

ABAP DEVELOPMENT

SAP ABAP 程式設計基礎課程 **第五堂課：使用內部資料表（Internal Table）**

歡迎您成為ABAP程式開發者,在本課程中,我們將瞭解到:

如何新增及使用Internal Table並介紹Internal Table的類型及使用時機

製作：周台誠
noah.chou@outlook.com

THE BEST-RUN BUSINESSES RUN SAP



- **Internal Table** 是一種**Data Structure**(數據結構)，類似於其它語言中的**Structure**，**Internal Table**與**Structure**(資料結構體)基本類似，它同樣是程式執行中被創建並於內容使用的一個存儲空間，但和**Structure**(資料結構體)不同的是：**Structure**只能儲存一筆資料，而**Internal Table** 它是一個可包含多條記錄的資料表，它可以由幾個不同類型的字段(**field**)組成，用來表示具有不同屬性的某一事物，單獨一筆數據表示某個事物，多筆數據表示具有相同屬性的多個事物。

- 在建立**Internal Table** 時可使用下列兩種方式來建立
 - 參照局部表類型來建立**Internal Table**
 - 參照全域數據庫資料表建立**Internal Table**
- 在建立**Internal Table**的標準寫法為：首先定義**Structure**(資料結構體)類型，然後參照該**Structure**類型定義**Internal Table**類型，最後再參照此**Internal Table**類型建立**Internal Table**。
- 而在實務上常使用定義**Structure**(資料結構體)類型後直接參照該**Structure**定義**Internal Table**。

```

1  TYPES: BEGIN OF mytext,
2          number TYPE i,
3          name(10) TYPE c,
4          END OF mytext.
5  *使用BEGIN-END 宣告一個資料結構，結構體名稱為：mytext，
6  *該結構體有number,name 共2個欄位
7  TYPES mytab TYPE STANDARD TABLE OF mytext WITH DEFAULT KEY.
8  *使用TYPES 指令建立一個Internal Table；類型為Standard
9  *參考的資料結構體為mytext；通過關鍵字：DEFAULT KEY指定預設字段為key列
10 *對於Standard tables，預設的key列是由所有CHAR-LIKE字段組成的(C,N,D,T,X,STIRNG,XSTRING)
11 DATA text TYPE mytext.
12 *使用DATA 宣告變數，名稱為：text、類型引用自行定義資料類型：mytext
13 DATA itab TYPE mytab.
14 *使用DATA 宣告變數，名稱為：itab、類型引用自行定義資料類型：mytab
15 text-number = 1. text-name = 'John'.      APPEND text TO itab.
16 text-number = 2. text-name = 'Paul'.      APPEND text TO itab.
17 text-number = 3. text-name = 'Ringo'.     APPEND text TO itab.
18 text-number = 4. text-name = 'George'.    APPEND text TO itab.
19 *將資料賦值給text(結構體)後再透過APPEND指令將text(結構體)的資料寫入itab(Internal Table )
20 LOOP AT itab INTO text.
21     WRITE: / text-number,text-name.
22 ENDLOOP.
23 *透過LOOP指令將itab(Internal Table )內容印出至螢幕。

