三	高雄縣	ZZ路300號	07-12345678	5000

資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... LIKE : 設定字串比對的篩選值

- 萬用字元的使用：在文字資料中常要找以某些文字開頭、某些文字結尾，或是字串中包含哪些文字的內容，此時即可
以使用 LIKE 運算子，並搭配以下萬用字元來進行查詢：
- LIKE 運算子搭配 WHERE 子句可以依一特定模式 (Pattern) 為條件來搜尋資料表中的特定資料。

萬用字元	說明	範例
_ (底線)	任何單一字元，一個中文 字也代表一個字元。	條件：「LIKE '文_閣」」，「文淵閣」符合，「文藏經閣」即 不符合。
%	任何含有零或多個字元的 字串。	條件：「LIKE '文 % 閣」」，「文淵閣」符合，「文藏經閣」 都符合。

- LIKE 語法

```
SELECT table_column1, table_column2, table_column3...
```

```
FROM table_name
```

```
WHERE column_name
```

```
LIKE pattern;
```



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... LIKE 範例

➤ 我們想從下面的 customers 資料表中取得住在台北縣市的顧客資料，我們可以使用這樣的 LIKE 查詢語句進行查詢：

```
SELECT *  
FROM customers  
WHERE Address  
LIKE '台北%';
```

C_Id	Name	City	Address
1	張一	台北市XX路100號	02-12345678
2	王二	新竹縣YY路200號	03-12345678
3	李三	高雄縣ZZ路300號	07-12345678
4	陳四	台北縣AA路400號	02-87654321

C_Id	Name	City	Address
1	張一	台北市XX路100號	02-12345678
4	陳四	台北縣AA路400號	02-87654321



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... WHERE ... LIKE 範例

➤ NOT LIKE 就是不包含在條件裡的的資料我通通要了，如上例多加上 NOT 查詢語句進行查詢

```
SELECT *  
FROM customers  
WHERE Address  
NOT LIKE '台北%';
```

C_Id	Name	City	Address
1	張一	台北市XX路100號	02-12345678
2	王二	新竹縣YY路200號	03-12345678
3	李三	高雄縣ZZ路300號	07-12345678
4	陳四	台北縣AA路400號	02-87654321

C_Id	Name	City	Address
2	王二	新竹縣YY路200號	03-12345678
3	李三	高雄縣ZZ路300號	07-12345678



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... ORDER BY：設定查詢結果的排序

- ORDER BY 的功能是用來設定欄位，進行排序查詢結果，它的基本語法如下：

SELECT 欄位名稱

FROM 資料表名稱

ORDER BY 指定排序的欄位 排序方式

- 其中排序方式有二種：

- ASC：遞增排序，由小排到大，也是未指定時預設的排序方法。
- DESC：遞減排序，由大排到小。

- 使用 ORDER BY 還可以對多個欄位進行排序，只要在每個指定欄位與排序方法間加上以「,」逗號區隔即可。

ORDER BY 指定排序的欄位 1 排序方式 1, 指定排序的欄位 2 排序方式 2, ...



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... ORDER BY : 設定查詢結果的排序

- 有時我們會需要將篩選出來的記錄依照遞增或遞減順序來進行排序，這時可以將 SELECT 取得的資料集依某欄位來作排序，而排序分別可以由小至大 (ascending; 預設)，或由大至小 (descending)。
- ORDER BY 語法與例子

```
SELECT table_column1, table_column2...
```

```
FROM table_name
```

```
ORDER BY column_name1 ASC|DESC, column_name2 ASC|DESC...
```

```
SELECT * FROM grade ORDER BY chinese ASC
```

```
SELECT * FROM grade ORDER BY chinese DESC
```

```
SELECT * FROM grade ORDER BY chinese DESC, math DESC
```



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... ORDER BY ... 範例

➤ 我們從下面的 employees 資料表中取出所有員工的資料並依職稱來作排序 (即字母順序) , 可以下這個 SELECT 敘述句 :

```
SELECT *  
FROM employees  
ORDER BY Title;
```

E_Id	Name	Title
1	Allen	crew
2	Tom	manager
3	Chris	crew
4	Bill	crew

E_Id	Name	Title
1	Allen	crew
3	Chris	crew
4	Bill	crew
2	Tom	manager

資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... ORDER BY ... 範例

- 若語句中沒加上 ASC 或 DESC 關鍵字，預設默認為 ASC。以上個例子為例，我們依職稱來作正向排序之後再依名稱來作反向排序：

```
SELECT *
FROM employees
ORDER BY Title ASC, Name DESC;
```

