



ASHTON·TATE™

dBASEIII PRACTICE

—***FUNDAMENTAL***

—***“PersonNEL Management”***

—***Example***

以“員工管理
系統”實例學習
dBASEIII PLUS
(演練篇)

整圖電腦軟體研究開發部門 編譯



全華科技圖書股份有限公司 總經銷

序 言

本書以“員工管理系統”為例，介紹資料庫建立的步驟，並從許多不同的觀點來展示，編譯或修改資料，至寫出一個完整的系統。根據實際上進行設計的每一個步驟，將 dBASE所提供的功能逐一介紹。

為使不同層次之讀者均能以最快的速度進入情況，我們將本書分為“進階篇”及“演練篇”，其中“演練篇”是以 Interpret方式操作的，而“進階篇”則以完整的程式進行系統的建立。本書適合完全不懂 dBASE 或對 dBASE不熟基礎不穩的讀者閱讀，更可作為補習班的教材或學校課本之用。

瑩圖電腦 R&D
七十六年五月

DBASE III PLUS 1.0 與 1.1 版之間的改進

@ ...CLEA TO

在 dBASE III PLUS 1.0 版中，雖然應用 @... TO命令可以在螢幕上畫出一條單獨的線，但是，應用 @... CLEAR TO卻不能從螢幕上清除一條單獨的線。@... CLEAR TO 最低限度只能清除兩行。

在 dBASE III PLUS 1.1 版中，可以使用 @... CLEAR TO清除螢幕中一條單獨的線。

@... GET/RANGE

在 dBASE III PLUS 1.0版本，如果您是在 STATUS ON的情況之下，cebar 清除 RANGE所顯示出來的錯誤訊息時，此訊息並不會消失。

現在，在 STATUS ON的情況之下，已可用 Spacebar 清除 RANGE所發出的錯誤訊息。

? / ??

在以前的版本，任何嘗試由工作區顯示一個備忘錄欄位不同於目前所選的工作區會產生錯誤訊息 "LIN Recoghized command Verb or phrase" 。

在 dBASE III PLUS 中，你可以由一個工作區 (work area) LIST 或 DISPLAY一個備忘錄欄位 (memo field) 而不是 SELECTed 的工作區。但你仍然無法使用 ?或??來拿取備忘錄欄位而不發生錯誤。

@..GET...READ

1. 在以前的版本，當你把數字放在數值欄位的小數點位置的右邊時，敲入小數點有時會使游標跳回欄位的開頭位置。這問題的發生是使用 @...GET...READ, APPEND, BROWSE, CHANGE, 或EDIT的資料檔案使用一個或多個索引檔 (index file)。

在 Developer's Release及 dBASE III PLUS, 當你敲入小數點後，你便可正確地輸入十進位數字。

2. 在以前的版本，如果你用 PICTURE子句 "999.99"來用 GET命令讀取一個記憶體變數，而且你並未用 PICTURE子句所定義的小數位來重新設定記憶體變數時，GET 命令並不會顯示十進位數字。如果你輸入一個小數點，你將會離開 GET命令。

在 Developer's Release及 dBASE III PLUS, READ 命令允許 PICTURE 子句決定在 GET中的小數位數。而且與記憶體變數是否設定無關。

@...SAY...GET

在以前的版本，如果你先用 @...SAY...GET顯示文字於螢幕，接著用@...SAY...GET 在同一行且不清除原先的字幕，先前的文字字元 (character)將在 SAY及 GET之間顯示出來。例如

```
STORE "123" TO mem
@10,0 SAY "XXXXXXXXXX"
@10,0 SAY "A" GET mem
```

顯示如下：

```
AX123XXXXXX
```

——— 這應該不會被顯示出來

在 dBASE III PLUS 時，@...SAY...GET命令將如所預期地清除SAY與 GET之間的螢幕。

APPEND

在以前的版本，你能在資料記錄中前後移動，而且修改資料，如果使用的資料檔有開索引檔，那麼當你修改索引檔時，索引檔的資料不會自動更新，中須再用 REINDEX來更新索引檔。

在 Developer's Release及 dBASE III PLUS,以這種方法對主要欄位的改變會反映給索引，而且當你離開 APPEND 後不再需要用 REINDEX。

APPEND FROM

如果你 APPEND FROM 一個DELIMITED的檔案，該檔的數值欄，尾隨一個空白，且空白在','之前，結果'0'會出現在數值欄裡，很明顯的空白被當成另一個以'0'為值的欄位。