```

```

1  John
2  Paul
3  Ringo
4  George

```

```
1  TYPES: BEGIN OF s_type,  
2         no(6)    TYPE c,  
3         name(10) TYPE c,  
4         part(15) TYPE c,  
5     END OF s_type.  
6  *建立所要參照的Structure(資料結構體)  
7  DATA gt_itab TYPE STANDARD TABLE OF s_type WITH  
8  *參照s_type(Structure)來建立Internal Table並指定TABLE類型為STANDARD  
9         NON-UNIQUE KEY no "建立索引/  
10        WITH HEADER LINE. "建立表頭  
11  gt_itab-no    = '0001'.  
12  gt_itab-name  = 'Noah'.  
13  gt_itab-part  = 'SAP MM Team'.  
14  APPEND gt_itab.  
15  *賦值並將資料寫回Internal Table內  
16  
17  LOOP AT gt_itab.  
18      WRITE: gt_itab-no,gt_itab-name,gt_itab-part.  
19  ENDLOOP.  
20  *使用LOOP語句來循環讀取Internal Table的內容
```

```
1 DATA gt_itab TYPE STANDARD TABLE OF scarr WITH NON-UNIQUE KEY carrid.  
2 *參照全域數據庫資料表scarr來建立Internal Table  
3 *其Internal Table跟scarr資料表有相同的資料結構  
4  
5 DATA gs_str LIKE LINE OF gt_itab.  
6 *參照Internal Table來建立工作區(Work Area)  
7  
8 SELECT * INTO TABLE gt_itab FROM scarr.  
9 *使用OPEN SQL 將scarr 內的資料寫入Internal Table  
10  
11 LOOP AT gt_itab INTO gs_str .  
12     WRITE:/ gs_str-carrid, gs_str-carrname.  
13 ENDLOOP.  
14 *使用LOOP語句來循環讀取Internal Table的內容  
15 *並將其內容讀取至工作區(Work Area)後顯示於螢幕
```

- 在**Internal Table**中處理數據的必須要在我們稱之為**工作區(Work Area)**之內進行，當在定義**Internal Table**有加上**WITH HEADER LINE** 語句時會自動產生工作區。
- 沒有使用**WITH HEADER LINE** 語句來產生表頭(工作區)的**Internal Table**需定義工作區來編輯**Internal Table**內的數據，而有帶表頭的**Internal Table**則可以直接用表的名稱來編輯數據。
- 自從**ABAP**中導入了物件導向的概念後；便不再建議使用帶有表頭的**Internal Table**。

- 在定義Internal Table有加上WITH HEADER LINE 語句和沒加上在書寫程式上的差異。

```
1  *有帶表頭的範例
2  TYPES: BEGIN OF t_str,
3          col1      TYPE i,
4          col2      TYPE i,
5      END OF t_str.
6  DATA gt_itab TYPE STANDARD TABLE OF t_str
7          WITH HEADER LINE.
8  DO 3 TIMES.
9      gt_itab-col1 = sy-index.
10     gt_itab-col2 = sy-index ** 2.
11     APPEND gt_itab.
12 ENDDO.
13
14 LOOP AT gt_itab.
15     WRITE:/ gt_itab-col1,gt_itab-col2.
16 ENDLOOP.
```

```
1  *沒有帶表頭的範例
2  TYPES: BEGIN OF t_str,
3          col1      TYPE i,
4          col2      TYPE i,
5      END OF t_str.
6  DATA gt_itab TYPE STANDARD TABLE OF t_str.
7  DATA gs_str LIKE LINE OF gt_itab.
8  *參照t_str建立名為gs_str的工作區
9  DO 3 TIMES.
10     gs_str-col1 = sy-index.
11     gs_str-col2 = sy-index ** 2.
12     APPEND gs_str TO gt_itab.
13 ENDDO.
14 *先寫至工作區再透過APPEND語句寫到gt_itab
15 LOOP AT gt_itab INTO gs_str.
16     WRITE:/ gs_str-col1,gs_str-col2.
17 ENDLOOP.
18 *先將gt_itab的數據存至gs_str工作區再由工作區來輸出
```

不帶表頭時使用的語句	帶表頭時使用的語句
針對所有類型的Internal Table(Standard, Sorted, Hashed Type)	
INSERT wa INTO TABLE itab.	INSERT TABLE itab.
COLLECT wa INTO itab.	COLLECT itab.
READ TABLE itab... INTO wa.	READ TABLE itab...
MODIFY TABLE itab FROM wa....	MODIFY TABLE itab...
MODIFY itab FROM wa... WHERE...	MODIFY itab... WHERE...
DELETE TABLE itab FROM wa.	DELETE TABLE itab.
LOOP AT itab INTO wa...	LOOP AT itab...
針對具有Index 特性的Internal Table(Standard, Sorted Type)	
APPEND wa To itab.	APPEND itab.
INSERT wa INTO itab...	INSERT itab...
MODIFY itab FROM wa...	MODIFY itab...