E_Id	Name	Title
1	Allen	crew
2	Tom	manager
3	Chris	crew
4	Bill	crew

E_Id	Name	Title
3	Chris	crew
4	Bill	crew
1	Allen	crew
2	Tom	manager



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... ORDER BY ... 範例

- SELECT 語句中用來作排序的欄位名稱，除了欄位名稱外，也可以使用 SELECT 語句查詢的欄位順序。如，SELECT 關鍵字後的第一個欄位 (table_column1) 為 1，第二個欄位 (table_column2) 則為 2。因此，我們也可以將上例中的 SQL 敘述句改寫成：

```
SELECT *  
FROM employees  
ORDER BY 3 ASC, 2 DESC;
```

E_Id	Name	Title
1	Allen	crew
2	Tom	manager
3	Chris	crew
4	Bill	crew

E_Id	Name	Title
3	Chris	crew
4	Bill	crew
1	Allen	crew
2	Tom	manager



資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... ORDER BY ... LIMIT：設定查詢顯示的筆數

- 有時符合查詢條件的記錄可能有很多筆，但我們並不需要看到所有記錄，只是想看看前幾筆記錄，那麼可以加上LIMIT語法來限制最多傳回筆數，它的基本語法如下，LIMIT 可以設定查詢後由哪一筆之後開始顯示，並顯示多少筆數。

SELECT 欄位名稱

FROM 資料表名稱

LIMIT 開始顯示的筆數，顯示多少筆資料

- LIMIT 是由查詢後的結果再進行擷取資料的動作，如果與 ORDER BY 進行排序搭配可以輕易取得最前的 10 筆資料或是最後的 10 筆資料的結果。也因為如此，LIMIT 在使用時必須放置在 ORDER BY 之後。
- TOP (SQL Server), LIMIT (MySQL), ROWNUM (Oracle) 這些語法其實都是同樣的功能，都是用來限制您的 SQL 查詢語句最多只影響幾筆資料，而不同的語法則只因不同的資料庫實作時採用不同的名稱。

資料查詢 - SELECT 敘述句

◆SELECT ... FROM ... ORDER BY ... LIMIT：設定查詢顯示的筆數

- 下面這個 資料表中，若只想取出前兩筆資料，可使用如下語法：

```
SELECT table_column1, table_column2...
FROM table_name
LIMIT number;
```

```
SELECT *
FROM customers
LIMIT 2;
```

```
SELECT *
FROM customers
LIMIT 2,2
```

略過前二筆，從第幾3筆開始，取回2筆，
所以會抓取第3~4筆，共2筆資料

E_Id	Name	Title
1	Allen	crew
2	Tom	manager
3	Chris	crew
4	Bill	crew

E_Id	Name	Title
1	Allen	crew
2	Tom	manager

E_Id	Name	Title
3	Chris	crew
4	Bill	crew



資料查詢 - SELECT 敘述句

◆加入統計欄位

- 若在結合資料表的查詢中加入了統計的欄位，非統計欄位的欄位都必須加入GROUP BY 的欄位，並以「,」逗號分隔，如此才能正確計算出結果。
- GROUP BY 出現的位置要在 FROM 及 WHERE 之後。



資料查詢 - SELECT 敘述句

◆ GROUP BY 語法

- GROUP BY 敘述句搭配聚合函數 (aggregation function) 使用，是用來將查詢結果中特定欄位值相同的資料分為若干個群組，而每一個群組都會傳回一個資料列。若沒有使用 GROUP BY，聚合函數針對一個 SELECT 查詢，只會返回一個彙總值。
- 聚合函數指的也就是 AVG()、COUNT()、MAX()、MIN()、SUM() 等這些內建函數。
- GROUP BY 語法如下

```
SELECT column_name(s), aggregate_function(column_name)  
FROM table_name  
WHERE column_name operator value  
GROUP BY column_name1, column_name2...;
```



資料查詢 - SELECT 敘述句

◆ GROUP BY 範例

➤ 我們想從下面的 orders 資料表中查詢個別顧客的訂單金額總合：我們可以下這個 SELECT 查詢敘述句，結果如下：

```
SELECT Customer,  
SUM(Price)  
FROM orders  
GROUP BY Customer;
```

