

# SQL語言處理資料庫

資料系  
林偉川

## 什麼是SQL語言

- SQL為「ANSI」(American National Standards Institute)標準的資料庫語言，SQL指令的語法可以存取和更新資料庫的記錄，目前Access、SQL Server、Informix、Oracle和Sybase等資料庫系統都支援ANSI的SQL語言。
- E. F. Codd建立**關連性資料庫**的觀念，同時就提出一個構想的**資料庫語言**，它是一種完整和通用的資料存取方式，雖然當時並沒有建立其語法，但這就是SQL語言的源起。

## SQL 語言種類

- SQL 語言的指令可以分為4個部分，如下所示：
  - 「資料定義語言DDL」(Data Definition Language)：資料表建立和欄位定義的SQL語法。
  - 「資料操作語言DML」(Data Manipulation Language)：資料表記錄**插入、刪除和更新**的相關指令。
  - 「資料查詢語言DQL」(Data Query Language)：資料表查詢指令，只有一個**SELECT**指令。
  - 「資料控制語言DCL」(Data Control Language)：**資料庫安全設定和權限管理**的相關指令。

3

## SQL 語言基本指令

- ASP.NET 網頁資料庫主要是使用SQL語言**資料操作和查詢指令**，總共只有**4個基本指令**，如下表所示：

指令	說明
INSERT	在資料表插入一筆新記錄
UPDATE	更新資料表的記錄，這些記錄是已經存在的記錄
DELETE	刪除資料表的記錄
SELECT	查詢資料表的記錄，使用條件子句查詢資料表符合條件的記錄

4

## Command物件的Execute()方法

- ADO.NET的Command物件提供3種Execute方法，如下表所示：

Execute 方法	說明
ExecuteNonQuery	執行的 SQL 指令但是不會傳回任何資料，通常是使用在資料庫操作指令 INSERT、DELETE 和 UPDATE
ExecuteScalar	執行 SQL 指令從資料表只取得 1 個欄位資料，如果是資料表，就是第 1 筆記錄的第 1 個欄位值
ExecuteReader	執行 SQL 指令傳回 DataReader 物件

5

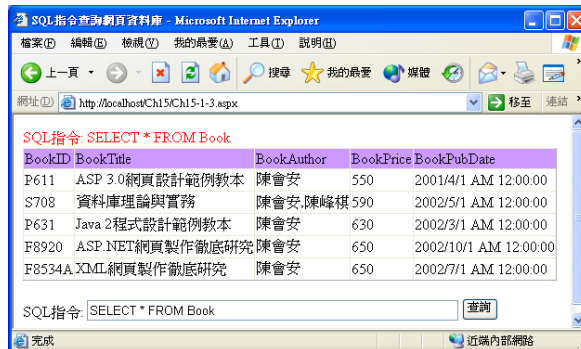
## ExecuteScalar()

- ExecuteScalar()方法，這個方法可以取得資料表指定記錄的單一欄位值，如下所示：  
`msg.Text = "查詢結果:" & objCmd.ExecuteScalar()`
- 程式碼執行Command物件objCmd的SQL指令，如果傳回的值不只一個，例如：SQL指令查詢的是多筆記錄，取得的是第1筆記錄的第1個欄位。

6

## SQL指令查詢網頁資料庫

- 為了SQL指令，筆者準備建立ASP.NET程式，只需在欄位輸入SQL指令，就可以使用DataGrid控制項，以表格顯示查詢的結果。



7

## SQL語言資料庫查詢指令

- SQL在資料查詢指令只有一個SELECT指令，完整的指令語法如下所示：

**SELECT column1, column2 FROM table WHERE conditions**

- column1~2為記錄欄位，table為資料表，而conditions為查詢條件，這個指令是「從資料表table取回符合WHERE條件所有記錄的欄位column1和column2」。
- conditions條件可以包含一些子句，例如：BETWEEN AND、ORDER BY等。

8

## SELECT設定查詢範圍

- SELECT敘述查詢資料表時可以只顯示部分欄位，SQL指令如下所示：

```
SELECT BookID, BookTitle, BookPrice FROM Book
```

- SELECT敘述顯示資料表Book的欄位BookID、BookTitle和BookPrice，欄位使用逗號分隔。

9

## SELECT查詢資料表所有的欄位

- SELECT敘述如果需要顯示記錄的所有欄位，我們並不用一一列出所有欄位，只需使用「\*」符號，就可以代表資料表記錄的所有欄位名稱，SQL指令如下所示：

```
SELECT * FROM Book
```

- SQL指令的執行結果是將資料表所有的記錄和所有的欄位都取回。

10

## SELECT查詢欄位不重複值

- 資料表記錄的欄位值如果擁有重複資料，在SELECT敘述可以加上**DISTINCT**指令，如此欄位擁有同值的記錄時，就只會顯示其中一筆，SQL指令如下所示：

```
SELECT DISTINCT BookPrice FROM Book
```

