

SQL SELECT

SQL 是用來做什麼的呢？一個最常用的方式是將資料從資料庫中的表格內選出。從這一句回答中，我們馬上可以看到兩個關鍵字：**從 (FROM)** 資料庫中的表格內**選出 (SELECT)**。(表格是一個資料庫內的結構，它的目的是儲存資料。在[表格處理](#)這一部分中，我們會提到如何使用 SQL 來設定表格。) 我們由這裡可以看到最基本的 SQL 架構：

SELECT "欄位名" FROM "表格名"

我們用以下的例子來看看實際上是怎麼用的。假設我們有以下這個表格：

Store_Information 表格

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999
Los Angeles	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

若要選出所有的店名 (store_name)，我們就鍵入：

```
SELECT store_name FROM Store_Information
```

結果：

store_name
Los Angeles
San Diego
Los Angeles
Boston

我們一次可以讀取好幾個欄位，也可以同時由好幾個表格中選資料。

SQL DISTINCT

SELECT 指令讓我們能夠讀取表格中一個或數個欄位的所有資料。這將把所有的資料都抓出，無論資料值有無重複。在資料處理中，我們會經常碰到需要找出表格內的不同資料值的情況。換句話說，我們需要知道這個表格/欄位內有哪些不同的值，而每個值出現的次數並不重要。這要如何達成呢？在 SQL 中，這是很容易做到的。我們只要在 **SELECT** 後加上一個 **DISTINCT** 就可以了。**DISTINCT** 的語法如下：

```
SELECT DISTINCT "欄位名"  
FROM "表格名"
```

舉例來說，若要在以下的表格，Store_Information，找出所有不同的店名時，

Store_Information 表格

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999
Los Angeles	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

我們就鍵入，

```
SELECT DISTINCT store_name FROM Store_Information
```

結果:

store_name

Los Angeles

San Diego

Boston

SQL WHERE

我們並不一定每一次都要將表格內的資料都完全抓出。在許多時候，我們會需要選擇性地抓資料。就我們的例子來說，我們可能只要抓出營業額超過 \$1,000 的資料。要做到這一點，我們就需要用到 **WHERE** 這個指令。這個指令的語法如下：

```
SELECT "欄位名"  
FROM "表格名"  
WHERE "條件"
```

若我們由以下的表格抓出營業額超過 \$1,000 的資料，

Store_Information 表格

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999
Los Angeles	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

我們就鍵入，

```
SELECT store_name  
FROM Store_Information  
WHERE Sales > 1000
```

結果:

store_name

Los Angeles

SQL AND OR

在上一頁中，我們看到 **WHERE** 指令可以被用來由表格中 有條件地選取資料。這個條件可能是簡單的（像上一頁的例子），也可能是複雜的。複雜條件是由二或多個簡單條件透過 **AND** 或是 **OR** 的連接而成。一個 SQL 語句中可以有無限多個簡單條件的存在。

複雜條件的語法如下：

```
SELECT "欄位名"  
FROM "表格名"  
WHERE "簡單條件"  
{[AND|OR] "簡單條件"}+
```

{}+ 代表{}之內的情況會發生一或多次。在這裡的意思就是 **AND** 加 簡單條件及 **OR** 加簡單條件的情況可以發生一或多次。另外，我們可以用 () 來代表條件的先後次序。

舉例來說，我們若要在 Store_Information 表格中選出所有 Sales 高於 \$1,000 或是 Sales 在 \$500 及 \$275 之間的資料的話，

Store_Information 表格

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999
San Francisco	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

我們就鍵入，

```
SELECT store_name  
FROM Store_Information  
WHERE Sales > 1000  
OR (Sales < 500 AND Sales > 275)
```

結果:

```
store_name  
Los Angeles  
San Francisco
```

SQL IN

在 SQL 中，在兩個情況下會用到 **IN** 這個指令；這一頁將介紹其中之一：與 **WHERE** 有關的那一個情況。在這個用法下，我們事先已知道至少一個我們需要的值，而我們將這些知道的值都放入 **IN** 這個子句。 **IN** 指令的語法為下：

```
SELECT "欄位名"  
FROM "表格名"  
WHERE "欄位名" IN ('值一', '值二', ...)
```

在括弧內可以有一或多個值，而不同值之間由逗點分開。值可以是數目或是文字。若在括弧內只有一個值，那這個子句就等於

WHERE "欄位名" = '值一'

舉例來說，若我們要在 Store_Information 表格中找出所有含蓋 Los Angeles 或 San Diego 的資料，Store_Information 表格

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999
San Francisco	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

我們就鍵入，

```
SELECT *  
FROM Store_Information  
WHERE store_name IN ('Los Angeles', 'San Diego')
```

結果:

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999

SQL BETWEEN

IN 這個指令可以讓我們依照一或數個不連續 (discrete) 的值的限制之內抓出 資料庫中的值，而 **BETWEEN** 則是讓我們可以運用一個範圍 (range) 內 抓出資料庫中的值。**BETWEEN** 這個子句的語法如下：

```
SELECT "欄位名"  
FROM "表格名"  
WHERE "欄位名" BETWEEN '值一' AND '值二'
```

這將選出欄位值包含在值一及值二之間的每一筆資料。

舉例來說，若我們由 Store_Information 表格中找出所有介於 January 6, 1999 及 January 10, 1999 中的資料，

Store_Information 表格

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999
San Francisco	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

