

# CHAP 07 建立資料表與資料庫圖表

- 7-1 使用 **SQL Server Management Studio** 建立資料表
- 7-2 使用 **SQL Server Management Studio** 修改資料表
- 7-3 使用 **SQL Server Management Studio** 建立資料表間的關聯
- 7-4 設定條件約束維護資料完整性
- 7-5 使用 **SQL Server Management Studio** 刪除資料表
- 7-6 資料庫圖表與圖表物件

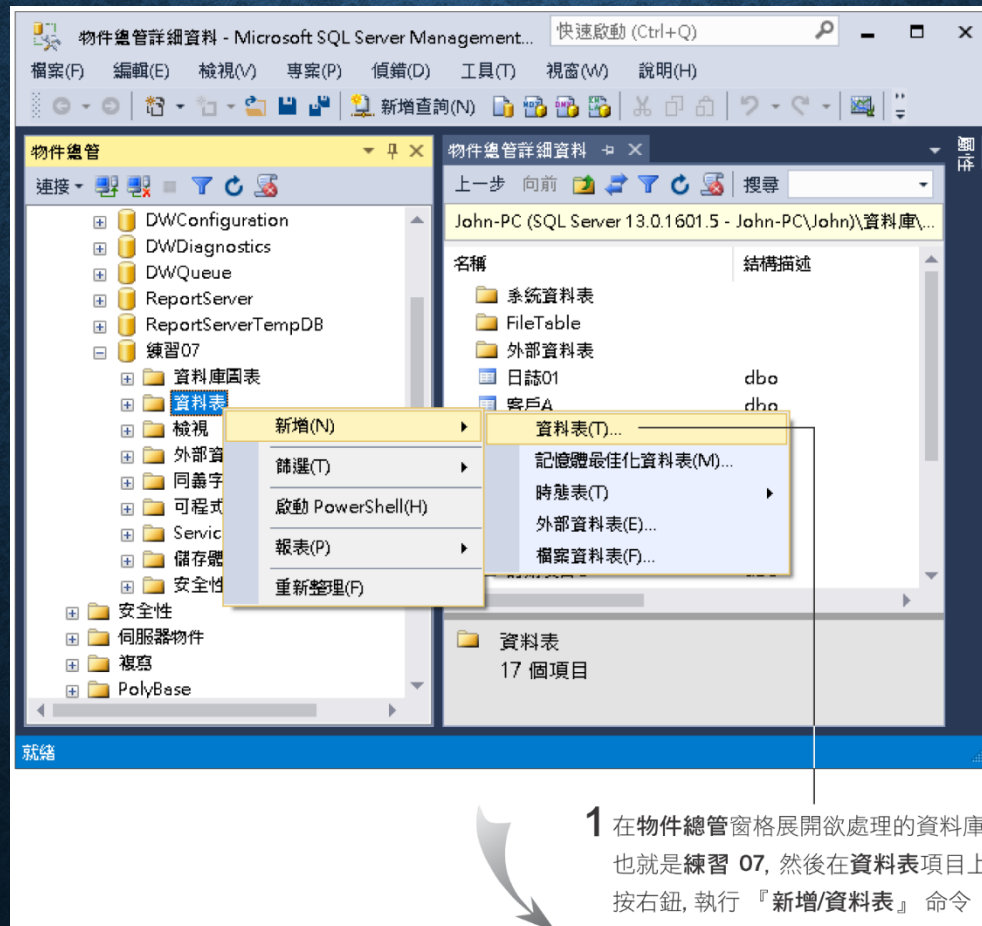
# CHAP 07 建立資料表與資料庫圖表

- 7-7 用 **CREATE TABLE** 敘述建立資料表
- 7-8 用 **ALTER TABLE** 敘述修改資料表
- 7-9 用 **DROP TABLE** 敘述刪除資料表
- 7-10 暫存資料表
- 7-11 自動紀錄資料異動 - **Temporal** 資料表

# 7-1 使用 SQL SERVER MANAGEMENT STUDIO 建立資料表

- 開啟建立資料表的視窗介面
- 定義資料表的欄位屬性
- 設定 **Primary key**
- 存檔 - 設定資料表名稱

# 開啟建立資料表的視窗介面



# 開啟建立資料表的視窗介面

2 請在此輸入欄位  
(資料行) 名稱

3 設定資料型別 (詳細  
說明請參考第 4 章)

4 設定是否允  
許 NULL 值

資料行名稱	資料類型	允許 Null
書籍編號	int	<input checked="" type="checkbox"/>

資料行屬性

(一般)

(名稱)	書籍編號
允許 Null	是
資料類型	int
預設值或繫結	

資料表設計工具

RowGuid	否
大小	4

(一般)

屬性

[資料表] dbo.Table\_1

(識別)

(名稱)	Table_1
伺服器名稱	john-pc
描述	
結構描述	dbo
資料庫名稱	練習07

資料表設計工具

Text/Image	PRIMARY
已複寫	否
可編索引	是

規則資料空間 PRIMARY

資料列 GUID	
鎖定擴大	資料表
識別欄位	

(識別)

就緒


此處可設定欄位 (資料行) 的屬性 (隨後會詳細介紹)

可在此設定資料表的屬性 (7-2 節會詳細介紹)。如果沒看到屬性窗格, 可按 **F4** 鍵將之顯示出來

# 定義資料表的欄位屬性

屬性	說明
長度	設定欄位的儲存空間, 單位是 byte。有些資料型別的長度是固定的, 例如 int、real、smallmoney 的長度固定為 4 bytes, 若欄位設為這類型別, 就不用再設定長度了 (使用這些長度固定的資料型別時, 下方窗格 <b>資料行屬性</b> 頁次中, 也不會出現 <b>長度</b> 欄位)。通常是字串類和二元碼類的型別才需設定長度
預設值或繫結	設定欄位的預設值。新增資料時, 如果沒有給予欄位值, 便填入預設值。除了 timestamp 型別外, 其它型別的欄位皆可設定此項屬性
整數位數	設定欄位的整數位數。只有 decimal 和 numeric 型別可設定這項屬性, bigint、int、smallint、tinyint、money、smallmoney、float、和 real 型別的精確度都是固定的
小數位數	設定欄位的小數位數。只有 decimal 和 numeric 型別可設定這項屬性
RowGuid	設定此欄位的值是全域 (全世界) 唯一的。若設為 <b>是</b> , 則我們通常還會將其 <b>預設值</b> 設為 newid(), 藉此函數產生全域唯一識別碼。只有 uniqueidentifier 型別的欄位才可以設定這項屬性 (詳見 7-41 頁)
可編索引	顯示此欄位是否能被索引, 關於索引的詳細說明請參考第 12 章。

# 定義資料表的欄位屬性

屬性	說明
已複寫	顯示此欄位是否已被設定複寫 (replicate) 到其他目的地
計算資料行規格/公式	設定計算欄位的運算式 (詳見 7-49 頁)
定序	設定欄位要套用的定序名稱及排序選項, 預設是使用 <b>資料庫預設值</b> 。將插入點移入此欄, 按下  鈕則會開啟一個交談窗, 讓您變更定序的設定。只有 char、varchar、text、nchar、nvarchar、ntext 型別可設定這項屬性
描述	可輸入欄位的補註說明, 每個欄位皆可設定這個屬性
識別規格/(為識別)	設定是否讓欄位值自動編號。有 2 個選擇： <b>否</b> 表示不自動編號、 <b>是</b> 表示欄位會根據 <b>識別值種子</b> 和 <b>識別值增量</b> 的設定自動編號產生欄位值。只有 bigint、int、smallint、tinyint、decimal、和 numeric 可設定這項屬性
識別規格/識別值種子	設定自動編號的起始值, 預設為 1。當 <b>識別屬性為是</b> 時才能設定此項
識別規格/識別值增量	設定自動編號的遞增值, 預設亦為 1。當 <b>識別屬性為是</b> 時才能設定此項

一個資料表中最多只能有一個識別屬性欄位及一個 RowGuid

# 定義資料表的欄位屬性

欄位名稱	資料型別	長度	允許 Null	屬性
書籍編號	int		否	識別規格/(為識別)：是 識別值種子：1 識別值增量：1
書籍名稱	nvarchar	50	是	
價格	smallmoney		是	
出版公司	nchar	20	是	



# 定義資料表的欄位屬性

欄位名稱	資料型別	長度	允許 Null	屬性
書籍編號	int		否	識別規格/(為識別)：是 識別值種子：1 識別值增量：1
書籍名稱	nvarchar	50	是	
價格	smallmoney		是	
出版公司	nchar	20	是	

資料行屬性

發佈的合併

資料類型扼要

識別規格

(為識別)

識別值種子

識別值增量

(為識別)

此欄不可修改

必須在這些子項目中修改

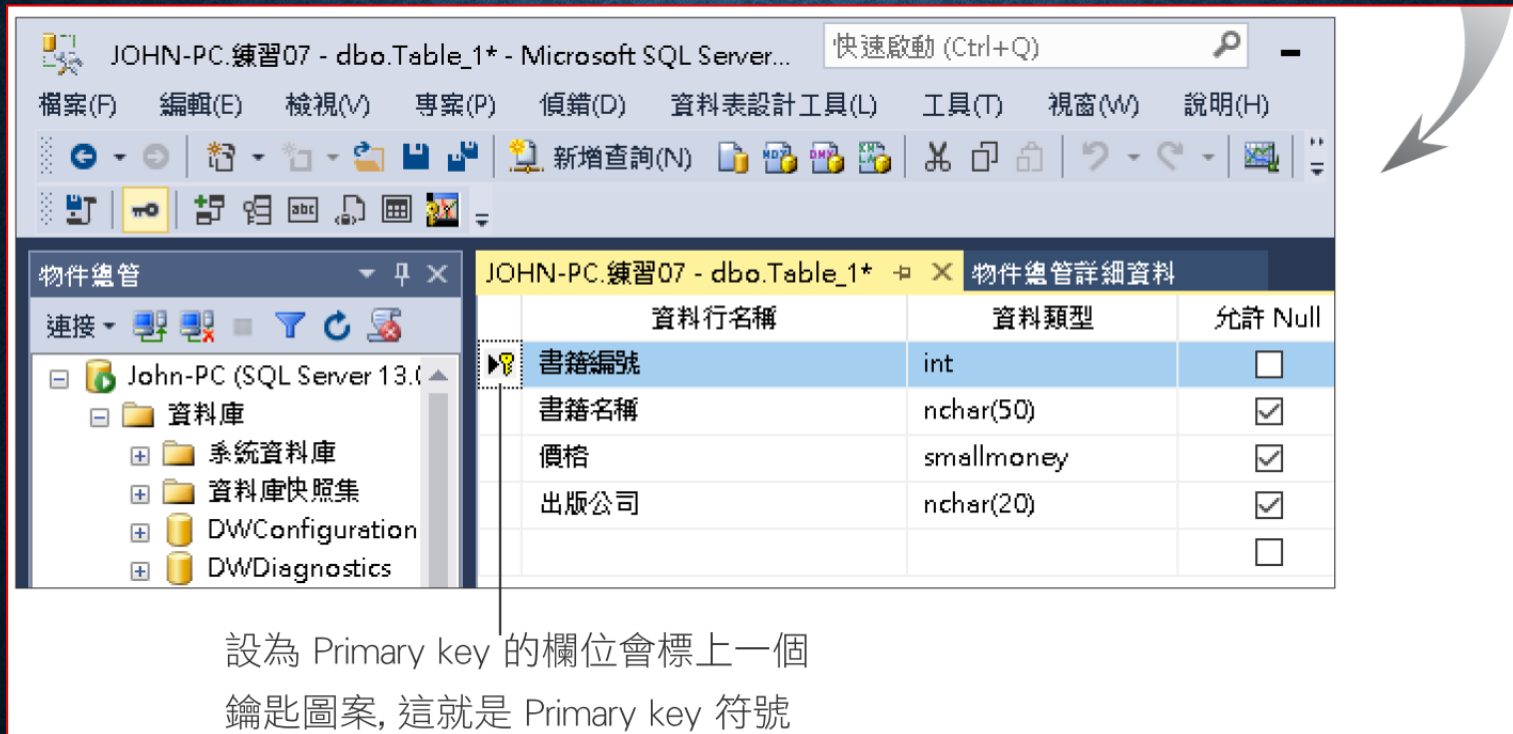
# 設定 PRIMARY KEY

1 按此鈕選取書籍編號欄位

2 按此鈕, 或在選取的欄位上按右鈕執行『設定主索引鍵』命令, 即可將選取的欄位設成 Primary Key

資料行名稱	資料類型	允許 Null
書籍編號	int	<input type="checkbox"/>
書籍名稱	nchar(50)	<input checked="" type="checkbox"/>
價格	smallmoney	<input checked="" type="checkbox"/>
出版公司	nchar(20)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

# 設定 PRIMARY KEY



The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The main window displays the '物件總管詳細資料' (Object Properties) for a table named 'JOHN-PC.練習07 - dbo.Table\_1\*'. The table structure is as follows:

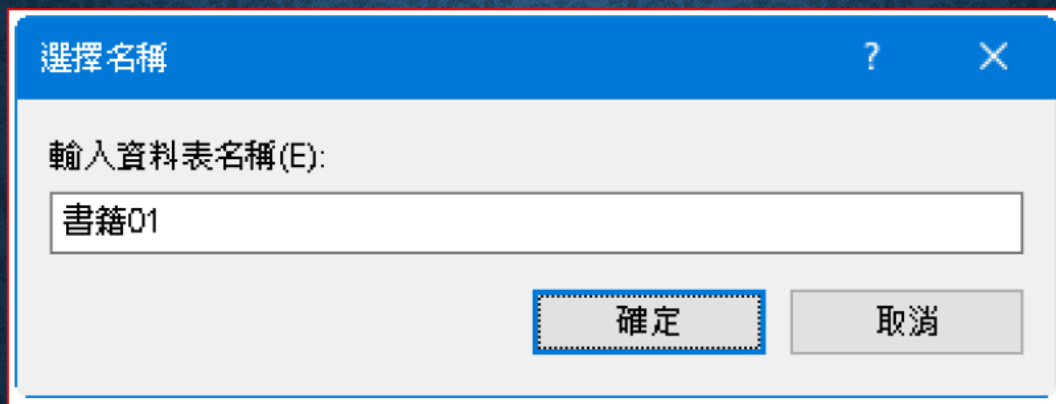
資料行名稱	資料類型	允許 Null
書籍編號	int	<input type="checkbox"/>
書籍名稱	nchar(50)	<input checked="" type="checkbox"/>
價格	smallmoney	<input checked="" type="checkbox"/>
出版公司	nchar(20)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

The '書籍編號' column is highlighted with a yellow background and a key icon in the first column, indicating it is the primary key. A line points from the text below to this key icon.

