

## 本書的使用方法

本書的概念是「一看就懂 SQL」，所以只要看看漫畫、插圖、語法與例句，就能自由地從資料庫取得需要的資料。

本書的學習環境為免費的 MySQL。MySQL 可於 Windows 環境安裝，而安裝方式請參考第 0 章的說明。



**Sayaka**

已任職三年的業務員。  
希望提升業績，每天與  
資料奮戰中。



**工程師**

系統開發部門的前輩，  
人很親切，也是 Sayaka  
的老師。

### ● 漫畫

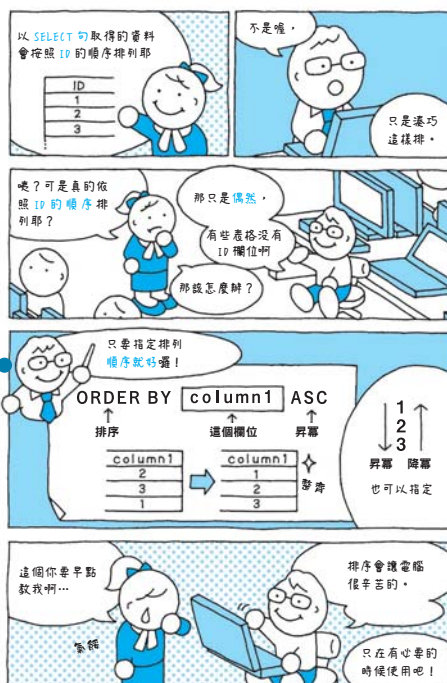
本書會先以漫畫簡略  
介紹該章的內容。

### ● 注意

無法於內容或 Point  
詳盡解說的重點，都會  
在這個部分補充。

### ● 冷知識

介紹一些記起來會很  
有用的關鍵字。



184

## ● 本書的目標讀者

- 從未使用過 SQL 的人
- 有時得到資料庫的人
- 開始資料分析的人
- 前台工程師或網頁設計師

## ● 本書執筆環境

- OS : Microsoft Windows 10 Home 64bit
- MySQL Community Edition 8.0.19

※ MySQL Community Edition 8.0.20 也可順利執行範例。

### 5

先排序再取得記錄

#### 語法 ORDER BY 的使用方法

SELECT  
欄位名稱 → 「什麼」  
FROM  
表格名稱 → 從「哪裡」  
ORDER BY  
欄位名稱 排列順序 → 以「何種順序」取得

排列順序可省略，省略時，預設為 ASC。

#### 例句 以 price 替 product 表格的所有欄位重新排序再取得記錄 (排列順序是昇冪)

```
SELECT  
FROM  
product  
ORDER BY  
price;
```

與「price ASC」的意思一樣

product_id	product_name	stock	price
1	藥用入浴劑	100	70
2	藥用手皂	23	700
3	溫泉之鄉草蓆	4	120
4	溫泉之鄉浴布院	23	120
5	草莓肥皂 100%	10	150
6	100% 牛奶入浴劑	15	140

排序 (昇冪)

product_id	product_name	stock	price
1	藥用入浴劑	100	70
3	溫泉之鄉草蓆	4	120
4	溫泉之鄉浴布院	23	120
6	100% 牛奶入浴劑	15	140
5	草莓肥皂 100%	10	150
2	藥用手皂	23	700

#### ● 語法

為了讓大家一看就懂，會使用指示線說明。

#### ● 例句

為了讓大家知道語法的功能，會以大量的插圖說明例句。

#### ● 執行結果

這是「例句」的執行結果。記錄的順序有時會與書裡的內容有出入。

第 1 章～第 8 章的結尾都有練習題喲。





# 01

第 1 章 利用 SQL 取得資料

## 資料庫與 SQL 到底是什麼？

資料庫到底是什麼？又該怎麼存取？讓我們簡單介紹一下。

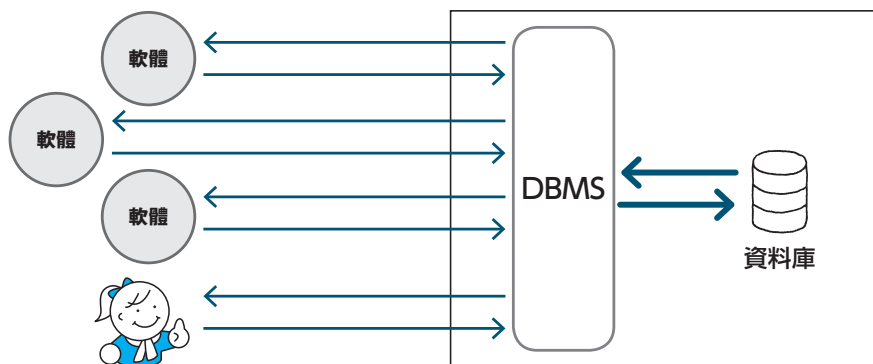
### 01-1 資料庫到底是什麼？

在電腦與網路就像空氣一樣的現代，所有的資訊都被當成「資料」儲存與使用。

如果資料的筆數與種類不多，可直接存成文字檔案，如果稍微多一點，也可使用試算表軟體儲存。

不過，如果是人力無法處理，試算表軟體也很難操作的大量資料，就可使用**資料庫（DB）**處理。簡單來說，資料庫就是「資料的集合體」，儲存在資料庫的資料會依照種類或用途分類，以便日後使用，這也是資料庫最明顯的特徵。

