



Austin, Texas, USA, by Frank S.C. Tseng
(<http://www2.nkfust.edu.tw/~imfrank>)

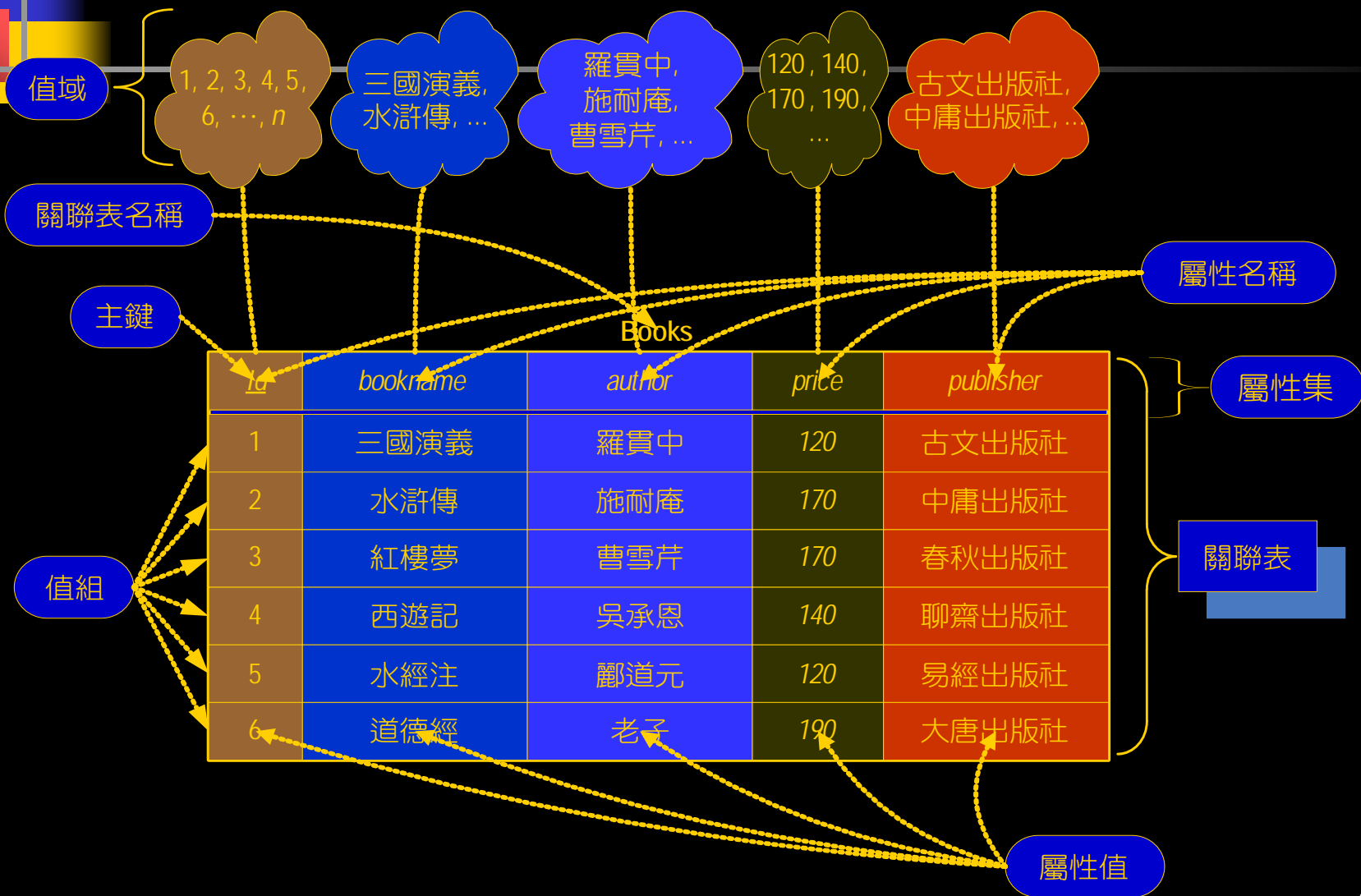
第四章 關聯式資料模式的資料結構



本章內容

- 4.1 關聯表綱要簡介
- 4.2 關聯表的內容
- 4.3 關聯表的正式定義
- 4.4 關聯表的特性
- 4.5 關聯表的種類
- 4.6 關聯式資料庫
- 4.7 在 MS SQL Server 2008 上建立表格
- 4.8 將資料匯入或匯出資料庫

關聯表綱要簡介






關聯表綱要上的名詞

- 關聯表名稱 (Relation Name) : Books
- 屬性 (Attribute) : id
- 屬性集 (Attribute Set) : {id, bookname, ...}
- 指定某個屬性子集為主鍵 (Primary Key) : id
- 屬性都會具有屬性名稱 (Attribute Name) 與資料類型 (Data Type) : id (int)
- 資料型態都會有其值域 (Domain) : {0, 1, 2, ...}



翻譯名詞上的說明

- 在數學函數 $f: A \rightarrow B$ 中， A 稱為 Domain，翻譯成「定義域」，數學上將 A 經過 f 映射後的結果集合 (可能是 B 的子集合) 稱為 Range，並翻譯成「值域」。集合 B 也稱為 Co-Domain (翻譯成「對應域」)
- 在關聯式資料庫中的 Domain 所代表的是「屬性資料值的領域」，所以我們覺得簡稱為「值域」較為恰當 (e.g. MS SQL Server 2000 的中文版手冊提到 Domain Integrity 也翻為「值域完整性」)。這就像 Domain 在網路應用中翻譯為「網域」而非「定義域」一般。
- 不過，請不要因此混淆函數中的 Domain 與 Range 間的關係。
- 如果不習慣將 Domain 翻譯成「值域」，那麼翻成「定義域」也無妨。但是，不論使用哪一個翻譯名稱，我們都建議在應付各種考試時都要加註原文，比較不會有爭議。



關聯表綱要上的名詞 (續)

- 關聯表的屬性數目稱為它的維度 (Degree)
- 一個關聯表中任兩個屬性名稱不可以重覆
- 「關聯表綱要」 (Relation Schema) 包含：
 - 關聯表名稱、
 - 屬性集宣告
 - 所有屬性的相對資料型態宣告
 - 主鍵的宣告
 - 外來鍵 (Foreign Key) 之宣告 (下一章說明)



關聯表綱要的用途

- 綱要：用來描述資料庫/關聯表中的資料 (Meta-data)
- 通常在定義時會省略最後的外來鍵

例如：

Books (*id*(int), *bookname*(string), *author*(string), *price*(int),
publisher(string))

- 更進一步省略：

Books (*id*, *bookname*, *author*, *price*, *publisher*)

關聯表的內容

- 存入的單位是一筆一筆的記錄 (Records)，正式名稱叫「值組」 (Tuple)



- 一筆記錄則是由許多屬性值所組成



--	--	--	--	--

- 關聯表中所含的值組數目叫「基數」 (Cardinality)



主鍵 (Primary Key)

- 由集合定義可知：關聯表中的值組不可以重覆
- 主鍵值是值組唯一的識別值 (Unique Identifier)
 - 由屬性集的子集所構成
 - 可能由一個以上的屬性 (Attributes) 所構成
 - 甚至可以由該關聯表的所有屬性所構成
- 關聯表中的主鍵只能有一個



值域 (Domains)

- 同一類型之數值 (但不可以是集合) 所構成的集合：
(0 可以) $\{1, 3, 5\}$ ，(X 不行) $\{\{1, 2\}, 1\}$
- 值域相同 (或相似，例如：整數與實數) 的值之間才能互相作比較
- 單一值域 (Simple Domain)：
 $D = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
- 複合值域 (Composite Domain)：將 A1 (值域為 D1) 與 A2 (值域為 D2) 兩個屬性當成一個屬性 A 來看待時，則 A 的值域為 $D = D1 \times D2$ ，
若 $D1 = \{1, 2, 3\}$, $D2 = \{1, 2\}$ 則
 $D1 \times D2 = \{(1,1), (1,2), (2,1), (2,2), (3,1), (3,2)\}$



屬性 (Attributes)

- 屬性是一個集合，其中所含的元素稱為屬性值 (Attribute Value)
- 屬性一定要定義於某個值域上
- 單一屬性 (Simple Attribute) : *bookname*
屬性值例如：'三國演義'
- 複合屬性 (Composite Attribute) : 將數個屬性看成一個,例如: 將 *year, month, day* 看成 *Date = (year, month, day)*, 屬性值例如：(1997, 3, 15)



