

ABAP/4 Reporting

Internal Tables

Compiled By : P. M. V. Subba Rao

Internal Table Concepts

- To create a sorted list with control levels, you can use internal tables or datasets generated with the EXTRACT statement.
- Use internal tables to process datasets with the same structure.
- Intermediate datasets are suitable for datasets that are structured differently.

Declaring an Internal table with Header line

```
TABLES SFLIGHT.  
TYPES: BEGIN OF T_ITAB,  
       CARRID LIKE SFLIGHT_CARRID,  
       CONNID LIKE SFLIGHT_CONNID,  
       FLDATE LIKE SFLIGHT_FLDATE,  
       PAYMENTSUM LIKE SFLIGHT_PAYMENTSUM,  
       END OF T_ITAB.  
  
DATA: ITAB TYPE T_ITAB OCCURS 100 WITH HEADER LINE.
```

- You declare an internal table using the **DATA** statement.
- To define an internal table with a header line, you can use one of the following possibilities:

You either use the fields of the table line between **BEGIN OF <tab> OCCURS <n> and END OF <tab>** or you refer to a structure type when defining the table. In the latter case, the **WITH HEADER LINE** addition sees to it that the header line is created.

The **OCCURS** parameter sees to it that an internal table is created, not a field string.

After **OCCURS**, you have to specify a numerical literal.

Note: When you know that your internal table will be smaller than 8 KB, you can specify the number of table lines in the OCCURS parameter.

SAP Document : ABAP/4 Reporting - Internal Tables

The memory area for this size is then allocated accordingly. This is of particular importance for nested structures. If the memory area is not sufficient, more blocks of 8 KB or 16 KB are created.

Note : If you do not know the size of the internal table beforehand, you can set the OCCURS parameter to 0.

Depending on the line width, blocks of 8 KB or 16 KB are created to accommodate the table lines. The new memory management stores the internal tables in the extended memory, a memory area you can address directly.

Filling an Internal Table with Header Line

- The **APPEND** statement stores the contents of the header line at the end of the internal table.
- The **COLLECT** statement includes the header line of an internal table as a new entry in the table or adds it to an existing entry with the same structure.
- ABAP/4 searches an entry in the table, which is identical in all fields other than fields of type P, I or F. If the system finds such an entry, it adds all header line fields of type P, I or F in the column accordingly to the corresponding fields of the table entry.
- If the system does not find such an entry, the contents of the header line is included as a new table entry at the table end.

Sorting an Internal Table

- To sort an internal table, use the **SORT** statement. If you do not specify a sort key, the table is sorted by fields ignoring fields of type P, I or F, in ascending order as declared.
- The additions BY <fieldname> and/or **ASCENDING** or **DESCENDING** allow you to sort by particular field. You can also determine the sort order and hierarchy (including type P, I, and F fields!). You can also set the sort key by using **SORT <itab> BY ... (fieldname)** at runtime.

SAP Document : ABAP/4 Reporting - Internal Tables

- You should restrict your sort criterion by using the BY parameter. ABAP/4 requires less main memory space to perform the sort procedure.

```
TABLES SFLIGHT.  
DATA ITAB TYPE T_ITAB OCCURS 100 WITH HEADER LINE.  
....  
* Fill Internal Table  
....  
SORT ITAB BY FLDATE CARRID CONNID.  
.  
SORT ITAB.  
SORT ITAB BY CARRID ASCENDING PAYMENTSUM  
DESCENDING.
```

Processing an Internal Table with Header

- You process an internal table using the **LOOP AT <tab>. ... ENDLOOP** loop statement.
- The ABAP/4 runtime system places the next table entry in the header line with each loop pass. When addressing table fields in the program, these are the header line fields.
- When working with tables without a header line, you need a work area, which corresponds to the structure of a table line.
- With **LOOP AT <tab> INTO <wa>** the internal table entries are transferred one after another to the work area <wa>.