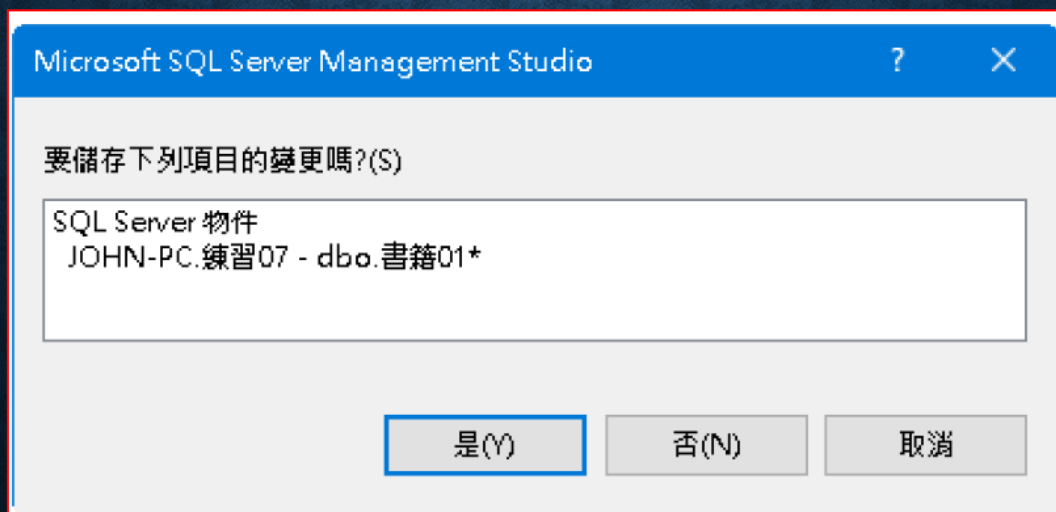
設為 Primary key 的欄位會標上一個  
鑰匙圖案, 這就是 Primary key 符號

若 **Primary key** 是由多個欄位組成，先選一個欄位後，按住 **Ctrl** 鍵，再選其他需要的欄位，最後再將這些欄位設為主鍵。

# 存檔 — 設定資料表名稱



若資料表已存在，經過更動後再儲存，則出現如下對話視窗



# 7-2 使用 SQL SERVER MANAGEMENT STUDIO 修改資料表

- 開啟資料表設定窗格
- 插入、搬移、刪除資料行
- 設定資料表屬性
- 更改資料表名稱

# 開啟資料表設定窗格

1 展開資料庫的資料表項目

2 在欲修改的資料表上按右鈕，如書籍 M

3 執行『設計』命令

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the '物件總管' (Server Enterprise Manager) tree is expanded to show the '資料表' (Tables) folder under the 'JOHN-PC.練習07 - dbo.' database. The 'dbo.書籍M' table is selected, and a context menu is open with the '設計(G)' (Design) option highlighted. The main pane shows the '物件總管詳細資料' (Server Enterprise Manager Details) for the selected table, displaying a list of columns with their names, data types, and whether they allow NULL values.

資料行名稱	資料類型	允許 Null
書籍編號	int	<input type="checkbox"/>
書籍名稱	varchar(50)	<input checked="" type="checkbox"/>
價格	smallmoney	<input checked="" type="checkbox"/>
出版公司	char(20)	<input checked="" type="checkbox"/>

資料行屬性

(一般)

(名稱)	書籍編號
允許 Null	否
資料類型	int
預設值或繫結	

資料表設計工具

RowGuid	否
大小	4

(一般)

可以重新修改資料表了

# 插入、搬移、刪除資料行

- 插入空白資料行

1 按此鈕選取價格欄位

資料行名稱	資料類型	允許 Null
書籍編號	int	<input type="checkbox"/>
書籍名稱	varchar(50)	<input checked="" type="checkbox"/>
價格	smallmoney	<input type="checkbox"/>
出版公司	char(20)	<input checked="" type="checkbox"/>

2 按右鈕執行『插入資料行』命令

在價格欄之上插入  
一列空白資料行了

資料行名稱	資料類型	允許 Null
書籍編號	int	<input type="checkbox"/>
書籍名稱	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>
價格	smallmoney	<input checked="" type="checkbox"/>
出版公司	char(20)	<input checked="" type="checkbox"/>

# 插入、搬移、刪除資料行

- 搬移資料行


資料行名稱	資料類型	允許 Null
書籍編號	int	<input type="checkbox"/>
書籍名稱	varchar(50)	<input checked="" type="checkbox"/>
簡介	varchar(50)	<input checked="" type="checkbox"/>
價格	smallmoney	<input checked="" type="checkbox"/>
出版公司	char(20)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

拉曳時會有一條粗黑線指示目前到達的地點



# 插入、搬移、刪除資料行

- 刪除資料行

- 先選取該欄位，然後按右鈕執行『刪除資料行』命令，或按  鍵

# 設定資料表屬性

此處可修改資料表的屬性

The screenshot shows the Microsoft SQL Server Management Studio interface. The '物件總管' (Object Explorer) on the left shows the database structure. The central pane displays the '資料行名稱' (Columns) table for the '書籍M' table. The '屬性' (Properties) window on the right is highlighted with a red box, showing the configuration for the table.

資料行名稱	資料類型	允許 Null
書籍編號	int	<input type="checkbox"/>
書籍名稱	varchar(50)	<input checked="" type="checkbox"/>
價格	smallmoney	<input checked="" type="checkbox"/>
出版公司	char(20)	<input checked="" type="checkbox"/>
簡介	varchar(50)	<input checked="" type="checkbox"/>

資料行屬性	
(一般)	
(名稱)	簡介
允許 Null	是
長度	50
資料類型	varchar
預設值或繫結	
(一般)	

[資料表] dbo.書籍M	
(識別)	
(名稱)	書籍M
伺服器名稱	john-pc
描述	
結構描述	dbo
資料庫名稱	練習07
資料表設計工具	
Text/Image 欄	PRIMARY
已複寫	否
可編索引	是
規則資料空間規	
資料列 GUID 資	
鎖定擴大	資料表
識別欄位	
(識別)	

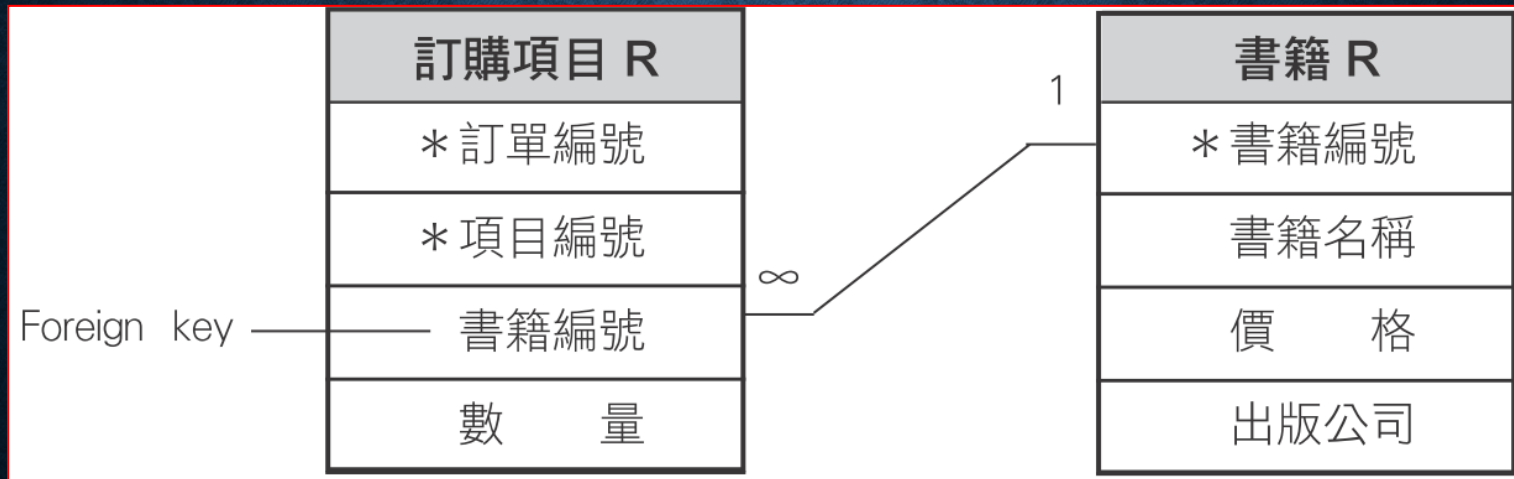
# 更改資料表名稱

1 在物件總管窗格中選取欲更改名的資料表

2 按右鈕執行『重新命名』命令

3 趁此時輸入新的名稱, 然後按 **Enter** 鍵就可以了

## 7-3 使用 SQL SERVER MANAGEMENT STUDIO 建立資料表間的關聯



# 建立關聯

- 先進入 **Foreign key** 所在的資料表的設計窗格

2 按關聯性鈕

The screenshot shows the SQL Server Enterprise Designer interface. On the left, the '物件總管' (Object Explorer) pane lists various database objects, with 'dbo.訂購項目R' highlighted in red. The main window displays the design view for 'JOHN-PC.練習07 - dbo.訂購項目R'. A table with columns '訂單編號', '項目編號', '書籍編號', and '數量' is shown. The '訂單編號' column is selected, and a relationship icon is visible in the '資料行名稱' column. Below the table, the '資料行屬性' (Column Properties) pane shows the properties for the selected column, including '(名稱)', '(名稱)', '允許 Null', '資料類型', and '預設值或繫結'. A red box highlights the '資料類型' property, which is set to 'int'. A red arrow points from the '資料類型' property to the '資料行名稱' column.

資料行名稱	資料類型	允許 Null
訂單編號	int	<input type="checkbox"/>
項目編號	int	<input type="checkbox"/>
書籍編號	int	<input type="checkbox"/>
數量	int	<input type="checkbox"/>

資料行屬性

(一般)

(名稱)	訂單編號
允許 Null	否
資料類型	int
預設值或繫結	

1 點選資料表內的任意資料行

# 建立關聯

4 用滑鼠左鈕點一下此欄位

正在編輯新關聯性的屬性。必須先填入'資料表及資料行規格'的屬性，才能接受新的關聯性。

選取的關聯性(S):  
FK\_訂購項目R\_訂購項目R\*

(一般)

- 資料表及資料行規格
- 檢查建立或重新啟用時的現有 是
- 資料表設計工具
- INSERT 及 UPDATE 規格
- 強制使用外部索引鍵條件約束 是
- 強制複寫 是
- 識別 (名稱) FK\_訂購項目R\_訂購項目R
- 描述

加入(A) 刪除(D) 關閉(C)

