

03

建立資料庫物件 (Creating Database Objects)

- 3-1 資料型態
- 3-2 建立資料庫物件
- 3-3 檢視 (views)
- 3-4 預存程序、函數

3-1. 資料型態

在SQL Server 資料庫中，無論是資料表的欄位、預存程序的參數、或程式中的變數，我們都必須為其設定一個適當的資料型態。資料型態的主要目的是用來宣告該欄位或變數所存放的資料性質，例如整數、字串、或日期等。

SQL Server 除了系統內建的資料形態之外，也提供使用者可以自訂資料型態。

◆ 系統內建的資料型態

▮**整數**：整數的資料型態用來定義整數資料。

資料型態	位元數 (Byte)	資料範圍
tinyint	1 Bytes	$0 \sim 2^8 - 1$ (0 ~ 255)
smallint	2 Bytes	$-2^{15} \sim 2^{15} - 1$ (-32,768 ~ 32,767)
int	4 Bytes	$-2^{31} \sim 2^{31} - 1$ (-2,147,483,648 ~ 2,147,483,467)
bigint	8 Bytes	$-2^{63} \sim 2^{63} - 1$ (-9,223,372,036,854,775,808 ~ 9,223,372,036,854,776,807)
bit	1 Bytes	0、1、Null

精確位數：精確位數的資料型態用來定義小數部分。

資料型態	位元數 (Byte)	資料範圍
numeric	視精確度而定	$-10^{38}+1 \sim 10^{38}-1$
decimal	視精確度而定	$-10^{38}+1 \sim 10^{38}-1$

浮點數：float 與real 資料型態可用來表示當數值很大或很小時的近似值，以科學記號表示，E+38 代表 10^{38} 。

資料型態	位元數 (Byte)	資料範圍
real	4 Bytes	$-3.40E+38 \sim 3.40E+38$ 最多可表示 7 位數
float	8 Bytes	$-1.79E+308 \sim 1.79E+308$ 最多可表示 15 位數

字串：字串資料型態用來存放字串資料，例如"apple123" 或" 微軟公司" 等由數字、英文字母大小寫、或特殊符號所組成的字串。

資料型態	位元數 (Byte)	資料範圍
char(n)	1 字元 1byte，為固定長度，未填滿的資料會以空白補上，若未指定 n，則預設值為 1。	1 ~ 8000 個字元
varchar(n)	1 字元 1byte，儲存空間視字元而定，n 位元即佔 nbyte，若未指定 n，則預設值為 1。	1 ~ 8000 個字元
varchar(max)	可變長度，儲存空間視字元而定，n 位元即佔 nbyte，最大可達 2GB。	1 ~ $2^{31} - 1$ 個字元
text	可變長度，儲存空間視字元而定，n 位元即佔 nbyte，最大可達 2GB。	1 ~ $2^{31} - 1$ 個字元

Unicode 字串：此資料型態與上述字串型態類似，但Unicode 字串是用來存放Unicode 字元，它是以 2 個bytes 為每個字元編碼，如此一來，可以有足夠的不同模式 (65,536) 來涵蓋大部份的語言，若資料類型是使用1byte (8bits) 編碼，則只能表示256 (2⁸) 種不同的字元，因此若資料中有包含多國語言，則可以使用Unicode 資料型態。

資料型態	位元數	資料範圍
nchar(n)	1 字元 2Byte，為固定長度，未填滿的資料會以空白補上，若未指定 n，則預設值為 1。	1 ~ 4000 個字元
nvarchar(n)	1 字元 2Byte，儲存空間視字元而定，n 位元即佔 2nbyte，若未指定 n，則預設值為 1。	1 ~ 4000 個字元
nvarchar(max)	1 字元 2Byte，儲存空間視字元而定，n 位元即佔 2nbyte，最大可達 2GB。	1 ~ 2 ³⁰ -1 個字元
ntext	1 字元 2Byte，可變長度，儲存空間視字元而定，n 位元即佔 2nbyte，最大可達 2GB。	1 ~ 2 ³⁰ -1 個字元

Binary 字串：此類型用來定義位元字串，二進位常數前面以 0x 開頭，後面接著以16 進位表示的位元模式。例如，0x5B 是指16 進位值 5B，這相當於10 進位值 91。當使用者需要儲存SID (安全性識別碼)、GUID (全域唯一識別碼) 時，便可以使用此資料型態

