

5장

2024년 4월 14일 일요일 오후 11:12

1 Internal Table

- 단일 프로그램에서 사용되는 내부테이블
- 선언

```
Types: Begin of s_type,  
       no(6) type c,  
       name(10) type c,  
       part(16) type c,  
End of s_type.
```

```
types t_type type standard table of s_type Initial size 100.  
data sawon type t_type.
```

- 생성

1) Local Table Type을 이용한 인터널 테이블 생성

```
REPORT Z05_01_031.
```

```
□ TYPES: BEGIN OF s_type,  
       no(6) TYPE c,  
       name(10) TYPE c,  
       part(16) TYPE c,  
END OF s_type.
```

) 구조 정의

[Local Type 정의]

```
data gt_itab type STANDARD TABLE OF s_type  
with NON-UNIQUE key no  
WITH HEADER LINE.  
  
gt_itab-no = '0001'.  
gt_itab-name = 'Easy ABAP'.  
gt_itab-part = 'SAP Team'.  
append gt_itab.
```

) 인터널 테이블 type
구조를 사용한 접근 방법

```
□ loop at gt_itab.  
    WRITE : gt_itab-no, gt_itab-name, gt_itab-part.  
ENDLOOP.
```

```

REPORT Z05_01_031.

"TYPES: BEGIN OF s_type,
DATA: BEGIN OF gs_type,
  no(6) TYPE c,
  name(10) TYPE c,
  part(16) TYPE c,
  "END OF s_type.
END OF gs_type.

" data gt_itab type STANDARD TABLE OF s_type
data gt_itab like STANDARD TABLE OF gs_type
  with NON-UNIQUE key no
    WITH HEADER LINE.

gt_itab-no = '0001'.
gt_itab-name = 'Easy ABAP'.
gt_itab-part = 'SAP Team'.
append gt_itab.

loop at gt_itab.
  WRITE : gt_itab-no, gt_itab-name, gt_itab-part.
ENDLOOP.

```

Entity Object 정의

Report Z05_01_031

0001 Easy ABAP SAP Team

2) Global ABAP Dictionary Type을 이용해 생성

ABAP dictionary 테이블이나 구조체 참고하여 인터널 테이블 생성

```

REPORT Z05_02_031.

DATA gt_itab type sorted table of scarr
  WITH UNIQUE key carrid.
DATA gs_str like line of gt_itab.

select * into table gt_itab
  from scarr.

loop at gt_itab into gs_str.
  WRITE : / gs_str-carrid, gs_str-carrname.
ENDLOOP.

```

sorted table
구조체

2 인터널 테이블 헤더 라인

- 헤더라인
 - 디버깅 시 모자 아이콘
- 선언
 - WITH HEADER LINE
 - 선언시 가장위에 있는 데이터가 itab이되고
 - 그 데이터들은 바디 itab[]이 된다.
- 비교
 - 헤더라인이 없으면 구조체를 선언 후 값을 복사하여 사용,

헤더라인이 있으면 인터널테이블의 이름은 ABAP프로그램 내 헤더라인을 의미

헤더라인이 없으면 Modify Table itab From wa.

있으면 Modify Table itab.

- 인터널 테이블은 Standard , Sorted, Hash Type 3가지만 존재

- 헤더 라인이 있는 인터널 테이블은 Work Area 대신 헤더라인을 이용한다.

Insert Table itab = Insert itab Into Table itab.

```
REPORT Z05_03_031.

□ TYPES: BEGIN OF t_str,
  coll type i,
  col2 type i,
  END OF t_str.

  DATA gt_itab type table of t_str WITH HEADER LINE.

□ do 3 TIMES.
  gt_itab-coll = sy-index.
  gt_itab-col2 = sy-index ** 2.
  append gt_itab.
ENDDO.

□ loop at gt_itab.
  write : / gt_itab-coll, gt_itab-col2.
ENDLOOP.

Report Z05_03_031
```

1	1
2	4
3	9

- 헤더라인이 존재하지 않는 인터널 테이블은 Work Area를 선언하여 데이터를 읽거나 변경

```
REPORT Z05_04_031.

□ TYPES: BEGIN OF t_str,
  coll TYPE i,
  col2 TYPE i,
  END OF t_str.

  DATA gt_itab type table of t_str.
  DATA gs_str like line of gt_itab.

□ do 3 times.
  gs_str-coll = sy-index.
  gs_str-col2 = sy-index * 2.
  append gs_str to gt_itab.
ENDDO.

□ loop at gt_itab into gs_str.
  WRITE : / gs_str-coll, gs_str-col2.
ENDLOOP.
```