要在資料庫新增或刪除資料，可使用專門用來管理資料的資料庫管理系統（**DataBase Management System : DBMS**）。



1

利用 SQL 取得資料

# 03

第 1 章 利用 SQL 取得資料

## 利用 SELECT 陳述句取得資料

接下來，開始正式地學習 SQL。第一步先學習從資料庫取得資料的基本知識。

### 03-1 執行 SELECT 陳述句

要從資料庫取得資料可執行 SQL 的 SELECT 語法。這種使用 **SELECT** 語法的 SQL 又稱為 **SELECT 陳述句**。

一般來說，資料庫會同時有很多張表格，而要取得資料，只需要指定從「哪裡」取得「什麼」資料即可。

比方說，要從儲存商品資訊的 `product` 表格取得 `product_id` 與 `product_name` 這兩個資訊，可將 SQL 寫成下列的內容。

**例句** 從 `product` 表格取得 `product_id` 與 `product_name`

```
SELECT
  product_id,
  product_name
FROM
  product;
```

「什麼」

從「哪裡」

取得

寫完 SELECT 之後，接著輸入相當於「什麼」部分的欄名。如果要從多個欄取得資料，可利用逗號間隔欄名。接著再寫 FROM，然後輸入相當於「哪裡」的表格。這些單字都要以半形字元或換行字元間隔。

# 04

第 1 章 利用 SQL 取得資料

## 將欄位變更為其他名稱

在取得資料時，可替欄位取一個別名。舉例來說，我們很難從「id」、「name」這類欄名看出這是什麼欄位，此時可試著替這類欄位命名一個更具體的名稱。

### 04-1 替欄位取一個更簡單易懂的名稱

SQL 的保留字都是英文字母，表格名稱與欄位名稱也通常都是英文字母。要利用 SELECT 陳述式取得資料時，可暫時替欄位取一個中文名稱或比較簡單易懂的名稱，取得的結果也會變得比較容易判讀。要替欄位另取新名可使用 AS 陳述式。

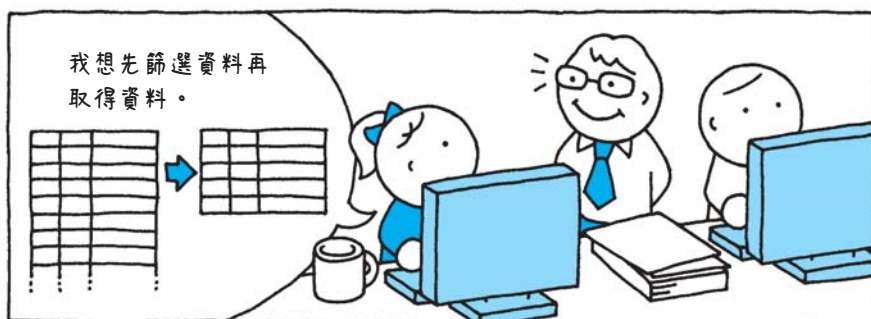
**例句** 以別名取得 **product\_id** 與 **product\_name**

```
SELECT
    product_id AS 商品ID,
    product_name AS 商品名稱
FROM
    product;
```

「什麼」  
以「別名」  
取得

上述的程式將 **product\_id** 指定為「商品 ID」這個別名，也將 **product\_name** 指定為「商品名稱」這個別名，取得的結果如下。



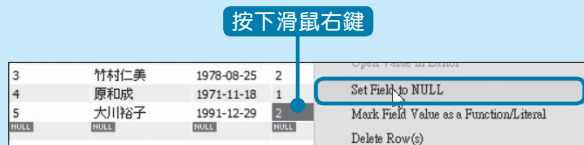




## 冷知識

### 設定 NULL

要在曾經存入其他值的欄位設定 NULL，可在該欄位按下滑鼠右鍵，選擇「Set Field to NULL」。



設定完成後，點選「Apply」套用。

2

取得  
資料

## 02-3 若利用比較運算子比較 NULL 會發生什麼事？

剛剛在說明比較運算子「<=>」的時候曾提到「a 等於 b (包含 NULL)」。

<=> 是可處理 NULL 的比較運算子。以 = 或 > 這類非 <=> 的運算子讓 NULL 與自己或其他值比較，都只會得到 NULL 的結果。要讓 NULL 與 NULL 或 NULL 以外的值比較是否相等時，可使用 <=> 運算子。