虛值 (Null Values)

- 資訊缺乏導致以「不明確」的值來存放
- 虛值不是 0 或空白，而是一個特殊標記
- 兩個虛值是無法比較大小的。
- 按虛值的意義與特性可以細分成：
 - 可應用的虛值 (Applicable Null Values)
 - 不可應用的虛值 (Inapplicable Null Values)
 - 完全不知道的虛值 (Totally Unknown)
- 虛值的變形：部份值 (Partial Values), 機率部份值 (Probabilistic Partial Values), ...等



可應用的虛值 (Applicable Null)

- 其意義為：這個值目前存在，但不知其值為何？

<u><i>eid</i></u>	<i>name</i>	<i>city</i>
1	老王	臺北市
2	老張	—



不可應用的虛值 (Inapplicable)

- 其意義為：這個值目前根本不存在, 使用者沒有東西可以存

<u><i>eid</i></u>	<i>name</i>	<i>spouse</i>
1	老王	吳小姐
2	老張	※



完全不知道的虛值 (Totally Unknown)

- 其意義為：不知道這個值目前是否存在, 因此即使這個值存在也不知道其值為何?

<u><i>eid</i></u>	<i>name</i>	<i>spouse</i>
1	老王	吳小姐
2	老張	#



虛值的變形

- 部份值 (Partial Values) — John Grant (1979)
 $[a, b, c]$ (a, b, c 中恰有一個是真的值)
- 部份值其實是「可應用的虛值」之推廣型式
- 「機率部份值」 (Probabilistic Partial Values)
 $[a^{0.3}, b^{0.2}, c^{0.5}]$ (a, b, c 中每一個值為真的機率, 全部機率加起來要等於 1)
- Inclusive Disjunctive Data : $[a, b, c]$
(a, b, c 中至少有一個是真的值)



資料庫管理系統並未完全支援虛值

- 虛值在資料庫中有存在的必要，但是卻會對查詢處理、統計分析造成很多困擾。
- 雖然很多研究人員嘗試著要解決此類困擾，但總是因複雜而難以完全實現，
- 目前市面上的資料庫管理系統大多只提供部份的虛值表示法，以及部份牽涉到虛值的運算，
- 仍然無法完全支援所有與虛值有關的運算與判斷機制，
- 使用者必須透過自行撰寫程式，配合自我規定的資料表示協定，自行區分各種虛值的意義。



關聯表的正式定義

- 一個關聯表包含標題 (Heading) 與本體 (Body) :
 - 標題 (Heading) : 就是關聯表 R 的綱要 (Schema)
標題部份是固定的，不太隨時間而變動
 $R(A_1(D_1), A_2(D_2), \dots, A_n(D_n))$
 - 本體 (Body) : 指的是關聯表中的資料部份，其內容與數量是隨時間而變動
 $\{ \langle A_1:V_1 \rangle, \langle A_2:V_2 \rangle, \dots, \langle A_n:V_n \rangle \}$



關聯表的特性

- 不含重覆的值組
- 值組之間是沒有順序的
- 屬性之間是沒有順序的
- 所有屬性值都是單元值 (Atomic Value)，不可以是一個集合
e.g. $\{x, y, z\}$ 不可以是一個屬性值
- 同一欄位內的所有資料都是來自同一個值域 (Domain)，都屬於同一種資料類型 (Data Type)。此項特性通常被認為是理所當然，因此不予討論



不含重複的值組

- 因為關聯表本身是一個集合
- 此一特性隱含了必須要有主鍵的事實
- 在實際應用上，系統可容許有重複的值組存在 (但我們不建議如此做)



值組之間是沒有順序的

- 按照集合的定義，集合中的元素沒有順序之分
- 有順序反而造成麻煩：
 - 順序本身也是一種資訊，所以要小心維護其順序
 - 系統必須提供「在第 n 與第 $n+1$ 筆間加入一筆」或「請刪除第八筆」的運算
 - 要以那一個屬性做為排順序的依據？還是要以加入的時間先後來排順序？
- 通常是查詢後對結果下排序命令



屬性之間是沒有順序的

- 屬性集也是一個集合
- 不過大部份的系統並沒有硬性規定
- 例如：在 VB 中 `Employees("id") = Employees(0)`

Employees

<u><i>id</i></u>	<i>name</i>	<i>salary</i>
1	Frank	50K

所有屬性值都是單元值

- (1) 簡化表示法
- (2) 容易實現關聯式模式的概念

Books

<u>id</u>	bookname	author	price	publisher
1	{天龍八部，鹿鼎記}	金庸	{250,300}	古文出版社
2	水滸傳	施耐庵	170	中庸出版社
3	紅樓夢	曹雪芹	170	春秋出版社
4	西遊記	吳承恩	140	聊齋出版社
5	水經注	酈道元	120	易經出版社
6	道德經	老子	190	大唐出版社

所有屬性值都是單元值 (續)

- 否則，系統要提供兩套運算—附加 (Append) 與新增 (Insert)
- 附加 (Append)：加入

1	神雕俠侶	金庸	250	古文出版社
---	------	----	-----	-------

- 新增 (Insert)：加入

7	三國演義	羅貫中	120	漢唐出版社
---	------	-----	-----	-------



第一正規化型式 (1NF)

- 在關聯式模式裡所有關聯表都要符合1NF

NewBooks

<u>id</u>	bookname	author	price	publisher
1	天龍八部	金庸	250	古文出版社
2	鹿鼎記	金庸	300	古文出版社
3	水滸傳	施耐庵	170	中庸出版社
4	紅樓夢	曹雪芹	170	春秋出版社
5	西遊記	吳承恩	140	聊齋出版社
6	水經注	酈道元	120	易經出版社
7	道德經	老子	190	大唐出版社

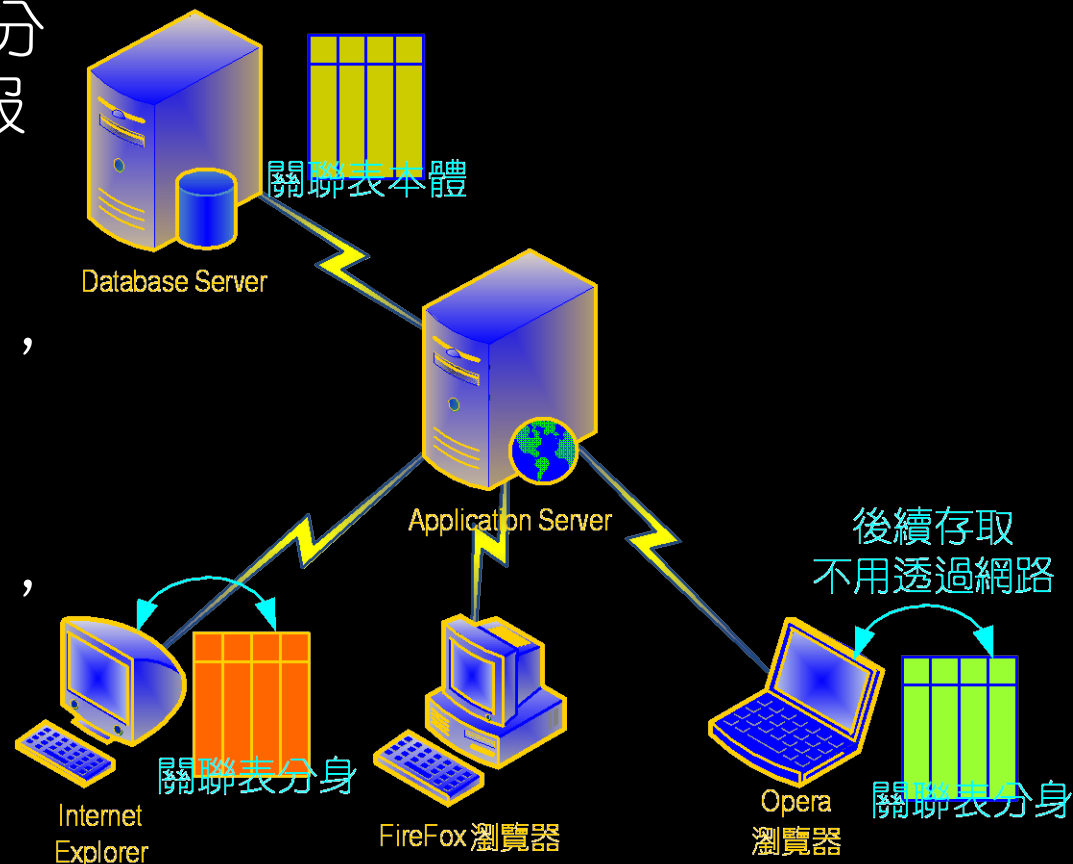


關聯表的種類

- 基底關聯表 (Base Relations)
- 導出 (衍生) 關聯表 — Derived Relations
- 視界 (Views)
- 瞬間關聯表 (Snapshots)
- 查詢結果 (Query Result)
- 查詢中間結果 (Intermediate Result)