- **Internal Table Type** 決定了讀取數據的方法，分為下列三種類型：

- **Standard tables**

標準表會自動為每一行數據編號，我們可以使用**index** 和 **key**兩種方式來讀取某一行的資料，但標準表的關鍵字(**key**)並非唯一，因此標準表中不能使用**WITH UNIQUE**語句。

- **Sorted tables**

排序表和標準表相同都具有索引，可以使用**index** 和 **key**兩種方式來讀取某一行的資料，但和標準表不同的是排序表內的數據是已經按照**key**欄位排序後保存的資料表，因此使用**Sort**語句會產生錯誤，並且在定義關鍵字時(**key**)可以使用**WITH UNIQUE**語句，也已經使用**BINARY SEARCH**功能，

- **Hashed tables**

雜湊表(哈希表)並沒有順序的索引而是用哈希運算所計算出來的值做為**key**來進行檢索，檢索速度和**tables**內的數據無關，定義這種表的時候**key**列必須是**UNIQUE**的，訪問的時候也只能通過**key access**訪問。

- 範例中使用了**DEFAULT KEY**語句會將**Tables**中以**CHAR**類型定義的前幾個**Field**定義為關鍵字，即使不寫出該語句預設還是會存在此語句，而在使用**READ TABLE**語句讀取**Internal Table**的方法有兩種，即是利用「索引」和「關鍵字」

```
1  TYPES: BEGIN OF t_line,  
2          field1 TYPE c LENGTH 5,  
3          field2 TYPE c LENGTH 4,  
4          field3 TYPE i,  
5          END OF t_line.  
6  DATA t_itab TYPE STANDARD TABLE OF t_line WITH NON-UNIQUE DEFAULT KEY.  
7  *建立STANDARD TABLE  
8  DATA s_str LIKE LINE OF t_itab.  
9  *建立工作區  
10 s_str-field1 = 'Noah'.  
11 s_str-field2 = 'ABAP'.  
12 s_str-field3 = 1.  
13 APPEND s_str TO t_itab.  
14 *先寫至工作區再透過APPEND語句寫到TABLE  
15  
16 READ TABLE t_itab INDEX 1 INTO s_str.  
17 WRITE:/ s_str-field1,s_str-field2,s_str-field3.
```

```
1 DATA t_itab TYPE STANDARD TABLE OF t_line WITH NON-UNIQUE KEY field1.
```

```
2 *自定關鍵字(User Define Key)為field1
```

```
3 READ TABLE t_itab WITH TABLE KEY field1 = 'Noah' INTO s_str.
```

```
4 *使用WITH TABLE KEY語句時，要使用內表中所有的關鍵字。
```

```
5 READ TABLE t_itab WITH KEY field3 = 1 INTO s_str.
```

```
6 *使用WITH KEY語句時，讀取關鍵字以外的欄位即可。
```

```
8 DATA t_itab TYPE STANDARD TABLE OF t_line WITH NON-UNIQUE DEFAULT KEY.
```

```
9 *因為使用DEFAULT KEY語句，因此field1,field2為關鍵字
```

```
10 READ TABLE t_itab WITH TABLE KEY field1 = 'Noah' field2 = 'ABAP' INTO s_str.
```

```
11 *使用WITH TABLE KEY語句時就要將所有關鍵字(field1,field2)全寫出。
```

```
12 READ TABLE t_itab WITH KEY field3 = 1 INTO s_str.
```

```
13 *但使用WITH KEY語句可以讀取關鍵字以外的欄位即可(是關鍵字欄位也可以)。
```