- SQL指令字串的欄位BookPrice如果有重複值，只會顯示其中一筆。

11

## SELECT敘述設定查詢範圍

- SELECT敘述查詢資料表時，在**DataGrid**控制項上方會顯示欄位標題，預設是欄位名稱，我們可以使用**AS**關鍵字設定別名，即使用指定的名稱來顯示，SQL指令如下所示：

```
SELECT BookID AS 書號, BookTitle AS 書名,  
BookPrice AS 書價 FROM Book
```

- SELECT敘述顯示資料表Book的欄位BookID、BookTitle和BookPrice，各欄位的別名分別為**書號**、**書名**和**書價**，欄位使用逗號分隔，可以看到顯示的欄位標題改為別名。

12

## WHERE條件子句

- SELECT敘述的**WHERE**條件子句才是**資料庫查詢**的主角，SELECT敘述只是指定從那個**資料表**和需要取得那些**欄位**，WHERE子句才是**資料庫查詢**的條件。

13

## WHERE條件運算子

- WHERE條件的欄位值可能為文字、數字或日期時間，可以使用的運算子，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
LIKE	包含子字串

14

## WHERE條件值為字串

- WHERE條件的欄位如果是字串需要加上單引號，條件是字串比較，可以使用的運算子和範例，如下表所示：

運算子	範例
=	SELECT * FROM Book WHERE BookID='P631'
>	SELECT * FROM Book WHERE BookID>'P631'
>=	SELECT * FROM Book WHERE BookID>='P631'
<	SELECT * FROM Book WHERE BookID<'P631'
<=	SELECT * FROM Book WHERE BookID<='P631'
<>	SELECT * FROM Book WHERE BookID<>'P631'

15

## WHERE條件包含字串

- 如果是文字內容的欄位，還可以使用【LIKE】包含運算子，LIKE運算子只需包含此子字串就符合條件，配合萬用字元可以建立字串範本來進行比對，如下表所示：

萬用字元	說明
%	代表任何的子字串或字元
_	代表一個字元

16



## WHERE條件包含字串

- 查詢子字串的WHERE條件，SQL指令如下：  
`SELECT * FROM Book WHERE BookID LIKE '%8%'`
- SQL指令的條件因為字元「%」可以代表其它任何字元或字串，只需欄位值擁有子字串「8」就符合條件。
- 「\_」萬用字元可以代表一個任何字元，SQL指令如下所示：  
`SELECT * FROM Book WHERE BookID LIKE 'P6_1'`
- SQL指令中BookID欄位的第3個字元可以是任何字元。

17

## WHERE條件值為數字

- WHERE條件如果為數字欄位就不要使用單引號，可以使用的運算子和範例，如下表所示：

運算子	範例
=	SELECT * FROM Book WHERE BookPrice=590
>	SELECT * FROM Book WHERE BookPrice>590
>=	SELECT * FROM Book WHERE BookPrice>=590
<	SELECT * FROM Book WHERE BookPrice<590
<=	SELECT * FROM Book WHERE BookPrice<=590
<>	SELECT * FROM Book WHERE BookPrice<>590

18

## WHERE條件值為日期時間

- 標準的ANSI-SQL擁有3種日期時間的資料類型，如下表所示：

資料類型	說明
DATE	日期格式，格式為 YYYY-MM-DD
TIME	時間格式，格式為 HH:MM:SS.nn
TIMESTAMP	日期時間格式，格式為 YYYY-MM-DD HH:MM:SS.nn

- Microsoft Jet-SQL只有一種**DATETIME**資料類型，能夠儲存ANSI-SQL的**DATE**、**TIME**和**TIMESTAMP**這3種資料類型。

19

## WHERE條件值為日期時間

- 如果條件值為日期時間資料時，請將**日期與時間**的字串使用「#」符號括起來，可以使用的運算子和範例，如下表所示：

運算子	範例
=	SELECT * FROM Book WHERE BookPubDate=#2002/3/1#
>	SELECT * FROM Book WHERE BookPubDate>#2002/3/1#
>=	SELECT * FROM Book WHERE BookPubDate>=#2002/3/1#
<	SELECT * FROM Book WHERE BookPubDate<#2002/3/1#
<=	SELECT * FROM Book WHERE BookPubDate<=#2002/3/1#
<>	SELECT * FROM Book WHERE BookPubDate<>#2002/3/1#

20

## AND與OR多條件AND運算子

- AND運算子連接的**前後條件都必須同時成立**，整個條件才成立，例如：圖書書號包含「1」且圖書名稱有「ASP」子字串，SQL指令如下所示：

```
SELECT * FROM Book WHERE BookID LIKE  
'%1%' AND BookTitle LIKE '%ASP%'
```

21

## AND與OR多條件OR運算子

- OR運算子連接的**前後條件只需任何一個條件成立即可**，例如：圖書書號包含「1」或圖書名稱有「ASP」子字串，SQL指令如下所示：

```
SELECT * FROM Book WHERE BookID LIKE  
'%1%' OR BookTitle LIKE '%ASP%'
```

22

## AND與OR多條件WHERE條件

- WHERE條件子句如果很複雜，我們可以同時使用**AND**和**OR**運算子連接3、4個或以上的**條件**。例如：查詢圖書名稱擁有「**ASP**」子字串或是圖書書號有「**3**」子字串且圖書價格小於650，SQL指令如下所示：

```
SELECT * FROM Book WHERE BookTitle LIKE '%ASP%' OR  
BookID LIKE '%3%' AND BookPrice < 650
```

