

Appendix

## C

## 全文檢索索引與搜尋

在資料庫中查詢資料，使用索引是提升效率的好方法。但索引一般都是建立在數值或是長度較短的字串欄位上，如編號欄位、姓名欄位（很少有人會用地址欄位建立索引），這麼做的原因，無非是要縮短更新索引所需的時間。

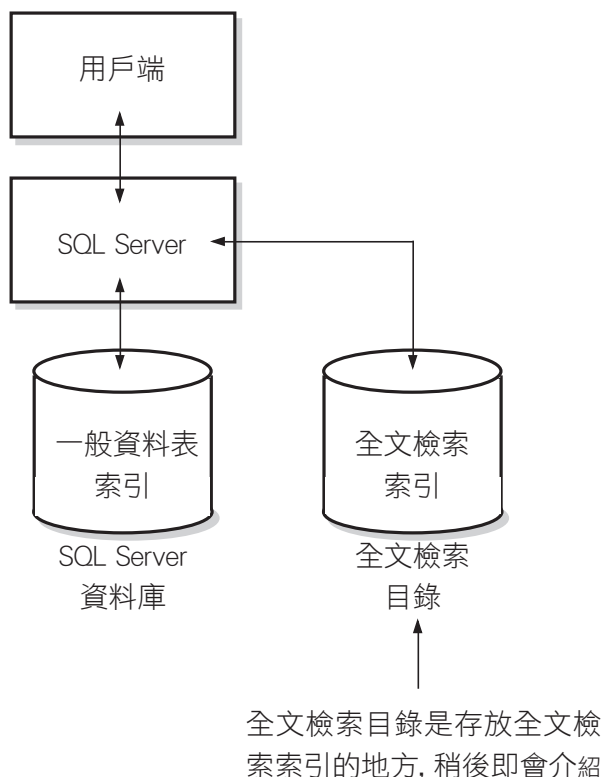
此外，如果是要針對字串中的某幾個字來查詢，例如要查詢包含 '忠孝' 的所有地址，那麼即使對地址欄做索引也沒用啊！因為只有針對地址由最前面開始的文字做查詢（例如 '台北市忠孝...'），才能使用以地址排序的索引。

可是難道這就表示，假如我們想要從長度較長的字串欄位中搜尋資料，就需捨棄效率慢慢來嗎？當然不是，本章要介紹的全文檢索索引功能，就是 SQL Server 專為這種需求所提出的解決方案。本章將使用**練習 CC** 資料庫為例說明，請依關於光碟中的說明，附加光碟中的資料庫到 SQL Server 中一起操作。

## C-1 全文檢索的架構

SQL Server 2016 的資料庫引擎已經內建全文檢索功能，不論是要建立**全文檢索索引**，或是要利用**全文檢索索引**搜尋資料，都可以直接透過資料庫引擎來完成工作。

和一般索引不同的是，**全文檢索索引**並不儲存在 SQL Server 的資料庫中，而是存放在**全文檢索目錄**中，所以當我們利用**全文檢索索引**來搜尋資料時，其運作流程和以往查詢資料時大不相同，請看下圖：



## C-2 全文檢索引目錄

在上一節提到，**全文檢索引目錄**是存放**全文檢索引**的地方。通常我們會先建立全文檢索引目錄，把存放地點準備好，下一步才是建立全文檢索引；之後，再利用全文檢索引從長字串欄位中快速搜尋所要的資料。這一節我們就先來看看如何建立全文檢索引目錄。

### 建立全文檢索引目錄

在動手建立全文檢索引目錄之前先提醒您，一個資料庫中可以建立多個全文檢索引目錄，而每個全文檢索引目錄則可存放多個全文檢索引；但是，每一個資料表只能建立一個全文檢索引（這部分下一節會詳細說明）。

當您在 SQL Server 2016 建立資料庫時，預設便會開啟資料庫的全文檢索引功能。不過在 SQL Server 2005 中，資料庫預設為關閉全文檢索引功能，所以如果將 SQL Server 2005 建立的資料庫檔案附加到 SQL Server 2016 時，必須先執行下面 T-SQL 語法，開啟該資料庫的全文檢索引功能：

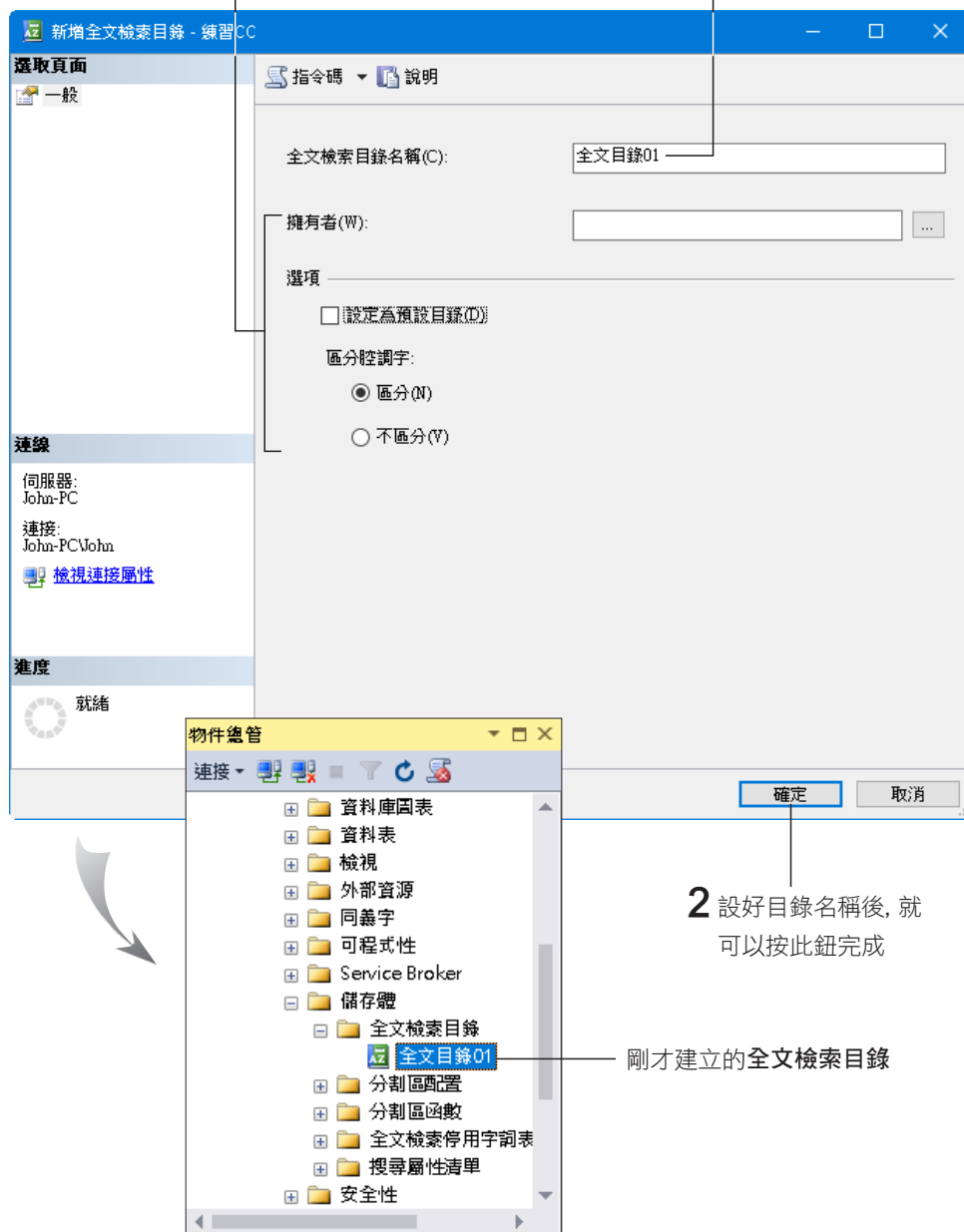
```
USE 練習 CC;           ← 使用練習 CC 資料庫
GO

sp_fulltext_database'enable'; ← 開啟全文檢索引功能
GO
```