瞬間關聯表 (Snapshots)

- 通常指：主-從式架構 (或分散式架構) 上由資料庫伺服器複製出來，放在客戶端的關聯表分身 (Replica)，
- 可減少客戶端的網路流量，加快查詢速度。
- 因為別的使用者可能會去更改伺服端的關聯表本體，所以要注意：「瞬間關聯表」在使用過程中，可能跟伺服端的關聯表本體內容不相同。





關聯式資料庫

- 由一群具有相互關係的正規化關聯表所組成
- 所有關係皆以資料值來聯繫，沒有任何鏈結指標（所以我使用「關聯表」而非「關連表」）
- 所有的屬性值都是單元值，不可以是集合
- 記錄之間沒有任何順序關係
- 請見課本的 **BOB** 資料庫 (後續皆以它做為範例)

Bookstores

<i>no</i>	<i>name</i>	<i>rank</i>	<i>city</i>
1	巨蟹書局	20	臺北市
2	射手書局	10	高雄市
3	水瓶書店	30	新竹市
4	天秤書局	20	臺中市
5	獅子書局	30	臺南市

Orders

<i>no</i>	<i>id</i>	<i>quantity</i>
1	1	30
1	2	20
1	3	40
1	4	20
1	5	10
1	6	10
2	1	30
2	2	40
3	2	20
4	2	20
4	4	30
4	5	40

Books

<i>id</i>	<i>bookname</i>	<i>author</i>	<i>price</i>	<i>publisher</i>
1	三國演義	羅貫中	120	古文出版社
2	水滸傳	施耐庵	170	中庸出版社
3	紅樓夢	曹雪芹	170	春秋出版社
4	西遊記	吳承恩	140	聊齋出版社
5	水經注	酈道元	120	易經出版社
6	道德經	老子	190	大唐出版社



BOB 暫時忽略真實應用的考量

- 在真實應用上，BOB 資料庫中的 Orders 關聯表原本應該還有一個欄位當作主鍵。例如：
 - 存放**訂單編號**的欄位，假設叫做「Order_id」，以便允許同一家書局可以有多筆訂單來訂購同一本書；或者
 - 也可以加入「Order_datetime」這個時間欄位用來存放**訂購的時間點**，這樣也是允許同一家書局可以有多筆訂單來訂購同一本書，而不會有主鍵重複的問題發生
- 但是為了簡潔起見，我想我們**先忽略此欄位的存在**，讓後續的例子能更容易說明。

Bookstores

<i>no</i>	<i>name</i>	<i>rank</i>	<i>city</i>
1	巨蟹書局	20	臺北市
2	射手書局	10	高雄市
3	水瓶書店	30	新竹市
4	天秤書局	20	臺中市
5	獅子書局	30	臺南市

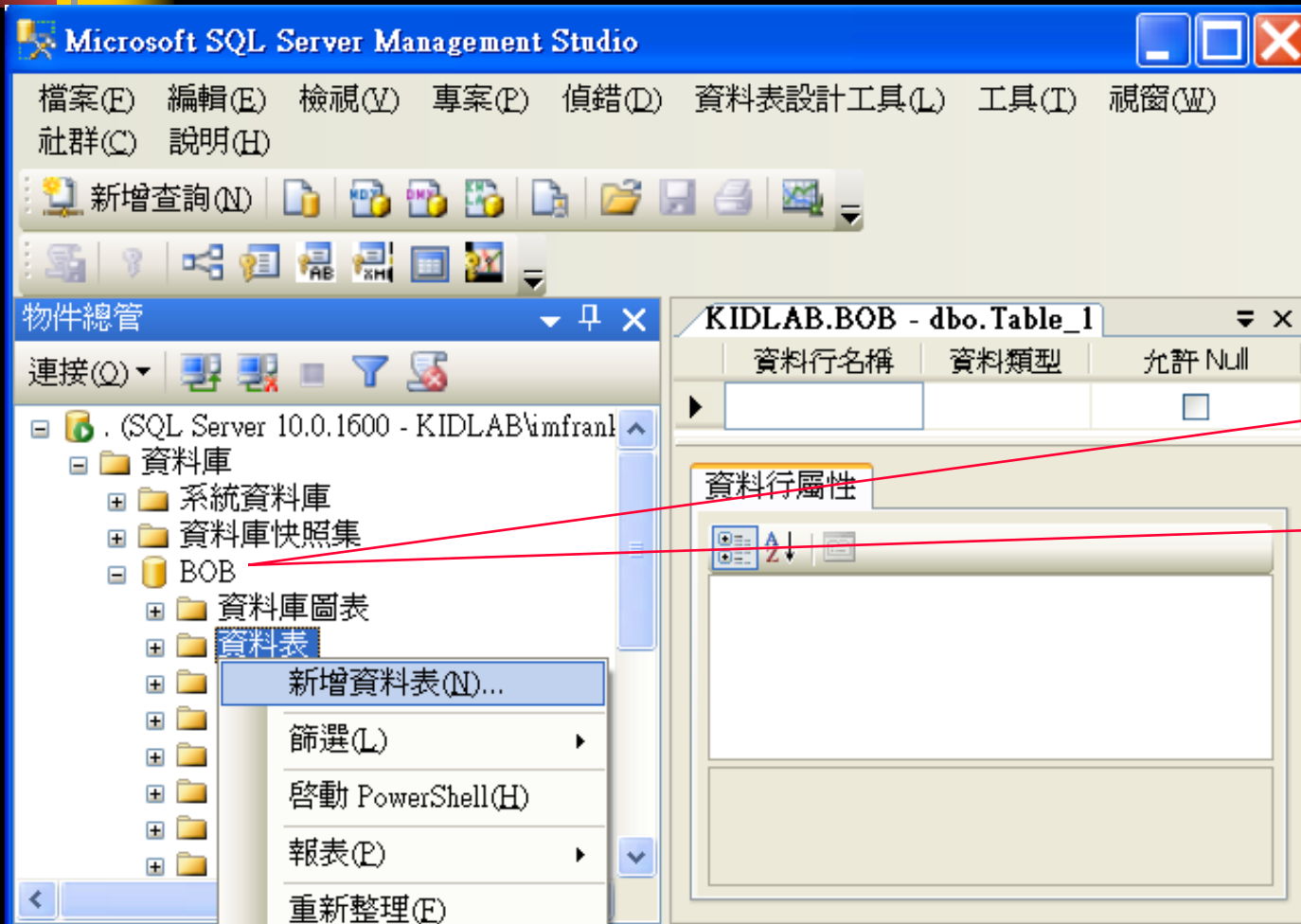
在真實應用上，Orders 關聯表應該還要有一個欄位當作主鍵

Books

<u><i>id</i></u>	<i>bookname</i>	<i>author</i>	<i>price</i>	<i>publisher</i>
1	三國演義	羅貫中	120	古文出版社
2	水滸傳	施耐庵	170	中庸出版社
3	紅樓夢	曹雪芹	170	春秋出版社
4	西遊記	吳承恩	140	聊齋出版社
5	水經注	酈道元	120	易經出版社
6	道德經	老子	190	大唐出版社