1	2
2	4
3	6

3 인터널 테이블 종류

- 인터널 테이블의 종류는 개별 Entry에 접근하는 방법을 결정, 3가지 종류
 - Standard / Sorted -> index table
 - Hashed
- Standard
 - o 순차적인 Index를 가지는 테이블 Tree 구조
index로 entry를 찾을 때
read, modify, delete 구문 사용 시
 - o index는 테이블에서 데이터가 위치하는 위치 순번
 - o 키는 항상 Non-unique 로 선언
with unique 구문 사용할 수 없다.
with non-unique key 가능
 - o Default key는 char 타입의 모든 컬럼들을 키 컬럼으로 정의
기본으로 포함된 것을 간주 standard key
 - o index를 이용하여 검색 -> 탐색 속도가 증가

REPORT Z05_05_031.

```

□ TYPES: begin of t_line,
|   field1 TYPE c length 5,
|   field2 type c length 4,
|   field3 type i,
| END OF t_line.

TYPES t_tab type STANDARD TABLE OF t_line WITH NON-UNIQUE DEFAULT KEY.

data gt_itab type t_tab WITH HEADER LINE.

gt_itab-field1 = 'ENJOY'.
gt_itab-field2 = 'ABAP'.
gt_itab-field3 = 1.

append gt_itab.           1번째 Line 추가

READ TABLE gt_itab INDEX 1.
WRITE : / gt_itab-field1, gt_itab-field2, gt_itab-field3.

```

- Sorted Table

- o 항상 정렬된 인터널 테이블 타입
- o key값으로 항상 정렬된 결과로 테이블에 저장
탐색 속도는 정적 상관관계
이미 정렬되어 있어 sort 명령어 오류 발생

- o key 값 선언시 unique / non-unique 명시
- o append 구문 사용시 Dump error
- o with non-unique key 수정시 중복 허용

`REPORT Z05_06_031.`

```
"structure type
types: BEGIN OF t_line,
|   col TYPE c,
|   seq type i,
END OF t_line.

"sorted table
TYPES t_tab TYPE SORTED TABLE OF t_line WITH UNIQUE key col.

"inter table
DATA gt_itab TYPE t_tab WITH HEADER LINE.

gt_itab-col = 'B'.
gt_itab-seq = '1'.
INSERT table gt_itab.

gt_itab-col = 'A'.
gt_itab-seq = '2'.
insert table gt_itab.

clear gt_itab.
read table gt_itab index 2.

write : / gt_itab-col, gt_itab-seq.|
```

Program Z05_06_031

B 1

- Hashed Table
 - o 순차적인 index 없이 Hash 값으로 계산된 key값으로 탐색 가능
 - o 응답속도 o(1)
 - o hash 알고리즘으로 메모리 저장된 주소 값으로 데이터를 읽음.
 - o 반드시 Unique
 - o Read talbe with table key / with key 구문
- 인터널 테이블 구식 선언법
 - o OCCURS 0
 - 3.0 버전 헤더라인을 가질때 사용, 호환성 문제로 INITIAL SIZE와 같은 효과, 메모리 할당을 의미 (최소화)

```

REPORT Z05_08_031.

types : begin of t_line,
  col type c,
  seq type i,
end of t_line.

data gt_itab type t_line occurs 0 WITH HEADER LINE.

gt_itab-col = 'A'.
gt_itab-seq = '1'.
insert table gt_itab.

clear gt_itab.
READ TABLE gt_itab index 1.
write : / gt_itab-col, gt_itab-seq.

```

```

data : begin of gt_itab occurs 0,
  col type c,
  seq type i,
end of gt_itab.

gt_itab-col = 'A'.
gt_itab-seq = '1'.
insert table gt_itab.

clear gt_itab.
READ TABLE gt_itab index 1.
write : / gt_itab-col, gt_itab-seq.