3 按加入鈕

5 按此鈕

8 SQL Server 會同步設定此關聯的名稱，如果您不滿意這個名稱，可自行更改

7 設定外部索引鍵資料表的 Foreign key

6 設定主索引鍵資料表與其 Primary key 欄位

資料表和資料行

關聯性名稱(N):  
FK\_訂購項目R\_書籍R

主索引鍵資料表(P):  
書籍R

外部索引鍵資料表:  
訂購項目R

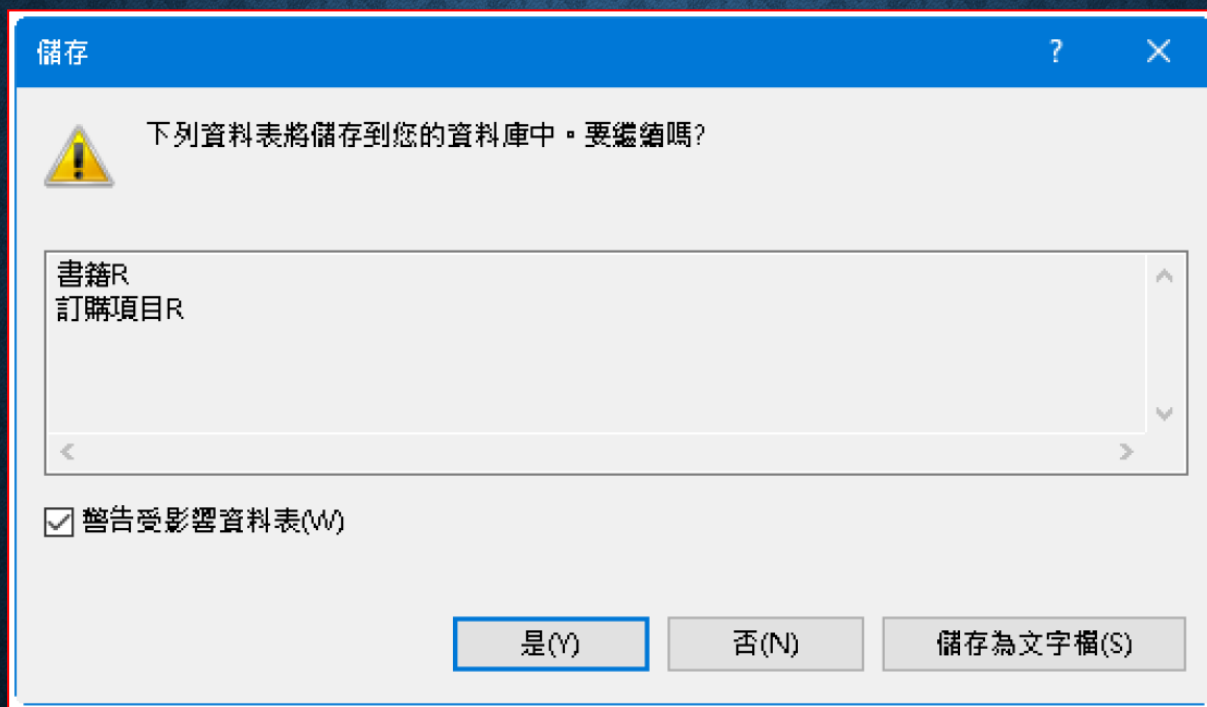
書籍編號  
<無>

確定 取消

10 按確定鈕完成設定

9 選擇無

# 建立關聯



# 關聯的屬性設定

- 檢查建立或重新啟用時的現有資料
- **INSERT** 及 **UPDATE** 規格
- 強制使用外部索引鍵條件約束
- 強制複寫
- (名稱)
- 描述



# 刪除關聯

外部索引鍵關聯性

選取的 關聯性(S):

FK\_訂購項目R\_書籍R

正在編輯現有關係性的屬性。

1 選取要刪除的關聯性名稱

2 按刪除鈕

3 按關閉鈕關閉此交談窗

加入(A) 刪除(D) 關閉(C)

▼ (一般)

- > 資料表及資料行規格  
檢查建立或重新啟用時的現有 是
- ▼ 資料表設計工具
- > INSERT 及 UPDATE 規格  
強制使用外部索引鍵條件約束 是  
強制複寫 是
- ▼ 識別

(名稱)	FK_訂購項目R_書籍R
描述	

## 7-4 設定條件約束維護資料完整性

- 條件約束的種類
- 在資料表設計視窗設定條件約束

# 條件約束的種類

- **Primary key**
- **Foreign key**
- **NULL、NOT NULL**
- **DEFAULT**
- **UNIQUE**
- **CHECK**

# 條件約束的種類

完整性種類	可用的對策
<b>實體完整性 (Entity Integrity)</b> 維持每筆記錄的唯一性	Primary key UNIQUE IDENTITY (識別屬性)
<b>區域完整性 (Domain Integrity)</b> 維持欄位資料的正確性	DEFAULT (預設值或繫結屬性) Foreign key CHECK NOT NULL
<b>參考完整性 (Referential Integrity)</b> 資料表間關聯的完整性	Foreign key CHECK
<b>使用者定義的完整性 (User -def ined)</b> 我們自訂的資料完整性	所有的條件約束 預存程序 (Stored procedures) 觸發程序 (Triggers)

## 在資料表設計視窗設定條件約束

- 設定 **CHECK** 條件約束：地址和電話至少要輸入一項資料。

資料行名稱	資料類型	允許 Null
▶ 客戶編號	int	<input type="checkbox"/>
客戶名稱	varchar(30)	<input type="checkbox"/>
聯絡人	char(10)	<input checked="" type="checkbox"/>
地址	varchar(50)	<input checked="" type="checkbox"/>
電話	char(12)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

# 在資料表設計視窗設定條件約束

2 按管理檢查條件約束鈕

1 點選資料表內的任意資料行

資料行名稱	資料類型	允許 Null
客戶編號	int	<input type="checkbox"/>
客戶名稱	varchar(30)	<input type="checkbox"/>
聯絡人	char(10)	<input checked="" type="checkbox"/>
地址	varchar(50)	<input checked="" type="checkbox"/>
電話	char(12)	<input checked="" type="checkbox"/>

4 輸入 CHECK 條件約束的運算式內容, 筆者輸入的 CHECK 條件約束運算式會同時對兩個欄位做檢查

正在編輯新檢查條件約束的屬性。必須先填入 '運算式' 的屬性, 才能接受新的檢查條件約束。

運算式 (地址 is not null or 電話 is not null)

資料表設計工具

於 INSERT 及 UPDATE 時強制複寫  是

檢查建立或重新啟用時的現有  是

識別 (名稱) CK\_客戶CU

描述

這 3 個選項的用途與關聯性設定交談窗上同名選項相同

5 SQL Server 會自動設定條件約束名稱, 若不滿意可自行更改

加入(A) 刪除(D) 關閉(C)

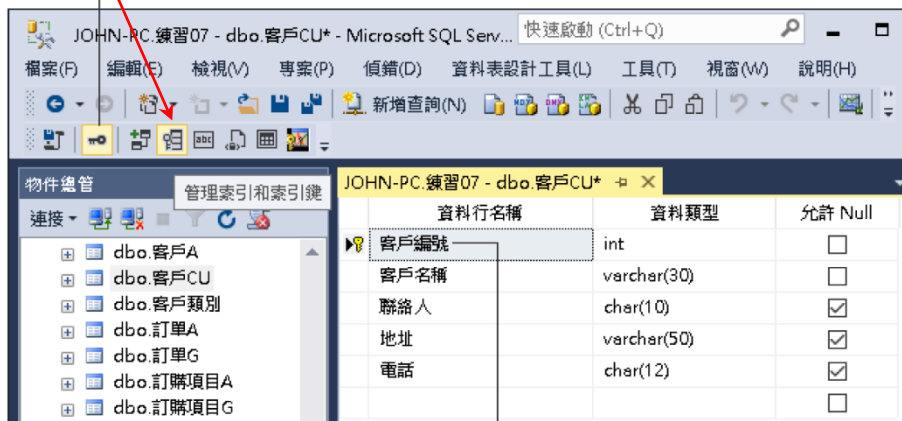
3 按下加入鈕

6 按下關閉鈕結束設定

# 設定UNIQUE條件約束

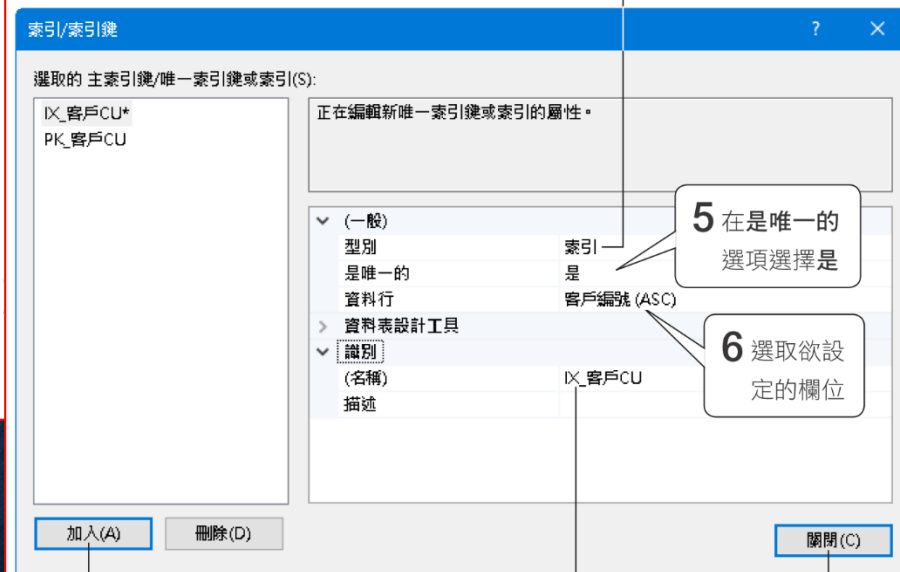
- 設定 **UNIQUE** 條件約束：防止輸入重複的聯絡人名稱

## 2 按管理索引和索引鍵鈕



1 點選資料表內的任意資料行

## 4 請確認此項目設定為索引型別



3 按下加入鈕

7 SQL Sserver 會自動設定名稱, 若不滿意可自行更改

8 按下關閉鈕結束設定

5 在是唯一的選項選擇是

6 選取欲設定的欄位

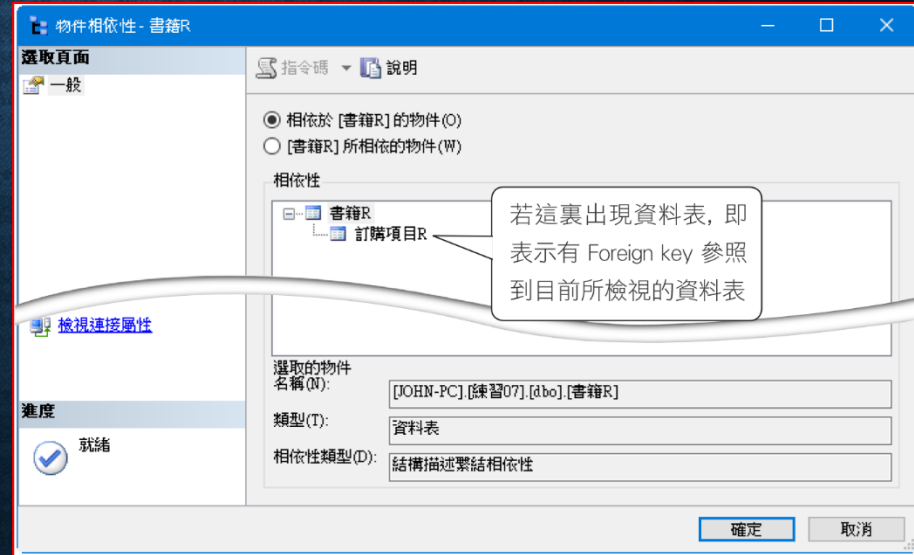
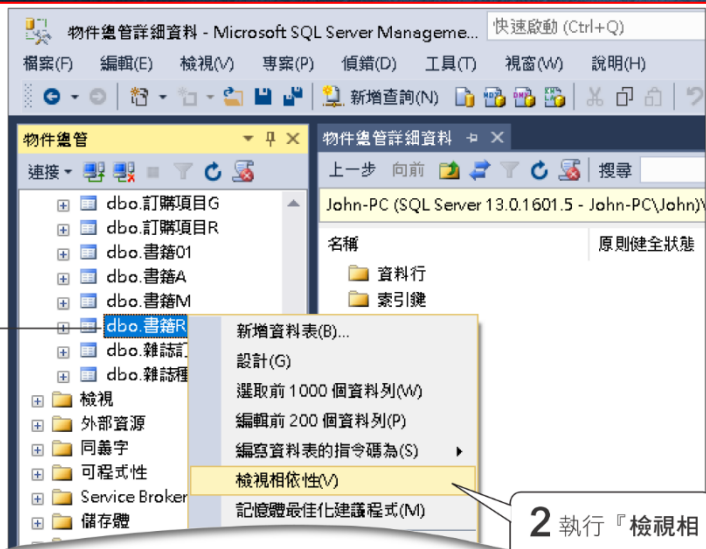
# 7-5 使用 SQL SERVER MANAGEMENT STUDIO 刪除資料表