資料型態	位元數 (Bytes)	資料範圍
binary(n)	固定長度 8000 Bytes，不足自動補上 0x00，若未指定 n，則預設值為 1	1 ~ 8000 Bytes
varbinary(n)	可變長度，儲存空間視輸入的資料而定 若未指定 n，則預設值為 1	1 ~ 8000 Bytes
varbinary(max)	可變長度，儲存空間視輸入的資料而定，最大可達 2GB	1 ~ 2 ³¹ -1 Bytes
image	可變長度，儲存空間視輸入的資料而定，最大可達 2GB	1 ~ 2 ³¹ -1 Bytes

日期 - 時間 - 貨幣 - Timestamp

資料型別	位元數	資料範圍
datetime	8 Bytes	1753/1/1 ~ 9999/12/31 ex: 2008-11-27 08:08:08.888
smalldatetime	4 Bytes	1900/1/1 ~ 2079/6/6 ex: 2008-11-24 15:11
money	8 Bytes	$-2^{63} \sim 2^{63} - 1$ 小數 4 位 (-922337203685477.5808 ~ 922337203685477.5807)
smallmoney	4 Bytes	$-2^{31} \sim 2^{31} - 1$ 小數 4 位 (-214748.3648 ~ 214748.3647)
timestamp	8 Bytes	8 Bytes 的 16 進位值
uniqueidentifier	16 Bytes	16 Bytes 的 16 進位值

3-2 建立資料庫物件

建立資料庫、在資料庫中建立資料表、插入資料至資料表、更新資料以及讀取資料。

◆ 建立資料庫

.....
語法
.....

```
CREATE DATABASE db_name
```

◆ 建立資料表

.....
語法
.....

```
CREATE TABLE table_name
```

◆ 條件約束

條件約束 (Constraint) 限制資料表的記錄，只有符合約束的記錄才可以儲存。

條件約束的種類：

- 1) 主索引條件約束
- 2) 外部索引鍵條件約束
- 3) 唯一條件約束
- 4) 檢查條件約束

3-3 檢視 (views)

「View」一般稱為「檢視」或稱為「視界」，由「View」所定義的表格，稱為「檢視表」。檢視表是不同於資料表的基底關聯表。資料表是資料庫中實際儲存資料的地方，但檢視表只是一個虛擬的表格，其資料並不需要真正存在磁碟理。

通常檢視表只是資料表的一部分而非全部，我們可以藉由控制不同使用者所能看到的資料內容，來提供資料安全性，以達到維護資料庫安全的目的。

◆ 檢視表可以提供使用者資料查詢，它具有以下優點及缺點：

▣ 檢視的優點

- 1) 使用者可以將比較常用的合併或比較複雜的查詢建立檢視表，以簡化日後查詢的作業。
- 2) 對使用者來說，不需要的資料不放在檢視表中，可清楚掌握資料的重點。
- 3) 可利用檢視，定義每位使用者查看資料庫的角度。
- 4) ▣ 檢視表的欄位名稱可以重新命名，可增加程式的可讀性。
- 5) 檢視表可以建立在其他檢視表之上，最多可包含250個欄位。
- 6) 檢視表於資料庫中的定義，不會因為所依據的表格被刪除而消失
- 7) 針對不同的使用者，可以限制欄位的存取，以增加資料的安全性

◆ 檢視表可以提供使用者資料查詢，它具有以下優點及缺點：

▫ 檢視的缺點

- 1) 檢視表的存取速度，較資料表的存取速度慢，因為檢視是當使用者需要使用查詢時，才會由資料表產生，必須經過轉換動作，因此會比直接從資料表查詢還要慢。
- 2) 無法對檢視表增加索引，其索引是用所依據表格的索引。
- 3) ▫使用者只能在目前的資料庫建立檢視表。

◆ 建立檢視 (CREATE VIEW)

使用CREATE VIEW 敘述可以建立檢視表，也就是由查詢定義其內容（資料行和資料列）的虛擬資料表。使用者可以使用CREATE VIEW敘述來建立資料庫中一個或多個資料表內資料的檢視。

檢視只能建立在目前資料庫中，檢視最多可有 1,024 個資料行。通常檢視有下列幾項目的：

- 1) 對焦 (Focus)、簡化和自訂每位使用者，查看資料庫的角度
- 2) 做為安全機制，讓使用者能夠透過檢視存取資料，但不將直接存取基底資料表的權限，授與使用者。
- 3) 提供回溯相容介面，以模擬其結構描述已變更的資料表。

◆ 修改檢視 (ALTER VIEW)

使用ALTER VIEW 敘述，用以修改先前建立的檢視。ALTER VIEW 敘述並不會影響相依的預存程序或觸發程序，且不會變更權限。若要執行 ALTER VIEW，至少需要 OBJECT 的 ALTER 權限。