- Sorted tables 不能使用APPEND語句，不然會發生排序問題的Dump Error，因此需改用INSERT語句

```
1  TYPES: BEGIN OF t_line,  
2          field1 TYPE c LENGTH 5,  
3          field2 TYPE c LENGTH 4,  
4          field3 TYPE i,  
5      END OF t_line.  
6  DATA t_itab TYPE SORTED TABLE OF t_line WITH NON-UNIQUE DEFAULT KEY.  
7  *建立SORTED TABLE  
8  DATA s_str LIKE LINE OF t_itab.  
9  *建立工作區  
10 s_str-field1 = 'Noah'. s_str-field2 = 'ABAP'. s_str-field3 = 1.  
11 INSERT s_str INTO TABLE t_itab.  
12 s_str-field1 = 'Judy'. s_str-field2 = 'ABAP'. s_str-field3 = 2.  
13 INSERT s_str INTO TABLE t_itab.  
14 s_str-field1 = 'Noah'. s_str-field2 = 'ABAP'. s_str-field3 = 3.  
15 INSERT s_str INTO TABLE t_itab.  
16  
17 LOOP AT t_itab INTO s_str.  
18     WRITE:/ s_str-field1,s_str-field2,s_str-field3.  
19 ENDLOOP.
```

Judy	ABAP	2
Noah	ABAP	3
Noah	ABAP	1

- 自定關鍵字欄位並使用**UNIQUE**語句

```

1  TYPES: BEGIN OF t_line,
2          field1 TYPE c LENGTH 5,
3          field2 TYPE c LENGTH 4,
4          field3 TYPE i,
5      END OF t_line.
6  DATA t_itab TYPE SORTED TABLE OF t_line WITH UNIQUE KEY field3.
7  *建立SORTED TABLE
8  DATA s_str LIKE LINE OF t_itab.
9  *建立工作區
10 s_str-field1 = 'Noah'. s_str-field2 = 'ABAP'. s_str-field3 = 1.
11 INSERT s_str INTO TABLE t_itab.
12 s_str-field1 = 'Judy'. s_str-field2 = 'ABAP'. s_str-field3 = 3.
13 INSERT s_str INTO TABLE t_itab.
14 s_str-field1 = 'Noah'. s_str-field2 = 'ABAP'. s_str-field3 = 2.
15 INSERT s_str INTO TABLE t_itab.
16
17 LOOP AT t_itab INTO s_str.
18     WRITE:/ s_str-field1,s_str-field2,s_str-field3.
19 ENDLOOP.

```

Noah	ABAP	1
Noah	ABAP	2
Judy	ABAP	3

```
1  TYPES: BEGIN OF t_line,  
2          field1 TYPE c LENGTH 5,  
3          field2 TYPE c LENGTH 4,  
4          field3 TYPE i,  
5      END OF t_line.  
6  DATA t_itab TYPE HASHED TABLE OF t_line WITH UNIQUE KEY field3.  
7  *建立HASHED TABLE  
8  DATA s_str LIKE LINE OF t_itab.  
9  *建立工作區  
10 s_str-field1 = 'Noah'. s_str-field2 = 'ABAP'. s_str-field3 = 1.  
11 INSERT s_str INTO TABLE t_itab.  
12 s_str-field1 = 'Judy'. s_str-field2 = 'ABAP'. s_str-field3 = 2.  
13 INSERT s_str INTO TABLE t_itab.  
14 s_str-field1 = 'Noah'. s_str-field2 = 'ABAP'. s_str-field3 = 3.  
15 INSERT s_str INTO TABLE t_itab.  
16 *HASHED TABLE不能使用APPEND語句  
17 READ TABLE t_itab WITH KEY field3 = 1 INTO s_str.  
18 WRITE:/ s_str-field1,s_str-field2,s_str-field3.  
19 *HASHED TABLE不存在索引，因此不能用READ TABLE~INDEX語句  
20 *需使用READ TABLE WITH TABLE KEY或是WITH KEY才能讀取。  
21 LOOP AT t_itab INTO s_str.  
22     WRITE:/ s_str-field1,s_str-field2,s_str-field3.  
23 ENDLOOP.
```