SAP Document : ABAP/4 Reporting - Internal Tables

```
TABLES SFLIGHT.  
DATA ITAB TYPE T_ITAB OCCURS 100 WITH HEADER LINE.  
....  
* Fill Internal Table  
* Output  
LOOP AT ITAB.  
    WRITE: / ITAB-CARRID COLOR COL_KEY,  
           ITAB-CONNID COLOR COL_KEY,  
           ITAB_FLDATE COLOR COL_KEY,  
           ITAB_PAYMENTSUM COLOR COL_NORMAL.  
ENDLOOP.
```

- When processing an internal table with the **LOOP** statement, you can perform control level processing with the **AT FIRST**, **AT NEW**, **AT END OF** and **AT LAST** statements.
- You have to conclude the statement sequence for each **AT** statement with **ENDAT**. This processing block is executed whenever the contents of <field> or of a component defined before <field> (that is, to the left of <field> in the record) change.
- If you use the **SUM** statement in an **AT ... ENDAT** block, the system calculates totals for the corresponding control level and/or grand total (for **AT FIRST** or **AT LAST**).

The system totals all fields of type P, I and F according to columns and places them in the corresponding fields in the header line.

The total fields have the same length as the fields to be totaled. This can result in an overflow. Increase the length of the fields affected.

You can use the SUM statement with control level headers and with control level footers.

- When formulating control level processing, you have to follow the above order concerning the individual control levels within the LOOP statement.
- Single record processing occurs outside the AT ... ENDAT statement.

SAP Document : ABAP/4 Reporting - Internal Tables

Intermediate Dataset

When working with several internal tables linked via key fields, you can avoid data redundancy to a large degree. In the example above, table **ITAB_SPFLI** is sorted by *CITYFROM CARRID CONNID*, table **ITAB_SFLIGHT** by *FLDATE*. During the processing of table **ITAB_SPFLI**, the corresponding records from **ITAB_SFLIGHT** are read and output for each entry. This requires extensive programming work.

Because of the disadvantages stated above, ABAP/4 provides another sorting procedure. With different record types, you can generate intermediate datasets. To do so, you have to define the required record types (the name is freely selectable; here: **CONNECTIONS**, **FLIGHTS**). Each record is automatically preceded by the sort key (record type **HEADER**, name is pre-defined). Fields with no values are set to Hex **00**. When sorting the dataset, these fields always precede the fields containing with values, irrespective of the sort sequence.

To define the record types, proceed as follows:

1. Declare the required record type (field groups, **FIELD-GROUPS**). The field group **HEADER** must always be specified. This group must contain all fields you want to use in the sort procedure.
2. Assign the required fields to the field groups (**INSERT ... INTO...**).

When executing the **EXTRACT** statement, the fields of the header and the fields of the corresponding field groups from the relevant work areas are supplied with contents. Database fields of dependent segments (e.g. **SFLIGHT-FLDATE** at the event **GET SPFLI**) are set to Hex **00**. Header and field group are written as a record to the intermediate dataset.

Sorting and Processing an Extract

- Intermediate datasets generated with the **EXTRACT** statement are typically sorted and processed at the **END-OF-SELECTION** event.
- You edit the sorted dataset within a **LOOP** pass during which the system provides you with record by record.
- Within **LOOP** you can use statements (**AT...**) which on the one hand address the single records of the sorted dataset (**AT <fieldgroup>**) and on the other hand allow control break processing (**AT FIRST, AT LAST, AT NEW <sortfield>, AT END OF <sortfield>**).

SAP Document : ABAP/4 Reporting - Internal Tables

- For group processing, the datasets have to be sorted correspondingly.
- The **AT** ... statements are control statements. The corresponding processing block must therefore conclude with **ENDAT**.
- When using the **SORT** statement without additions, the system sorts by the entire header. You are recommended to restrict the search criterion by using the BY parameter. You will improve the performance and need less working memory.
- Within **AT END OF ... ENDAT** and **AT LAST. ... ENDAT** you can address the fields **SUM**(<numfield>) and **CNT** (<headerfield>):
- **SUM**(<numfield>) calculates totals by groups. The work field required is provided automatically and initialized with 0 at each control break.
- **CNT**(<headerfield>) is increased by 1 when the contents of the specified header field changes. With any control break, the field addressed with **CNT, which is responsible for the control break**, must be subordinated in the sort order.