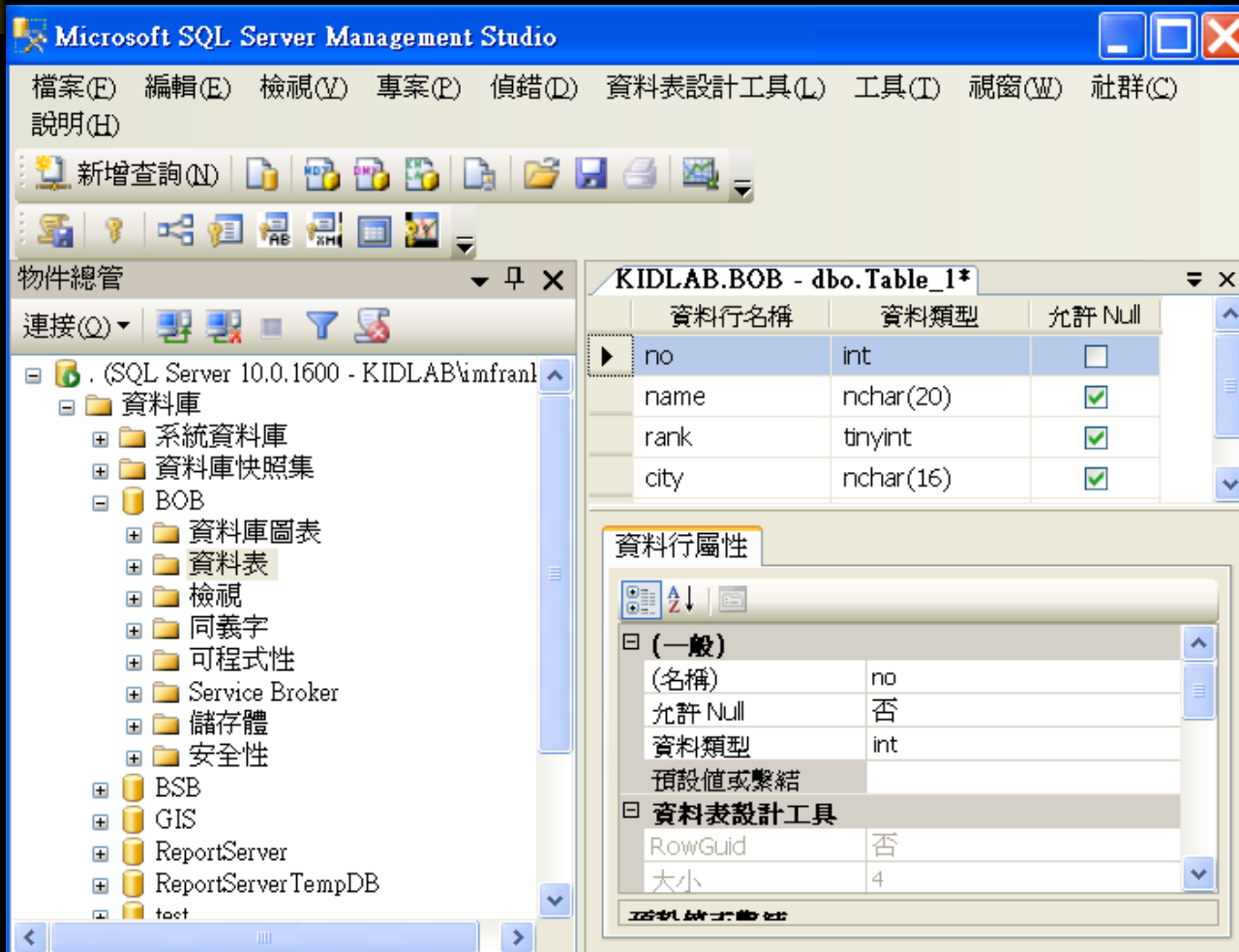
<u><i>Order id</i></u>	<i>no</i>	<i>id</i>	<i>quantity</i>
070101-1	1	1	30
070101-2	1	2	20
070101-3	1	3	40
070102-1	1	4	20
070102-2	1	5	10
070103-1	1	6	10
070103-2	2	1	30
070103-3	2	2	40
070104-1	3	2	20
070104-2	4	2	20
070104-3	4	4	30
070107-1	4	5	40