	Standard tables	Sorted tables	Hashed tables
用索引讀取(Index Access)	Yes	Yes	NO
用關鍵字讀取(Key Access)	Yes	Yes	Yes
關鍵字(Key Values)	NON-UNIQUE	NON-UNIQUE or UNIQUE	UNIQUE
建議讀取方式	Index Access	Key Access	Key Access
速度	寫入=速度快 讀取=速度慢	寫入=速度最慢 讀取=速度快	寫入=速度慢 讀取=速度最快

- **Internal Table**和其他變數一樣可以使用**MOVE**語句來賦值，但是要特別決定的是有帶表頭的**Internal Table**要區分是賦值給表頭還是表體。

```
1  TYPES: BEGIN OF s_type,  
2          col1 TYPE i,col2 TYPE i,  
3          END   OF s_type.  
4  DATA gt_itab1 TYPE STANDARD TABLE OF s_type WITH NON-UNIQUE DEFAULT KEY  
5          WITH HEADER LINE. "建立表頭  
6  DATA gt_itab2 TYPE STANDARD TABLE OF s_type WITH NON-UNIQUE DEFAULT KEY  
7          WITH HEADER LINE. "建立表頭  
8  DO 5 TIMES.  
9      gt_itab1-col1 = SY-INDEX. "當前LOOP循環過的次數.  
10     gt_itab1-col2 = SY-INDEX * 2.  
11     APPEND gt_itab1.  
12 ENDDO.  
13 MOVE gt_itab1[] TO gt_itab2[].  
14 *因為帶表頭的Internal Table其表頭和表體名稱相同。  
15 *因此為了區分是賦值給表體其名稱後面需加上[]來代表  
16 LOOP AT gt_itab2.  
17     WRITE:/ gt_itab2-col1,gt_itab2-col2.  
18 ENDLOOP.
```

- 和其他變數一樣Internal Table也可以清空其數據，初始化內表的語句有**CLEAR**，**REFRESH**，**FREE**這三種。而這三種語句對是否有無帶表頭的Internal Table用法也不同。
 - **CLEAR** 刪除Internal Table數據並釋放記憶體空間，如果是用在帶表頭的Internal Table；僅清空HEADER LINE，並不影響Internal Table數據存儲空間。
 - **REFRESH**刪除Internal Table數據但不釋放記憶體空間
 - **REFRESH** <itab> **FROM TABLE** <dbtab> 刪除Internal Table數據，從數據庫表所獲數據。
 - **FREE**釋放記憶體空間，用在 **REFRESH**語句之後

- **REFRESH <itab> FROM TABLE <dbtab>** 使用範例

```
1  TABLES: scarr.  
2  *宣告所要使用的所要使用的全域數據庫資料表  
3  DATA gt_itab TYPE STANDARD TABLE OF scarr WITH NON-UNIQUE KEY carrid.  
4  *參照全域數據庫資料表scarr來建立Internal Table  
5  *其Internal Table跟scarr資料表有相同的資料結構  
6  DATA gs_str LIKE LINE OF gt_itab.  
7  *參照Internal Table來建立工作區(Work Area)  
8  REFRESH gt_itab FROM TABLE scarr.  
9  *清空gt_itab的存儲空間，並從scarr中查找數據填充到gt_itab  
10 LOOP AT gt_itab INTO gs_str .  
11     WRITE:/ gs_str-carrid, gs_str-carrname.  
12 ENDLOOP.  
13 *使用LOOP語句來循環讀取Internal Table的內容  
14 *並將其內容讀取至工作區(Work Area)後顯示於螢幕
```

- 對Internal Table進行排序，只能用Standard tables和Hashed tables在可以指定具體的排序欄位、排序方式（昇冪/降冪），語法如下。
 - 用Internal Table已定義好的關鍵字進行排序，預設排序方式為**ASCENDING**
SORT itab [ASCENDING / DESCENDING].
 - 任意指定欄位進行排序
SORT itab [BY f1 f2 ... fn] [ASCENDING / DESCENDING].
 - ◆ 指定欄位最多**250**個
 - ◆ 若f1欄位的值存在null時；則會排除此列
 - 穩定排序法(**stable sorting**)，如果鍵值相同之資料，在排序後相對位置與排序前相同時
SORT itab STABLE.

