

# 1 簡介

SQL(Structured Query Language 結構化查詢語言) 是一種特定目的的程式語言，用於管理關聯式資料庫系統 (RDBMS)，或在關係流資料管理系統 (RDSMS) 種進行流處理。SQL 與資料庫有著很大的關聯，資料庫包括以檔案為基礎 (File-based) 的小型簡易資料庫，如 DBASE、FoxPro、Access 資料庫等，以及主從架構 (Client-Server) 為基礎的中大型資料庫，如甲骨文的 Oracle、Microsoft 的 SQL Server、IBM 的 DB2、10gen 的 MongoDB 等，每一種的資料庫都有自己的特色和優缺點。使用資料庫儲存資料進行管理搭配前後台我們可以輕易發展出生活中的種種應用，如訂票系統、資訊查詢系統、人事薪資系統等。

資料進入資料庫後可以使我們更為方便的集中管理這些資料，為了要能夠將資料庫內的資料進行存取，因此我們需要透過 SQL 來對資料庫內的資料進行相關的操作，資料的基本操作為查詢 (Select)、新增 (Insert)、更新 (Update) 及刪除 (Delete)，本次實習課將會帶領大家如何操作資料庫內的資料。

# 2 資料庫

Microsoft 提供了免費的 SQL Server Express 版本可供使用，免費版本與付費版本除了有儲存容量和些許功能上的限制外，其餘功能皆沒有太大的差異，如果我們只是要開發一個小型的專案或是練習如何操作資料庫，那麼免費的 SQL Server Express 就已經很夠用了，它可以讓我們儲存 10GB 的資料量，我們可以透過<https://www.microsoft.com/zh-tw/sql-server/sql-server-editions-express> 進行下載，本次使用的版本是 SQL Server 2017 Express。

下載並安裝完成後，在我們的電腦裡面就有一個可以儲存資料的資料庫了，即便如此，我們仍然需要一個管理資料庫的軟體來對資料庫進行相關的 SQL 操作，因此還需要額外下載 SQL Server Management Studio(SSMS) 軟體，下載網址為 <https://docs.microsoft.com/zh-tw/sql/ssms/download-sql-server-management-studio-ssms?redirectedfrom=MSDN&view=sql-server-ver15>。

有了資料庫以及資料庫管理軟體之後，我們就可以新增一個資料庫並對該資料庫塞一些資料來做一些簡單的資料操作，Microsoft 也在 GitHub 提供了一些簡單的資料庫範例，有興趣的同學可以自行前往<https://github.com/microsoft/sql-server-samples/tree/master/samples/databases>，本次實習課要用到的資料庫是 northwind-pubs 裡面的 instwnd 資料，資料庫及資料的 SQL 語法可以至 <https://raw.githubusercontent.com/microsoft/sql-server-samples/master/samples/databases/northwind-pubs/instwnd.sql> 進行下載，資料庫及資料匯入操作流程如下：

1. Open web URL and copy all instructions.
2. Open SSMS
3. Connect to your MS-SQL database
4. Press ctrl + N to open new query interface
5. Paste all instructions that you have copied from the website
6. Press F5 to run all instructions in new query interface

### 3 SQL Instruction

在 SQL 裡面指令的大小寫並不會影響操作，所以 Insert、INSERT 和 InSeRt 在 SQL 都視為插入的指令，在 SQL 下指令時我們通常會習慣用全部大寫以示區別。

### 4 查詢 (SELECT)

當我們想要對資料庫內的資料進行查詢時，例如我們想要知道某學生的基本資料或是我們想要知道該學生在某一個學期某一科目的成績，我們便會用到 SELECT 這個指令，而 SELECT 指令的基本語法是

```
1 SELECT column1 , column2 ...  
2 FROM table_name  
3 [WHERE conditions ]  
4 [ORDER BY column list ]
```

SELECT 後面接的是我們需要的欄位名稱，FROM 後面接的是我們要操作的資料表名稱，WHERE 後面接的是我們要查詢的條件，ORDER BY 後面接的是我們希望針對哪幾個欄位進行排序，WHERE 和 ORDER BY 的指令是可選擇的 (optional)，當我們在 SELECT 時不下 WHERE 和 ORDER BY 的指令，表示我們希望將該資料表的資料全部撈出並且不做任何排序。下面的 SELECT 指令顯示了我們要從 Northwind 的 Categories 資料表撈出 CategoryID、CategoryName、Description 和 Picture 欄位的所有資料。

```
1 SELECT CategoryID , CategoryName , Description , Picture  
2 FROM Northwind.dbo.Categories
```