現在我們來為**練習 CC** 資料庫建立一個全文檢索引目錄。請在物件總管窗格內**練習 CC** 資料庫的**儲存體/全文檢索引目錄**項目上按滑鼠右鈕，執行『**新增全文檢索引目錄**』，然後如下操作，建立新的全文檢索引目錄：

這些設定請保留預設值即可

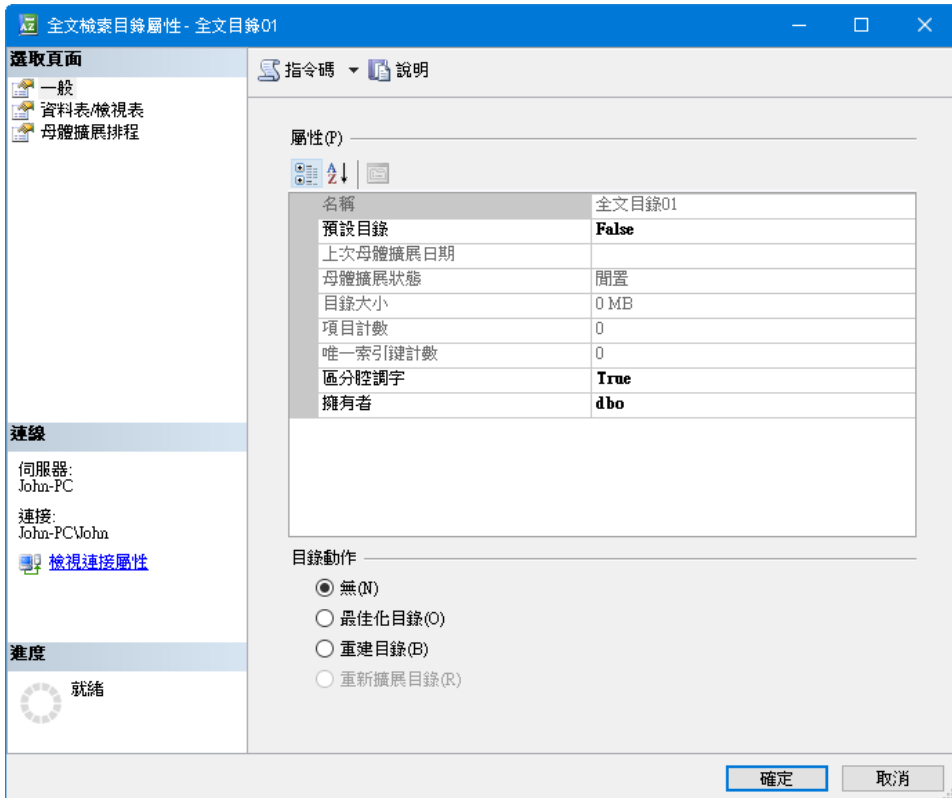
1 在此輸入全文檢索目錄的名稱



使用 create fulltext catalog 敘述亦可建立全文檢索目錄, 例如"create fulltext catalog 全文目錄 02"。

## 檢視全文檢索引目錄的屬性

請展開資料庫的儲存體/全文檢索引目錄項目，然後在欲檢視的全文檢索引目錄名稱上按滑鼠右鈕執行『屬性』命令，即可檢視該目錄的屬性內容：



屬性交談窗中有 3 個頁面：

- ◎ 一般頁面記載當初建立目錄的設定以及目錄現在的狀態。
- ◎ 資料表/檢視表頁面則會列示存放在此目錄中的所有全文檢索引，可惜目前還是空的。
- ◎ 母體擴展排程頁面則可檢視此目錄及其中全文檢索引的擴展排程，也就是 "更新" 的時間表 (有關排程的說明請參閱 C-4 節)。

## 刪除全文檢索目錄

若要刪除某個全文檢索目錄，請先切換到資料庫的**儲存體/全文檢索目錄**項目，選取欲刪除的目錄名稱，然後按滑鼠右鈕執行『**刪除**』命令即可。

## C-3 建立全文檢索索引

接著我們來談如何建立全文檢索索引。首先介紹為資料表建立全文檢索索引時，所需具備的條件以及一些限制：

- ◎ 要建立全文檢索索引的資料表，必須設有 Primary Key，或是具備唯一性的 Key 才行，而且不得為多個欄位組合的 Key。
- ◎ 每個資料表只能建立一個全文檢索索引，但全文檢索索引中可包含多個欄位。
- ◎ 包含在全文檢索索引中的欄位，即是您想進行全文搜尋的欄位，如地址欄位、備註欄位... 等。這些欄位只能是 char、varchar、nchar、nvarchar、text、ntext、image、xml 和 varbinary (含第 7-7 節介紹的 FILESTREAM 資料) 等型別的欄位。