```

Program Z05_08_031

A 1

- internal table 속도 비교

	STANDARD TABLE	SORTED TABLE	HASHED TABLE
Key access (single entry)	Table scan ↓	Binary search	Hash function
Access costs for n entries	선형적으로 증가 <u>O(n)</u>	로그함수 증가 <u>O(log(n))</u>	항상 동일 <u>O(1)</u>
Access using	mainly <u>index</u>	mainly <u>key</u>	<u>only key</u>
Uniqueness	<u>NON-UNIQUE</u>	<u>UNIQUE, NON-UNIQUE</u>	<u>UNIQUE</u>

- o sorted table 탐색 속도가 빠르다.

```

REPORT Z05_09_031.

□ TYPES: BEGIN OF t_line,
  | col1 type i,
  | col2 type i,
  | col3 type i,
  END OF t_line.

DATA gt_itab TYPE STANDARD TABLE OF t_line WITH HEADER LINE.

□ do 10000000 TIMES.
  | gt_itab-col1 = sy-index.
  | gt_itab-col2 = sy-index.
  | gt_itab-col3 = sy-index.
  | insert table gt_itab.
enddo.

READ TABLE gt_itab with key col1 = 99999.

```

```

REPORT Z05_10_031.

□ TYPES: BEGIN OF t_line,
  | col1 type i,
  | col2 type i,
  | col3 type i,
  END OF t_line.

DATA gt_itab TYPE sorted TABLE OF t_line
  WITH HEADER LINE with non-unique key col1.

□ do 10000000 TIMES.
  | gt_itab-col1 = sy-index.
  | gt_itab-col2 = sy-index.
  | gt_itab-col3 = sy-index.
  | insert table gt_itab.
enddo.

READ TABLE gt_itab with key col1 = 99999.

```

- sorted read는 빠르지만 append가 느림
- standard는 append 빠르지만 read가 느림

- Binary Search
 - Standard / binary search
 - 이진트리를 이용하여 키값 기준으로 정렬
 - sort 명령어로 정렬한 후에 수행 / 아니면 데이터를 찾을 수 없음
 - Read table <itab> with key col1 binary search.
 - sorted table은 키값으로 정렬되어 있어 기본으로 포함되어 있음

```

REPORT Z05_11_031.

□ TYPES: BEGIN OF t_line,
  | col1 TYPE i,
  | col2 TYPE i,
  | col3 TYPE i,
  | END OF t_line.

  DATA itab TYPE STANDARD TABLE OF t_line WITH key col1 WITH HEADER LINE.

□ do 10000000 times.
  | itab-col1 = sy-index.
  | itab-col2 = sy-index.
  | itab-col3 = sy-index.
  | insert table itab.
  | enddo.

  sort itab by col1 col2 col3.

  read table itab with key col1 = 5000 col2 = 1000 col3 = 999999.

```

```

REPORT Z05_12_031.

□ TYPES: BEGIN OF t_line,
  | col1 TYPE i,
  | col2 TYPE i,
  | col3 TYPE i,
  | END OF t_line.

  DATA itab TYPE STANDARD TABLE OF t_line WITH key col1 WITH HEADER LINE.

□ do 10000000 times.
  | itab-col1 = sy-index.
  | itab-col2 = sy-index.
  | itab-col3 = sy-index.
  | insert table itab.
  | enddo.

  sort itab by col1 col2 col3.

  read table itab with key col1 = 5000 col2 = 1000 col3 = 999999 BINARY SEARCH.

```

- Binary search read 속도 빠름
- loop 이전에 sort하고 binary search를 하자
- Binary search vs sorted table

```

REPORT Z05_13_031.

□ TYPES: BEGIN OF t_line,
  | col1 TYPE i,
  | col2 TYPE i,
  | col3 TYPE i,
  | end of t_line.

  DATA itab TYPE SORTED TABLE OF t_line WITH UNIQUE key col1 with HEADER LINE.

□ do 1000000 times.
  | itab-col1 = sy-index.
  | itab-col2 = sy-index.
  | itab-col3 = sy-index.
  | INSERT table itab.
  | enddo.

  READ TABLE itab WITH key col1 = 5000 col2 = 100 col3 = 999999.