在以前的版本，如果你 APPEND FROM一個包函在數值欄位後且在逗號之前在空白格的定界 (delimited)檔案時，在數值欄位分界的數值之間的空白格並不與 APPEND FROM功能衝突。

APPEND FROM SDF

APPEND FROM SDF 自動假設原始文字檔案的記錄能以 "回返" /跳行 (ODH OAH)秩序作為終結。在以前的版本，如果這次序相反(OAH ODH)，dBASE III 會將 "回返" (carriage return)之後的字元忽略，且目的資料庫檔案中第一個記錄之後的每一個記錄的第一個字元(character)都會被忽略。

在 Developer's Release及 dBASE III PLUS,回返及跳行的次序相反並不影響 APPEND FROM SDF。

ASSIST

在 dBASE III PLUS 1.0 版中，使用定位功能表中的定位選擇對一個資料錄實行全螢幕的編輯時，不可以繼續作其他的選擇。當你從全螢幕的操作中返回時，因不再有反白區，進而不能存取。於 dBASE III PLUS 1.1版中，在全螢幕編輯操作之後，可以繼續定位功能表的選擇。

1. 在以前的版本，如果你在 ASSIST 命令中做選擇，它允許你在資料庫檔案中使用 FOR/WHILE子句（AVERAGE, DISPLAY, REPORT, 等等），且有27個欄位。你不能用下移箭頭的 FOR/WHILE子句拿取這組欄位的最後幾個欄位。在螢幕上顯示時除了最後那組欄位外，一切都很好。最後那組欄位會在下個操作螢幕上顯示。

在 Developer's Release及 dBASE III PLUS,經由 ASSIST 命令可拿取所有的欄位資料，而管它是在此資料庫檔案中的第幾個欄位。

2. 在以前的版本，如果 SET INTENSITY被關閉時，你不能在ASSIST 命令中作選擇。這是因為 ASSIST 必須在反白的映像 (reverse video)才能操作。

在 Developer's Release及 dBASE III PLUS, SET INTENSITY的狀態 (status) 不會影響 ASSIST 的運作。

AVERAGE / SUM

1. 在以前的版本，不論在螢幕上或列印時，SUM 及 AVERAGE都會在命令終結時加一個額外的跳行 (linefeed)。

在 Developer's Release及 dBASE III PLUS, SUM及 AVERAGE在終結處送一個跳行。

2. 在以前的版本，如果 SUM TO 的結果設定到一個公用的變數會有錯誤的結果，想要將宣告為 PUBLIC 的記憶體變數加起來時會有錯誤的結果。在 SUM命令之前先重新起始變數的話，有時會更正此錯誤，但並不全是如此。另外，在 SUM命令中可使用運算式(expression)的數目也會實際上的被限制。

為了使用 SUM命令更可靠，下列規則必須遵守：

- a. 將SUM TO 的結果先存在 PRIVATE 變數。
- b. 將 SUM命令中的運算式數目限制為三個或更少。

在 Developer's Release及 dBASE III PLUS, PLBUIC 並不影響S或 AVERAGE的結果。運算式式數目的限制可能命令行的長度的限制。

BROWSE

在 dBASE III PLUS 1.0版本中用 BROWSE 命令選擇 NOMENU 可鎖定游標控制功能表，如需恢復可按下 F1 鍵而非書上所說的 F10鍵。

現在，NOMENU選擇項已可很正確的顯現 BROWSE 功能表橫列。

CALL

在 dBASE III PLUS 1.0版本中，當 CALL 一個常式 (ROUTINE)但出錯 WITH 時，會導致電腦當機，須用 "暖啟動" (warm boot) 重新開機電腦。

現在，假如您拼錯 WITH，在 dBASE III PLUS 1.1版本會顯示出錯誤訊息 "Syntax error"。

CATALOG (ASSIST)

在 dBASE III PLUS 1.0版本中，於 CATALOG被使用中的狀況下，企圖從另一個次目錄中選擇一個索引檔時，會顯示出不適當的錯誤訊息，例如 "File does not exist"或 "Index file does match database."。這是因為在 Catalog的路徑欄位 (path field) 中的路徑 (path) 並沒有指到此索引檔的所在位置。