O_Id	Price	Customer
1	1000	張一
2	2000	王二
3	500	李三
4	1300	張一

Customer	SUM(Price)
張一	2300
王二	3800
李三	500



多資料表關聯查詢

◆結合資料表的查詢

- SQL JOIN (連接) 是利用不同資料表之間欄位的關連性來結合多資料表之檢索。也就是結合多個資料表而組成一抽象的暫時性資料表以供資料查詢，在原各資料表中之紀錄及結構皆不會因此連接查詢而改變。
- 若要結合二個資料表的基本語法如下：

SELECT 顯示欄位…

FROM 資料表 A, 資料表 B

WHERE 資料表 A. 相關欄位 = 資料表 B. 相關欄位



多資料表關聯查詢

◆ 使用 JOIN 結合資料表

- 若要 JOIN 語法結合二個資料表的基本語法如下：

SELECT 顯示欄位…

FROM 資料表 A [INNER] JOIN 資料表 B

ON A. 相關欄位 = 資料表 B. 相關欄位

- 另一個方式為：

SELECT 顯示欄位…

FROM 資料表 A [INNER] JOIN 資料表 B

USING (相關欄位)

- 在這個語法中，JOIN 也可以使用 INNER JOIN 取代，功能相同。
- 由語法的結構來看，ON 的使用讓整個 SQL 語法更容易閱讀，並將條件式獨立在 WHERE 的子句中。



多資料表關聯查詢

◆ 使用 JOIN 結合資料表

SQL 的 JOIN 查詢有以下幾種類型，以下將介紹主要幾種類型。

- INNER JOIN 內部連接
- LEFT (OUTER) JOIN 左外部連接
- RIGHT (OUTER) JOIN 右外部連接
- FULL (OUTER) JOIN 全部外部連接
- CROSS JOIN 交叉連接
- NATURAL JOIN 自然連接



多資料表關聯查詢

◆ 使用JOIN結合資料表

- 以下為一個客戶資料表 `customers` 及產品訂單的資料表 `orders`，其中，`C_Id` 是客戶資料表中的主鍵 (Primary Key) 欄位
- 我們將以這兩張表為例進行說明怎麼將這兩張不同的資料表依相關欄位來作個連接結合以便查詢

C_Id	Name	City	Address	Phone
1	張一	台北市	XX路100號	02-12345678
2	王二	新竹縣	YY路200號	03-12345678
3	李三	高雄縣	ZZ路300號	07-12345678

O_Id	OrderNo	C_Id
1	2572	3
2	7375	3
3	7520	1
4	1054	1
5	1257	5



多資料表關聯查詢

◆INNER JOIN 語法

- INNER JOIN (內部連接) 為等值連接，必需指定等值連接的條件，而查詢結果只會返回符合連接條件的資料。

```
SELECT table_column1, table_column2...
FROM table_name1
INNER JOIN table_name2
ON table_name1.column_name=table_name2.column_name;
```

```
SELECT table_column1, table_column2...
FROM table_name1
INNER JOIN table_name2
USING (column_name);
```



多資料表關聯查詢

◆INNER JOIN 範例

- 我們想列出所有客戶的訂單編號資料，這時可以使用 INNER JOIN 查詢，其中點號連接之 XXX.YYY 表示 XXX 資料表中的 YYY 欄位。查詢結果只會返回符合連接條件的資料。

```
SELECT customers.Name, orders.Order_No  
FROM customers  
INNER JOIN orders  
ON customers.C_Id=orders.C_Id;
```

Name	Order_No
李三	2572
李三	7375
張一	7520
張一	1054



多資料表關聯查詢

◆LEFT JOIN 、 RIGHT JOIN 語法

- 無論用上述何種方式結合資料表，在二邊的資料表都要有資料才會顯示，只要有一方沒有即不會出現在結果中。
- 例如若有一個學生沒有登錄成績，就不會在結果中顯示這個學生的資訊，如此很容易會有漏失資訊的情況發生。
- 此時可以使用 LEFT JOIN 或 RIGHT JOIN 的方法來執行資料表連結，語法為：