在 SQL Server 上建表格

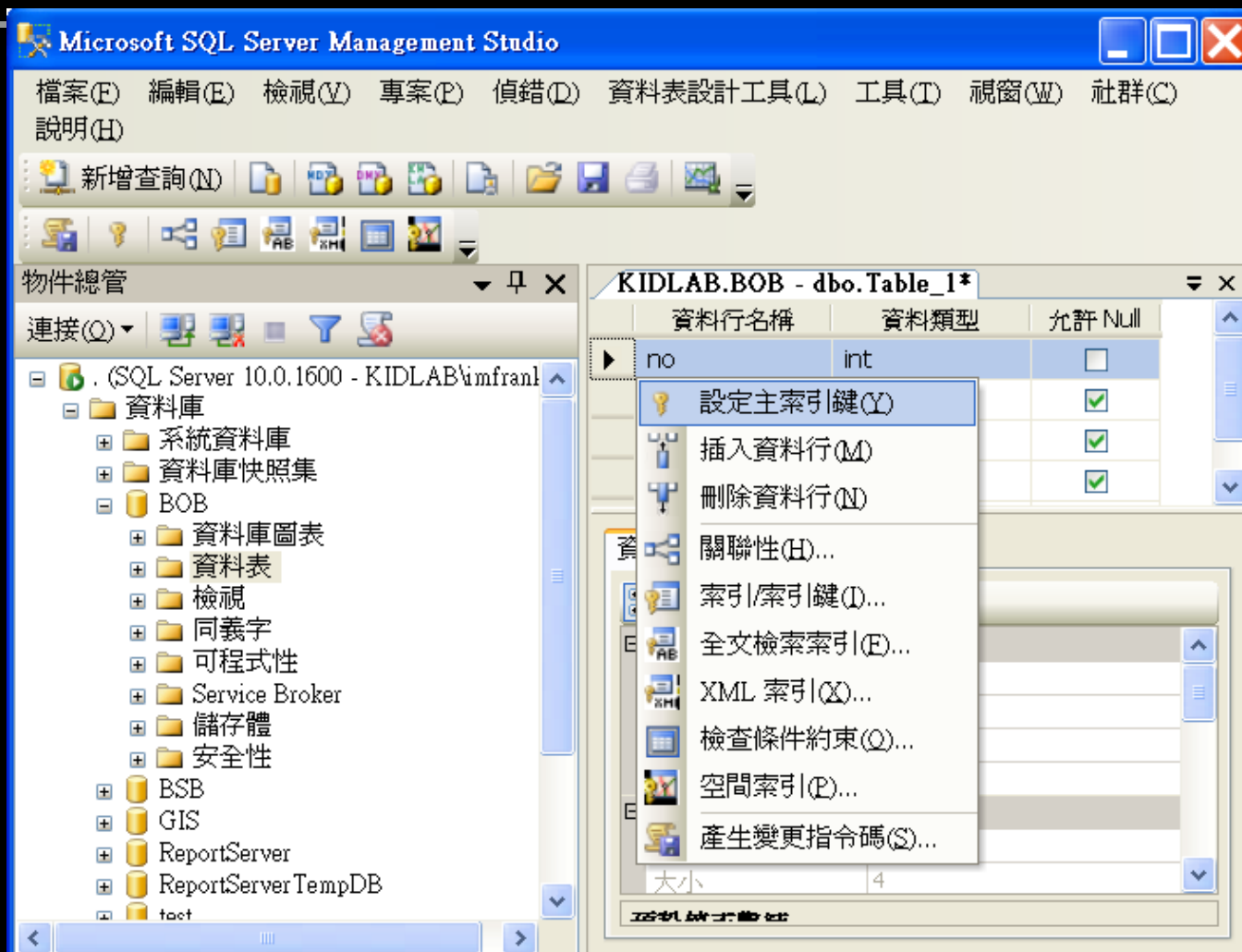


BOB
資料庫
要先建立

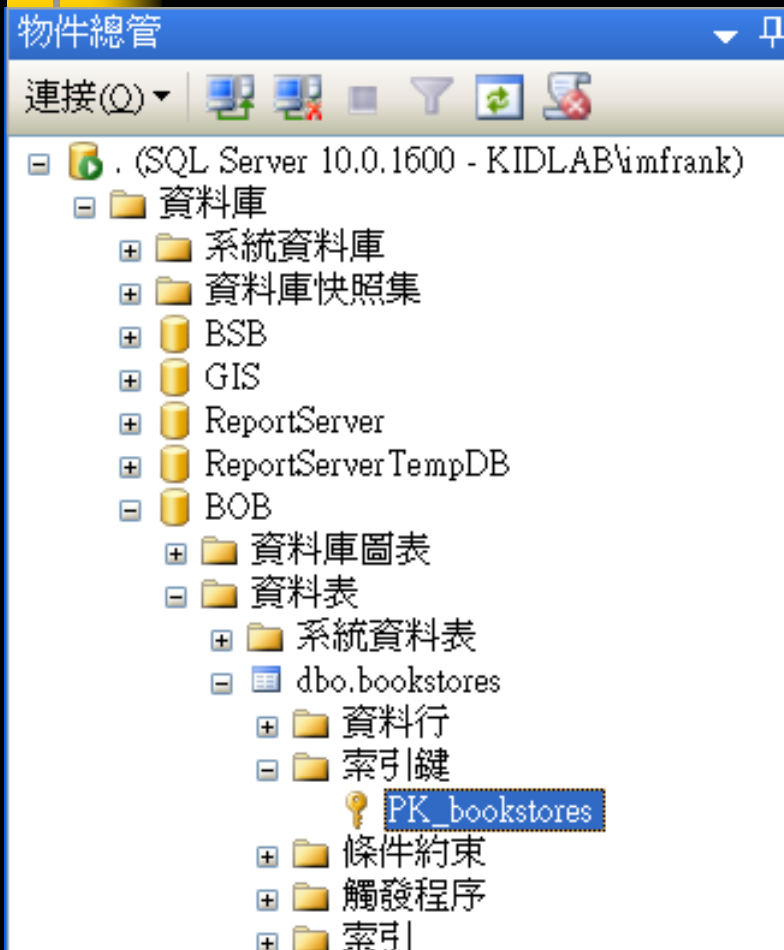
建立 Bookstores 表格



設定主鍵之限制條件



表格主鍵已設定



查詢限制條件設定

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "Microsoft SQL Server Management Studio". The menu bar includes "檔案(F)", "編輯(E)", "檢視(V)", "查詢(Q)", "專案(P)", "偵錯(D)", "工具(T)", "視窗(W)", "社群(C)", and "說明(H)". The toolbar contains icons for "新增查詢(N)", "執行(O)", and others. The "物件總管" (Object Explorer) on the left shows a tree structure for "KIDLAB\imfrank", including "資料庫" (Databases) and "安全性" (Security). The "SQL Query2.sql" window shows the query "sp_helpconstraint Bookstores". The "結果" (Results) pane at the bottom displays the output of the query, which is a table with columns "constraint_type" and "constraint_na...". The first row of the results is "PRIMARY KEY (clustered)" and "PK_bookstores".

Microsoft SQL Server Management Studio

檔案(F) 編輯(E) 檢視(V) 查詢(Q) 專案(P) 偵錯(D) 工具(T) 視窗(W) 社群(C) 說明(H)

新增查詢(N) 執行(O)

物件總管

連接(Q)

SQL Query2.sql - (loc...IDLAB\imfrank (51))

sp_helpconstraint Bookstores

結果 訊息

Object Name
1 Bookstores

constraint_type	constraint_na...
1 PRIMARY KEY (clustered)	PK_bookstores

1 KIDLAB\imfrank (51) BOB 00:00:00 2 個資料列

建立 Books 的畫面

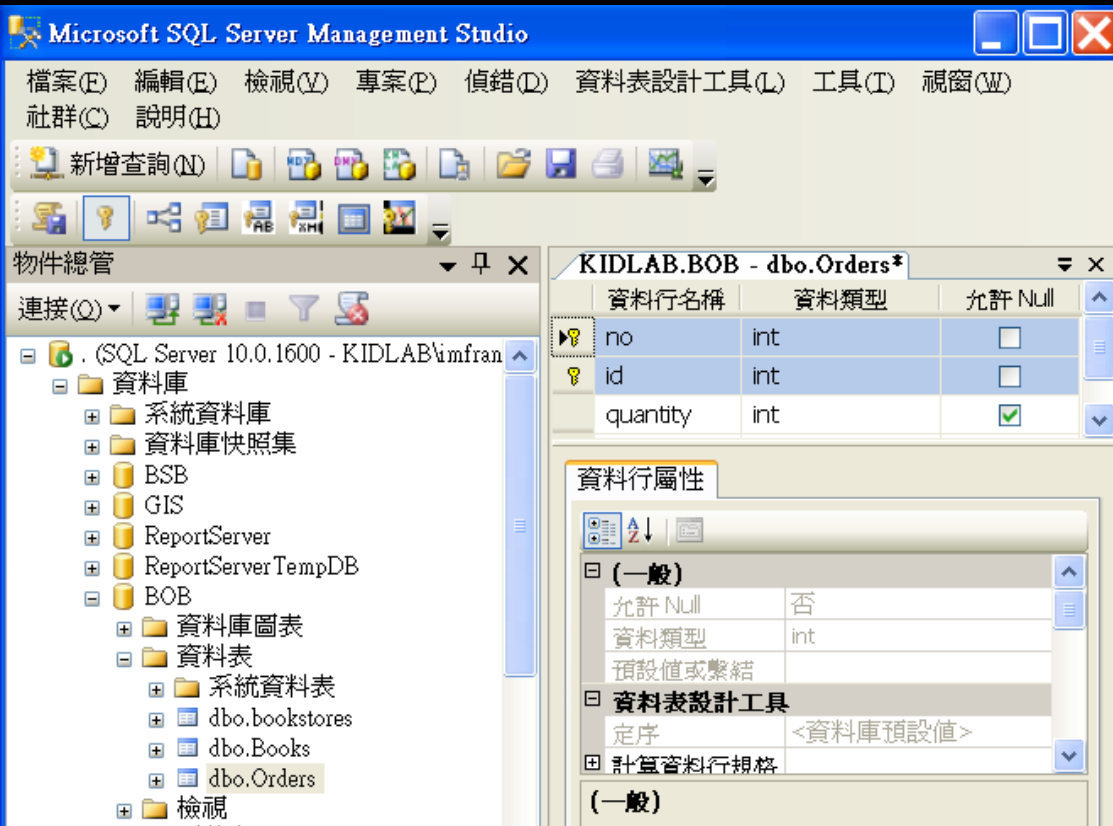
The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the '物件總管' (Object Explorer) with the 'KIDLAB.BOB' database selected. The 'dbo.Books' table is highlighted. The right pane shows the 'KIDLAB.BOB - dbo.Books*' table structure with the following columns:

資料行名稱	資料類型	允許 Null
id	int	<input type="checkbox"/>
bookname	nchar(50)	<input checked="" type="checkbox"/>
author	nchar(30)	<input checked="" type="checkbox"/>
price	smallint	<input checked="" type="checkbox"/>
publisher	nchar(20)	<input checked="" type="checkbox"/>

Below the table structure, the '資料行屬性' (Column Properties) window is open for the 'id' column. It shows the following properties:

屬性	值
(名稱)	id
允許 Null	否
資料類型	int

建立 Orders 的畫面



注意：因為主鍵 (*no, id*) 包含兩個欄位，要同時選取這兩個欄位 (先選一個後按住 [Shift] 或 [Ctrl] 鍵再選取另一個)，再按滑鼠右鍵，選擇 **[設定主索引鍵 (Y)]** 才能設定成功