### TIP

有關為 varbinary、image 型別的欄位建立全文檢索索引，請參閱 C-6 節。

## 建立全文檢索索引

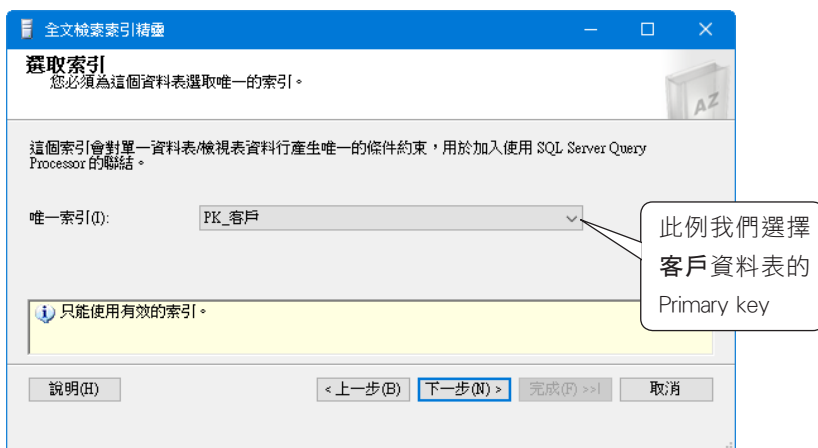
了解建立全文檢索索引的條件及限制後，下面我們就以**練習 CC** 資料庫的**客戶**資料表為例，說明建立全文檢索索引的操作方法。

請切換到**練習 CC** 資料庫的資料表項目，選取**客戶**資料表，然後按滑鼠右鈕執行『**全文檢索索引/定義全文檢索索引**』命令，接著會啟動全文檢索索引精靈來帶領我們：

**Step1** 第一個畫面是精靈的介紹，看過後就按下一步鈕繼續。



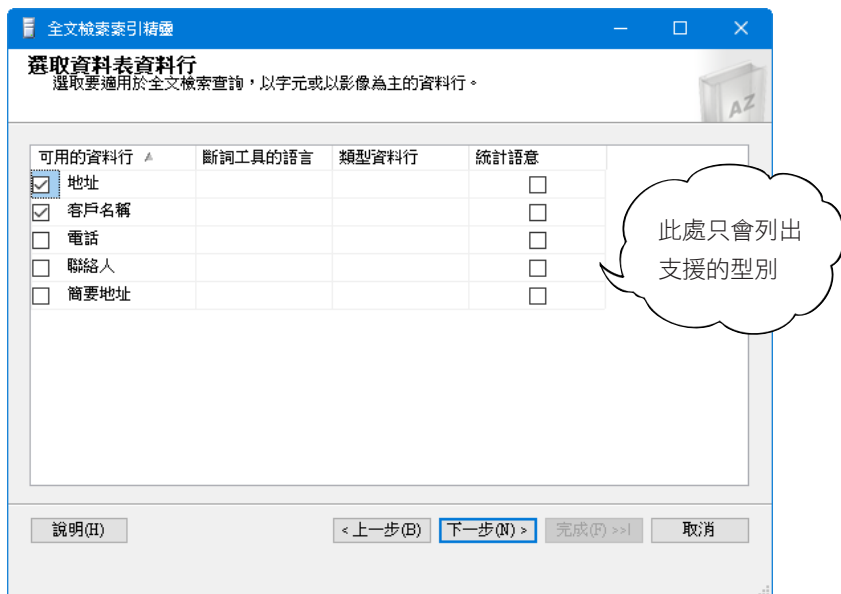
**Step2** 從資料表中選出一個具有唯一性的 key，作為全文檢索引的唯一索引鍵。這個 key 的欄位越小越好，如此可減少全文檢索引所佔用的儲存空間。