```

- Binary보다 Sorted가 Read 빠름

- Hashed Table

REPORT Z05_14_031.

```

□ TYPES : BEGIN OF t_line,
  |   col1 TYPE i,
  |   col2 TYPE i,
  |   col3 TYPE i,
  | END OF t_line.

□ *
  | * data itab TYPE STANDARD TABLE OF t_line with key col1 WITH HEADER LINE.
  |
  | * data itab TYPE SORTED TABLE OF t_line with UNIQUE key col1 WITH HEADER LINE.

DATA itab TYPE HASHED TABLE OF t_line WITH UNIQUE key col1 WITH HEADER LINE.

□ DO 1500000 TIMES.
  | itab-col1 = sy-index.
  | itab-col2 = sy-index.
  | itab-col3 = sy-index.
  | INSERT TABLE itab.
ENDDO.

READ TABLE itab WITH key col1 = 999999.

```

구문	STANDARD TABLE	SORTED TABLE	HASHED TABLE
APPEND 속도	422ms 438ms 438ms	1,438ms 1,454ms 1,459ms	1,313ms 1,328ms 1,297ms
READ 속도	125ms 110ms 109ms	78ms 94ms (dilash) 93ms	63ms 46ms 47ms
결론	APPEND 속도 빠름 READ 속도 느림	APPEND 속도 가장 느림 READ 속도 빠름	APPEND 속도 가장 느림 READ 속도 가장 빠름

	Standard Table	Sorted Table	Hashed Table
Index Access	Yes	Yes	No X
Key Access	Yes	Yes	Yes
Key Values	Not Unique	Unique 또는 Not Unique <i>반복될 수 있음</i>	Unique
권장되는 Access 방법	Index 권장	Key 권장	Key만 사용

5 인터널 테이블 명령어

- 인터널 테이블 값 할당

move itab1 to itab2 (=) itab2 = itab1 (header)
move itab1[] to itab2[] (=) itab2[] = itab1[] (body)

move-corresponding itab1 to itab2
 인터널 테이블 타입이 같아야 사용.
 다르면 칼럼 순서대로 할당 서로다른 경우 에러.

```

REPORT Z05_15_031.

DATA : BEGIN OF t_line,
      col1 TYPE i,
      col2 TYPE i,
      END OF t_line.

DATA: gt_itabl TYPE STANDARD TABLE OF t_line WITH HEADER LINE,
      gt_itab2 TYPE STANDARD TABLE OF t_line,
      gs_wa like line of gt_itab2.

do 5 TIMES.
  gt_itabl-col1 = sy-index.
  gt_itabl-col2 = sy-index * 2.
  INSERT table gt_itabl.
enddo.

move gt_itabl[] to gt_itab2.

loop at gt_itab2 into gs_wa.
  WRITE : / gs_wa-col1, gs_wa-col2.
ENDLOOP.

```

Report Z05_15_031

1	2
2	4
3	6
4	8
5	10

- 헤더라인이 있으면 헤더라인과 인터널 테이블 이름이 같음.
- 헤더라인이 없는 인터널 테이블을 이름은 자기자신이 됨.

- 인터널 테이블 초기화
 - clear , refresh, free
 - 헤더라인이 없을시 body삭제
 - clear
 - 메모리 공간을 반환
 - clear body - 헤더 라인 삭제 / clear body[] - 바디 삭제
 - refresh
 - 데이터만 지움 메모리 유지
 - free
 - 메모리 반환
 - refresh itab. 테이블 내용 삭제
 - free itab. 메모리 반환

```

REPORT Z05_16_031.

DATA : BEGIN OF gs_line,
      col1 TYPE i,
      col2(1) TYPE c,
      END OF gs_line.

DATA : gt_itab like STANDARD TABLE OF gs_line WITH HEADER LINE.

```

```
REPORT Z05_16_031.

DATA : BEGIN OF gs_line,
      col1 TYPE i,
      col2(1) TYPE c,
    END OF gs_line.

DATA : gt_itab like STANDARD TABLE OF gs_line WITH HEADER LINE.

gs_line-col1 = 1.
gs_line-col2 = 'A'.
INSERT gs_line into TABLE gt_itab.

refresh gt_itab.

* free gt_itab.
* clear gt_itab.
if gt_itab is INITIAL.
  WRITE : 'internal table has no data.'.
  FREE gt_itab.
ENDIF.
```

Report Z05_16_031

internal table has no data.