

LOOP

`[CLOSE] LOOP [(r)]`

Dieses Kapitel behandelt folgende Themen:

- Funktion
 - Einschränkung
 - Syntax-Beschreibung
 - Beispiele
-

Funktion

Das Statement `LOOP` dient dazu, eine Verarbeitungsschleife zu schließen. Es bewirkt, dass der aktuelle Schleifendurchlauf beendet wird und die Kontrolle wieder an den Anfang der Schleife übergeben wird.

Sobald die Verarbeitungsschleife, auf die sich das `LOOP`-Statement bezieht, beendet ist (d.h. sobald alle Datensätze verarbeitet und alle Schleifendurchläufe ausgeführt sind), wird die Verarbeitung mit dem auf das `LOOP`-Statement folgenden Statement fortgesetzt.

Referenzierung von Datenbankvariablen

Neben dem Schließen der Schleife(n) bewirkt das `LOOP`-Statement, dass alle Referenzierungen von Feldern, die in `FIND`-, `FIND FIRST`-, `FIND UNIQUE`-, `READ`- und `GET`-Statements innerhalb der geschlossenen Schleife(n) verwendet werden, eliminiert werden.

Ein Feld, das in einem View enthalten ist, kann auch außerhalb einer mit `LOOP` geschlossenen Schleife referenziert werden, und zwar indem bei der Referenzierung der View-Name angegeben wird.

Einschränkung

- Dieses Statement gilt nur für Reporting Mode.
- Ein `LOOP`-Statement darf nicht an eine logische Bedingung wie etwa ein `IF`- oder `AT BREAK`-Statement geknüpft werden.

Syntax-Beschreibung

| | |
|-----------------------------|--|
| LOOP (<i>r</i>) | Statement-Referenzierung: Sollen mehrere Schleifen geschlossen werden, so kann ein bestimmtes Statement per Statement-Label oder Sourcecode- Zeilennummer referenziert werden (Notation (<i>r</i>)); in diesem Falle werden durch das LOOP-Statement dann die referenzierte Verarbeitungsschleife sowie alle innerhalb der referenzierten Schleife befindlichen Schleifen geschlossen. |
|-----------------------------|--|

Anmerkung:

Im Reporting Mode werden durch ein END-Statement alle noch aktiven Verarbeitungsschleifen, die noch nicht explizit durch ein LOOP-Statement beendet wurden, automatisch geschlossen.

Beispiele

Beispiel 1

```
0010 FIND ...
0020  READ ...
0030   READ ...
0040 LOOP (0010)    /* closes all loops
```

Beispiel 2

```
0010 FIND ...
0020  READ ...
0030   READ ...
0040    LOOP        /* closes loop initiated on line 0030
0050  LOOP          /* closes loop initiated on line 0020
0060 LOOP           /* closes loop initiated on line 0010
```