

UPDATE

Structured Mode-Syntax

```
UPDATE [RECORD] [IN] [STATEMENT] [(r)]
```

Reporting Mode-Syntax

```
UPDATE [RECORD] [IN] [STATEMENT] [(r)]
      [
        SET
        WITH
        USING
      ] { SAME [RECORD]
        { operand1=operand2 } ... }
```

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Einschränkungen
- Datenbankspezifische Anmerkungen
- Syntax-Beschreibung
- 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 | BACKOUT TRANSACTION | BEFORE BREAK PROCESSING | DELETE | END TRANSACTION | FIND | GET | GET SAME | GET TRANSACTION DATA | HISTOGRAM | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | STORE

Gehört zur Funktionsgruppe: *Datenbankzugriffe und Datenbankänderungen*

Funktion

Das UPDATE-Statement dient dazu, die in der Datenbank gespeicherten Werte eines oder mehrerer Felder eines Datensatzes zu verändern. Der betreffende Datensatz muss vorher mit einem FIND-, GET- oder READ-Statement (oder bei Adabas auch mit einem STORE-Statement) ausgewählt werden.

Hold-Status

Das UPDATE-Statement bewirkt, dass jeder mit dem betreffenden FIND-Statement oder READ-Statement gelesene Datensatz in den "Hold"-Status gestellt wird.

Die *Hold-Logik* ist im *Leitfaden zur Programmierung* beschrieben.

Einschränkungen

- Das UPDATE-Statement darf nicht in derselben Sourcecode-Zeile stehen wie das Statement, mit dem der zu aktualisierende Datensatz ausgewählt wird.
- Mit Entire System Server ist das UPDATE-Statement nicht verfügbar.

Datenbankspezifische Anmerkungen

DL/I	<p>Das UPDATE-Statement kann dazu verwendet werden, ein Segment einer DL/I-Datenbank zu aktualisieren. Erforderlichenfalls wird die Länge des Segments vergrößert, um alle mit dem UPDATE-Statement angegebenen Felder aufnehmen zu können.</p> <p>Ist ein multiples Feld oder eine Periodengruppe mit variabler Länge definiert, so werden nur die im UPDATE-Statement angegebenen Ausprägungen in das Segment geschrieben.</p> <p>Der DL/I-AIX-Feldname darf in einem UPDATE-Statement nicht verwendet werden; AIX-Felder können aktualisiert werden, indem man sich auf das Sourcefeld bezieht, welches das betreffende AIX-Feld enthält.</p> <p>DL/I-Folgefelder können aufgrund von DL/I- Beschränkungen nicht aktualisiert werden.</p> <p>Wegen GSAM-Beschränkungen ist das UPDATE-Statement nicht auf GSAM-Datenbanken anwendbar.</p>
VSAM	<p>VSAM-Primärschlüssel können aufgrund von VSAM-Beschränkungen nicht aktualisiert werden.</p>
SQL	<p>Mit dem UPDATE-Statement können Sie eine Reihe einer Datenbanktabelle aktualisieren. Das UPDATE-Statement entspricht dem SQL-Statement UPDATE WHERE CURRENT OF CURSOR (Positioned Update), d.h. nur die zuletzt gelesene Reihe kann aktualisiert werden.</p> <p>Auf Großrechnern werden nur Spalten (Felder) aktualisiert, die innerhalb des Programms geändert wurden, sowie Spalten, die außerhalb des Programms (z.B. als Eingabefelder in Maps) geändert worden sein könnten (aber nicht notwendigerweise auch geändert wurden). Auf allen anderen Plattformen werden alle Spalten aktualisiert.</p> <p>Bei den meisten SQL-Datenbanken kann eine mit FIND SORTED BY oder READ LOGICAL gelesene Reihe nicht aktualisiert werden.</p>

Syntax-Beschreibung

Operanden-Definitionstabelle:

Operand	Mögliche Struktur			Mögliche Formate								Referenzierung erlaubt	Dynam. Definition		
<i>operand1</i>		S	A		A	N	P	I	F	B	D	T	L	nein	nein
<i>operand2</i>	C	S	A		A	N	P	I	F	B	D	T	L	ja	nein

Syntax-Element-Beschreibung:

(r)	<p>Statement-Referenzierung:</p> <p>Mit der Notation (<i>r</i>) können Sie das Statement referenzieren, mit dem der Datensatz, der aktualisiert werden soll, gelesen wurde. <i>r</i> kann als Statement-Label oder Sourcecode-Zeilenummer angegeben werden. Falls keine Referenzierung erfolgt, bezieht sich das UPDATE-Statement auf die innerste aktive READ- bzw. FIND-Verarbeitungsschleife.</p> <p>Ist keine READ- oder FIND-Schleife aktiv, bezieht es sich auf das letzte vorhergehende GET-Statement (bzw. STORE-Statement).</p> <p>Anmerkung: Das UPDATE-Statement muss innerhalb der READ- bzw. FIND-Schleife stehen, auf die es sich bezieht.</p>
USING SAME	<p>USING SAME-Klausel:</p> <p>Diese Klausel ist nicht erlaubt, wenn ein DEFINE DATA-Statement verwendet wird, da sich in diesem Fall das UPDATE-Statement immer auf den gesamten im DEFINE DATA-Statement definierten View bezieht.</p> <p>Das Layout des Satzpuffers oder des Formatpuffers kann mit dem OBTAIN-Statement deklariert werden.</p> <p>Mit USING SAME geben Sie im Reporting Mode an, dass dieselben Felder aktualisiert werden sollen, die mit dem Statement, welches vom UPDATE-Statement referenziert wird, gelesen wurden; dies bedeutet, dass zur Aktualisierung der Felder die Werte verwendet werden, die den Datenbankfeldern zuletzt zugeordnet waren. Ist kein neuer Wert zugeordnet worden, wird der alte verwendet.</p> <p>Wenn das zu aktualisierende Feld ein Array-Bereich eines multiplen Feldes oder einer Periodengruppe ist und Sie einen variablen Index für diesen Array-Bereich verwenden, wird der zuletzt gültige Array-Bereich aktualisiert. Wenn die Indexvariable modifiziert wird, nachdem der Datensatz gelesen wurde, aber bevor das UPDATE USING SAME- (Reporting Mode) bzw. UPDATE-Statement (Structured Mode) ausgeführt wird, bedeutet dies, dass ein anderer Array-Bereich aktualisiert wird als gelesen wurde.</p>

SET/WITH <i>operand1=operand2</i>	SET/WITH-KLausele: <p>Mit dieser Klausel können Sie im Reporting Mode die Felder angeben, die geändert werden sollen, sowie die neuen Werte für diese Felder.</p> <p>Diese Klausel ist nicht erlaubt, wenn ein DEFINE DATA-Statement verwendet wird, da sich in diesem Fall das UPDATE-Statement immer auf den gesamten im DEFINE DATA-Statement definierten View bezieht.</p> <p>Anmerkung: Bei DL/I-Datenbanken: Wenn diese Klausel verwendet wird, dürfen nur I/O-sensitive Felder angegeben werden. Ein Segmentfolgefeld darf nicht mit einem UPDATE-Statement verändert werden (verwenden Sie hierzu die Statements DELETE und STORE).</p>
---	---

Beispiel

```

** Example 'UPDEX1S': UPDATE (structured mode)
**
** CAUTION: Executing this example will modify the database records!
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 CITY
*
1 #NAME (A20)
END-DEFINE
*
INPUT 'ENTER A NAME:' #NAME (AD=M)
IF #NAME = ' '
  STOP
END-IF
*
FIND EMPLOY-VIEW WITH NAME = #NAME
  IF NO RECORDS FOUND
    REINPUT WITH 'NO RECORDS FOUND' MARK 1
  END-NOREC
  INPUT 'NAME:          ' NAME (AD=O) /
        'FIRST NAME:' FIRST-NAME (AD=M) /
        'CITY:           ' CITY (AD=M)

  UPDATE
  END TRANSACTION
END-FIND
*
END

```

Ausgabe des Programms SUBEX1S:

```
ENTER A NAME: BROWN
```

Nach Eingabe und Bestätigung des Namens:

```
NAME: BROWN
FIRST NAME: KENNETH
CITY: DERBY
```

Äquivalentes Reporting-Mode-Beispiel: UPDEX1R.