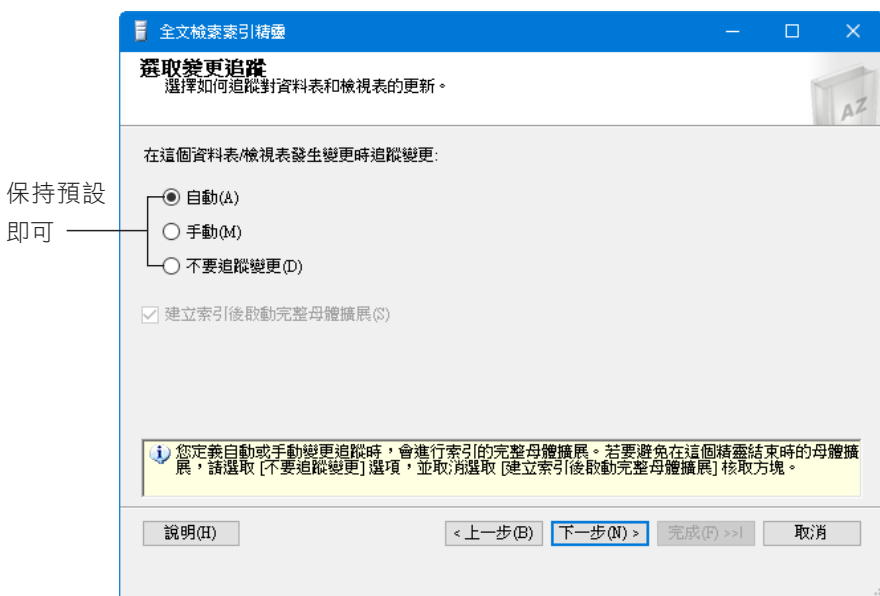
**TIP**

如果步驟 2 直接顯示 "沒有可用的唯一索引"，表示資料表中沒有唯一的 key，或 key 包含 2 個以上欄位，這時只好回去修改資料表之後再重來了。

### Step3 勾選要加入全文檢索引中的欄位，可以選擇多個欄位。



### Step4 設定變更追蹤。





**Step5** 選擇全文檢索引要放在哪一個全文檢索引目錄中。

此例選擇放在上一節建立的全文檢索引目錄中

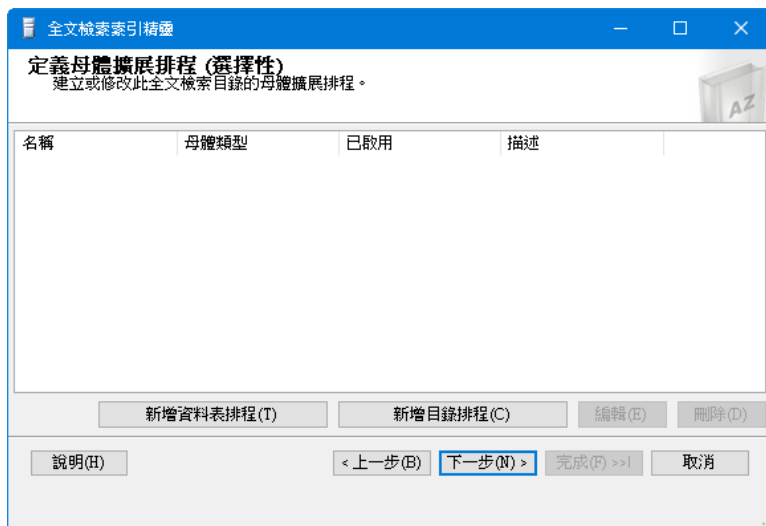
若勾選此項，可立即新增一個全文檢索引目錄來存放

### 選擇全文檢索引目錄的建議

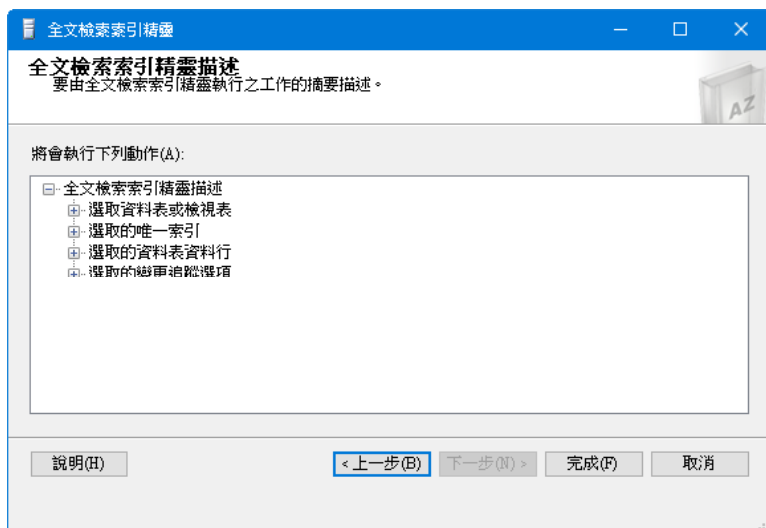
雖然說，一個全文檢索引目錄中可以存放多個全文檢索引索引，但是如果全文檢索引索引的內容相當大（例如有幾百萬筆記錄時），最好考慮將這個全文檢索引索引單獨放在一個全文檢索引目錄中，以避免影響到其它全文檢索引索引的效率。

此外，我們還可以根據每個全文檢索引索引需要的更新時間來替它們分類，將更新時間相同的全文檢索引索引放在同一個全文檢索引目錄中。

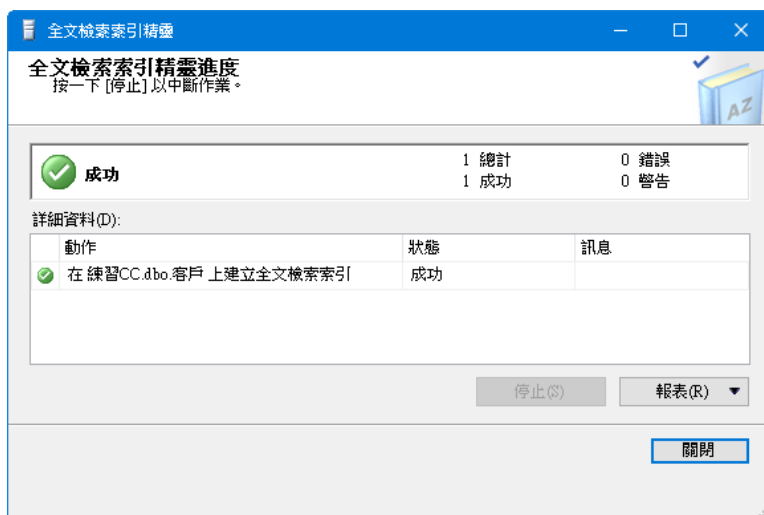
**Step6** 選擇或建立全文檢索引及全文檢索引目錄的擴展排程，這部份我們留待下一節再詳細介紹，請按下一步鈕繼續：



**Step7** 設定完畢，按下完成鈕結束吧！



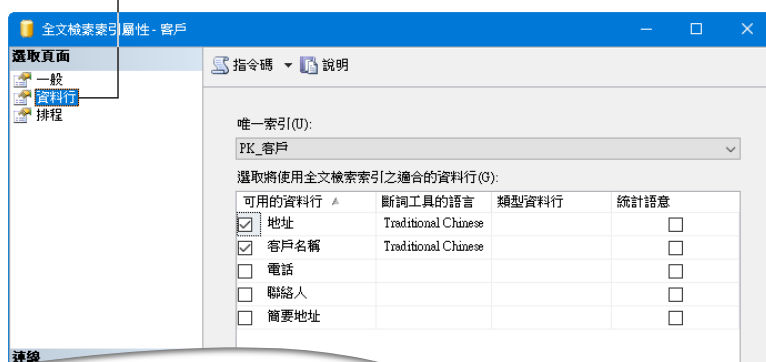
**Step8** 接著便會開始依照剛才的設定建立全文檢索引，完成後會出現如下的交談窗來通知我們，按下**關閉**鈕即可。



## 檢視資料表的全文檢索引設定

我們可以選取資料表然後按滑鼠右鈕執行『全文檢索引/屬性』命令，開啟全文檢索引屬性交談窗來檢視該資料表的全文檢索引設定：

請切換到**資料行**頁面



## 編輯與刪除全文檢索引

假如在建立全文檢索引時，有地方設錯了，例如少勾了一個欄位、放錯全文檢索引目錄... 等等，可以事後在要修改的全文檢索引目錄上按滑鼠右鈕執行『**屬性**』命令開啟**全文檢索引屬性**交談窗，然後切換到**資料表/檢視表**頁面，您就可在選取的物件屬性窗格中修改。

如果要刪除資料表的全文檢索引，則在該資料表上按滑鼠右鈕執行『**全文檢索引/刪除全文檢索引**』命令，此時會出現訊息讓我們再次確認，按下**確定**鈕就可刪除。

## C-4 擴展全文檢索引目錄與索引

所謂“**擴展**”，就是更新的意思，目的是讓全文檢索引目錄及全文檢索引能夠反映最新的狀況。新建立的全文檢索引目錄或全文檢索引，我們都會先擴展一次再開始使用。而若資料表的記錄內容有增減、或全文檢索引的欄位有變動，我們也會進行擴展的動作。