1	原有資料：	5	4	3	5	2
2	加上編號：	5 (1)	4 (2)	3 (3)	5 (4)	2 (5)
3	經過排序：	2 (5)	3 (3)	4 (2)	5 (1)	5 (4)

- 想要得知Internal Table的相關屬性；可利用**DESCRIBE** 語句來取得。
語法： **DESCRIBE TABLE** itab [**LINES** gv_line] [**OCCURS** gv_init] [**KIND** gv_kind]
 - **LINES** 取得Internal Table 的資料筆數
 - **OCCURS** 取得Internal Table 的初始大小
 - **KIND** 取得Internal Table 的類型，其中「T」代表Standard tables，
 「S」代表Sorted tables，「H」代表Hashed tables

- 插入一行新的資料至Internal Table的命令有 **INSERT**, **APPEND**, **COLLECT** 這三種，其中 **INSERT**和**APPEND**都是將新增的資料插入Internal Table的最後一行，但是**INSERT**語句根據Internal Table類型不同，其執行結果也會有所不同。而**COLLECT** 語句用於合計關鍵字相同的數字類型的欄位，如果關鍵字不同的將直接追加數據。
- 不同的Internal Table類型所具有各自不同的**INSERT**效果

Standard tables	Sorted tables	Hashed tables
追加數據至Internal Table的最後一行	按照Internal Table排序好的順序來追加數據	按照關鍵字的Hashed 索引順序追加數據
與APPEND語句的效果相同	若是關鍵字設為NON-UNIQUE的類型，重複的數據會追加到相同的數據的上一行	

- 不同的Internal Table類型所具有各自不同的**APPEND**效果

Standard tables	Sorted tables	Hashed tables
追加數據至Internal Table的最後一行	按照Internal Table排序好的順序來追加數據，否則會產生Dump Error	不能使用APPEND語句
可利用SORTED BY 語句以關鍵字來排序(DESCENDING)		

- 利用關鍵字來追加一筆數據

語法: **INSERT line INTO TABLE itab.**

INSERT語句執行成功時，系統變數 **SY-SUBRC**會傳回0，若**Internal Table**存有唯一性的設定 (**UNIQUE KEY**)，若追加已存在相同關鍵字的數據時，系統變數 **SY-SUBRC**會傳回4，並且不會產生**Dump Error**。

- 利用關鍵字來追加多筆數據

語法: **INSERT LINES OF itab1 [FROM n1] [TO n2] INTO TABLE itab2.**

- 利用索引來追加一筆數據

語法: **INSERT line INTO TABLE itab [Index idx].**

利用**Index**語句可以在 **Index**指定的位置追加一筆數據，注意此方法不適用於**Hashed tables**，語句執行成功，系統變數 **SY-SUBRC**會傳回0，同時系統變數 **SY-TABIX**會傳回**Index**值。