23

## 擁有括號WHERE條件

- WHERE條件的**括號擁有較高的優先順序**，所以加上括號將產生不同的查詢結果，例如：查詢圖書名稱有「ASP」子字串或圖書編號有「3」子字串，這兩個條件有括號，而且圖書價格需要小於650的記錄，SQL指令如下所示：

```
SELECT * FROM Book WHERE (BookTitle LIKE '%ASP%'  
OR BookID LIKE '%3%') AND BookPrice < 650
```

24

## Order By排序子句

- SQL指令的查詢結果如果需要**排序顯示**，可以指定依欄位值由小到大或由大到小進行排序。

25

## Order By由小到大排序

- SQL指令只需加上ORDER BY子句指定排序欄位，例如：查詢書價大於等於590元的記錄，使用圖書書價欄位進行排序，SQL指令如下所示：  

```
SELECT * FROM Book WHERE BookPrice>=590 ORDER BY BookPrice ASC
```
- SQL指令使用圖書書價BookPrice欄位**由小到大**進行排序，此為預設方式。

26

## Order By由大到小排序

- 如果排序順序需要倒過來由大到小，只需在最後加上DESC關鍵字，如下所示：

```
SELECT * FROM Book WHERE BookPrice >= 590 ORDER  
BY BookPrice DESC
```

- SQL指令使用圖書書價BookPrice欄位，由大到小進行排序。

27

## BETWEEN ... AND 資料範圍子句

- BETWEEN AND可以定義SELECT敘述WHERE條件子句的範圍，範圍值可以是文字、數字和日期時間，SQL指令如下所示：

```
SELECT * FROM Book WHERE BookPubDate BETWEEN  
#2002/1/1# AND #2002/9/30#
```

- SQL指令為日期時間範圍，查詢2002年1月1日到9月30日出版的圖書。

28

## BETWEEN ... AND 資料範圍子句

- BETWEEN AND使用BookPrice欄位數字範圍的SQL指令，如下所示：

```
SELECT * FROM Book WHERE BookPrice BETWEEN 500  
AND 640
```

- SQL指令為數字範圍，查詢圖書價格在500到640之間的圖書。

29

## IN運算子(文字)

- IN運算子可以設定文字和數字清單，欄位值必須為其中之一才符合條件，例如：圖書書號需要是指定的幾個字串，SQL指令如下所示：

```
SELECT * FROM Book WHERE BookID IN ('P611', 'P631',  
'F8920')
```

- SQL指令只有書號欄位BookID屬於P611、P631和F8920才符合條件。

30

## IN運算子(數字)

- 查詢書價為550和650的記錄，SQL指令如下所示：  
`SELECT * FROM Book WHERE BookPrice IN (550, 650)`
- SQL指令只有書價欄位BookPrice為550和650才符合條件。

31

## NOT運算子

- NOT運算子可以搭配前面的WHERE條件子句，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

32



## NOT運算子

- 圖書書號除了幾個指定字串外的記錄，SQL指令如下所示：

```
SELECT * FROM Book WHERE BookID NOT IN ('P611',  
    'P631', 'F8920')
```

- SQL指令只有書號欄位BookID 不屬於P611、P631和F8920才符合條件。

33

## 聚合函數

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

34

## SQL語言的資料庫操作

- 在資料表插入、刪除和更新記錄的操作需要使用**Command物件**執行SQL指令，使用的是**ExecuteNonQuery()**方法，如下所示：

```
count = objCmd.ExecuteNonQuery()
```

- 指令左邊的變數**count**可以取得影響的記錄數，這個方法執行SQL指令，但是**並不會傳回任何記錄資料**。

35

## 插入資料表記錄

- SQL插入記錄操作指令就是**新增一筆記錄**到資料表，INSERT指令的基本語法，如下所示：

```
INSERT INTO table (column1,column2,...) VALUES ('value1',  
'value2',...)
```

- SQL指令的**table**為準備插入記錄的**資料表名稱**，**column1~n**為資料表內的欄位名稱，**不需要全部的欄位**，**value1~n**是對應的欄位值。

36

## 更新資料表記錄

- SQL更新記錄指令是將資料表內符合條件的記錄，更新指定欄位的內容，UPDATE指令的基本語法，如下所示：

**UPDATE table SET column1 = 'value1' WHERE conditions**

- table是資料表，SET子句column1是資料表的欄位名稱，不用全部只需指定需要更新的欄位，value1是更新的欄位值。
- 如果更新的欄位不只一個，請使用逗號分隔，如下所示：

**UPDATE table SET column1 = 'value1' , column2 = 'value2' WHERE conditions**

37

## 刪除資料表記錄

- SQL刪除記錄指令是將資料表內符合條件的記錄刪除掉，DELETE指令的基本語法，如下所示：

**DELETE FROM table WHERE conditions**

- 指令table是資料表，WHERE子句conditions為刪除記錄的條件，就是「將符合conditions條件的記錄刪除掉」。

38