### 擴展的對象與作業類型

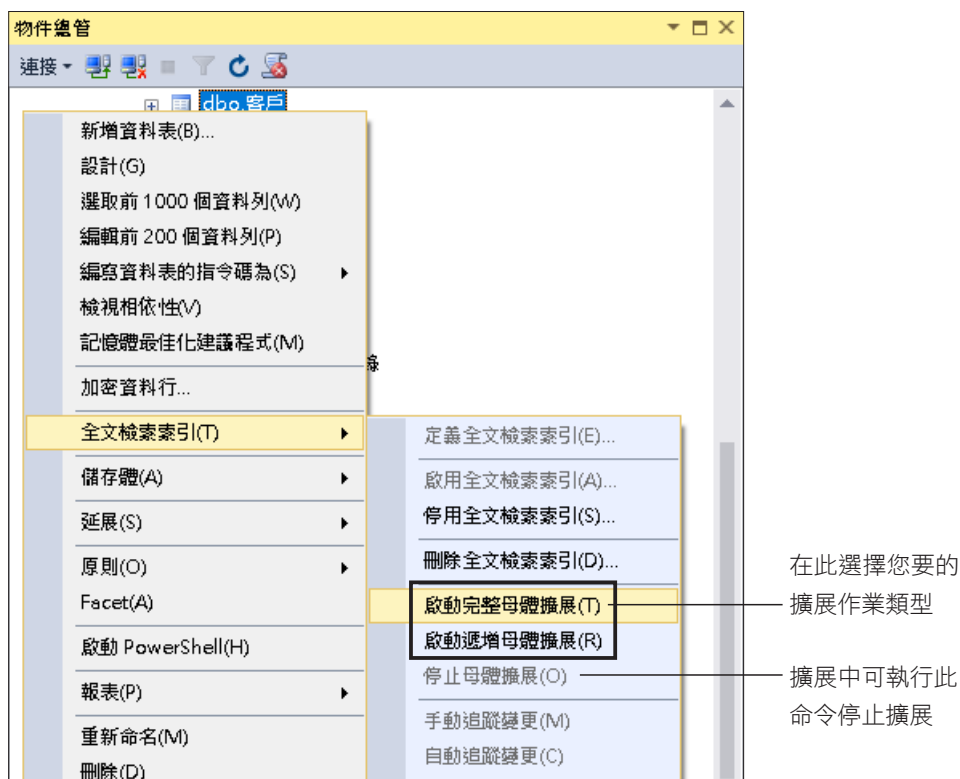
擴展的對象可以是全文檢索引或全文檢索引目錄，擴展全文檢索引其實就是同步擴展該目錄中所有的全文檢索引。至於擴展還可分成下列 3 種作業類型：

- ◎ **完全擴展**：全部更新重建。通常發生在初次擴展全文檢索引或全文檢索引目錄的時候，以後就可以用**遞增擴展**或**變更追蹤擴展**來節省擴展的時間。
- ◎ **遞增擴展**：僅調整自上一次擴展後有變動的地方。要用這種方式，則建立全文檢索引的資料表中必須有 timestamp 型別的欄位，如此才能判斷是否要更新。
- ◎ **變更追蹤擴展**：平時將全文檢索引的變更維護於記錄檔之中，待適當時機再將變更填入全文檢索引中。全文檢索引目錄無法使用這種方式來擴展。

## 手動擴展全文檢索引

進行擴展有兩種方式：一種是**手動擴展**；另一種是**建立排程**，待時間一到就自動進行擴展。這裏先介紹**手動擴展**的方式。

如果想立即擴展某個全文檢索引，則選取該全文檢索引所在的資料表，然後按滑鼠右鈕後，將滑鼠移至『**全文檢索引**』項目，如下操作：



若全文檢索引要使用**變更追蹤擴展**，但在 C-9 頁的第 4 步驟卻設定為**不要追蹤變更**，則必需先啟動**變更追蹤**功能，以記錄全文檢索引的任何變動：請選取資料表，然後按滑鼠右鈕執行『**全文檢索引/自動追蹤變更**』命令就可以啟動**變更追蹤**功能。

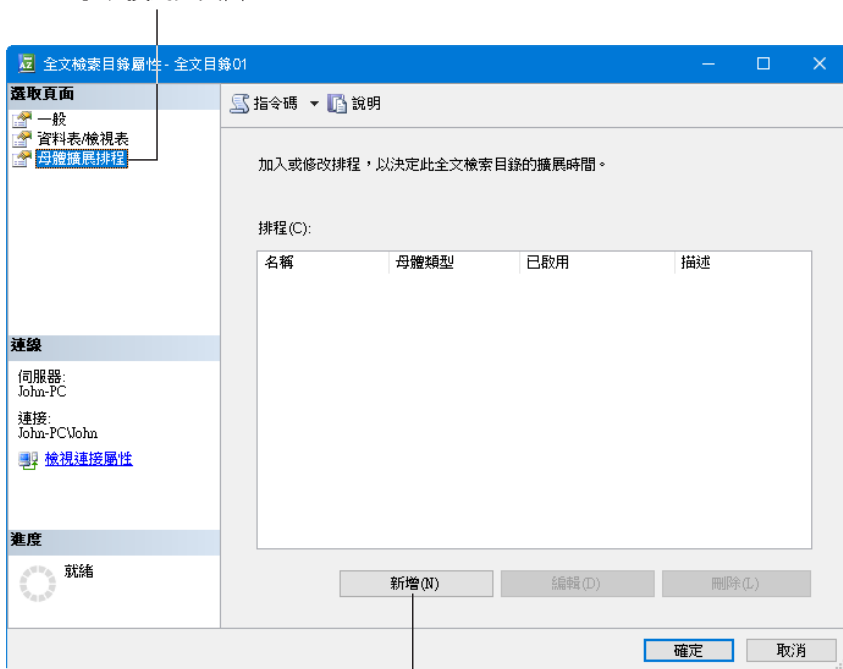
一段時間後，若要將變更追蹤所記錄的變動反映到全文檢索引上，則選取資料表，然後按滑鼠右鈕執行『全文檢索引/套用追蹤變更』命令即可。

## 建立排程自動擴展全文檢索引目錄

由於全文檢索引所包含的欄位都是長度較長的字串欄位，擴展時總是要花費不少時間，所以通常我們會建議在比較沒有人使用 SQL Server 的時候，才進行擴展工作。當然，不會要各位在半夜爬起來親手啟動擴展；而是，我們可以建立排程，只要時間一到就自動進行擴展，完全不會打擾到睡眠時間，而且白天的工作也不致因為擴展而中斷了。