- 不同的**Internal Table**類型所具有各自不同的**INSERT**效果

Standard tables	Sorted tables	Hashed tables
追加數據至 Internal Table 的最後一行	按照 Internal Table 排序好的順序來追加數據	按照關鍵字的 Hashed 索引順序追加數據
與 APPEND 語句的效果相同	若是關鍵字設為 NON-UNIQUE 的類型，重複的數據會追加到相同的數據的上一行	

```
1 DATA: BEGIN OF gs_line,  
2     col1 TYPE c, col2 TYPE i,  
3     END OF gs_line.  
4 DATA gt_itab1 LIKE STANDARD TABLE OF gs_line WITH NON-UNIQUE KEY col1.  
5 DATA gt_itab2 LIKE SORTED TABLE OF gs_line WITH NON-UNIQUE KEY col1.  
6 *建立gt_itab1(STANDARD TABLE)和gt_itab2(SORTED TABLE)  
7 gs_line-col1 = 'B'. gs_line-col2 = '1'.  
8 INSERT gs_line INTO TABLE gt_itab1.  
9 gs_line-col1 = 'A'. gs_line-col2 = '2'.  
10 INSERT gs_line INTO TABLE gt_itab1.  
11 gs_line-col1 = 'C'. gs_line-col2 = '3'.  
12 INSERT gs_line INTO TABLE gt_itab1.  
13 *在gt_itab1(STANDARD TABLE)中寫入三筆資料，注意其資料排序為何？  
14 INSERT LINES OF gt_itab1 INTO TABLE gt_itab2.  
15  
16 LOOP AT gt_itab2 INTO gs_line.  
17     WRITE:/ gs_line-col1,gs_line-col2,gs_line-col3.  
18 ENDLOOP.  
19 *寫入gt_itab2(SORTED TABLE)後其資料排序為何？
```

- **INSRET**語句可以利用關鍵字及索引來追加數據，但是**APPEND**語句只能用索引來追加數據，並且**Hashed tables**不能使用該語句。
- 追加一筆數據
語法: **APPEND line TO itab.**
執行**APPEND**語句後，系統變數**SY-TABIX**中會傳回所追加數據的行號，即索引編號。
- 追加多筆數據
語法: **APPEND LINES itab1 [FROM n1] [TO n2] TO itab2.**
可將**itab1**的索引**N1~N2**之間的數據追加至**itab2**
- 不同的**Internal Table**類型所具有各自不同的**APPEND**效果

Standard tables	Sorted tables	Hashed tables
追加數據至Internal Table的最後一行	按照Internal Table排序好的順序來追加數據，否則會產生Dump Error	不能使用APPEND語句
可利用SORTED BY 語句以關鍵字來排序(DESCENDING)		

- **COLLECT** 語句的用途在於合計**Internal Table**中數字類型的欄位

語法: **COLLECT** wa **INTO** itab.

除了關鍵字以外的欄位需要都是數字類型(f,i,p)，執行**COLLECT** 語句時當存在相同關鍵字的數據時，會合計其數字類型的欄位，如果關鍵字不同的將直接追加數據，沒有關鍵字的**Internal Table**，則會把**CHAR**類型的字段作為關鍵字後執行相同的動作。

```
1  TYPES: BEGIN OF COMPANY,  
2         NAME(20) TYPE C,  
3         SALES    TYPE n,  
4         END OF COMPANY.  
5  
6  DATA: COMP      TYPE COMPANY,  
7         COMPTAB TYPE COMPANY OCCURS 0 .  
8  
9  COMP-NAME = 'MOBILE'.    COMP-SALES = 10. COLLECT COMP INTO COMPTAB.  
10 COMP-NAME = 'NOTEBOOK'. COMP-SALES = 20. COLLECT COMP INTO COMPTAB.  
11 COMP-NAME = 'MOBILE'.    COMP-SALES = 30. COLLECT COMP INTO COMPTAB.  
12 "兩行數據NAME值相同合併，SALES字段值彙總
```

- 如果要修改Internal Table內的數據時，可使用MODIFY語句用關鍵字或是索引來修改相應行的數據。
- 使用關鍵字來修改一筆數據

語法: **MODIFY TABLE** itab **FROM** wa [**TRANSPORTING** f1 g2]

以關鍵字為基準來修改Internal Table內的數據時，如該Internal Table關鍵字並非唯一時(NON-UNIQUE)，也就是存有重複資料的狀況，執行MODIFY語句時會修改第一筆的數據。

