

END TRANSACTION

`END [OF] TRANSACTION [operand1 ...]`

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Einschränkung
- Syntax-Beschreibung
- Betroffene Datenbanken
- Datenbank-spezifische Anmerkungen
- Beispiele

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 | BACKOUT TRANSACTION | BEFORE BREAK PROCESSING | DELETE | 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 Statement `END TRANSACTION` dient dazu, das Ende einer logischen Transaktion zu markieren. Eine logische Transaktion ist die kleinste (vom Benutzer definierte) logische Arbeitseinheit, die vollständig ausgeführt werden muss, damit die logische Konsistenz der Daten auf der Datenbank gewährleistet ist.

Die erfolgreiche Ausführung eines `END TRANSACTION`-Statements bewirkt, dass alle im Verlaufe der Transaktion durchgeführten Datenänderungen physisch auf der Datenbank durchgeführt worden sind (bzw. werden) und von einem anschließenden Abbruch, sei er durch den Benutzer, Natural, die Datenbank oder das Betriebssystem herbeigeführt, nicht mehr beeinflusst werden können. Wenn das `END TRANSACTION`-Statement nicht erfolgreich ausgeführt wird, d.h. wenn die logische Transaktion nicht vollständig ausgeführt ist, werden alle im Laufe der Transaktion bereits durchgeführten Datenänderungen automatisch wieder rückgängig gemacht.

`END TRANSACTION` bewirkt außerdem, dass alle während der Transaktion im Hold-Status gehaltenen Datensätze wieder freigegeben werden.

Die Ausführung des `END TRANSACTION`-Statements kann an eine logische Bedingung geknüpft werden.

Weitere Informationen hierzu finden Sie im Kapitel *Datenbankzugriffe* im *Leitfaden zur Programmierung*.

Einschränkung

Das Statement `END TRANSACTION` kann nicht mit Entire System Server benutzt werden.

Syntax-Beschreibung

Operanden-Definitionstabelle:

Operand	Mögliche Struktur		Mögliche Formate										Referenzierung erlaubt	Dynam. Definition						
<i>operand1</i>	C	S			N	A	U	N	P	I	F	B	D	T					ja	nein

Syntax-Element-Beschreibung:

<i>operand1</i>	<p>Speicherung von Transaktionsdaten:</p> <p>Bei einer Transaktion auf einer Adabas-Datenbank oder auf einer DL/I-Datenbank in einer batch-orientierten BMP-Region (nur in IMS-Umgebungen) können Sie mit diesem Statement auch transaktionsbezogene Daten speichern. Diese Daten dürfen maximal 2000 Bytes lang sein und können mit einem <code>GET TRANSACTION DATA</code>-Statement wieder gelesen werden.</p> <p>Die Transaktionsdaten werden auf die mit dem Profilparameter <code>ETDB</code> angegebene Datenbank geschrieben.</p> <p>Wenn der <code>ETDB</code>-Parameter nicht gesetzt ist, werden die Transaktionsdaten auf die mit dem Profilparameter <code>UDB</code> angegebene Datenbank geschrieben. Ausnahme: Auf Großrechnern werden die Transaktionsdaten auf die Datenbank geschrieben, auf der sich die Natural-Security-Systemdatei (<code>FSEC</code>) befindet (ist <code>FSEC</code> nicht angegeben, dann ist sie identisch mit der Natural-Systemdatei <code>FNAT</code>; ist Natural Security nicht installiert, dann werden die Transaktionsdaten auf die Datenbank geschrieben, auf der sich <code>FNAT</code> befindet).</p>
-----------------	---

Betroffene Datenbanken

Ein `END TRANSACTION`-Statement ohne Transaktionsdaten (d.h. ohne *operand1*) wird nur ausgeführt, wenn eine Datenbanktransaktion unter Kontrolle von Natural stattgefunden hat. Für welche Datenbanken das Statement ausgeführt wird, hängt davon ab, wie der Natural-Profilparameter `ET` gesetzt ist.

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

Ein `END TRANSACTION`-Statement mit Transaktionsdaten (d.h. mit Angabe von *operand1*) wird immer ausgeführt, und die Transaktionsdaten werden wie unten beschrieben auf einer bestimmten Datenbank gespeichert. Für welche Datenbanken das Statement außerdem ausgeführt wird, hängt vom `ET`-Parameter (siehe oben) ab.

Datenbank-spezifische Anmerkungen

DL/I Databases	Da die PSB-Initialisierung durch eine Syncpoint-Anfrage beendet wird, speichert Natural die PSB-Position, bevor das END TRANSACTION-Statement ausgeführt wird. Bevor das nächste Kommando ausgeführt wird, re-initialisiert Natural den PSB und versucht, die PCB-Position so zu setzen, wie sie vor dem END TRANSACTION-Statement war. Die PCB-Position kann vorverlegt werden, falls in der Zeit zwischen END TRANSACTION und dem nachfolgenden Kommando ein adressiertes Segment gelöscht wurde.
VSAM Databases	Informationen zur Transaktionslogik, die beim Zugriff auf VSAM gilt, finden Sie in der <i>Natural for VSAM</i> -Dokumentation.
SQL Databases	Da die meisten SQL-Datenbanken bei Beendigung einer logischen Arbeitseinheit alle Cursor schließen, darf ein END TRANSACTION-Statement nicht innerhalb einer datenbankverändernden Verarbeitungsschleife stehen, sondern muss nach einer solchen plaziert werden.

Beispiele

- Beispiel 1 — END TRANSACTION-Statement
- Beispiel 2 — END TRANSACTION-Statement mit ET-Daten

Beispiel 1 — END TRANSACTION-Statement

```

** Example 'ETREX1': END TRANSACTION
**
** CAUTION: Executing this example will modify the database records!
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 CITY
  2 COUNTRY
END-DEFINE
*
FIND EMPLOY-VIEW WITH CITY = 'BOSTON'
  ASSIGN COUNTRY = 'USA'
  UPDATE
  END TRANSACTION
/*
AT END OF DATA
  WRITE NOTITLE *NUMBER 'RECORDS UPDATED'
END-ENDDATA
/*
END-FIND
END

```

Ausgabe des Programms ETREX1:

```
7 RECORDS UPDATED
```

Beispiel 2 — END TRANSACTION-Statement mit ET-Daten

```

** Example 'ETREX2': END TRANSACTION (with ET data)
**
** CAUTION: Executing this example will modify the database records!
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
  2 FIRST-NAME
  2 CITY
*
1 #PERS-NR (A8) INIT <' '>
END-DEFINE
*
REPEAT
  INPUT 'ENTER PERSONNEL NUMBER TO BE UPDATED:' #PERS-NR
  IF #PERS-NR = ' '
    ESCAPE BOTTOM
  END-IF
  /*
  FIND EMPLOY-VIEW PERSONNEL-ID = #PERS-NR
  INPUT (AD=M)   NAME / FIRST-NAME / CITY
  UPDATE
  END TRANSACTION #PERS-NR
  END-FIND
  /*
END-REPEAT
END

```

Ausgabe des Programms ETREX2:

```
ENTER PERSONNEL NUMBER TO BE UPDATED: 20027800
```

Nach Änderung und Bestätigung der Personalnummer:

```
NAME LAWLER
FIRST-NAME SUNNY
CITY MILWAUKEE
```