- 檢視資料表的相依性
- 刪除資料表



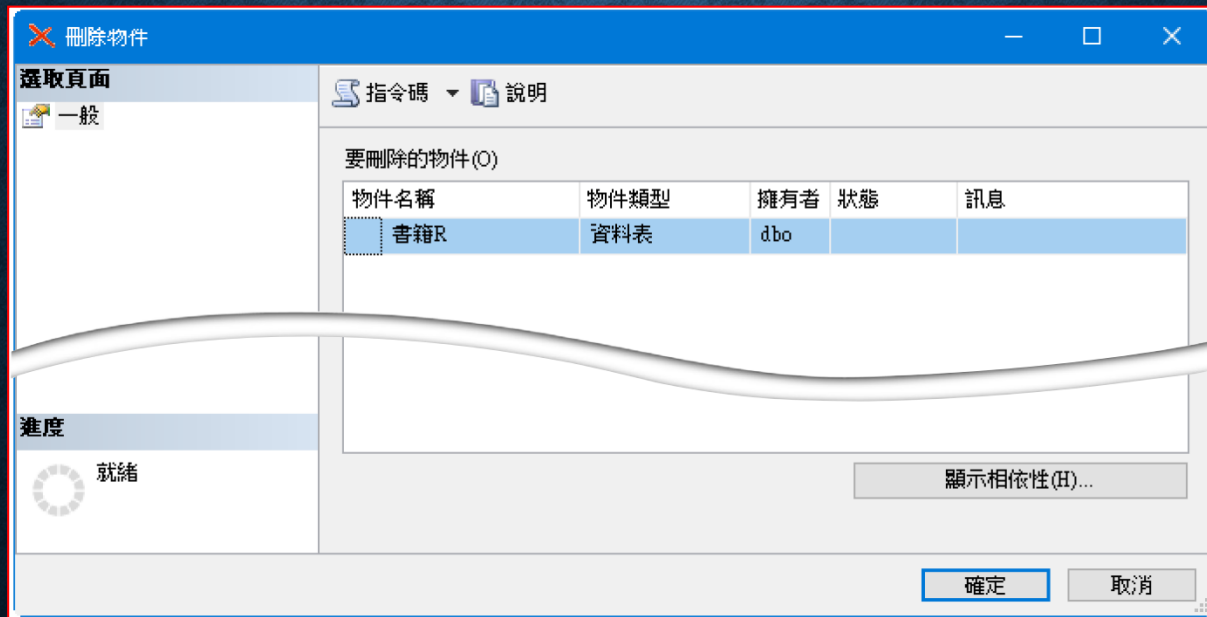
# 檢視資料表的相依性

1 在欲檢視的資料表上按右鈕



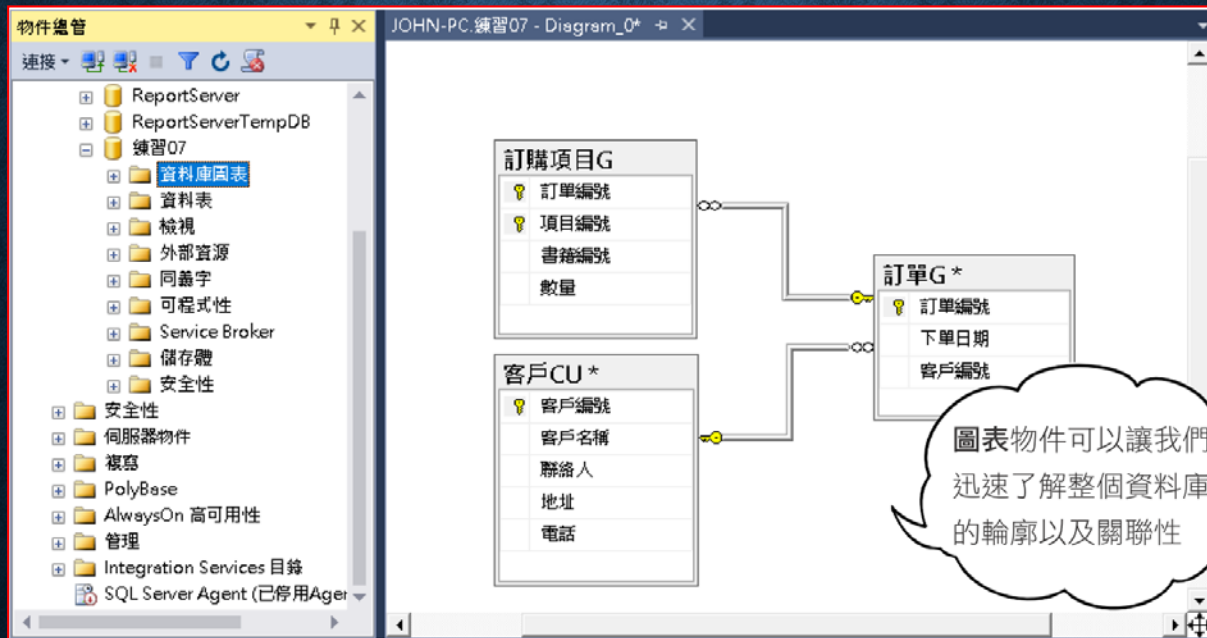
# 刪除資料表

- 在物件總管中選取要刪除的資料表，按右鍵執行『刪除』。

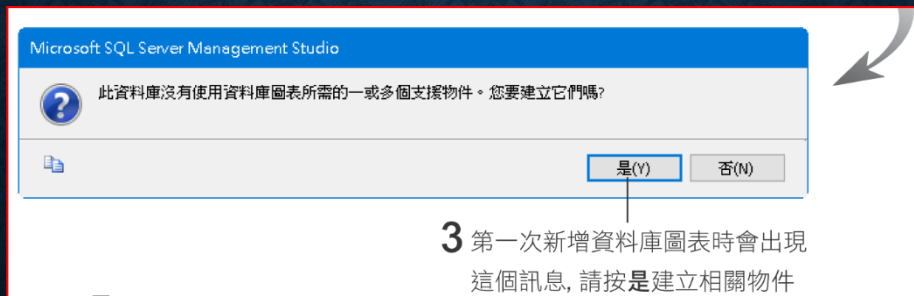
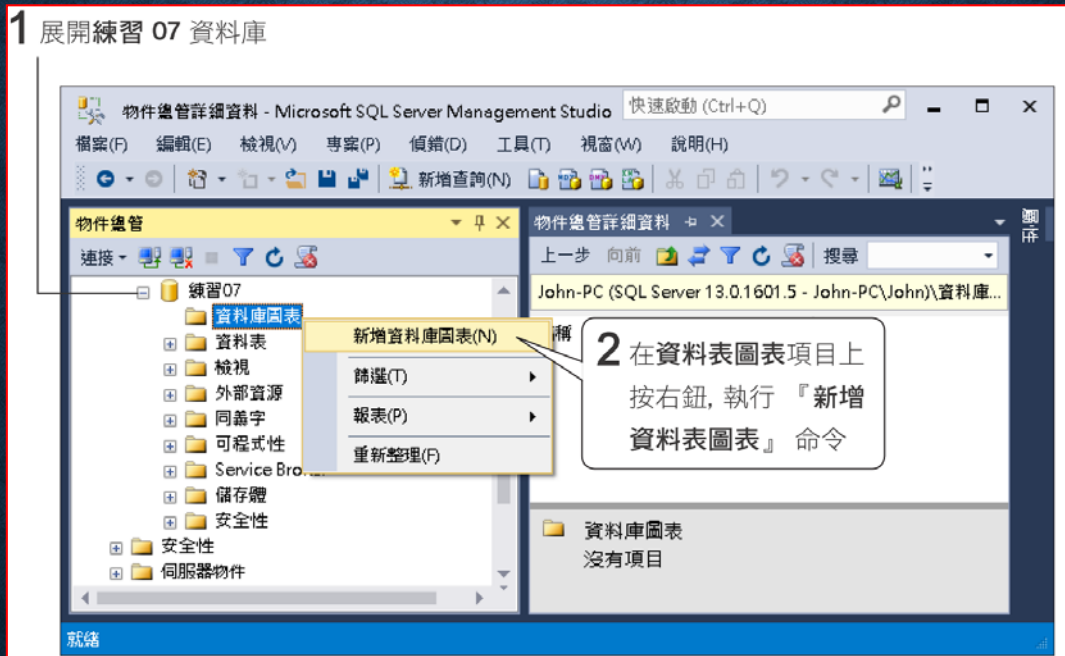


## 7-6 資料庫圖表與圖表物件

- 以圖形的方式顯示資料庫中資料表的關聯性。

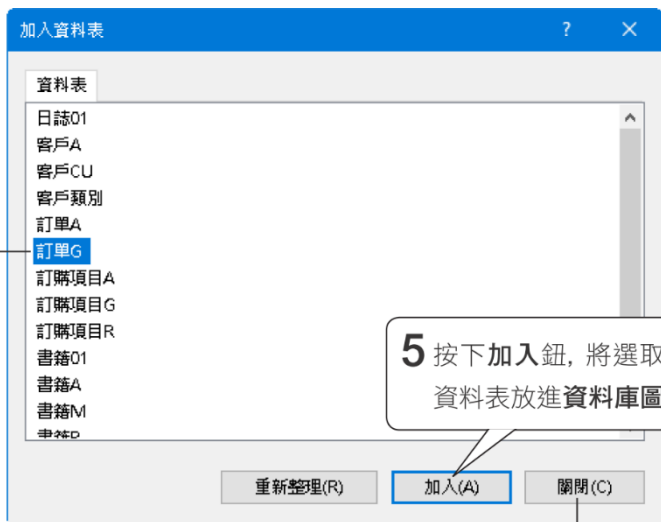


# 新增資料庫圖表及載入資料表



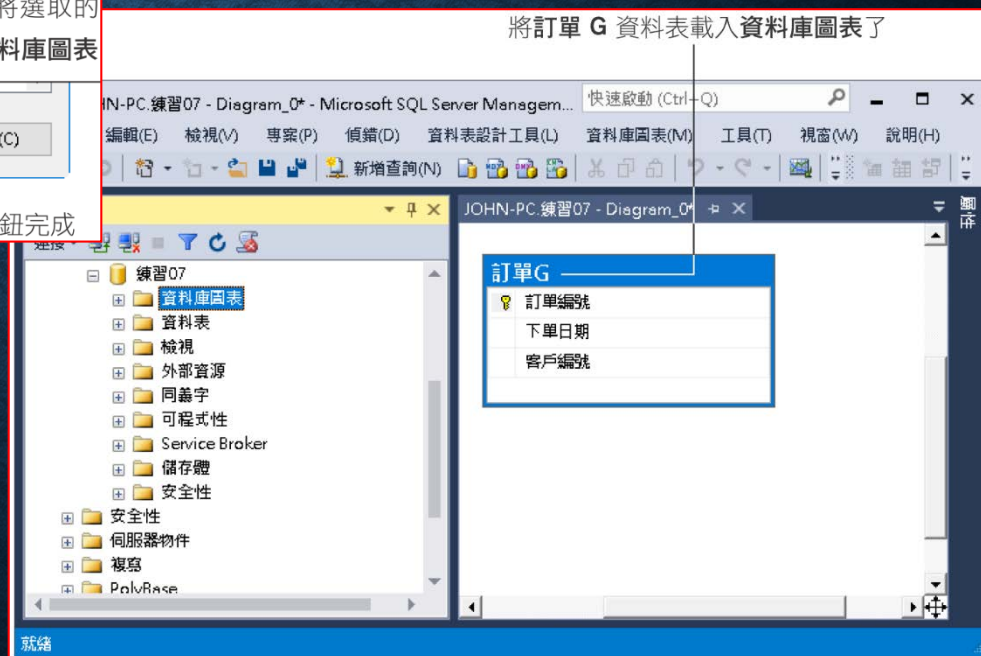
# 新增資料庫圖表及載入資料表

4 在此列示窗中選取欲載入的資料表 (可搭配 **Ctrl** 鍵選取多個資料表)



