

BACKOUT TRANSACTION

BACKOUT [TRANSACTION]

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Einschränkung
- Datenbank-spezifische Anmerkungen
- Beispiel

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Verwandte Statements: ACCEPT/REJECT | AT BREAK | AT START OF DATA | AT END OF DATA | BEFORE BREAK PROCESSING | DELETE | END TRANSACTION | FIND | GET | GET SAME | GET TRANSACTION DATA | HISTOGRAM | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | STORE | UPDATE

Gehört zur Funktionsgruppe: *Datenbankzugriffe und Datenbankänderungen*

Funktion

Das BACKOUT TRANSACTION-Statement bewirkt, dass alle Datenbankänderungen, die während der laufenden, noch nicht abgeschlossenen logischen Transaktion ausgeführt wurden, rückgängig gemacht werden; außerdem bewirkt es, dass alle während der Transaktion gehaltenen Datensätze wieder freigegeben werden.

Das BACKOUT TRANSACTION-Statement wird nur ausgeführt, wenn eine Datenbanktransaktion unter Kontrolle von Natural stattgefunden hat. Für welche Datenbanken das Statement ausgeführt wird, hängt davon ab, wie der Profilparameter ET (Ausführung von END/BACKOUT TRANSACTION-Statements) gesetzt ist:

- Ist ET=OFF gesetzt, wird das Statement nur für die von der Transaktion betroffene Datenbank ausgeführt.
- Ist ET=ON gesetzt, wird das Statement für alle Datenbanken ausgeführt, die seit der letzten Ausführung eines BACKOUT TRANSACTION- oder END TRANSACTION-Statements referenziert wurden.

BACKOUT TRANSACTION über Abbruchkommando

Unterbricht der Benutzer mit einem Natural-Terminalkommando (Kommando %% oder CLEAR-Taste) eine gerade aktive Natural-Operation, dann führt Natural ein BACKOUT TRANSACTION-Statement aus.

Weitere Informationen siehe Terminalkommando %% in der *Terminalkommando*-Dokumentation.

Weitere Informationen

Weitere Informationen zur Natural-Transaktionslogik und zum Beenden/Abbrechen einer logischen Transaktion finden Sie im Kapitel *Datenbankzugriffe* im *Leitfaden zur Programmierung*.

Einschränkung

Mit Entire System Server kann dieses Statement nicht verwendet werden.

Datenbank-spezifische Anmerkungen

SQL-Datenbanken	Da die meisten SQL-Datenbanken bei Beendigung einer logischen Arbeitseinheit alle Cursor schließen, darf ein BACKOUT TRANSACTION-Statement nicht innerhalb einer datenbankverändernden Verarbeitungsschleife stehen, sondern muss nach einer solchen platziert werden.
XML-Datenbanken	Ein BACKOUT TRANSACTION-Statement darf nicht in einer datenbankverändernde Programmschleife, sondern muss nach einer solchen platziert werden.

Beispiel

```

** Example 'BOTEX1': BACKOUT TRANSACTION
**
** CAUTION: Executing this example will modify the database records!
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 DEPT
  2 LEAVE-DUE
  2 LEAVE-TAKEN
*
1 #DEPT (A6)
1 #RESP (A3)
END-DEFINE
*
LIMIT 3
INPUT 'DEPARTMENT TO BE UPDATED:' #DEPT
IF #DEPT = ' '
  STOP
END-IF
*
FIND EMPLOY-VIEW WITH DEPT = #DEPT
  IF NO RECORDS FOUND
    REINPUT 'NO RECORDS FOUND'
  END-NOREC
  INPUT 'NAME:          ' NAME (AD=O) /
        'LEAVE DUE:      ' LEAVE-DUE (AD=M) /
        'LEAVE TAKEN:' LEAVE-TAKEN (AD=M)

  UPDATE
END-FIND

```

```
*  
INPUT 'UPDATE TO BE PERFORMED? YES/NO:' #RESP  
DECIDE ON FIRST #RESP  
  VALUE 'YES'  
  END TRANSACTION  
  VALUE 'NO'  
  BACKOUT TRANSACTION  
  NONE  
  REINPUT 'PLEASE ENTER YES OR NO'  
END-DECIDE  
*  
END
```

Ausgabe des Programms BOTEX1:

DEPARTMENT TO BE UPDATED: **MGMT30**

Ergebnis für die Abteilung MGMT30:

```
NAME:          POREE  
LEAVE DUE:     45  
LEAVE TAKEN:   31
```

Aufforderung zur Bestätigung:

UPDATE TO BE PERFORMED YES/NO: **NO**