請切換到資料庫的儲存體/全文檢索引目錄項目，選取欲設定的目錄名稱，如全文目錄 01，然後按滑鼠右鈕執行『屬性』命令：

1 切換到此頁面



3 給排程取個名稱

4 選擇排程的類型, 此例我們選擇**重複執行**

名稱(N): 練習CC目錄排程

排程類型(S): 重複執行

已啟用(B) ☒

僅執行一次

日期(D): 2016/12/18 時間(T): 上午 12:00:00

頻率

發生於(C): 每日

重複頻率(R): 1 天

每日頻率

☒ 執行一次於(A): 下午 12:29:55

☐ 重複執行於每(V): 1 小時

開始時間(I): 下午 12:29:55

結束時間(G): 下午 12:29:55

持續時間

開始日期(D): 2016/12/18

☐ 結束日期(E): 2016/12/18

☒ 沒有結束日期(O):

摘要

描述(P): 每天於 下午 12:29:55 發生。排程會從 2016/12/18 開始使用。

確定 取消 說明

勾選此項才會  
啟用此排程

5 這裡可以設定執行的頻率以及持續的時間

6 按**確定**鈕繼續

全文檢索目錄屬性 - 全文目錄01

選取頁面

一般 資料表/檢視表 母體擴展排程

指令碼 說明

加入或修改排程, 以決定此全文檢索目錄的擴展時間。

排程(C):

名稱	母體類型	已啟用	描述
全文目錄01.練習CC目錄排程	目錄 - 最佳化	<input checked="" type="checkbox"/>	每天於 下午 12:2...

進度

就緒

新增(N) 編輯(E) 刪除(L)

確定 取消

建立一個索引  
目錄排程了7 按**確定**鈕完成設定

## C-5 使用全文檢索索引搜尋資料

假設各位所需的全文檢索索引都設定好了，也都擴展過了，那現在我們來看如何透過全文檢索索引來搜尋資料。

進行全文檢索搜尋時，同樣是透過 `SELECT` 敘述，但在設定搜尋條件時有點小小的變化。假如搜尋條件是要放在 `WHERE` 子句中做判斷，那麼可用 `CONTAINS` 或 `FREETEXT` 這兩個述詞。`CONTAINS` 主要是針對單字或句子做查詢，`FREETEXT` 則會先將指定的句子分解為一個個單字，然後再查詢這些單字。

如果搜尋條件是要放在 `FROM` 子句中以便直接傳回搜尋到的資料，則可使用 `CONTAINSTABLE` 或 `FREETEXTTABLE` 函數。底下我們就針對這 4 種方式分別來說明。

### 使用 `CONTAINS` 搜尋

`CONTAINS` 使用於 `WHERE` 子句中，可用來搜尋資料中的一個單字，例如 "Flag" 或 "忠"、一個句子，例如 "Flag Publishing Company" 或 "忠孝東路"、衍生字，例如 "Publish" 與 "Publishing"、以及位置相近的字串，例如 "忠孝 X 路" 的 "路" 是在 "忠孝" 附近。

`CONTAINS` 的基本語法如下：

```
CONTAINS ( column , 'contains_search_conditions' )
```

↑  
指定欲搜尋的欄位名稱，此欄位必須已加入全文檢索索引中。若用\*，表示要搜尋全文檢索索引中的所有欄位

↑  
設定搜尋的條件，因為是字串，所以必須用單引號括起來



CONTAINS 述詞的搜尋條件有 5 種類型，底下我們就一一為您介紹。

- ◎ **簡單的形式**：例如單字、句子以及加上邏輯運算，如 AND、OR、AND NOT。

```
SELECT *
FROM 客戶
WHERE CONTAINS (地址, '師大路')
```

← 找出地址欄位中有師大路字串單字的記錄

```
SELECT *
FROM 客戶
WHERE CONTAINS (地址, ' "Parkway Fremont" ')
```

← 搜尋條件中的字串若包含空白或標點符號，須再以雙引號括起來

```
SELECT *
FROM 客戶
WHERE CONTAINS (地址, ' "Parkway Fremont" ' OR "師大路" ')
```

← 加上邏輯運算的搜尋條件

當搜尋條件中使用多個 AND、OR、AND NOT 時，可以用**小括號**將要優先運算的部份括起來，否則會以 NOT > AND > OR 的順序運算 (NOT 最先)。不過 NOT 必須接於 AND 之後才行，不可單獨使用或和 OR 共用。

### 使用句子查詢時

若搜尋條件是一個句子時，進行搜尋時會依句子中出現的單字依序做比對，必須每個字都符合才會找出來，但在比對時會將半型逗號及英文的無義助詞（如 a, an, the 等冠詞）都忽略掉。例如 CONTAINS (備註, ' "a computer failure" '), 則 "This is a computer, faulure to work." 也會被找出來。

- ◎ **英文字首加上萬用字元**：例如只知道某單字的開頭是 Pa，則可以用 **Pa\*** 為條件來搜尋，這個條件需用**雙引號**括起來。

```
SELECT *  
FROM 客戶  
WHERE CONTAINS (地址, 'Pa*')
```

**TIP**

這種類型的條件一定要指明字首字元，如 "Pa\*"，不可以用 "\*ar\*" 或 "\*way\*" 這類的條件。

假如我們的搜尋條件是一個句子加上萬用字元，則句子中的每個單字均會當成字首做比對。例如 CONTAINS (備註, ' "local wine\*" '), 則"local winery"、"locally wined and dined" 等都會找出來。

- ◎ **位置相近的字串**：使用記錄中位置相近的兩個字串來搜尋，例如要搜尋美國加州的客戶，但又不確定完整地址時，就可使用這種類型的條件來尋找。

```
SELECT *  
FROM 客戶  
WHERE CONTAINS (地址, ' "CA" NEAR "USA" ')
```

**TIP**

請注意，若要使用英文字串，必須輸入完整的單字或在字尾加上萬用字元，例如 "Flag" NEAR "Company"，或是 "Flag" NEAR "Com\*"；若是使用 "Flag" NEAR "pany"，那就可能找不到了。

另外，我們也可以指定多個位置相近的字串，例如 ' "Flag" NEAR "Pu\*" NEAR "Comp\*" '。在搜尋時，這些字串間的距離越相近，則符合的程度越高。

- ◎ **衍生字**：英文單字有現在式、過去式、未來式、分詞、以及單複數等不同的形式（中文就沒這麼麻煩了）。在 CONTAINS 中可以用一個單字來找尋資料中所有包括此單字衍生字的記錄，例如搜尋條件是 drive，則包含有 drives、drove、driving、driven 等衍生字的記錄都會被找出來。

```
SELECT *
FROM 客戶
WHERE CONTAINS (客戶名稱, 'FORMSOF (INFLECTIONAL, flag)', LANGUAGE 1033 )
```