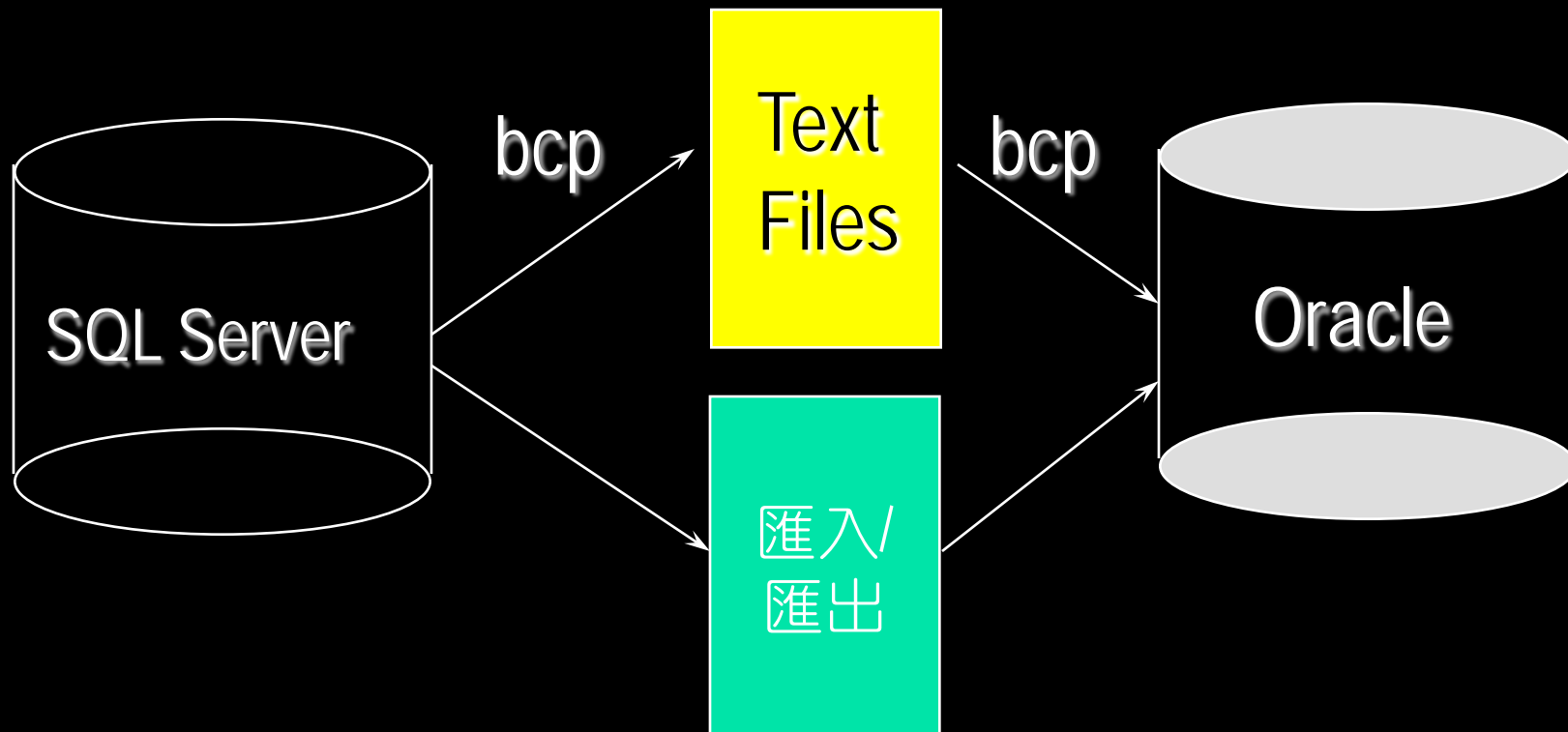


將資料匯入或匯出資料庫

- 在 SQL Server 中有一個 Dos下的 bcp 命令，速度很快
(位於 *C:\Program Files\Microsoft SQL Server\100\Tools\Binn*下)
 - fast bcp 模式：不記錄異動記錄
 - slow bcp 模式：當表格中有建立索引或觸發程序時使用
- 在 SQL Server 2008 中也可以使用 匯入/匯出工具來做，很好用但速度較慢

匯入或匯出資料庫

- 將某類型的資料庫 (如：Sybase) 轉移到不同類型的資料庫 (例如：Oracle) 時，不可或缺





使用 bcp 的注意事項

- 為了加速效能，資料庫中如下的 'Select into/bulkcopy' 選項最好先設定成 True
- 在 SQL Query Analyzer 下使用指令來設定

```
sp_dboption Bob, 'select into/bulkcopy', true
```



使用 bcp 的注意事項

- 在作業系統下面下指令

BCP.EXE *dbtable* {in | out} *datafile*
[-m *maxerrors*] [-f *formatfile*] [-e *errfile*]
[-F *firstrow*] [-L *lastrow*] [-b *batchsize*]
[-n *native type*] [-c *character type*] [-q *quoted identifier*]
[-t *field terminator*] [-r *row terminator*]
[-i *inputfile*] [-o *outfile*] [-a *packet size*]
[-E *explicit identity*] [-U *username*] [-P *password*]
[-S *server*] [-v *version*] [-T *trusted connection*]



使用 bcp 的注意事項

- *dbtable* 通常的寫法是 *database..tablename*
- -q：指定每一個欄位資料的引號為何？(例如：以雙引號做為欄位標示則使用 -q")
- -t：指定欄位資料之間的分隔符號為何？(例如：以逗號做為欄位標示則使用 -t,)。以[Tab]來做為分隔符號時，則以 C 語言的語法，再用雙引號夾起來(例如：-t"\"代表以[Tab]做為區隔符號)。



使用 bcp 的注意事項

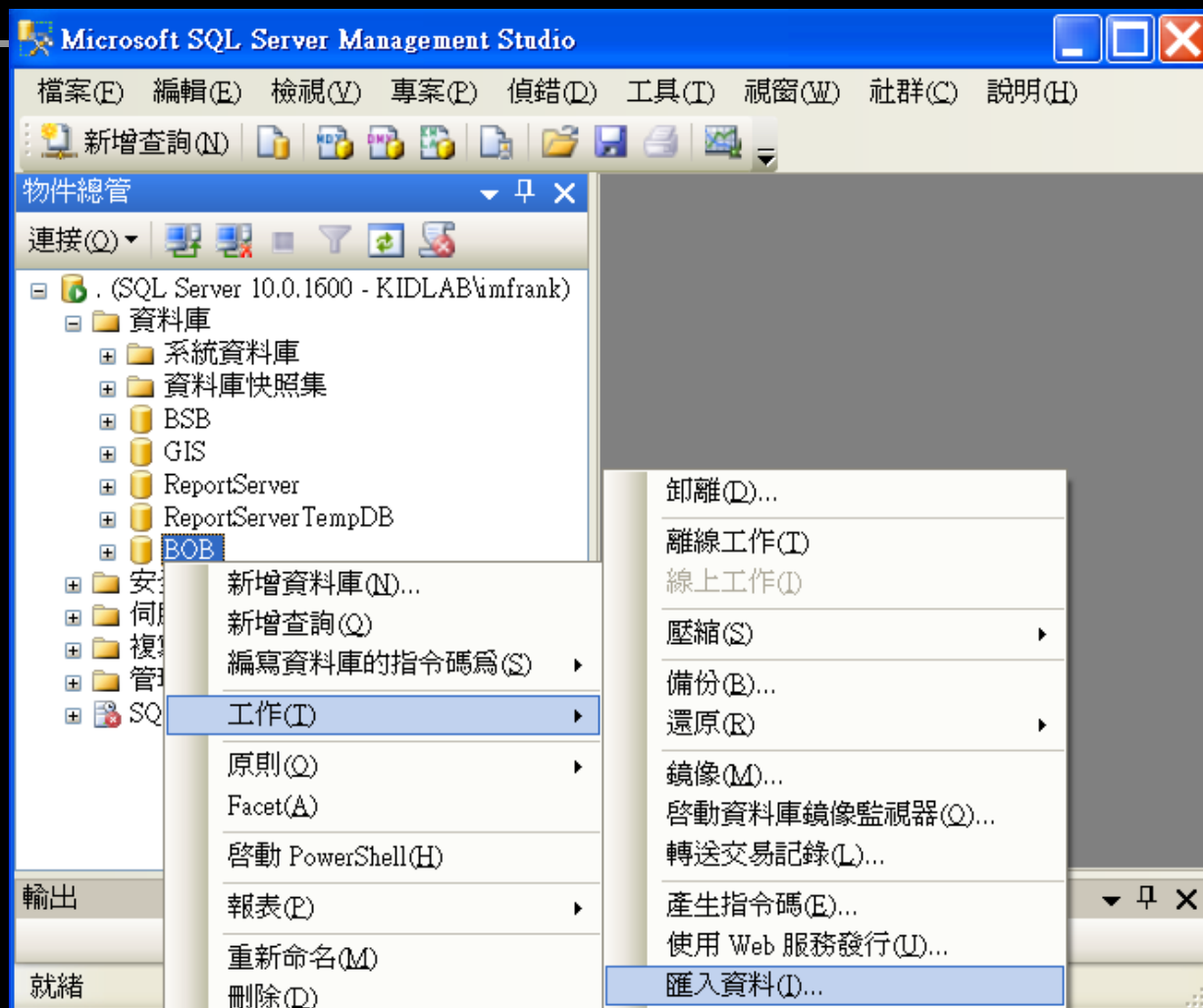
- -r：用來指定值組資料之間的分隔符號為何？通常不寫表示使用換行字元 (Carriage Return)
- -U：指定使用者名稱。例如：-Usa 代表是 sa
- -P：用來指定密碼為何？例如：-Ptitanic 代表密碼是 titanic
- -S：用來選定要對那一部 SQL Server 做 BCP 動作？

bcp 的使用範例

- 於文字編輯器中先行將資料庫中的資料輸入檔案 books.txt 中，並以 “,” 隔開各欄位。
- c:\>bcp BOB..Books in books.txt -c -t, -Usa -Pxxx -Syyy

```
1,三國演義,羅貫中,120,古文出版社↵  
2,水滸傳,施耐庵,170,中庸出版社↵  
3,紅樓夢,曹雪芹,170,春秋出版社↵  
4,西遊記,吳承恩,140,聊齋出版社↵  
5,水經注,酈道元,120,易經出版社↵  
6,道德經,老子,190,大唐出版社↵  
↵
```