如果先前所用的檢視是利用ALTER VIEW 來修改的，Database Engine會取得檢視的獨佔結構描述鎖定。當授與鎖定時，檢視並沒有使用中的使用者，Database Engine 會從程序快取中，刪除檢視的所有副本。現有參考這份檢視的計畫會保留在快取中，但在叫用它時，會重新編譯。

您可以將ALTER VIEW 套用在索引檢視上；不過，ALTER VIEW 會無條件地刪除檢視的所有索引。

◆ 刪除檢視 (DROP VIEW)

若使用者不再需要檢視，可以使用DROP VIEW 敘述從目前資料庫移除一個或多個檢視。使用DROP VIEW 敘述只會刪除檢視的定義，並不會影響資料表本身。執行DROP VIEW 敘述者必須是該檢視的擁有者。

當使用者刪除檢視時，檢視的定義、檢視的其他相關資訊、以及檢視的所有權限都會被刪除。若要執行 DROP VIEW，至少需要SCHEMA 的 ALTER 權限，或 OBJECT 的 CONTROL 權限。

3-4 預存程序、函數

◆ 預存程序 (Stored Procedure)

預存程序類似程式語言的副程式，是一種儲存在資料庫中的可執行物件。使用者可以事先編譯SQL 程式，並將它儲存，日後只要使用EXECUTE 指令便可執行該程序。預存程序允許使用者宣告變數、傳遞參數，也可以重複被呼叫、執行，執行後傳回結果。不同的預存程序可以互相呼叫，但無法在運算式中直接呼叫預存程序。

預存程序可以包含資料存取、資料的輸入輸出、流程控制、邏輯處理、錯誤處理……等，使用時非常具有彈性。預存程序的優點包括：

1. 提升執行速度。
2. 降低網路負載。
3. 增加資料安全性。
4. 模組化的程式設計。

不同的資料庫管理系統，提供不同的預存程序之語法，因此在SQL Server 使用的存程序不能在其他系統上使用，這是預存程序可攜性較差的缺點。

若要建立程序，使用者必須擁有資料庫的 **CREATE PROCEDURE** 權限，修改預存程序使用 **ALTER PROCEDURE** 指令，執行預存程序則用 **EXECUTE** 指令。

◆ 觸發程序

MicrosoftSQL Server 提供兩種機制，來強制執行商務規則和資料完整性

1. 條件約束
2. 觸發程序

觸發程序是一種特殊的預存程序，它會在執行語言事件時自動生效

SQL Server 包含三種一般的觸發程序：

1. DML 觸發程序
2. DDL 觸發程序
3. 登入觸發程序

◆ **DML 觸發程序有下列幾種用途：**

- 1) 它們可以串聯資料庫中相關資料表的變更，不過，透過串聯的參考完整性條件約束，來執行這些變更會更有效率。
- 2) 它們可以預防惡意或錯誤的 INSERT、UPDATE 和 DELETE 作業，並且強制執行其他比 CHECK 條件約束，所定義的限制更複雜的限制。
- 3) 與 CHECK 條件約束不同的是，DML 觸發程序可以參考其他資料表中的資料行。
- 4) 它們可以評估資料修改前後的資料表狀態，並依據這些差異採取動作。
- 5) 資料表中相同類型（INSERT、UPDATE 或 DELETE）的多個 DML 觸發程序，允許對相同的修改陳述式，執行多種不同的動作。
- 6) 未特別說明，一般觸發程序是指 DML 觸發程序。

◆ 函數

函數與預存程序的功能類似，它可以接受參數、執行複雜計算、以及傳回各種資料型態的值。SQL Server 提供兩種函數，分別是「內建函數」與「使用者自訂函數」。使用函數的好處如下：

- 1) 模組化的程式設計。
- 2) 函數只需建立一次，便可將它儲存在資料庫中，日後可以無限次呼叫該函數。
- 3) 可加快執行速度。
- 4) 可加快執行速度。
- 5) 可降低網路負載。
- 6) 可降低網路的資料流量。

◆ 使用者自訂函數雖與預存程序的功能類似，但還是有些許差異：

- 1) 使用者的自訂函數，適用於經常需要進行計算或資料存取的作業，再將結果傳回給運算式、SELECT 子句或FROM 子句中使用。
- 2) 預存程序，較適用於對資料庫進行操作或設定，通常不需要將執行結果傳回，或將結果傳回到執行該預存程序的應用程式中。
- 3) 函數能傳回一個值，傳回的值可以是單一值或資料表，預存程序可以使用多個傳出參數。