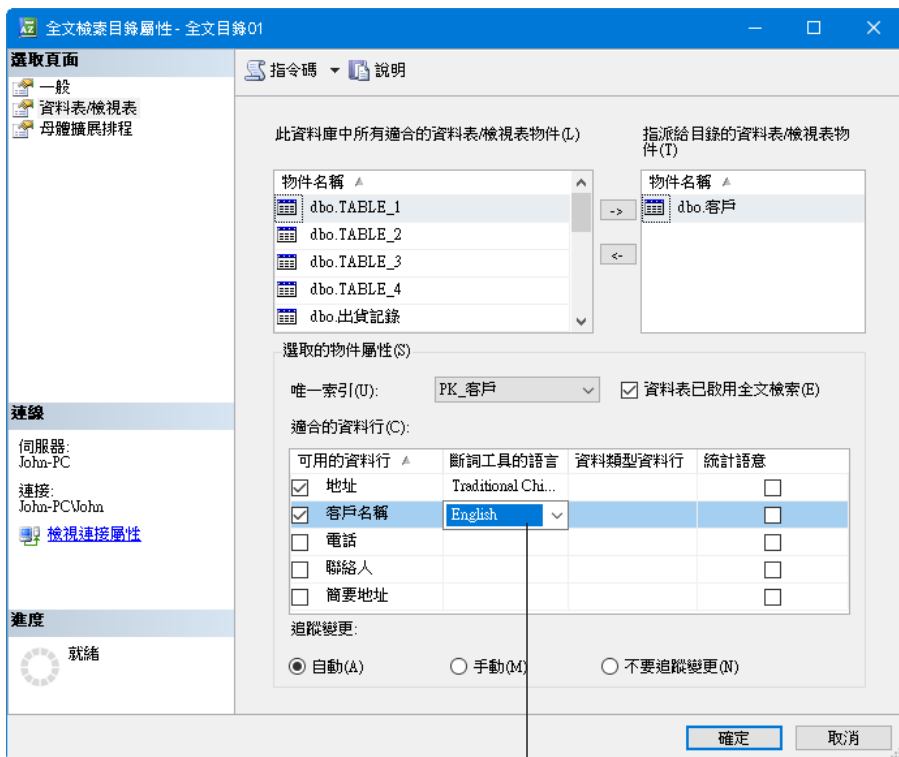
CONTAINS 括號中第 3 個參數 "LANGUAGE 1033" 是用來表示進行衍生字、斷詞（見下頁）等處理時所使用的語言，"LANGUAGE" 後面要接的參數，必須使用 sys.syslanguages 系統檢視表中所列的 langid 數值或 alias 名稱。例如 1033 就是美式英語的 langid，而其 alias 則為 'English'。

### 全文檢索索引欄位的語言設定

若將上例 CONTAINS 括號中第 3 個參數 "LANGUAGE 1033" 刪除再執行一次，將會出現錯誤訊息。這是因為當初在為客戶資料表建立全文檢索索引時，並沒有要各位設定欄位的語言，在這種情形下，會直接以 SQL Server 的語言版本來設定，例如我們是使用中文版，欄位的語言便設為 **Traditional Chinese**，亦即以中文的規範來定義文字分界（斷字）以及動詞變化。因為中文沒有衍生字，使得衍生字這種搜尋條件對中文沒有作用。

以前文的範例為例，請各位切換到資料庫的**儲存體/全文檢索目錄**，在全文目錄 01 上按滑鼠右鈕，執行『**屬性**』命令後切換到**資料表/檢視表**頁面，將**客戶名稱**欄位的**斷詞工具的語言**改為 **English**，然後擴展全文檢索索引，再執行未指定語言的衍生字查詢，就會有結果出來了。

接下頁



改為 English

### TIP

試過之後，請務必要將客戶名稱欄位的斷詞工具的語言改回 **Traditional Chinese**。

當然，欄位的斷詞工具的語言可不是胡亂設的，該欄位中的資料是什麼語言，就要設為那個語言才對。假如欄位資料包含多種語言，或是不支援的語言時，請設為中立語言 **Neutral**。

- ◎ **給字串加權計分**：當搜尋數個字串時，我們可以給字串不同的加權值 (weight, 須介於 0.0 至 1.0 之間)。

```
SELECT *
FROM 客戶
WHERE CONTAINS ( 地址,
                'ISABOUT (USA weight (0.8),
                           重慶南路 weight(0.5),
                           師大路 weight(0.1))'
                )
```

## 使用 FREETEXT 搜尋

FREETEXT 會將一個句子中的每個單字分開來搜尋，其搜尋結果和用 CONTAINS 搜尋比起來比較不精確。FREETEXT 的語法為：

```
FREETEXT (column | *, 'freetext_string', LANGUAGE lang_term )
```

例如：

```
SELECT *
FROM 客戶
WHERE FREETEXT ( *, 'Fremont CA USA', LANGUAGE 1033 )
```

FREETEXT 會將條件中的字串拆解成 "Fremont"、"CA"、"USA"，因此上述查詢只要有符合 "Fremont"、"CA"、"USA" 其中之一的記錄就會找出來。

## 使用 CONTAINSTABLE 函數搜尋

CONTAINSTABLE 函數會傳回符合條件的多筆記錄，而記錄中只包含 KEY 及 RANK 兩個欄位。KEY 欄位用來放置全文檢索引的唯一索引鍵（就是建立全文檢索引時指定的具有唯一性的 key）；而查詢完畢後，每一筆找到的記錄會依查詢符合的程度而產生一個 RANK 值，存放在 RANK 欄位中。

CONTAINSTABLE 函數的語法如下：

```
CONTAINSTABLE (table, { column | * }, 'contains_search_conditions'
               [ , top_n_by_rank ]
```

語法其實和 CONTAINS 差不多，只不過增加了下列兩個參數：

- ◎ **table**：指定全文檢索引所在的資料表名稱。
- ◎ **top\_n\_by\_rank**：指定只傳回 n 個最高順序的記錄，依遞減順序傳回。