現在，當您從 Set up:Database檔案功能表中選擇一個索引檔，如果其路徑 (path) 已設定在 Catalog中，則此索引檔就可很正確的載入。

CLOSE PROCEDURE

在以前的版本，如果你在程序檔案 (Procedure file) 中用到CLOSE PROCEDURE 或 SET PROCEDURE TO [<檔名>]時，dBASE III 將停止執行。那你將必須重新暖機啟動 (warm boot)你的電腦。

當此情形發生時，Developer's Release 及dBASE III PLUS將回返錯誤訊息 "檔案已開啓"。

CONFIG.DB

在以前的版本，CONFIG.DB 檔案中的 WP 或 TEDIT不接受超過 8個字元的參數。超過 8個字元的部份將被捨去。想要使用 MODIFY 命令 (或編輯一個備忘錄欄位) 時，只會顯示作業系統的訊息 "命令或檔名錯誤"，且將脫離編輯畫面，返回 dBASE的提示符號 (prompt)。

在 Developer's Release及 dBASE III PLUS,在 CONFIG.DB的 WP 及 TEDIT參數可任意長度，只要不超過一行即可。

CONFIG.DB/DEFAULT

在版本 1.0 的 dBASE Administrator，假如在 CONFIG.DB 檔或在第一次被呼叫的 .PRG 檔內設定另外的磁碟機而試著去載入原定的磁碟機，如此就會招致 "Unauthorized duplicate" 的錯誤，並且把控制權交還給 DOS。

在 1.1 版中的 dBASE 管理系統，此種情況不會發生。

CONFIG.DB/PROMPT

在 dBASE III PLUS 1.0 版本中可用 PROMPT= <字串> 設定一 20 個字元長的字串在 Config.DB 中。如果字串長度超過 20 個字元，則會出現錯誤訊息 "<␣ truncated"，並且此 PROMPT 會被截斷成 20 個字元長。

現在，如果您所設定的 PROMPT 長度超過 20 個字元，並不會影響執行，但是 PROMPT 仍然保留為 20 個字元長，所以，如果您在 PROMPT 下，仍會收到錯誤訊息 "<␣ truncated"。

CONFIG.DB/DELIMITERS

在 dBASE III PLUS 1.0 版本中，您可以在 config.DB 使用 SET DELIMITERS ON 或 OFF，但是您無法設定分界字元 (delimiter character)。

在 1.1 版你可以 SET DELIMITER TO "字元"。

COPY TO DELIMITED

在以前的版本，如果資料庫檔案的結構包含大量欄長一元組文字型或數值欄位時，COPY TO DELIMITED 命令將產生不可靠的結果。

在 Developer's Release及 dBASE III PLUS, COPY TO <檔名> DLIMITED 將正確地拷貝單一字元的多個欄位到至多 128個字元長度的欄位。

COPY TO SDF

在以前的版本，如果你拷貝資料檔且有備忘錄欄位到一個標準的文字格式檔，則在備忘錄欄位之後的欄位資料。無法抄到個檔中，即使抄過去，資料他不正確。但是，如果你將備忘錄欄位之後的欄位用FIELDS子句定義時，這問題不會發生。

在 Developer's Release及 dBASE III PLUS,你能拷貝一個在備忘錄欄位之後的欄位。

CREATE

在以前的版本，如果你建立 (create) 一個資料庫檔案，你必須利用此檔案來建立檔名當作預定的化名 (alias)。否則此資料庫檔案沒有其化名，當參考化名時會回應錯誤訊息 "沒有發現化名"。

在 Developer's Release及 dBASE III PLUS,當資料庫檔案創建時，檔名立即變成預定的化名。所以不需要再利用檔名來定義化名了。

當列印時的 Ctrl_C

在以前的版本，如果你在列印時按下 Ctrl_C 時你將回返到作業系統。LIST TO PRINT, DISPLAY TO PRINT, 及 SET PRINT TO, REPORT FROM PRINT 的執行便會受到影響。

在 Developer's Release及 dBASE III PLUS,列印時按下 Ctrl_C 並不會影響印出也不會回到作業系統。

CREATE / MODIFY SCREEN

在 dBASE III PLUS 中，如果你正在背景區編輯，於行列中移動游標到特殊字元（！，？，...，@）左邊，不小心按了 ctrl-T，則電腦會當機，必須暖啓動，才能再開機。在 dBASE III PLUS 1.1 版中，可以在背景區中按 ctrl-T 而不會有問題。

於 dBASE III PLUS 1.0 版，執行 CREATE / MODIFY SCREEN 時，在“黑板”（Black board）中編修螢幕畫面，然後使用 CTRL-U 鍵刪除欄位時，所產生的 “Do you wish to delete field”錯誤訊息還沒完全地解決。