5 按下加入鈕，將選取的資料表放進資料庫圖表

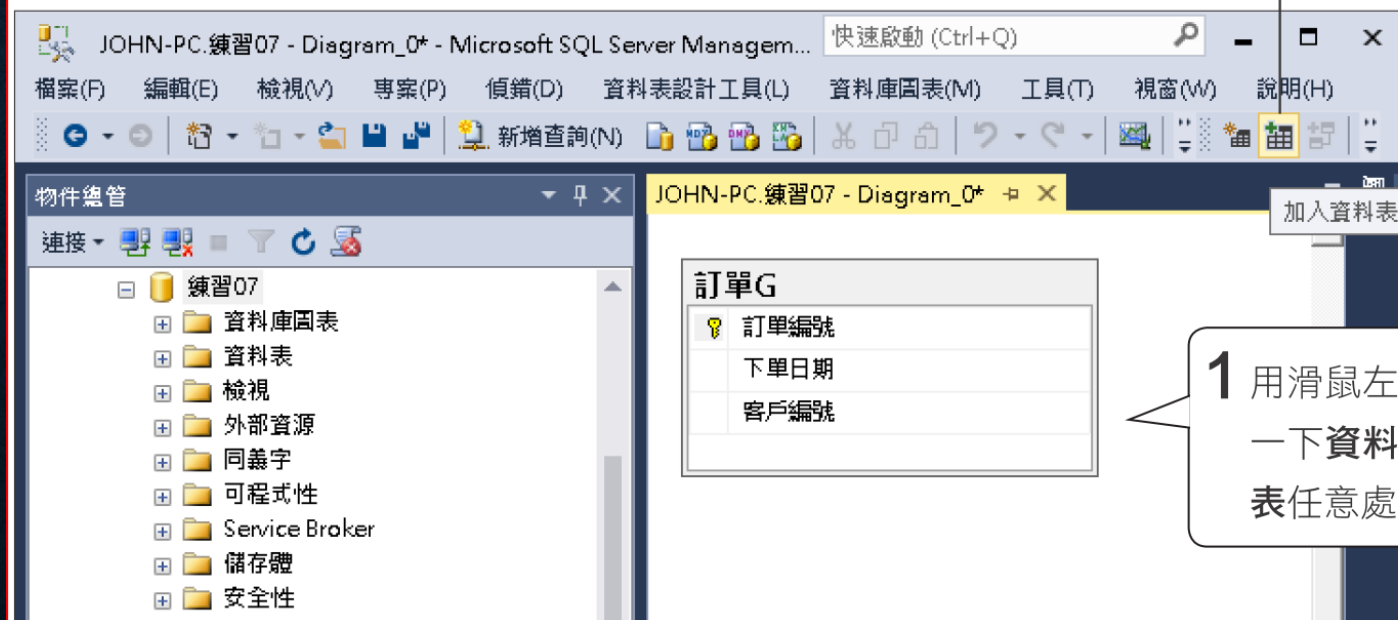
6 按關閉鈕完成



# 新增資料庫圖表及載入資料表

- 可以在資料庫圖表載入其他的資料表

2 按工具列上的加入資料表鈕

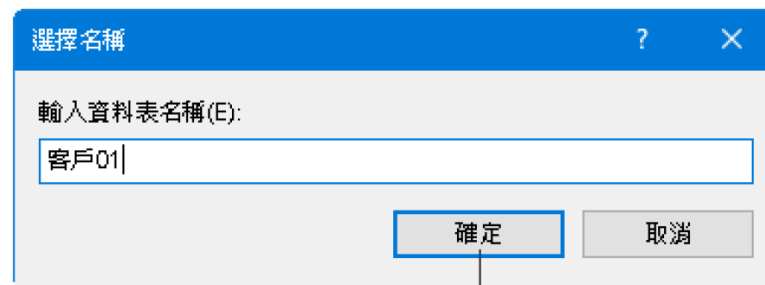
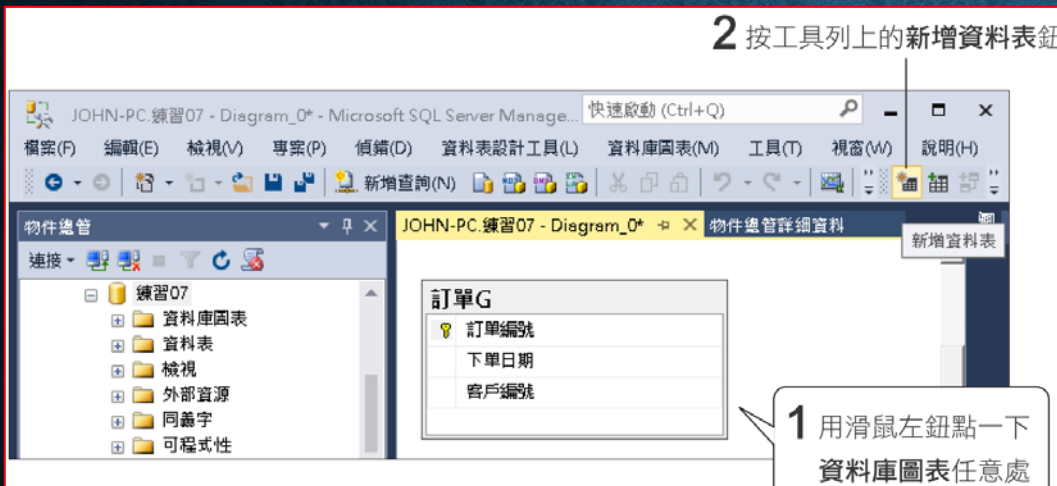


# 資料庫圖表移除資料表

- 將資料表移出資料庫圖表
  - 選取該資料表 (在資料表名稱列按一下)，然後按右鈕執行『從圖表移除』命令，將該資料表移出資料庫圖表

# 在資料庫圖表中建立新資料表

2 按工具列上的新增資料表鈕



3 輸入資料表的名稱後按確定鈕



# 在資料庫圖表中建立新資料表

- 6 按設定主索引鍵鈕, 將客戶編號欄位設為 Primary key 當資料庫名稱列出現 \*, 表示資料表有所變更

物件總管

JOHN-PC.練習07 - Diagram\_0\*

設定主索引鍵

資料行名稱	資料類型	允許 Null
客戶編號	int	<input checked="" type="checkbox"/>
客戶名稱	varchar(50)	<input checked="" type="checkbox"/>
聯絡人	char(10)	<input checked="" type="checkbox"/>
地址	varchar(50)	<input checked="" type="checkbox"/>
電話	nchar(10)	<input type="checkbox"/>

中斷點 輸出

就緒

5 選取客戶編號欄位

4 請依圖輸入資料表的各個欄位

# 在資料庫圖表中建立新資料表

- 資料表及欄位的屬性設定

1 按 **F4** 鍵開啟屬性窗格

資料行名稱	資料類型
客戶編號	int
客戶名稱	varchar(50)
聯絡人	char(10)
地址	varchar(50)
電話	nchar(10)

2 如果要修改資料表屬性，請用滑鼠選取要修改的資料表名稱列；若要修改欄位屬性，則請選取資料表內的欄位

3 在屬性窗格中修改資料表或欄位的屬性

# 資料庫圖表中更改資料表檢視

- 改變資料表檢視：

◎ 索引鍵檢視



僅顯示 Primary key 欄位

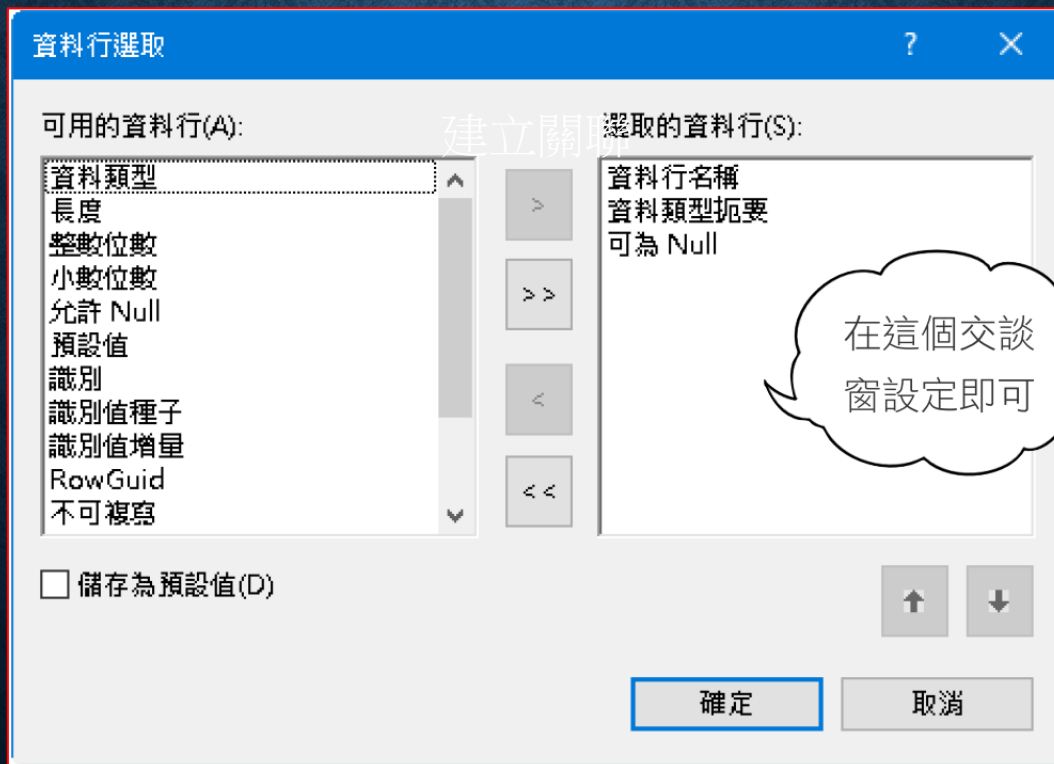
◎ 僅顯示名稱檢視



僅剩資料表名稱

# 資料庫圖表中更改資料表檢視

- 修改自訂檢視：滑鼠點選資料表名稱列，按右鍵，資料表檢視/修改自訂



# 建立關聯

1 在此欄位上  
按住滑鼠左  
鈕不放

2 然後拉曳到**訂  
單  
G** 的**訂  
單  
編  
號**欄  
位上（兩表間會  
出現一條虛線）

3 放開滑鼠左鈕

4 確認無誤就請  
按下**確定**鈕

神奇吧！相關欄位  
已經都設定好了

資料表和資料行

關聯性名稱(N):  
FK\_訂購項目G\_訂單G

主索引鍵資料表(P): 訂購G  
外部索引鍵資料表: 訂購項目G

訂購編號 訂購編號

確定 取消

# 建立關聯

外部索引鍵關聯性

選取的 關聯性(S):

FK\_訂購項目G\_訂單G\*

正在編輯新關聯性的屬性。必須先填入 '資料表及資料行規格' 的屬性, 才能接受新的 關聯性。

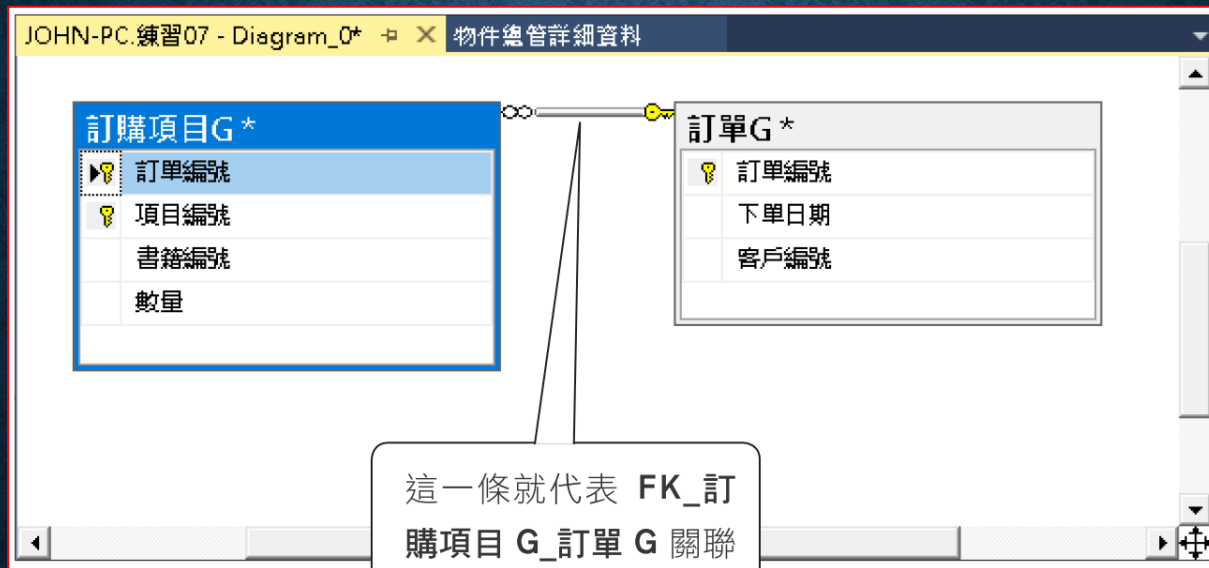
(一般)

- > 資料表及資料行規格
  - 檢查建立或重新啟用時的現有 是
- > 資料庫設計工具
- > INSERT 及 UPDATE 規格
  - 強制使用外部索引鍵條件約束 是
  - 強制複寫 是
- > 識別
  - (名稱) FK\_訂購項目G\_訂單G
  - 描述

確定 取消

6 按確定鈕完成 5 設定其他關聯性的選項

# 建立關聯



# 建立關聯

- 關聯的屬性設定，再按一次顯示關聯性標籤鈕，即可取消關聯性名稱顯示。

1 按顯示關聯性標籤鈕

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the '物件總管' (Object Explorer) pane displays a folder structure for '練習07', including '資料庫圖表' (Database Diagrams), '資料表' (Tables), '檢視' (Views), '外部資源' (External Resources), '同義字' (Synonyms), '程式性' (Programmability), 'Service Broker', '儲存體' (Storage), '安全性' (Security), and '伺服器物件' (Server Objects). The main window displays a diagram with two tables: '訂單G\*' (Orders) and '訂購項目G\*' (Order Items). The '訂單G\*' table has columns '訂單編號' (Order ID), '下單日期' (Order Date), and '客戶編號' (Customer ID). The '訂購項目G\*' table has columns '訂單編號' (Order ID), '項目編號' (Item ID), '書籍編號' (Book ID), and '數量' (Quantity). A relationship line connects the '訂單編號' column in '訂單G\*' to the '訂單編號' column in '訂購項目G\*'. The relationship is named 'FK\_訂購項目G\_訂單G'. A red box highlights the relationship name label on the connecting line.