使用 SSIS 匯入/匯出資料



SQL Server 匯入和匯出精靈

選擇資料來源
選取要複製資料的來源。

資料來源(D): SQL Server Native Client 10.0

伺服器名稱(S): (local)

驗證

☒ 使用 Windows 驗證(W)

☐ 使用 SQL Server 驗證(Q)

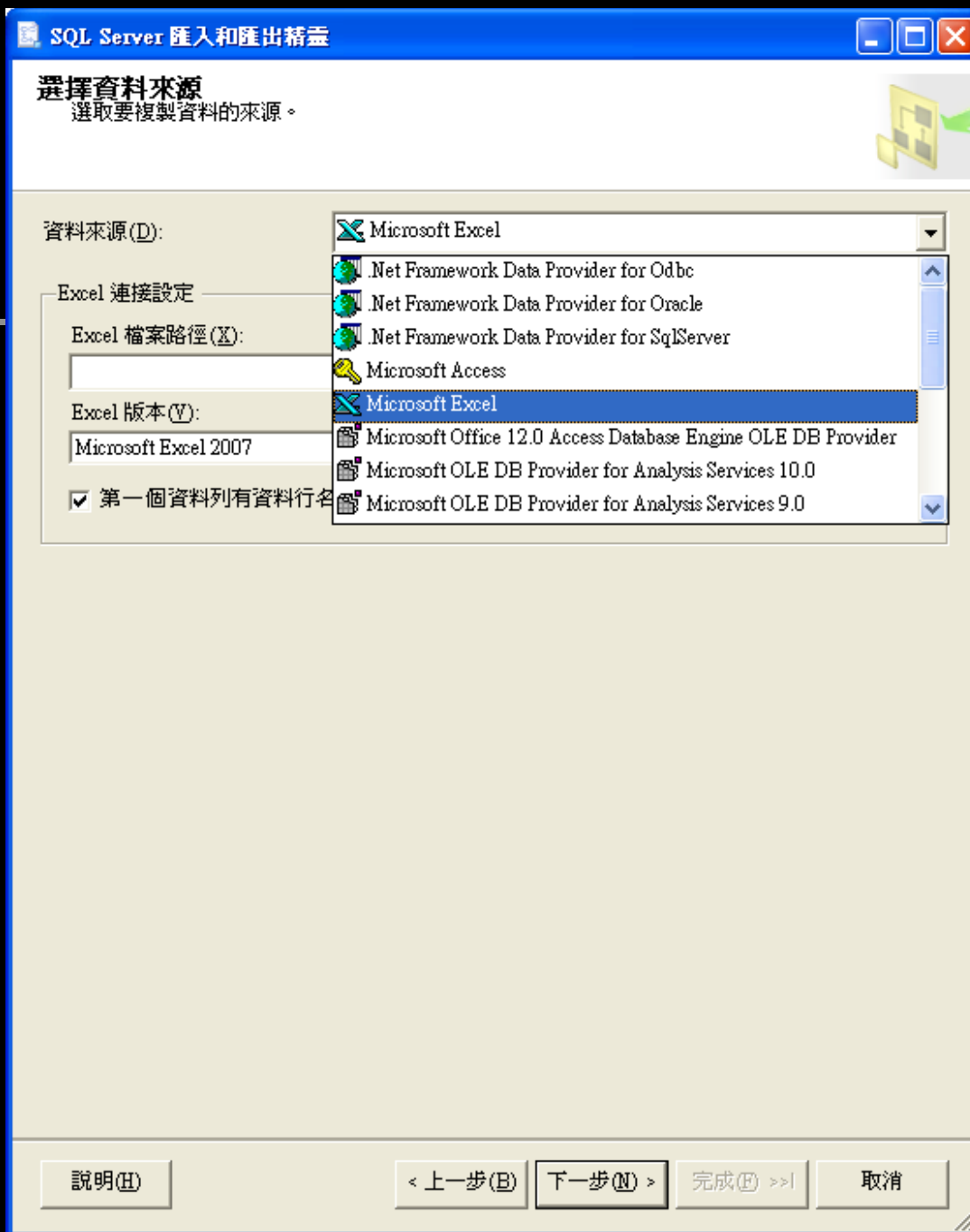
使用者名稱(U):

密碼(P):

資料庫(I): <預設值> 重新整理(R)

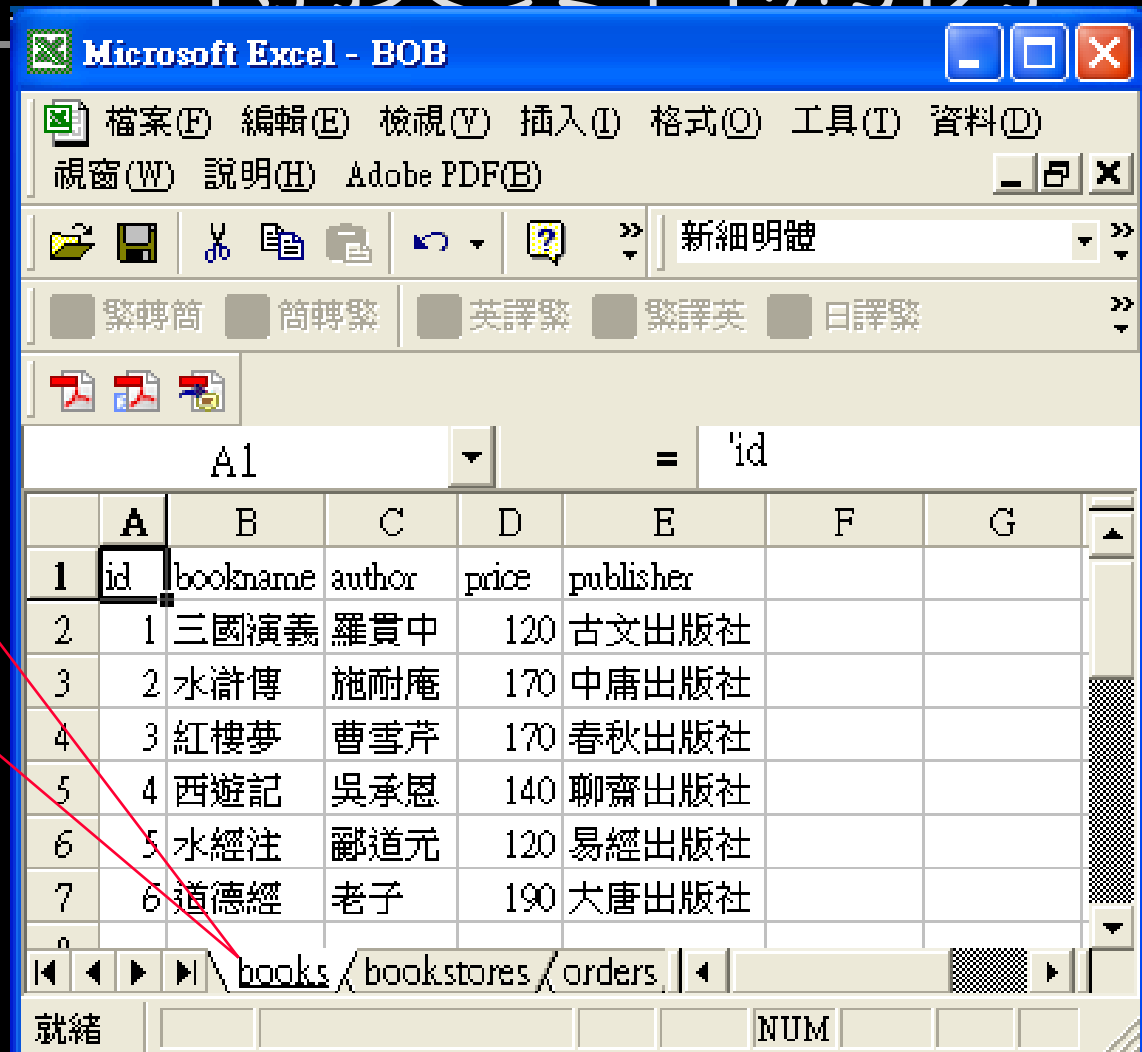
說明(H) < 上一步(B) 下一步(N) > 完成(F) >> 取消

多樣化的資料來源轉換



從 Excel 轉換資料為例

先建好
Books,
Bookstores,
Orders
的資料
在 Excel 中



Microsoft Excel - BOB

檔案(F) 編輯(E) 檢視(V) 插入(I) 格式(O) 工具(T) 資料(D)
視窗(W) 說明(H) Adobe PDF(B)