我們就鍵入，

```
SELECT *
FROM Store_Information
WHERE Date BETWEEN 'Jan-06-1999' AND 'Jan-10-1999'
```

請讀者注意：在不同的資料庫中，日期的儲存法可能會有所不同。在這裡我們選擇了其中一種儲存法。

結果：

store_name	Sales	Date
San Diego	\$250	Jan-07-1999
San Francisco	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

SQL 萬用字元

有的時候，我們需要依照由字串模式中找出相符的資料。要滿足這個需求，我們就需要用到萬用字元 (wildcard) 的做法。SQL 中有兩個萬用字元：

% (百分比符號)：代表零個、一個、或數個字母。

_ (底線)：代表剛好一個字母。

萬用字元是與 [LIKE](#) 關鍵字一起使用的。

以下是幾個萬用字元的例子：

- 'A_Z': 所有以 'A' 起頭，另一個任何值的字元，且以 'Z' 為結尾的字串。'ABZ' 和 'A2Z' 都符合這一個模式，而 'AKKZ' 並不符合 (因為在 A 和 Z 之間有兩個字元，而不是一個字元)。
- 'ABC%': 所有以 'ABC' 起頭的字串。舉例來說，'ABCD' 和 'ABCABC' 都符合這個模式。
- '%XYZ': 所有以 'XYZ' 結尾的字串。舉例來說，'WXYZ' 和 'ZZXYZ' 都符合這個模式。
- '%AN%': 所有含有 'AN' 這個模式的字串。舉例來說，'LOS ANGELES' 和 'SAN FRANCISCO' 都符合這個模式。
- '_AN%': 所有第二個字母為 'A' 和第三個字母為 'N' 的字串。舉例來說，'SAN FRANCISCO' 符合這個模式，而 'LOS ANGELES' 則不符合這個模式。

SQL LIKE

LIKE 是另一個在 **WHERE** 子句中會用到的指令。基本上，**LIKE** 能讓我們依據一個模式 (pattern) 來找出我們要的資料。相對來說，在運用 **IN** 的時候，我們完全地知道我們需要的條件；在運用 **BETWEEN** 的時候，我們則是列出一個範圍。**LIKE** 的語法如下：

```
SELECT "欄位名"
FROM "表格名"
WHERE "欄位名" LIKE {模式}
```

{模式} 經常包括萬用字元 (wildcard)。在上一頁中，我們看到了好幾個[萬用字元](#)的例子。以下我們用一個實例來看萬用字元和 **LIKE** 是如何被運用的：

Store_Information 表格

store_name	Sales	Date
LOS ANGELES	\$1500	Jan-05-1999

SAN DIEGO	\$250	Jan-07-1999
SAN FRANCISCO	\$300	Jan-08-1999
BOSTON	\$700	Jan-08-1999

要找出所有名稱內包含 AN 這兩個字串的資訊，我們就鍵入，

```
SELECT *
FROM Store_Information
WHERE store_name LIKE '%AN%'
```

結果:

store_name	Sales	Date
LOS ANGELES	\$1500	Jan-05-1999
SAN FRANCISCO	\$300	Jan-08-1999
SAN DIEGO	\$250	Jan-07-1999

SQL ORDER BY

到目前為止，我們已學到如何藉由 **SELECT** 及 **WHERE** 這兩個指令將資料由表格中抓出。不過我們尚未提到這些資料要如何排列。這其實是一個很重要的問題。事實上，我們經常需要能夠將抓出的資料做一個有系統的顯示。這可能是由小往大 (ascending) 或是由大往小 (descending)。在這種情況下，我們就可以運用 **ORDER BY** 這個指令來達到我們的目的。

ORDER BY 的語法如下：

```
SELECT "欄位名"
FROM "表格名"
[WHERE "條件"]
ORDER BY "欄位名" [ASC, DESC]
```

[] 代表 **WHERE** 子句不是一定需要的。不過，如果 **WHERE** 子句存在的話，它是在 **ORDER BY** 子句之前。**ASC** 代表結果會以由小往大的順序列出，而 **DESC** 代表結果會以由大往小的順序列出。如果兩者皆沒有被寫出的話，那我們就會用 **ASC**。

我們可以照好幾個不同的欄位來排順序。在這個情況下，**ORDER BY** 子句的語法如下(假設有兩個欄位)：

```
ORDER BY "欄位一" [ASC, DESC], "欄位二" [ASC, DESC]
```

若我們對這兩個欄位都選擇由小往大的話，那這個子句就會造成結果是依據 "欄位一" 由小往大排。若有好幾筆資料 "欄位一" 的值相等，那這幾筆資料就依據 "欄位二" 由小往大排。

舉例來說，若我們要依照 Sales 欄位的由大往小列出 Store_Information 表格中的資料，

Store_Information 表格

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999

San Diego	\$250	Jan-07-1999
San Francisco	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

我們就鍵入：

```
SELECT store_name, Sales, Date
FROM Store_Information
ORDER BY Sales DESC
```

結果：

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
Boston	\$700	Jan-08-1999
San Francisco	\$300	Jan-08-1999
San Diego	\$250	Jan-07-1999

在以上的例子中，我們用欄位名來指定排列順序的依據。除了欄位名外，我們也可以用欄位的順序（依據 SQL 句中的順序）。在 **SELECT** 後的第一個欄位為 1，第二個欄位為 2，以此類推。在上面這個例子中，我們用以下這句 SQL 可以達到完全一樣的效果：

```
SELECT store_name, Sales, Date
FROM Store_Information
ORDER BY 2 DESC
```