2 關連線條上顯示關聯性的名稱



# 刪除關聯

- 刪除關聯
  - 在欲刪除的關聯線條上按右鈕，執行『從資料庫中刪除關聯性』命令

# 設定 CHECK 與 UNIQUE 條件約束

- 在資料庫圖表中為資料表設定 **CHECK** 條件約束，請先選取資料表，然後按下**管理檢查條件約束**鈕，就可開啟檢查條件約束交談窗來進行設定，或者開啟**索引/索引鍵**交談窗來進行設定。

# 刪除資料表

- 在資料庫圖表刪除資料表，請在欲刪除的資料表上按右鈕，執行『從資料庫刪除資料表』命令

# 儲存圖表物件

選擇名稱

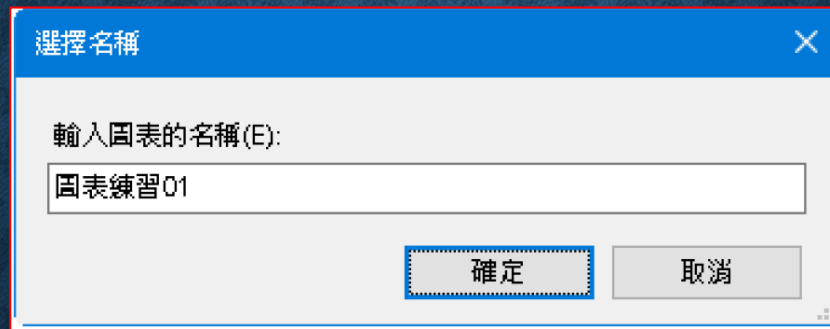
輸入圖表的名稱(E):

圖表練習01

確定 取消

# 儲存圖表物件

- 儲存圖表物件，按下儲存鈕。



- 建好的圖表物件會放在資料庫的資料庫圖表項目下，若無法及時顯示，按一下 **F5**，重新整理即可。

# 7-7 用 **CREATE TABLE** 敘述 建立資料表

- **CREATE TABLE** 敘述的語法
- 設定資料表名稱
- 定義欄位屬性
- 設定欄位的條件約束
- 設定資料表條件約束
- 指定檔案群組
- 建立計算欄位

# CREATE TABLE 敘述的語法

```
CREATE TABLE
```

```
[databasae_name.[ schema ].| schema. ] table_name ← 設定資料表名稱  
( { <column_definition> ← 定義欄位屬性與條件約束  
  | column_name AS computed_column_expression ← 定義計算欄位  
  | <table_constraint> } ← 設定資料表條件約束  
[ , ...n ] )
```

```
[ON { filegroup | "default" } ]
```

```
[TEXTIMAGE_ON {filegroup | "default"} ]
```

```
[FILESTREAM_ON { filegroup | "default" } ]
```

} 指定存放資料表資料  
的檔案群組

# 設定資料表名稱

- **database\_name**：資料表所在的資料庫名稱
- **schema**：資料表所屬的結構描述
- **table\_name**：新建的資料表名稱

CREATE TABLE 練習 07.dbo.客戶 ←—— 最完整的表達方式

CREATE TABLE 練習 07..客戶 ← 省略 schema, 則 schema 預設為建立者在目前資料庫中的  
預設結構描述

CREATE TABLE dbo.客戶 ← 省略 database\_name 和 schema, 則預設為目前的資料庫

CREATE TABLE 客戶 ← 省略 database\_name 和 schema, 則 database\_name 預設為目前的資料  
庫, schema 預設為建立者在目前資料庫中的預設結構描述



# 定義欄位屬性

```
< column_definition > ::=  
column_name <data_type>  
  [ NULL | NOT NULL ]  
  [ COLLATE collation_name ]  
  [  
    [ CONSTRAINT constraint_name ] DEFAULT constant_expression  
    | IDENTITY [ ( seed , increment ) ] [ NOT FOR REPLICATION ]  
  ]  
  [ ROWGUIDCOL ]  
  [ FILESTREAM ]  
  [ < column_constraint > [ ...n ] ]
```

# 定義欄位屬性

- **column\_name data\_type** : 定義欄位名稱以及資料型別
- **NULL**、**NOT NULL** 條件約束 : 定義好欄位的屬性後，直接把 **NULL** 或者是 **NOT NULL** 加在後面

( 書籍名稱 varchar(40) NOT NULL ,  
價格 smallmoney NULL )

若未指定 NULL 或 NOT  
NULL, 則預設為 NULL

# 定義欄位屬性

- **COLLATE collation\_name** : 設定欄位要套用的定序名稱，用以指定資料的排序方式
- **DEFAULT constant\_expression** : 設定欄位的預設值或繫結屬性

```
( 商品編號 int,  
 折扣 numeric(3, 2) CONSTRAINT 預設折扣 DEFAULT 0.8 )
```

# 定義欄位屬性

- **IDENTITY (seed, increment)**：設定識別屬性，讓欄位值自動編號
- **ROWGUIDCOL**：設定此欄位為資料表的 **GUID** (全域唯一識別碼) 欄位

```
CREATE TABLE 圖庫
( 區域編號 int IDENTITY (1001, 1),           ← 由 1001 開始自動編號
  全域編號 uniqueidentifier DEFAULT NEWID() ROWGUIDCOL, ← GUID 欄位
  檔名 nvarchar(20) NOT NULL,                ← 不可為 NULL
  建檔日 date DEFAULT CONVERT(date, GETDATE()) ← 預設為今天
)
```

設定 **ROWGUIDCOL** 的欄位必須是 **uniqueidentifier** 型別，系統不會自動輸入其值，因此我們通常會設定預設值為 **NEWID()**，產生全域的唯一值。

**CONVERT(型別, 資料)** 函數，可將資料轉換為指定的型別。  
例子中將 **GETDATE()** 函數傳回的值，轉換為 **date** 的資料型別。

# 定義欄位屬性

```
INSERT 圖庫 (檔名) VALUES ( 'Fl ag01.jpg') ← 新增一筆記錄  
SELECT * FROM 圖庫 ← 查詢資料表內容
```



	區域編號	全域編號	檔名	建檔日
1	1001	72F93684-E04E-42D4-9E54-B50F355E3A15	Flag01.jpg	2016-11-15

```
SELECT $ROWGUID  
FROM 圖庫  
WHERE $IDENTITY = 1001
```

使用這個方法, 可以不必知道相關的欄位名稱



	全域編號
1	72F93684-E04E-42D4-9E54-B50F355E3A15

# 定義欄位屬性

- **FILESTREAM**：指定此欄位的值要以『檔案』形式個別儲存

```
CREATE DATABASE 練習 fs      -- 請先建立 C:\data 資料夾
ON
PRIMARY
    (NAME = 練習 fs, FILENAME = 'C:\data\練習 fs.mdf'),
FILEGROUP fs_group1
    CONTAINS FILESTREAM
    (NAME = fs1, FILENAME = 'C:\data\fs1')
LOG ON
    (NAME = 練習 fs_log, FILENAME = 'C:\data\練習 fs.ldf')

GO
```

# 定義欄位屬性：FILESTREAM

```
CREATE TABLE 練習 fs.dbo.文件  
( 編號 uniqueidentifier DEFAULT NEWID()  
  ROWGUIDCOL NOT NULL UNIQUE,  
  檔名 nvarchar(20) NOT NULL,  
  內容 varbinary(max) FILESTREAM NULL  
)
```

```
INSERT 練習 fs.dbo.文件 (檔名, 內容) ← 新增一筆資料  
VALUES ( 'Doc01.txt' , CONVERT(varbinary(max), '測試儲存' ) )
```

```
SELECT CONVERT(varchar(max), 內容) ← 查詢 FILESTREAM 欄的內容  
FROM 練習 fs.dbo.文件
```



	(沒有資料行名稱)
1	測試儲存

新增一筆資料，檔名為 **Doc01.txt**，內容"測試儲存"，所以使用 **CONVERT()** 函數將茲串資料轉換為 **varbinary(max)**。

# 定義欄位屬性

所有 FILESTREAM 資料都會以檔案形式儲存在此資料夾中

此檔案的內容為：測試儲存



# 定義欄位屬性

```
SELECT 檔名, 內容.PathName() ← 使用『欄名.PathName()』來查詢路徑  
FROM 練習 fs.dbo.文件
```



	檔名	(沒有資料行名稱)
1	Doc01.txt	WJOHN-PC\MSSQLSERVER\w02-A60EC2F8-2B24-11DF-9CC3...

這就是 Doc01.txt 的虛擬路徑, 在前端應用程式中可用此路徑來存取檔案內容。限於篇幅, 若需進一步的說明請參閱 SQL Server 線上叢書

# 設定欄位的條件約束

```
< column_constraint > ::= [ CONSTRAINT constraint_name ] ← 設定條件約束的名稱
[ { PRIMARY KEY | UNIQUE }
  [ CLUSTERED | NONCLUSTERED ]
  [ WITH FILLFACTOR = fillfactor ]
  [ ON { filegroup | "default" } ] ]
| [ [ FOREIGN KEY ]
    REFERENCES ref_table [ ( ref_column ) ]
    [ ON DELETE { NO ACTION | CASCADE ... } ]
    [ ON UPDATE { NO ACTION | CASCADE ... } ]
    [ NOT FOR REPLICATION ] ]
| CHECK [ NOT FOR REPLICATION ]
  ( logical_expression ) }
```

設定 PRIMARY KEY 或  
UNIQUE 條件約束

設定 FOREIGN KEY  
條件約束

設定 CHECK 條件約束

# 設定欄位的條件約束

- 設定條件約束的名稱，若省略**PK\_**訂單編號，系統會自動命名

```
( 訂單編號 int CONSTRAINT PK_訂單編號 PRIMARY KEY )
```

- **PRIMARY KEY**、**UNIQUE** 條件約束

```
( 客戶編號 int NOT NULL PRIMARY KEY,  
  聯絡人 char(10) NULL UNIQUE )
```

- **FOREIGN KEY** 條件約束

```
CREATE TABLE 訂單 01
```

```
( 訂單編號 int PRIMARY KEY,
```

```
  下單日期 date ,
```

```
  客戶編號 int CONSTRAINT FK_ 訂單與客戶 01 ← 設定 FOREIGN KEY 條件約束的名稱
```

```
    FOREIGN KEY ← 這個參數可以省略
```

```
    REFERENCES 客戶 01 (客戶編號)) ← 設定參照到的資料表及欄位名稱
```

# 設定欄位的條件約束：**FOREIGN KEY**

- **FOREIGN KEY** 條件約束還有 3 項屬性設定
  - **ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }**：設定當此欄位所參考到的記錄被刪除時
  - **ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }**：設定當此欄位所參考到的欄位值被修改時
  - **NOT FOR REPLICATION**：若加上這項參數，則在複寫程式將資料輸入資料表時，不使用 **CHECK** 條件約束。

# 設定欄位的條件約束：**CHECK**

- **CHECK** 條件約束

```
( 價格 smallmoney CONSTRAINT CK_價格範圍  
CHECK (價格 > 1 AND 價格 < 3000) )
```

logical\_expression 的部份  
要用小括號 ( ) 括起來

# 設定資料表條件約束：多個欄位

```
< table_constraint > ::= [ CONSTRAINT constraint_name ] ← 設定條件約束的名稱
{ [ { PRIMARY KEY | UNIQUE }
  [ CLUSTERED | NONCLUSTERED ]
  { ( column [ASC | DESC] [ , ...n ] ) }
  [ WITH FILLFACTOR = fillfactor ]
  [ ON { filegroup | "default " } ] ]
| FOREIGN KEY
  [ ( column [ , ...n ] ) ]
  REFERENCES ref_table [ ( ref_column [ , ...n ] ) ]
  [ ON DELETE { NO ACTION | CASCADE... } ]
  [ ON UPDATE { NO ACTION | CASCADE... } ]
  [ NOT FOR REPLICATION ]
| CHECK [ NOT FOR REPLICATION ]
  ( logical_expression ) }
```

設定 PRIMARY KEY  
或 UNIQUE 條件約束

設定 FOREIGN  
KEY 條件約束

設定 CHECK 條件約束

# 設定資料表條件約束：多個欄位

```
CREATE TABLE 客戶 02
( 客戶編號 int IDENTITY PRIMARY KEY,
  身份證字號 char(10) NOT NULL UNIQUE,
  年齡 int CHECK (年齡 > 0) DEFAULT 25,
  地址 varchar(50),
  電話 varchar(12),
  雜誌編號 int
      REFERENCES 雜誌種類 (雜誌編號),
  訂戶編號 int NOT NULL,

  FOREIGN KEY (雜誌編號, 訂戶編號)
      REFERENCES 雜誌訂戶 (雜誌編號, 訂戶編號),
  CHECK (地址 is not null or 電話 is not null)
)
```

這裏為資料表條件約束

# 指定檔案群組

- **ON { filegroup | "default" }**：指定資料表的資料
- **TEXTIMAGE\_ON { filegroup | "default" }**：指定 **text**、**ntext**、**image**、**xml**、**varchar(max)**、**nvarchar(max)** 及 **varbinary(max)** 型別的欄位資料要存放的檔案群組
- **FILESTREAM\_ON { filegroup | "default" }**：指定 **varbinary(max) FILESTREAM** 欄位資料要存放的 **FILESTREAM** 檔案群組



# 建立計算欄位

```
CREATE TABLE 估價  
( 編號 int IDENTITY,  
  單價 numeric ( 5, 1 ),  
  數量 int ,  
  總價 AS 單價 * 數量 ← 總價欄位即為計算欄位  
)
```

```
INSERT 估價 VALUES (21.5, 8)  
INSERT 估價 VALUES (12, 3)  
SELECT * FROM 估價
```