新細明體

繁轉簡 簡轉繁 英譯繁 繁譯英 日譯繁

A1 = 'id'

	A	B	C	D	E	F	G
1	id	bookname	author	price	publisher		
2	1	三國演義	羅貫中	120	古文出版社		
3	2	水滸傳	施耐庵	170	中庸出版社		
4	3	紅樓夢	曹雪芹	170	春秋出版社		
5	4	西遊記	吳承恩	140	聊齋出版社		
6	5	水經注	酈道元	120	易經出版社		
7	6	道德經	老子	190	大唐出版社		

books / bookstores / orders

就緒 NUM

SQL Server 匯入和匯出精靈

選擇目的地

指定複製資料的目的地。

目的地(D): SQL Server Native Client 10.0

伺服器名稱(S): .

驗證

☒ 使用 Windows 驗證(W)

☐ 使用 SQL Server 驗證(Q)

使用者名稱(U):

密碼(P):

資料庫(I): BOB

重新整理(R)

新增(N)...

說明(H) < 上一步(B) 下一步(N) > 完成(F) >>| 取消

SQL Server 匯入和匯出精靈

指定資料表複製或查詢

指定是否要從資料來源複製一或多個的資料表/檢視，或複製查詢的結果。

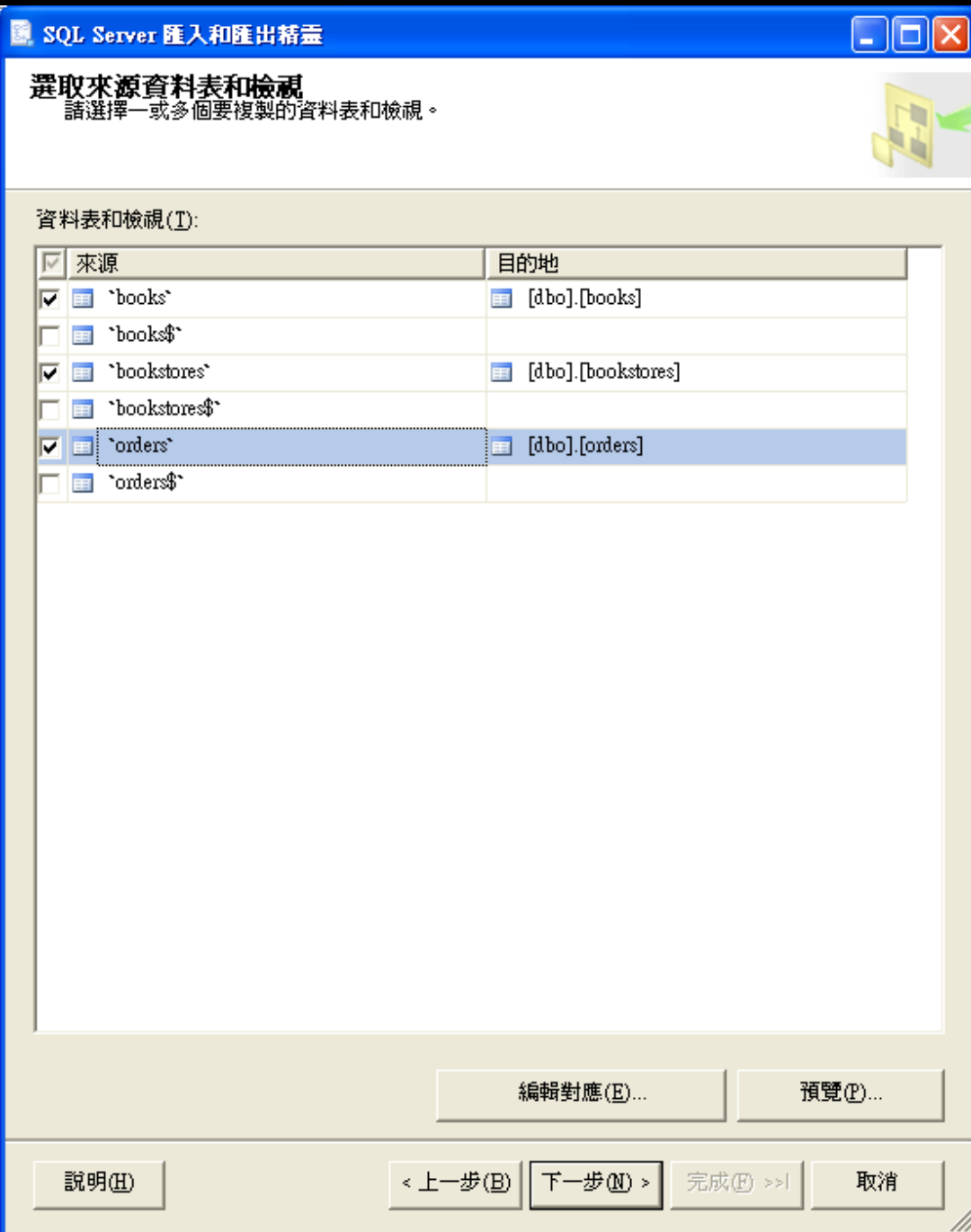
☒ **從一或多個資料表或檢視複製資料(C)**

使用這個選項可以複製來源資料庫中現有資料表或檢視的所有資料。

☐ **寫入查詢來指定要傳送的資料(W)**

使用這個選項可以寫入 SQL 查詢來操作或限制複製作業的來源資料。

說明(H) < 上一步(B) 下一步(N) > 完成(F) >>| 取消



SQL Server 匯入和匯出精靈

儲存並執行封裝
指出是否儲存 SSIS 封裝。

☒ 立即執行(E)

☐ 儲存 SSIS 封裝(S)

☒ SQL Server(Q)

☐ 檔案系統(F)

封裝保護等級(L):

機密資料以使用者金鑰加密

密碼(P):

再次輸入密碼(R):

說明(H) < 上一步(B) 下一步(N) > 完成(F) >>I 取消

可能出現的錯誤訊息

SQL Server 匯入和匯出精靈

檢閱資料類型對應
選取資料表，檢閱它的資料類型對應到目的地資料類型的方式，並且選取精靈處理轉換問題的方式。

資料表:

來源	目的地
! `books`	[dbo].[books]
! `bookstores`	[dbo].[bookstores]
! `orders`	[dbo].[orders]

資料類型對應:

來源資料行	來源類型	目的地資料行	目的地類型	轉換	錯誤時
! id	Double	id	int	<input checked="" type="checkbox"/>	使用全域
! bookname	VarChar	bookname	char	<input checked="" type="checkbox"/>	使用全域
! author	VarChar	author	char	<input checked="" type="checkbox"/>	使用全域
! price	Double	price	smallint	<input checked="" type="checkbox"/>	使用全域
! publisher	VarChar	publisher	char	<input checked="" type="checkbox"/>	使用全域

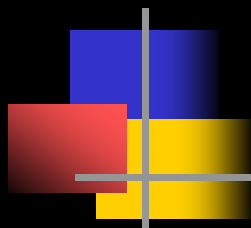
若要檢視轉換細節，請按兩下含有要轉換的資料行來源類型的資料列。

錯誤時 (全域)

交易時 (全域)

說明(H) < 上一步(B) 下一步(N) > 完成(F) >> 取消





本章結束
The End.