當我們需要撈取某張資料表裡面所有欄位的全部資料時，我們應該要把該張資料表的所有欄位都寫在 SELECT 後面，若該張資料表在設計時或是因需求變更而演化成超多欄位時，那麼我們就要在 SELECT 後面接這麼多的欄位，聽起來似乎不是一個很有效率的方法，因此在 SQL 裡面也提供了可以操作此方法的指令「\*」，如下面指令所示。

```
1 SELECT *  
2 FROM Northwind.dbo.Categories
```

下面的這個指令則顯示了我們需要從 Northwind 資料庫裡面的 Categories 資料表找出符合 CategoryID=1 的所有欄位資料。

```
1 SELECT *  
2 FROM Northwind.dbo.Categories  
3 WHERE CategoryID=1
```

當我們希望能從某張資料表找出相似符合條件的欄位資料時，我們可以在 WHERE 條件式底下加入 LIKE 的指令並搭配「%」這個符號，下面的這個指令顯示了我們需要從 Northwind 資料庫裡面的 Categories 找出 Description 欄位裡面是 S 開頭的所有欄位值，查詢結果會以 CategoryName 來進行由大至小的排序 (DESC)。SQL 指令的排序可以分為 AESC 和 DESC 兩種排序方式，AESC 指的是欄位值會由小至大進行排序，DESC 指的則是欄位值會由大至小進行排序。

```
1 SELECT *  
2 FROM Northwind.dbo.Categories  
3 WHERE Description LIKE 'S%'
```

#### 4 ORDER BY CategoryName DESC

查詢除了上述的基本操作指令外，我們還可以做出更為複雜的操作，例如查詢出不重複的資料 (DISTINCT)、查詢出前幾筆的資料 (TOP n)、結合多張資料表進行整合式的查詢 (INNER JOIN、OUTER JOIN)，或是針對查詢結果的欄位直接進行相關運算等。基本的運算操作指令說明如下表所示。

Operator	Explain
COUNT(expression)	Calculate how many records selected.
AVG(expression)	Calculate the average for selected field.
MAX(expression)	Get the maximum value of selected field.
MIN(expression)	Get the minimum value of selected field.
SUM(expression)	Calculate the summary value of selected field.

下面的指令顯示了一個基本的查詢計算操作，該指令會將 Northwind 資料庫的 Categories 資料表內所有資料總列數進行計數並回傳計數結果。

```
1 SELECT COUNT(*)
2 FROM Northwind.dbo.Categories
```

而 WHERE 指令除了上述我們所看到的等於及相似外，它也提供了不等於、大於、小於等相關操作，相關的操作指令如下表所示。

Operator	Explain
=	Equal
<>, !=	Not equal
>	Greater than
>=	Greater or equal to
<	Less than
<=	Less than or equal to
!<	Not less than
!>	Not greater than

除了上述的操作外，WHERE 也提供了基本的邏輯操作指令，相似查詢就是一個邏輯的操作，其它的邏輯操作指令說明如下表所示。

Operator	Explain
LIKE	contain
BETWEEN ...AND ...	in a range
IN	one of the list
NOT	
AND	
OR	

如果我們今天想要查詢的資訊跨了兩張資料表，我們可以考慮使用 INNER JOIN 的指令來結合兩張資料表的資料，下面的 SQL 指令顯示了結合 Northwind 資料庫的 Categories 和 Products 資料表的操作方式，該指令會將 Categories 資料表與 Products 資料表內的資料，將 Categories 的 CategoryID 欄位值與 Products 的 CategoryID 欄位值相等之資料進行結合，查詢出來的結果筆數會以儲存最多筆數的資料表為主。

```
1 SELECT *
2 FROM Northwind.dbo.Categories
```

```
3 INNER JOIN Northwind.dbo.Products
4 ON Northwind.dbo.Categories.CategoryID = Northwind.dbo.Products
   .CategoryID
```

## 5 新增 (INSERT)

如果今天你訂了一張車票，那麼在車票的訂票管理系統應該要在你訂了車票之後直接把你的訂票紀錄寫進資料庫裡面，而這個寫進去資料庫的動作我們就會使用到 SQL 新增 (INSERT) 的指令，新增的基本操作指令如下：

```
1 INSERT INTO table_name(column1, column2, ...)
2 VALUES ('value1', 'value2' ...)
```

上面指令的意思是將 value1, Value2... 寫入到 table 裡面相對應的 column1, column2 ...，如果你今天要寫入的欄位值數量剛好等於 table 裡面相對應的欄位數量，那麼你其實可以省略 table 後面的欄位名稱，你唯一需要注意的地方是確認插入的欄位值順序與 table 的欄位數量順序是否一致。

```
1 INSERT INTO table_name
2 VALUES ('value1', 'value2' ...)
```

