



## Aggregation mit VALUE, GROUPS und FOR

Als SAP-Programmierer hat man häufig mit Datenstrukturen zu tun, die aus einer Belegnummer und einer Positionsnummer bestehen. Hieraus ergibt sich dann häufig die Aufgabenstellung, alle Belegnummern in einer separaten Tabelle zu sammeln, also auf Belegnummer zu aggregieren.

Der folgende Code-Schnipsel erledigt das für eine Tabelle mit Hilfe der VALUE-Anweisung in Verbindung mit FOR und GROUPS.

Zuerst die Datenstruktur der Tabelle, die aus Belegnummer (Document = D) und Position ( Item = IT) besteht.

```
TYPES: BEGIN OF _dit,  
        doc TYPE n LENGTH 10,  
        itm TYPE posnr,  
END OF _dit,  
_dits TYPE SORTED TABLE OF _dit WITH UNIQUE KEY doc itm.
```

Zum Testen fülle ich die Tabelle mit ein paar Testdaten:

```
DATA(dits) = VALUE _dits(  
  ( doc = 100 itm = 1 )  
  ( doc = 100 itm = 2 )  
  ( doc = 102 itm = 2 )  
  ( doc = 103 itm = 4 )  
  ( doc = 103 itm = 5 ) ).
```

Die VALUE-Anweisung macht übrigens einen INSERT INTO TABLE, was bedeutet, dass die einzufügenden Werte nicht sortiert vorliegen müssen!

Die folgende Anweisung verursacht also keinen Shortdump:

```
DATA(dits) = VALUE _dits( ( doc = 200 ) ( doc = 100 ) ).
```

*Info zur Anwendung VALUE*

Die Zielstruktur, sieht wie folgt aus:

```
TYPES: BEGIN OF _doc,  
       no TYPE n LENGTH 10,  
       END OF _doc,  
       _docs TYPE SORTED TABLE OF _doc WITH UNIQUE KEY no.
```

Das Coding für die Aggregation lautet folgendermaßen:

```
DATA(docs) = VALUE _docs(  
  FOR GROUPS d OF line IN dits  
  GROUP BY line-doc ( no = d ) ).
```

Es ist so kompakt, dass man es auch noch vertretbar in einer Zeile stehen lassen kann.

```
DATA(docs) = VALUE _docs( FOR GROUPS d OF line IN dits GROUP BY line-doc ( no  
= d ) ).
```

## **DATA vs. FIELD-SYMBOLS**

Die Syntax lässt zwei Schreibweisen zu: einmal mit einer Workarea (DATA) und einmal mit Feldsymbolen. In meinem Beispiel unterscheidet sich die Schreibweise also nur für ein Element:

- LINE (DATA)
- <LINE> (Field-Symbols)

Die Verwendung von Field-Symbols ist generell schneller. Die Performance hängt stark davon ab, wie breit die Tabelle ist. Bei meinem Beispiel, das nur die beiden Felder DOC und ITM enthält, ist der Vorsprung von Field-Symbols nur minimal. Ändere ich die Breite der Struktur jedoch, indem ich z.B.

die Tabelle VBAP einbinde (ca. 3700 Zeichen breit), dann verlängert sich die Laufzeit bei Field-Symbols um ca. 1/3. Bei DATA ist die Laufzeit mehr als drei Mal so hoch!

Empfehlung also: Verwende immer die Variante mit Field-Symbols! Einen praktischen Nutzen habe ich bisher noch nicht finden können. Normalerweise kann man bei der Verwendung von Feldsymbolen mit ihrer Hilfe die zugrunde liegenden Daten direkt verändern. Bei dieser Variante mit GROUPS ist das meines Wissens jedoch nicht möglich.

## **Gruppierung mit SELECT**

Seit Release 7.52 ist es möglich, einen SELECT auf eine interne Tabelle abzusetzen. Und auch so kann man natürlich gruppieren:

```
SELECT DISTINCT doc AS no  
FROM @dits AS itab  
INTO TABLE @DATA(docs).
```

## **Gruppierung mit LOOP**

In diesem Artikel habe ich beschrieben, wie die Gruppierung bei einer LOOP-Schleife funktioniert.

Loop At ITAB Group By