SELECT 顯示欄位…

FROM 資料表 A LEFT|RIGHT JOIN 資料表 B

ON A. 相關欄位 = 資料表 B. 相關欄位

- 若改為 LEFT JOIN 方式連結資料表，會發現主資料表的值在另一個資料表中還沒有資料，這時會以「NULL」顯示。



多資料表關聯查詢

◆LEFT JOIN 語法

- LEFT JOIN 可以用來建立左外部連接，查詢的 SQL 敘述句 LEFT JOIN 左側資料表 (table_name1) 的所有記錄都會加入到查詢結果中，即使右側資料表 (table_name2) 中的連接欄位沒有符合的值也一樣。
- 語法如下，有些資料庫的語法會是 LEFT OUTER JOIN 。

```
SELECT table_column1, table_column2...
FROM table_name1
LEFT JOIN table_name2
ON table_name1.column_name=table_name2.column_name;
```



多資料表關聯查詢

◆LEFT JOIN 範例

- 我們想查詢所有客戶與對應訂單狀況的資料，我們可以作一個 LEFT JOIN 查詢，結果如下，LEFT JOIN 會返回左側資料表中所有資料列，就算沒有符合連接條件，而右側資料表中如果沒有匹配的資料值就會顯示為 NULL。

```
SELECT customers.Name, orders.Order_No  
FROM customers  
LEFT JOIN orders  
ON customers.C_Id=orders.C_Id;
```

Name	Order_No
張一	7520
張一	1054
王二	
李三	2572
李三	7375



多資料表關聯查詢

◆RIGHT JOIN 語法

- RIGHT JOIN 可以用來建立右外部連接，查詢的 SQL 敘述句 RIGHT JOIN 右側資料表 (table_name2) 的所有記錄都會加入到查詢結果中，即使左側資料表 (table_name1) 中的連接欄位沒有符合的值也一樣。
- 語法如下，有些資料庫的語法會是 RIGHT OUTER JOIN 。

```
SELECT table_column1, table_column2...
FROM table_name1
RIGHT JOIN table_name2
ON table_name1.column_name=table_name2.column_name;
```



多資料表關聯查詢

◆RIGHT JOIN 範例

- 我們想查詢所有訂單與對應客戶之資料，我們可以作一個 RIGHT JOIN 查詢，結果如下，RIGHT JOIN 會返回右側資料表中所有資料列，就算沒有符合連接條件，而左側資料表中如果沒有匹配的資料值就會顯示為 NULL。

```
SELECT customers.Name, orders.Order_No  
FROM customers  
RIGHT JOIN orders  
ON customers.C_Id=orders.C_Id;
```

Name	Order_No
李三	2572
李三	7375
張一	7520
張一	1054
	1257



多資料表關聯查詢

◆ UNION 語法

- UNION 運算子用來將兩個(以上) SQL 查詢的結果合併起來，而由 UNION 查詢中各別 SQL 語句所產生的欄位需要是相同的資料型別及順序。
- UNION 查詢只會返回不同值的資料列，有如 SELECT DISTINCT 。
- UNION 就是像是 OR (聯集)，如果紀錄存在於第一個查詢結果集或第二個查詢結果集中，就會被取出。
- UNION 與 JOIN 不同的地方在於，JOIN 是作橫向結合 (合併多個資料表的各欄位)；而 UNION 則是作垂直結合 (合併多個資料表中的紀錄)。
- UNION語法如下，其查詢返回的資料集欄位名稱，通常會依據第一個 SELECT 查詢的欄位名稱 。

```
SELECT column_name(s)  
FROM table_name1  
UNION SELECT column_name(s)  
FROM table_name2;
```



多資料表關聯查詢

◆UNION 範例

- 下表是公司在台灣地區銷售的產品資料表 `products_taiwan` 以及在日本地區銷售的產品資料表 `products_japan`，我們想查出公司所有產品類別的資料，可以作一個 UNION 查詢，結果如下：

P_Id	P_Name
1	LCD
2	CPU
3	RAM

P_Id	P_Name
1	Keyboard
2	CPU
3	LCD

```
SELECT P_Name  
FROM products_taiwan  
UNION SELECT P_Name  
FROM products_china;
```

P_Name
LCD
CPU
RAM
Keyboard



多資料表關聯查詢

◆UNION 範例

➤ UNION ALL 則會列出所有的資料，不論是否重複。我們可以作一個 UNION ALL 查詢，結果如下：

P_Id	P_Name
1	LCD
2	CPU
3	RAM

P_Id	P_Name
1	Keyboard
2	CPU
3	LCD

```
SELECT P_Name  
FROM products_taiwan  
UNION ALL  
SELECT P_Name  
FROM products_china;
```

P_Name
LCD
CPU
RAM
Keyboard
CPU
LCD

Q & A