在 dBASE III PLUS 1.1 版中，這個錯誤訊息決不再產生。

於 dBASE III PLUS 1.0 版，樣本程式的目錄檔（Triolog.CAT）的 CODE 當這個目錄檔是啓用狀態，而你企圖選擇的資料檔欄位中有錯誤的資料錄。

現在 1.1 版的 dBASE III PLUS CODE 欄位會填上適當的值，以便當你從目錄中選擇一有關的檔案時，不會發生錯誤。

CREATE SCREEN

在 dBASE III PLUS 1.0 版本中，如果您按 RETURN 插入一空白列而使“黑板”（blackboard）上的資料往下移一行時，游標會跑到狀態列（Status Line）下面，唯一的方法是按鍵使游標回到原來的位置。

現在，在 dBASE III PLUS 1.1 版本中，當游標到達“黑板”的最後一行時，會停在原位而資料則會繼續的往上移。

DIR

在 1.0 版本的 dBASE Administrator 中，用 DIR 命令來讀取 "唯讀檔案"，會發生 "not readable" 的錯誤。

在 1.1 版本中，允許用 DIR 命令去讀取一個只讀檔案的目錄，而不會發生錯誤！

DO WHILE

1. 在以前的版本，如果你的 DO WHILE 以分號將敘述延續到第二行且將時，dBASE III 在執行迴圈的 (loop) 第二次時執行此第二行敘述。

在 Developer's Release 及 dBASE III PLUS，DO WHILE 命令行可以分號執行不會有問題。

2. 在以前的版本，一個 DO WHILE...ENDDO 命令內的 RETURN 的架構並不能將 DO WHILE <condition> 程式結束。因此，條件式會不休止的執行。例如

```
*...A.PRG
expA="1"
DO WHILE expA="1"
  ? "Hi!"
  DO B -----> *...B.PRG
ENDDO
*EOP A.PRG
expB="Z"
DO WHILE expB="Z"
  ? "How are you ?"
  expA="X"
  RETURN
ENDDO
*EOP B.PRG
```

執行 A.PRG將導致永無止盡的迴路。dBASE III 繼續執行 B.PRG的條件式。想要正確地執行，B.PRG 必須寫成以下型式：

```
*...B.PRG
expB="Z"
DO WHILE expB="Z"
    ? "How are you ? "
    expA="X"
    EXIT -----> 注意！EXIT在這裡。
ENDDO
RETURN -----> RETURN 在這裡。
*EOP B.PRG
```

在 Developer's Release及 dBASE III PLUS,由 DO WHILE 迴路離開後會將此迴路關閉。

FILE()

在以前的版本，FILE() 函數 (function) 只找尋目前所在目錄，而且 SET PATH TO命令並不會影響。如果要去找尋其他目錄時，路徑 (path) 必須為函數中引數的一部份。

在 Developer's Release及 dBASE III PLUS,如果檔案不在目前的目錄上時，FILE() 會依據 SET PATH TO命令定義的路徑去找尋。

FIND

在 dBASE III PLUS 1.0 版中，FIND命令如果在巨指令變數及說明之間有空格，而於行中使用說明 (& &) 會失敗，舉例如下：

```
FIND &MEMVAR
```

這種用法將不能找到 memvar 變數中的值。

在 dBASE III PLUS 1.1 版中，可在 FIND 命令行的任何地方加上說明，而不會有錯誤。

在以前的版本，巨集命令 FIND & MEMVAR，如果 FIND & MEMVAR 的變數中有尾隨的空白，得到的訊息是 "NOT FOUND"。使用巨集命令時會有空白尾隨到變數裡，造成字串找尋不到。當 FIND 在命令檔案或在提示符號 (prompt) 時會發生此情形。當使用 FIND 命令時，此尾隨空白會與巨型記憶體變數串接在一起。因為有了這個額外的尾隨空白，故新的字串將找不到。

在 Developer's Release 及 dBASE III PLUS，FIND & memvar 的動作就如同 SEEK memvar。尾隨空白並不包含在此字串中。

FOUND()

在 dBASE III PLUS 1.0 版中，如果設定 SET DELETED ON，並以一鍵執行 FIND 或 SEEK 多次，雖然能找到相配的資料錄，但是第一次發生時，會被清除，而 FOUND() 傳回了一個 .F. 值。