- 當修改有帶表頭的Internal Table時可省略FROM wa語句

- TRANSPORTING語句可修改指定字串的值。

```
1 DATA: BEGIN OF gs_line,  
2     |   coll TYPE c, coll2 TYPE i,  
3     |   END OF gs_line.  
4 DATA gt_itab1 LIKE STANDARD TABLE OF gs_line WITH NON-UNIQUE KEY coll.  
5 gs_line-coll = 'B'. gs_line-coll2 = '1'.  
6 INSERT gs_line INTO TABLE gt_itab1.  
7 gs_line-coll = 'A'. gs_line-coll2 = '2'.  
8 INSERT gs_line INTO TABLE gt_itab1.  
9  
10 gs_line-coll = 'A'. gs_line-coll2 = '1'.  
11 MODIFY TABLE gt_itab1 FROM gs_line.  
12  
13 LOOP AT gt_itab1 INTO gs_line.  
14     |   WRITE:/ gs_line-coll,gs_line-coll2,gs_line-coll3.  
15 ENDLOOP.
```

- 使用**WHERE** 條件來修改多筆數據

語法: **MODIFY** itab **FROM** wa [**TRANSPORTING** f1 g2] **WHERE** cond.

修改一筆以上的數據時可使用**WHERE** 語句，並且可以使用多個欄位的檢索條件。

- 使用索引來修改一筆數據

語法: **MODIFY** itab **FROM** wa [**INDEX** idx] [**TRANSPORTING** f1 g2].

利用索引可以修改對應行的數據，但因為使用索引的關係；只能用在**Standard tables**和**Sorted tables**上，在**LOOP**語句中省略**INDEX** 選項，系統會自動修改目前的索引行編號。

- 使用**DELETE** 語句可以刪除Internal Table內的一筆數據，並可使用關鍵字或是索引作為查詢條件來刪除數據。
- 使用關鍵字來刪除一筆數據
語法: **DELETE TABLE** itab [FROM wa].
在關鍵字沒有唯一的Standard tables中使用**WITH TABLE KEY** 語句來刪除重複數據中的某一筆
語法: **DELETE TABLE** itab **WITH TABLE KEY** k1 = v1 ... kn = vn.
- 使用索引字來刪除一筆數據
語法: **DELETE** itab **INDEX** idx.
相同的因為使用索引的關係; **Hashed tables**無法使用該語句。
- 使用索引字來刪除多筆數據
語法: **DELETE** itab **FROM** idx1 TO idx2.
DELETE itab **FROM** idx1.
刪除從idx1開始以下所有的筆數
DELETE itab **FROM** TO idx2.
刪除從開始到idx2之間所有的筆數

- 使用**ADJACENT DUPLICATES** 語句來刪除重複資料行
語法: **DELETE ADJACENT DUPLICATES FROM itab [COMPARING f1 f2 ...|ALL FIELDS].**
在使用**ADJACENT DUPLICATES** 語句來刪除重複數據前必須先進行排序(**SORT**)後; 才能得到所要的結果。

- 使用**READ**語句到讀取Internal Table內數據，要注意的是當有表頭存在時，所對應的資料會存在表頭中，其餘則存在工作區。
- 使用關鍵字來讀取Internal Table的數據
語法：**READ TABLE** itab **FROM** wa **INTO** result.
READ TABLE itab **WITH [TABLE] KEY** k1 = v1 ... kn = vn **INTO** result.
執行成功，系統變數**SY-SUBRC** 傳回0，失敗傳回4。**Result**是將結果存入工作區，如果有是有自帶表頭的Internal Table，會直接將表頭當成工作區，因此可以省略**INTO result** 語句。
- 使用索引來讀取Internal Table的數據
語法：**READ TABLE** itab **INDEX** idx **INTO** result.
執行成功，系統變數**SY-SUBRC** 傳回0，失敗傳回4。相同的因為使用索引的關係；**Hashed tables**無法使用該語句。