假設我們要在 Northwind 資料庫的 Categories 資料表新增一筆資料，我們的新增指令可以這樣下，執行完指令後再次對 Categories 進行查詢，可以發現該資料表的資料多了一筆。

```
1 INSERT INTO Northwind.dbo.Categories
2 VALUES('test', 'test', 'test')
```

[HINT] 仔細觀察 Categories 的欄位數量有 4 個，可是我們在 VALUES 卻只有輸入 3 個值，為什麼不需要在 Northwind.dbo.Categories 指定欄位名稱呢？

## 6 更新 (UPDATE)

延續剛剛訂車票的例子，如果你訂完車票也付完款項並且取走車票，那麼在車票的資料庫裡面該車次的剩餘車票應該要少一張，這樣其他人在查詢該車次的剩餘車票時才會是正確的數量。該車次剩餘車票要少一張的這個動作就會使用到 SQL UPDATE 的指令，UPDATE 指令的基本操作語法如下。如同 SELECT 查詢指令一樣，WHERE 條件式是可選擇的，當我們更新指令不加入 WHERE 條件式進行資料篩選並更新資料時，表示我們的更新指令是對整張資料表內的資料進行更新，這樣的舉動是非常危險的，所以在做更新指令時必須非常的注意。

```
1 UPDATE table_name
2 SET column1 = 'value1', column2 = 'value2'...
3 [WHERE conditions]
```

上面 SQL 指令的意思是將資料表 (table) 內符合 WHERE 條件式的 column1 設定為 value1，column2 設定為 value2，依此類推。

假設我們要在 Northwind 資料庫的 Categories 資料表更新 CategoryID=9 的 Category-Name 欄位資料，我們的 SQL 指令可以這樣下。執行完指令後再次對 Categories 資料表進行查詢，可以發現 CategoryID=9 的資料，它的 CategoryName 已經被設定成 test2 了。

```
1 UPDATE Northwind.dbo.Categories
2 SET CategoryName = 'test2'
3 WHERE CategoryID = 9
```

## 7 刪除 (DELETE)

同樣延續訂車票的例子，假設你已經訂好某一天的車票但是你同時也訂到了那一天的另一班車次，所以兩車次的發車時間、行駛時間、到站時間及票價會是你兩班車次間取捨的主要因素，當然，你不太可能會在同一天內搭乘不同的班次前往同樣的目的地，因此你如果決定了其中一班的車次，另一班車次的預訂票就必須要取消，否則你會花了相當於兩倍的價錢來搭車，而這個取消預訂票的動作，在資料庫內就叫做刪除 (DELETE)，基本的 SQL 刪除語法如下。

```
1 DELETE FROM table_name
2 [WHERE conditions]
```

上面 SQL DELETE 指令的意思是將資料表 (table) 內符合 WHERE 條件式的資料進行刪除。如同 SELECT 以及 UPDATE 指令一樣，WHERE 條件式是可選擇的，當我們刪除指令不加入 WHERE 條件式進行資料篩選並刪除資料時，表示我們的刪除指令是對整張資料表內的資料進行刪除，這樣的舉動同樣是非常危險的，所以在做刪除指令時也必須非常的注意。

假設我們要在 Northwind 資料庫的 Categories 資料表刪除 CategoryID=9 的資料，我們的 SQL 指令可以這樣下。執行完指令後再次對 Categories 資料表進行查詢，可以發現 CategoryID=9 的資料已經成功被我們刪除了。

```
1 DELETE Northwind.dbo.Categories
2 WHERE CategoryID = 9
```

## 8 課堂練習

1. 使用 SQL 指令查詢 Northwind 資料庫內的 Order 資料表 EmployID=5 的資料。
2. 使用 SQL 指令查詢 Northwind 資料庫內的 Order 資料表 EmployID 介於 3 到 6 的資料，並且將查詢結果根據 EmployID 進行由大至小的排序。
  - [HINT]>= AND <= or BETWEEN...AND...
3. 使用 SQL 指令計算 Northwind 資料庫內的 Order 資料表共有幾筆訂單。
  - [HINT]SELECT COUNT
4. 使用 SQL 指令在 Northwind 資料庫內的 Products 資料表新增一筆 Products 的資料，內容不限。
  - [HINT] 需注意資料欄位內的資料型態
5. 使用 SQL 指令在剛剛新增的 Products 資料將 ProductName 設定為 Egg，UnitPrice 設定為 12.00。

6. 使用 SQL 指令刪除 Northwind 資料庫內的 Products 資料表你所新增的 Products 資料。
7. \* 使用 SQL 指令將 Northwind 資料庫內的 Categories、Order Details 以及 Products 資料表進行結合，查詢出每筆訂單詳細的產品名稱 (ProductName)、種類名稱 (CategoryName) 及單位價格 (UnitPrice)。
  - [HINT](nested) INNER JOIN