HELP

在 dBASE III PLUS 1.0 版本中，如果 HELP.DBS 檔案無法找到時，會出現錯誤訊息 "Help text not found"，但隨即消失，因此使用者來不及看清楚。

現在，除了會出現錯誤出現之外，螢幕會停止並增加訊息 "Press a key to continue"。

INDEX

1. 在 dBASE III PLUS 1.0 版中，INDEX 或 REINDEX 有時候會傳回錯誤訊息“檔案不能存取”並毀掉正在使用的索引檔。當鍵的長度及索引檔的路徑超過 100 個字元時，就會發生這種情形。對於較小的資料庫，除非再增加較多的資料錄，將不會發生這種錯誤。注意：並非所有的鍵值及路徑的組合超過 100 個字元時，都會發生這種錯誤。

在 dBASE III PLUS 1.1 版中，卻能建立鍵的表示及路徑組合總長超過 100 個字元的索引檔，而不會發生錯誤。

2. 在 dBASE III PLUS 1.0 版中，為了建立索引檔，而使用 SET CENTURY ON 及 DTOC()，兩者不相容。設定 SET CENTURY ON，而對日期欄位 DTOC() 應用 INDEXing ON 於索引檔中是正確的。以此方法建立的索引檔，應用 FIND 或是 SEEK 命令能夠存取，但是使用 BR-OWSE、EDIT 或 GOTO 於此索引檔的資料錄時，會產生錯誤訊息，“資料錄不能索引”。

在 dBASE III PLUS 1.1 版中，設定 SET CENTURY ON 並應用 DTOC() 於日期欄位中，能建立有效的索引檔，而不會有錯誤發生。

INDEX

1. 在以前的版本，如果用來建立索引檔的索引鍵是由一連串的 STR() 功能所串指的運算式，而此建陳式 (expression) 是由一連串的 STR() 函數所組成時，INDEX ON 命令所產生的索引檔案將以記錄次序安排而非索引鍵次序。特別是當此索引鍵的運算式超過 80 個字元及全由 STR() 函數所組成時發生此情形。

在 Developer's Release 及 dBASE III PLUS，INDEX。INDEX ON 命令可在用於任何合法的陳式而不論所用的 STR() 函數有多少。

2. 在以前的版本，鍵陳式及鍵本身的最大長度是 100個字元。如果索引鍵或索引鍵式子的長度超過 100個字元時，dBASE III PLUS 將回應：“INDEX IS TOO BIG”然而當索引鍵式子是由 STR()串指起來的，程式中無法發現這種問題，但建立索引檔時會有錯誤訊息產生。如果當你嘗試這種方法時，雖然沒有回應錯誤訊息，但事實上將發生二種錯誤中的一種。不是如前所述的建立的索引是以記憶記錄次序安排就是 dBASE III將回應隨機的錯誤訊息。

在 Developer's Release及 dBASE III PLUS,索引鍵式子長度的限制是 220個字元。任何索引鍵式子超過此限制時將回應“索引陳式太大(最多 220個字元)。

INDEX / UNIQUE

在 dBASE III PLUS 1.0 版，要建一個索引檔時，用 SET UNIQUE 或用 INDEX ON UNIQUE在小的資料庫檔案沒問題(小於 100筆資料錄)，但是對一個大的資料庫檔建一個索引檔會由於下列原因之一而失敗。

1. INDEX 帶有 UNIQUE 子句看似可執行，但所建的索引檔既不唯一也不完整。例如，列出一個檔案內容時，若建索引檔時用 SET UNIQUE ON 和用 SET UNIQUE OFF 兩者所顯現的資料錄會不一樣。
2. 用 INDEX ON UNIQUE 或用 SET UNIQUE OFF會使電腦停滯，導致磁碟上一個資料叢的資料遺失了，並留下了兩個檔，一個是 .NDX 檔，它包含 0位元組，另一個是較大的 .W44 檔。 .W44 檔是做 INDEX時內部暫時性的檔，在正常的情況之下，當INDEX做完，.W44即被刪除。
3. 對大的資料庫檔(1650筆以上的資料錄)做 INDEX UNIQUE 有時不會使電腦停滯，但會繼續建索引檔直到以用的磁碟空間用完為止，然後產生 "Disk is full When writing file: <filename>." 假如此種情況發生，你只有按 ESC鍵結束 INDEX命令的執行且用 ERASE 指令去清除 .NDX, .T44 和 .W44 三個檔。