SELECT

```
1 = NULL, 1 <=> NULL, NULL <=> NULL,  
1 != NULL, 1 <> NULL, 1 < NULL;
```

1=NULL	1<=>NULL	NULL<=>NULL	1!=NULL	1<>NULL	1<NULL
NULL	0	1	NULL	NULL	NULL

由於只有 <=> 可處理 NULL，所以在比較兩側的值之後，將傳回 1 或 0 的結果。若以其他運算子比較 NULL，只會得到 NULL 這個結果。



要注意的是，若是從其他工具複製「\」再貼入 MySQL Workbench，有可能會因為出現亂碼而無法進行逸脫處理。

**例句** 從 product 表格取得 product\_name 有 '100%' 的記錄

```
SELECT
  *
FROM
  product
WHERE
  product_name LIKE '%100\%%';
```

與「%100\%%一致」

果汁100%

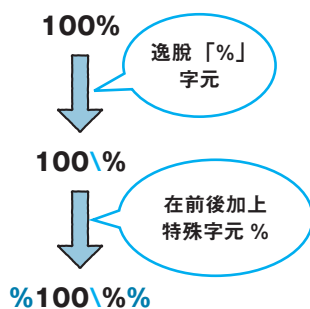
○

100%果汁

○

果汁100橘子

×



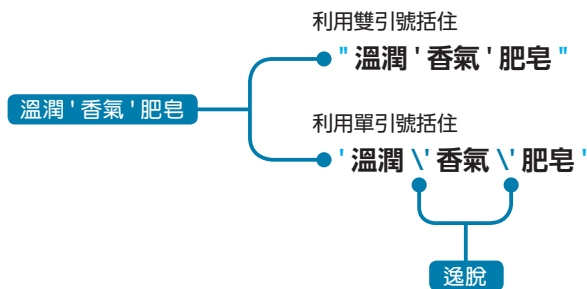
product_id	product_name	stock	price
5	草莓肥皂 100%	10	150
6	100% 牛奶入浴劑	15	140

請確認是否取得具有 '100%' 字串的記錄。除了「%」之外，還有其他需要逸脫處理的特殊字元，這些字元都具有特殊意義，有的還無法直接從鍵盤輸入。

### 需要逸脫處理的主要特殊字元

逸脫字元	意義	逸脫字元	意義
\%	字元「%」	\n	換行字元
\_	字元「_」	\t	定位點字元
\\	字元「\」	\b	退後鍵字元
\'	字元「'」	\r	歸位換行字元
\"	字元「"」		

除了 LIKE 之外，若是遇到需要在字串輸入特殊字元時，就可使用上述的語法。比方說，要在字串輸入單引號，可直接利用雙引號括住字串，但如果就是想用單引號來寫，可對字串的內容進行逸脫處理。



### 03-5 試著比較大小

= 運算子可於字串、日期或其他資料類型的資料使用。以此類推，> 或 <= 這類比較運算子也能用來比較字串或日期的大小。

## 01

第3章 在○○取得類似△△的資料

## 將多個條件串起來

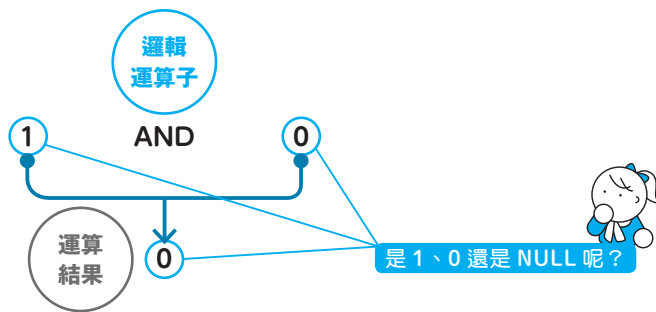
我們可以指定取得資料的條件，一次指定多重條件能讓條件變得更嚴格或寬鬆。

## 01-1 什麼是邏輯運算子？

我們在上一章學會了指定條件再取得資料的方法，不過當時只指定了一個條件，但其實可一次指定兩個條件以上。

要指定「大於等於 ○ 而且小於等於 △」這種兩個條件組合而成的條件必須使用專用的運算子。

這種專用的運算子稱為**邏輯運算子**。邏輯運算就是以 1 (=TRUE) 或 0 (=FALSE) 這種邏輯值進行的運算，結果不是 1 就是 0，而 SQL 除了 1 與 0，還有 NULL 這個值。



本書將所有的邏輯運算子整理成下面這張表格。

3

在○○取得類似△△的資料



不是取得的時候統整，而是統整  
SELECT 的結果啊？

對啊，因為 SELECT 句會依序執行每行程式，這個觀念非常重要喲！



使用 DISTINCT 的時候，也可以同時指定多個欄位。此時這些欄位的內容若有重複，也會統整為單筆資料。

#### 例句

以沒有重複的方式，從 inquiry 表格取得 pref 與 age 欄位的資料

```
SELECT
  DISTINCT pref, age
FROM
  inquiry;
```

「什麼」  
從「哪裡」  
無重複  
取得

pref	age
東京都	20
神奈川縣	30
埼玉縣	40
神奈川縣	20
東京都	30
東京都	20

DISTINCT → ?

pref	age
東京都	20
神奈川縣	30
埼玉縣	40
神奈川縣	20
東京都	30

這個組合有兩個