CONTAINSTABLE 的搜尋條件與 CONTAINS 一樣有 5 種類型，請參考前面的說明，此處不再重複。但 CONTAINSTABLE 是用在 FROM 子句中，而且如果想要顯示出原資料表的欄位內容，還必須與原資料表做 JOIN，我們直接用範例來說明比較清楚：

```
SELECT 客戶編號, 地址, My_Table.*
FROM 客戶
      INNER JOIN
      CONTAINSTABLE (客戶, 地址, ' 羅斯福 OR 一段 OR USA ' )
      AS My_Table
ON 客戶編號 = My_Table.[KEY]
ORDER BY My_Table.RANK DESC
```

**TIP**

因為 KEY 是 SQL 的關鍵字，此例我們是用來指名 KEY 欄位，所以必須加上中括號。


上述範例中由 CONTAINSTABLE 函數定義了一個 My\_Table 資料表，並且 My\_Table 的 KEY 欄位與客戶資料表的客戶編號欄位要相同。然後使用 ORDER BY 子句來排列 RANK 欄位，其結果如下所示：

	客戶編號	地址	KEY	RANK
1	9	270 Bayside Parkway Fremont, CA 94538, USA	9	64
2	2	台北市敦化南路一段1號	2	48
3	5	台北市重慶南路一段57號	5	48

再來看一個使用加權方式搜尋的範例：

```
SELECT 地址, My_Table.RANK
FROM 客戶
    INNER JOIN
        CONTAINSTABLE (客戶, 地址, ' ISABOUT ( 南京 WEIGHT(0.1),
            一段 WEIGHT(0.4), 敦化 WEIGHT(0.7) ) ' , ③)
    AS My_Table
ON 客戶編號 = My_Table.[KEY]
```

只傳回前面 3 筆  
符合條件的記錄



	地址	KEY	RANK
1	台北市中山北路六段88號	6	70
2	270 Bayside Parkway Fremont, CA 94538, USA	9	70

依符合的程度再經加  
權, 可得到 **RANK** 值

## 使用 FREETEXTTABLE 函數搜尋

FREETEXTTABLE 函數也只傳回 KEY 與 RANK 欄位, 語法與 FREETEXT 類似, 但多了 table 和 top\_n\_by\_rank 兩個參數：

```
FREETEXTTABLE (table, {column | *}, 'freetext_string'
    [ , top_n_by_rank ] )
```

例如：

```
SELECT 地址, My_Table.*
FROM 客戶
    INNER JOIN FREETEXTTABLE (客戶, 地址, '88 Bayside', LANGUAGE 1033)
    AS My_Table
ON 客戶編號 = My_Table.[KEY]
```



	地址	KEY	RANK
1	台北市中山北路六段88號	6	70
2	270 Bayside Parkway Fremont, CA 94538, USA	9	70

## C-6 varbinary、image 型別欄位的全文檢索搜尋

全文檢索也可以對二進位資料 (含 FILESTREAM) 進行全文檢索，前提是必須在另一個欄位中儲存該資料的類型，例如 Word、Excel、HTML (網頁) 等。

### 支援的檔案類型

我們可以將檔案直接存入 varbinary、image 型別的欄位中，不過，若要進行全文檢索搜尋，SQL Server 預設支援 50 種檔案類型，右表列出幾種常用類型：

#### TIP

您可以執行『SELECT \* FROM sys.fulltext\_document\_types』來列出所有支援的檔案類型。

檔案類型	副檔名
Word 文件	doc、dot
Excel 活頁簿	xls
PowerPoint 簡報	ppt
網頁	htm、html
純文字檔	txt
XML 檔	xml
WordPad 文件	rtf

此外，資料表還要再設置一個字串型別的欄位，用來記錄存放在 varbinary、image 欄位中的檔案副檔名。例如 varbinary、image 欄位中存放的是 Word 文件，此欄就要輸入 doc。當進行搜尋時，SQL Server 便會根據這個欄位使用適當的篩選器來萃取資料。

#### TIP

請注意，記錄副檔名的欄位，除了副檔名外不能有任何的空白，所以我們建議此欄位最好設為 varchar 型別。

在**練習 CC** 資料庫我們已經依照上述的條件建立了**應徵者**資料表，其內容如下，待會兒就利用這個資料表來為您示範：



						varbinary 型別的欄位	記錄副檔名的欄位
	編號	姓名	性別	地址	電話	自傳	文件類型
1	1	戚莉秀	女	台北市新興路一段10-1號10樓	(02)22796641	0xD0CF11E0A...	DOC
2	2	吳玉婷	女	台北市民有街150巷2號1樓	(02)23745877	0xD0CF11E0A...	DOC
3	3	林雅君	女	台北市信義路四段300號2-2樓	(02)29312098	0xD0CF11E0A...	DOC
4	4	黃彥政	男	台北市博愛路二段130號3樓	(02)27778344	NULL	NULL
5	5	黃家瑋	男	台北市忠信路65號3樓	(02)27276535	NULL	NULL
6	6	汪羽新	男	台北市漢生路一段24號	(02)23916643	NULL	NULL
7	7	陳弘順	男	台北市復興北路15-1號	(02)23939800	NULL	NULL
8	8	賴培娥	女	台北市育英街130號5樓	(02)27376544	NULL	NULL
9	9	王惠馨	女	台北市復興北路65號	(02)27743998	NULL	NULL
10	10	陳曉萍	男	台北市忠孝西路一段3號4樓	(02)23928890	0xD0CF11E0A...	DOC

## 建立包含 varbinary、image 欄位的全文檢索引

首先我們要為應徵者資料表建立全文檢索引，方法和 C-3 節介紹的一樣，但是當進行到選擇資料表資料行交談窗的時候，請如下設定：

### 1 勾選 varbinary 或 image 型別的欄位



接著就請繼續往下進行，完成全文檢索引的建立。

## 全文搜尋 varbinary、image 欄位的資料

假設各位都已建好應徵者資料表的全文檢索引，同時也擴展過了（因為資料較多，可能要等一會兒），那現在我們就用上一節介紹的方法，來搜尋存放在 varbinary 或 image 欄位中的檔案資料。

```
SELECT 編號, 姓名, 性別  
FROM 應徵者  
WHERE CONTAINS (自傳, 'xml')
```



	編號	姓名	性別
1	10	陳曉萍	男

```
SELECT 編號, 姓名, 性別  
FROM 應徵者  
WHERE CONTAINS (自傳, '"SQL Server" OR HTML' )
```



	編號	姓名	性別
1	1	戚莉秀	女
2	2	吳玉婷	女
3	3	林雅君	女
4	10	陳曉萍	男

### TIP

根據筆者的測試，在二進位資料搜尋中文字詞時，可能會發生搜尋『興趣』可以查到結果，但是搜尋『興』卻找不到結果的問題。