← 輸入兩筆資料, 但計算欄位不必給值  
← 顯示估價資料表的記錄內容



	編號	單價	數量	總價
1	1	21.5	8	172.0
2	2	12.0	3	36.0

← 計算欄位的值會自動算出

計算欄位是唯讀，不可以輸入或修改。

# 7-8 用 ALTER TABLE 敘述 修改資料表

ALTER TABLE table	← 指定欲進行修改的資料表名稱
[ ALTER COLUMN column_name { new_data_type [ ( precision [ , scale ] ) ] [ COLLATE collation_name ] [ NULL   NOT NULL ]   { ADD   DROP } ROWGUIDCOL } ]	修改欄位屬性
ADD { [ <column _ definition> ]   column_name AS computed_column_expression } [ , ...n ]	新增欄位或新增計算欄位
[ WITH CHECK   WITH NOCHECK ] ADD { < table_constraint > } [ , ...n ]	新增資料表條件約束
DROP { [ CONSTRAINT ] constraint_name   COLUMN column } [ , ...n ]	刪除條件約束或刪除欄位
{ CHECK   NOCHECK } CONSTRAINT { ALL   constraint_name [ , ...n ] }	啟動或關閉條件約束
{ ENABLE   DISABLE } TRIGGER { ALL   trigger_name [ , ...n ] }	啟動或關閉觸發程序 (此處 不介紹, 請參閱第 18 章)
SET (FILESTREAM_ON = { filegroup   "default "   "NULL" } )	更改 FILESTREAM 欄位 所存放的檔案群組

# 用 **ALTER TABLE** 敘述 修改資料表

- 修改欄位屬性
- 新增欄位/計算欄位
- 新增資料表條件約束
- 刪除條件約束/欄位
- 啟動/關閉條件約束
- 更改欄位與資料表的名稱

# 修改欄位屬性

```
ALTER TABLE table
  [ ALTER COLUMN column_name
    { new_data_type [ (precision [ , scale ] ) ]
      [ COLLATE collation_name ]
      [ NULL | NOT NULL ]
      | {ADD | DROP} ROWGUIDCOL } ]
```

# 修改欄位屬性

- **column\_name**
  - 指定欲修改屬性的欄位名稱。
- **new\_data\_type ( precision [, scale] )**
  - 為欄位指定新的資料型別。
- **{ADD | DROP} ROWGUIDCOL**
  - 為欄位加上 (**ADD**) 或移除 (**DROP**) **RowGuidCol**屬性。

```
ALTER TABLE 客戶 A
    ALTER COLUMN 聯絡人
        varchar (30) NULL
```

# 不能變更屬性的欄位

- **timestamp** 型別的欄位
- **text**、**ntext**、**image** 型別的欄位只能進行以下的更改。

原來的型別	可更改為
text	varchar(max)、nvarchar(max)、或 xml
ntext	varchar(max)、nvarchar(max)、或 xml
image	varbinary(max)

- 已具備 **RowGuidCol** 屬性的欄位
- 計算欄位，或者計算欄位用到的欄位
- 用於 **PRIMARY KEY** 或 **FOREIGN KEY** 條件約束中的欄位
- 用於 **CHECK** 或 **UNIQUE** 條件約束中的欄位，但是長度可變
- 用於 **DEFAULT** 中的欄位，但是長度可變

# 變更型別的注意事項

- 新資料型別必須與原資料型別相容，亦即能夠進行隱含式轉換，可參閱 10-5 節。
- 不能變更為 **timestamp** 型別。
- 若變更屬性的欄位原就具備識別規格中的屬性，新資料型別也必須要能夠支援該屬性。

# 新增欄位 / 計算欄位

```
ALTER TABLE table
  ADD
    { [ <column_definition> ]                                     ← 新增一般欄位
      | column_name AS computed_column_expression }             ← 新增計算欄位
    [ , ...n ]

< column_definition > : := { column_name data_type }
  [ [ DEFAULT constant_expression ] [ WITH VALUES ]
    | [ IDENTITY [ (seed, increment) [ NOT FOR REPLICATION ] ] ] ]
  [ ROWGUIDCOL ]
  [ COLLATE collation_name ]
  [ FILESTREAM ]
  [ < column_constraint > ] [ ...n ]
```



# 新增欄位 / 計算欄位

	NOT NULL	NULL
有 WITH VALUE	填入預設值	填入預設值
無 WITH VALUE	填入預設值	填入 NULL

```
ALTER TABLE 客戶 A
ADD
    類別編號 int
    DEFAULT 1 WITH VALUES
    CONSTRAINT FK_類別編號
    FOREIGN KEY
    REFERENCES 客戶類別(類別編號)
```

**WITH VALUES** 用意在資料表就有紀錄的新增欄位中填入預設值

# 新增資料表條件約束

```
ALTER TABLE table
  [ WITH CHECK | WITH NOCHECK ] ADD
  { < table_constraint > } [ , ...n ]
```

```
< table_constraint > ::= [ CONSTRAINT constraint_name ]
  { [ { PRIMARY KEY | UNIQUE }
    { ( column [ , ...n ] ) }
  | FOREIGN KEY
    [ ( column [ , ...n ] ) ]
    REFERENCES ref_table [ ( ref_column [ , ...n ] ) ]
    [ ON DELETE { NO ACTION | CASCADE ... } ]
    [ ON UPDATE { NO ACTION | CASCADE ... } ]
    [ NOT FOR REPLICATION ]
  | DEFERRED CHECK
    FOR column [ WITH VALUES ]
  | CHECK [ NOT FOR REPLICATION ]
    ( logical_expression ) }
```

← 必須是在同一敘述中用 ADD 新增的欄位, 才能指定 WITH VALUE 選項

# 新增資料表條件約束

```
ALTER TABLE 訂購項目 A  
  WITH CHECK ADD  
  CONSTRAINT PK_訂購項目 A  
  PRIMARY KEY (訂單編號, 項目編號)
```

**WITH CHECK** 表示要使用新增的資料表條件約束去檢查舊有紀錄。

# 刪除條件約束 / 欄位

```
ALTER TABLE table
```

```
  DROP
```

```
    { [ CONSTRAINT ] constraint_name ← 刪除條件約束, 指定欲刪除的條件約束  
                                           名稱即可
```

```
    | COLUMN column } [ , ...n ] ← 刪除欄位, 指定欲刪除的欄位名稱即可
```

```
ALTER TABLE 訂購項目 A
```

```
  DROP CONSTRAINT PK_訂購項目 A
```

```
ALTER TABLE 訂購項目 A
```

```
  DROP COLUMN 訂單編號, 項目編號
```

# 啟動/關閉條件約束

- 設定功能只能用在 **FOREIGN KEY**

```
ALTER TABLE table
    { CHECK | NOCHECK } CONSTRAINT
    { ALL | constraint_name [ , ...n ] }
```

```
ALTER TABLE 客戶 A
    NOCHECK CONSTRAINT FK_類別編號
```

```
ALTER TABLE 客戶 A
    CHECK CONSTRAINT FK_類別編號
```

# 使用 **sp\_rename** 更改欄位與資料表的名稱

```
sp_rename 'object_name' , 'new_name' [ , 'object_type' ]
```

↑  
原物件名稱

↑  
新物件名稱

↑  
物件型態

object_type	說明
COLUMN	變更欄位名稱
DATABASE	變更資料庫名稱
INDEX	變更使用者自訂的索引名稱
OBJECT	變更條件約束 (CHECK、FOREIGN KEY、PRIMARY KEY、UNIQUE)、資料表、檢視表、預存程序、觸發程序的名稱
USERDATATYPE	變更使用者自訂資料型別物件的名稱

EXEC sp\_rename ' 訂單 A' , ' 訂購單 A' ← 將**訂單 A** 資料表改為**訂購單 A** 資料表

EXEC sp\_rename ' 客戶 A.地址' , ' 通訊處' , 'COLUMN' ← 將**客戶 A** 資料表的**地址**欄位更名為**通訊處**欄位

## 7-9 用 **DROP TABLE** 敘述刪除資料表

```
DROP TABLE table_name
```

```
DROP TABLE 書籍 A
```

刪除資料表要先刪除其關聯，否則無法刪除

# 刪除資料表

資料庫中已經有一個名為 '...' 的物件。

```
DROP TABLE IF Exists table_name;
```

```
DROP TABLE IF Exists 暫存 A
```



## 7-10 暫存資料表

- 暫存資料表的種類：
  - 區域暫存資料表 (以 # 開頭)，只有建立它的人可使用，當該使用者離線後，**SQL Server** 會自動刪除。
  - 全域暫存資料表 (以 ## 開頭)，所有人都可以使用，當建立者離線後，其他使用者即無法開啟此資料表，但是使用中的使用者仍可繼續使用，直到所有使用他的指令都結束後，**SQL Server** 會自動刪除。
- 暫存資料表會存放在 **tempdb** 資料庫中。

```
CRATE TABLE # 訂單 ( 編號 int, 數量 int )      ←—— 建立區域暫存資料表
CREATE TABLE ## 客戶 ( 編號 int, 姓名 char(10) ) ←—— 建立全域暫存資料表
```

# 7-11 自動紀錄資料異動— **TEMPORAL** 資料表

- 建立 **Temporal** 資料表
- 查詢異動紀錄
- 刪除或修改 **Temporal** 資料表
- 將現存資料表改成 **Temporal** 資料表

**Temporal** 是系統控制版本資料表，可以自動記錄資料表的異動歷程，作為回復資料的依據。

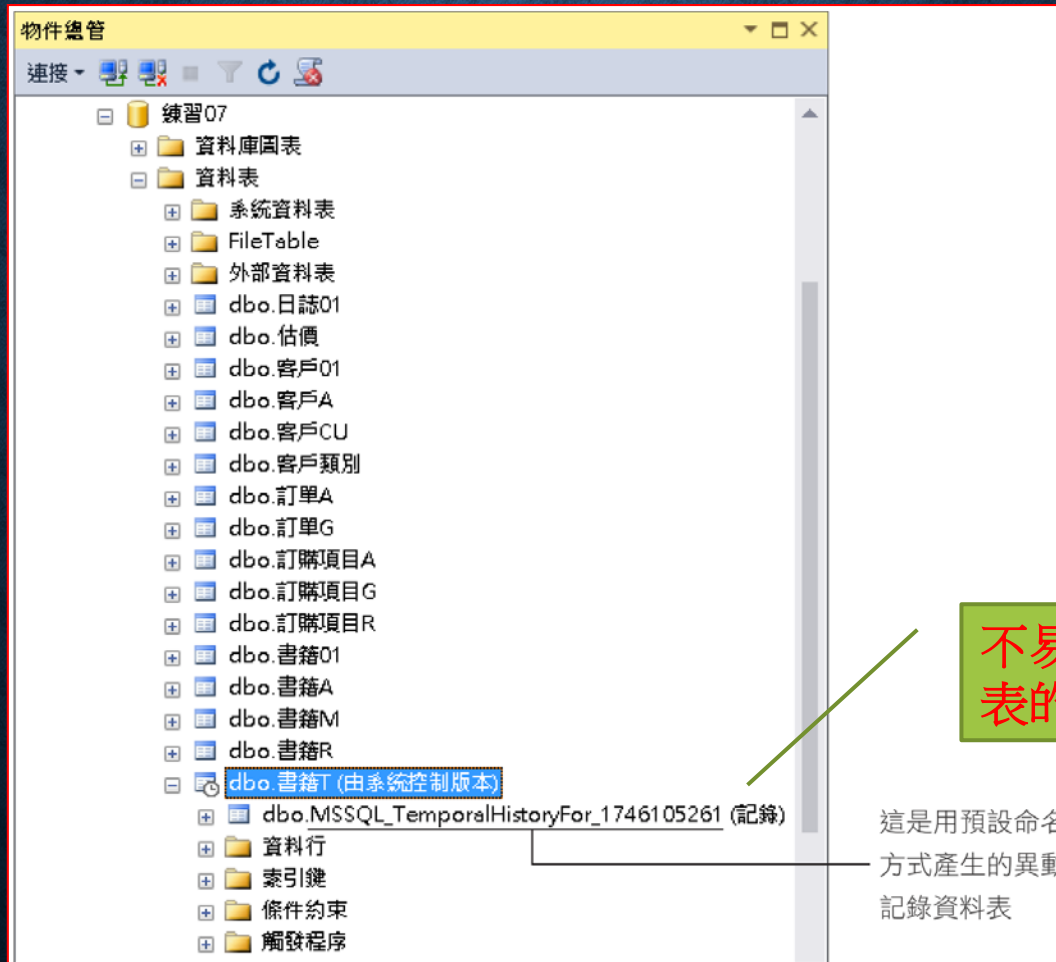
# 建立 **TEMPORAL** 資料表

- 資料表必須有主索引鍵 (**Primary Key**)
- 必須有 2 個資料類型為 **DATETIME2** 的欄位，並且設定參數為 **GENERATED ALWAYS AS ROW**
- 資料表必須加上 **SYSTEM\_VERSIONING** 的參數，並且設置為 **ON**，啟用『系統控制版本』功能

```
CREATE TABLE table_name ← 設定資料表名稱
(
  id int NOT NULL PRIMARY KEY, ← 定義主索引鍵
  ...
  time1_name DATETIME2 GENERATED ALWAYS AS ROW,
  time2_name DATETIME2 GENERATED ALWAYS AS ROW,
  PERIOD FOR SYSTEM_TIME (StartTime, EndTime)
)
WITH (SYSTEM_VERSIONING = ON); ← 資料表最後加上『SYSTEM_VERSIONING = ON』參數啟用系統控制版本
```

定義 2 個 DATETIME2 類型的欄位，並啟用『GENERATED ALWAYS AS ROW』參數，欄位名稱自訂

# 建立 TEMPORAL 資料表



# 建立 TEMPORAL 資料表

...

```
WITH(SYSTEM_VERSIONING = ON (HISTORY_TABLE = [資料表名稱] ) )
```

```
CREATE TABLE 書籍 T
```

將此欄位設定為主索引鍵, Temporal  
資料表一定要有主索引鍵

```
(
```

```
  書籍編號 INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
```

```
  書籍名稱 VARCHAR(50) NULL,
```

```
  價格 SMALLMONEY NULL,
```

```
  出版公司 CHAR(20) NULL,
```

```
  起始時間 DATETIME2 GENERATED ALWAYS AS ROW START HIDDEN,
```

```
  迄止時間 DATETIME2 GENERATED ALWAYS AS ROW END HIDDEN,
```

```
  PERIOD FOR SYSTEM_TIME (起始時間, 迄止時間)
```

```
) WITH(SYSTEM_VERSIONING = ON (HISTORY_TABLE = dbo.書籍 T_History));
```

加入此參數, 以系統時間寫入指定的欄位, 即我們加入 2 個 DATETIME2 欄位

此參數可以隱藏欄位, 操作資料表時我們可以忽略這些隱藏欄位

加入 2 個 DATETIME2  
資料類型欄位

← 加入此參數以自動寫入系統時間

將 History 資料表命名為 **書籍 T\_History**, 不過必須使用「兩部分名稱格式」, 亦即前面需加上 **dbo.**

加入此參數以啟用系統控制版本功能

# 建立 TEMPORAL 資料表

物件總管

連接 ▾

- [-] 練習07
  - [+] 資料庫圖表
  - [-] 資料表
    - [+] 系統資料表
    - [+] FileTable
    - [+] 外部資料表
    - [+] 書籍01
    - [+] dbo.書籍01
    - [+] dbo.書籍A
    - [+] dbo.書籍M
    - [+] dbo.書籍R
    - [+] **dbo.書籍T (由系統控制版本)**
    - [+] dbo.書籍T\_History (記錄)
    - [+] 資料行
    - [+] 索引鍵

1 這是剛才建立的 Temporal 資料表, 我們以 Current 資料表來稱呼

2 按此處展開資料表選單

3 這是系統自動建立, 用來紀錄書籍 T 異動歷程的資料表, 我們以 History 資料表來稱呼

4 指定 History 資料表名稱, 2 個資料表的關係一目瞭然

# 查詢異動紀錄

```
insert into 書籍 T (書籍名稱, 價格, 出版公司) values ('Windows 使用手冊', 500, '旗旗出版公司')
insert into 書籍 T (書籍名稱, 價格, 出版公司) values ('Office 使用手冊', 500, '標標出版公司')
insert into 書籍 T (書籍名稱, 價格, 出版公司) values ('Linux 使用手冊', 500, '旗旗出版公司')
```

主索引鍵為自動增  
值的編號, 不必寫入

2 個 DATETIME2 欄  
位不需要寫入資料

插入 3 筆資料

```
SELECT * FROM 書籍 T
```



	書籍編號	書籍名稱	價格	出版公司
1	1	Windows 使用手冊	500.00	旗旗出版公司
2	2	Office 使用手冊	500.00	標標出版公司
3	3	Linux 使用手冊	500.00	旗旗出版公司

— 剛才寫入的 3 筆資料

# 查詢異動紀錄

```
SELECT *,
```

```
    起始時間,  
    迄止時間
```

```
FROM 書籍 T
```



查看隱藏的欄位，  
必須明確指定。

隱藏欄位也顯示出來了

	書籍編號	書籍名稱	價格	出版公司	起始時間	迄止時間
1	1	Windows 使用手冊	500.00	旗旗出版公司	2016-12-01 04:52:15.5358979	9999-12-31 23:59:59.9999999
2	2	Office 使用手冊	500.00	標標出版公司	2016-12-01 04:52:15.5358979	9999-12-31 23:59:59.9999999
3	3	Linux 使用手冊	500.00	旗旗出版公司	2016-12-01 04:52:15.5358979	9999-12-31 23:59:59.9999999

這個欄位紀錄了  
資料寫入時間

這個時間為 DATETIME2 的最  
大值, 代表該筆資料尚未到期



# 查詢異動紀錄

```
SELECT * FROM 書籍 T_History
```



書籍編號	書籍名稱	價格	出版公司	起始時間	迄止時間
------	------	----	------	------	------

因為尚未更動，因此無資料顯示。

```
UPDATE 書籍 T SET 價格 = 600 WHERE 書籍編號 = 3 ← 更新一筆資料
```

```
SELECT *, 起始時間, 迄止時間 FROM 書籍 T
```

```
SELECT * FROM 書籍 T_History
```



	書籍編號	書籍名稱	價格	出版公司	起始時間	迄止時間
1	1	Windows 使用手冊	500.00	旗旗出版公司	2016-12-01 04:52:15.5358979	9999-12-31 23:59:59.9999999
2	2	Office 使用手冊	500.00	標標出版公司	2016-12-01 04:52:15.5358979	9999-12-31 23:59:59.9999999
3	3	Linux 使用手冊	600.00	旗旗出版公司	2016-12-01 04:59:04.8719364	9999-12-31 23:59:59.9999999

	書籍編號	書籍名稱	價格	出版公司	起始時間	迄止時間
1	3	Linux 使用手冊	500.00	旗旗出版公司	2016-12-01 04:52:15.5358979	2016-12-01 04:59:04.8719364

這筆資料的起始時間變成  
剛才執行更新的時間

迄止時間仍是  
未到期的狀態

History 資料表記錄  
了異動前的資料

從 2 個時間得知這筆資料的到期  
時間也就是我們更新資料的時間

# 查詢異動紀錄

Delete 書籍 T WHERE 書籍編號 = 3 ← 刪除一筆資料

SELECT \*, 起始時間, 迄止時間 FROM 書籍 T

SELECT \* FROM 書籍 T\_History



	書籍編號	書籍名稱	價格	出版公司	起始時間	迄止時間
1	1	Windows 使用手冊	500.00	旗旗出版公司	2016-12-01 04:52:15.5358979	9999-12-31 23:59:59.9999999
2	2	Office 使用手冊	500.00	標標出版公司	2016-12-01 04:52:15.5358979	9999-12-31 23:59:59.9999999
1	3	Linux 使用手冊	500.00	旗旗出版公司	2016-12-01 04:52:15.5358979	2016-12-01 04:59:04.8719364
2	3	Linux 使用手冊	600.00	旗旗出版公司	2016-12-01 04:59:04.8719364	2016-12-01 05:07:39.0027278

Current 資料表已經  
看不到刪除的資料了

這筆資料到期的時間，也  
就是我們刪除資料的時間

# 刪除或修改 **TEMPORAL** 資料表

```
DROP TABLE 書籍 T
```

訊息 13552，層級 16，狀態 1，行 1

在資料表 '練習 07.dbo.書籍 T' 上進行卸除資料表作業失敗，因為在系統設定版本的時態表上，不支援此作業。

刪除 **TEMPORAL** 資料表，須以下列方式進行：

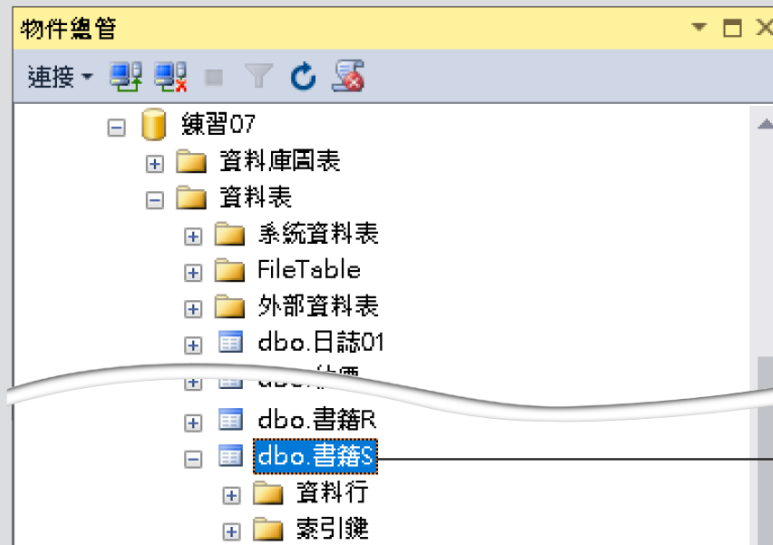
```
ALTER TABLE 書籍 T  
    SET (SYSTEM_VERSIONING = OFF)
```

```
DROP TABLE 書籍 T
```

```
DROP TABLE 書籍 T_History
```

# 將現存資料表改成 **TEMPORAL** 資料表

```
CREATE TABLE 書籍 S  
(  
    書籍編號 INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,  
    書籍名稱 VARCHAR(50) NULL,  
    價格 SMALLMONEY NULL,  
    出版公司 CHAR(20) NULL  
)
```



書籍 S 是一  
般的資料表

# 將現存資料表改成 **TEMPORAL** 資料表

--加入 2 個 DATETIME2 欄位, 並指定寫入系統時間

```
ALTER TABLE 書籍 S
```

```
ADD PERIOD FOR SYSTEM_TIME (起始時間, 迄止時間), ← 設定系統時間寫入 2 個欄位
```

```
起始時間 DATETIME2 GENERATED ALWAYS AS ROW START HIDDEN,
```

```
迄止時間 DATETIME2 GENERATED ALWAYS AS ROW END HIDDEN
```

新增 2 個 Datetime2 欄位

```
GO
```

--啟用 SYSTEM\_VERSIONING 參數

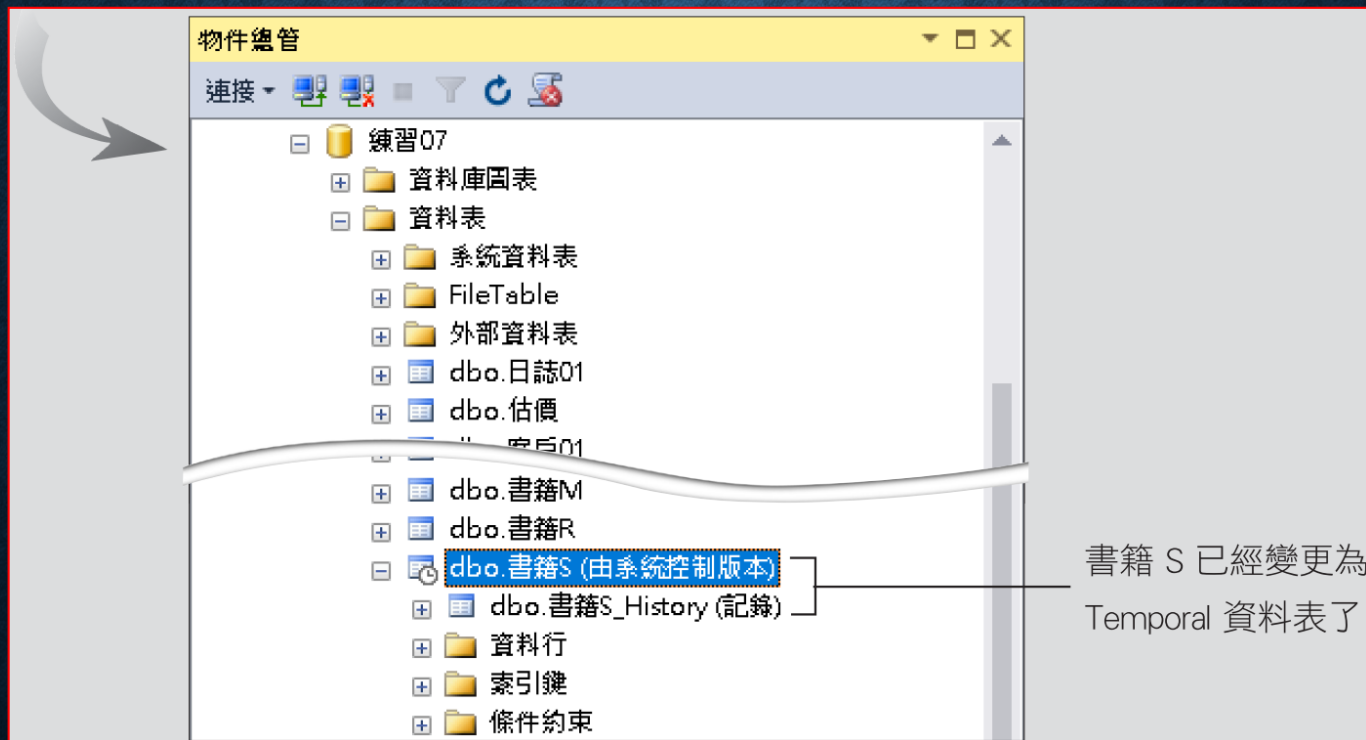
```
ALTER TABLE 書籍 S
```

```
SET (SYSTEM_VERSIONING = ON (HISTORY_TABLE = dbo.書籍 S_History)) ←
```

啟用 SYSTEM\_VERSIONING 參數, 並指定 History 資料表名稱

```
GO
```

# 將現存資料表改成 **TEMPORAL** 資料表



**TEMPORAL** 資料表雖然好用，但是對 SQL Server 的效能和磁碟用量勢必產生影響，尤其是資料表中有二進位資料 (binary、image等)。