UPDATE - SQL

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Beispiele

Funktion

Das SQL-Statement UPDATE dient dazu, Reihen in einer Tabelle zu ändern, ohne einen Cursor zu verwenden ("Searched" UPDATE), oder Spalten in der Reihe zu ändern, auf die der Cursor zeigt ("Positioned" UPDATE).

Syntax-Beschreibung

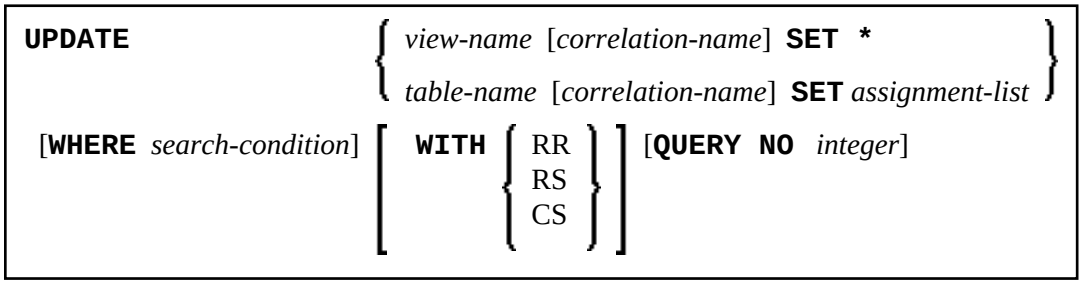
Es sind zwei unterschiedliche Strukturen möglich:

- Syntax 1 — Searched UPDATE
- Syntax 2 — Positioned UPDATE

Syntax 1 — Searched UPDATE

Searched UPDATE ist ein eigenständiges Statement, das unabhängig von einem SELECT-Statement verwendet werden kann. Mit einem einzigen Statement können Sie keine, eine, mehrere oder alle Reihen einer Tabelle ändern. Welche Reihen geändert werden, bestimmen Sie mit der Suchbedingung (*search-condition*), die auf die Tabelle angewendet wird. Außerdem ist es möglich, einem Tabellen- oder View-Namen einen *correlation-name* zuzuweisen.

Anmerkung:
Die Anzahl der Reihen, die mit einem Searched UPDATE tatsächlich geändert wurden, kann mit der Systemvariablen *ROWCOUNT (siehe *Systemvariablen*-Dokumentation) ermittelt werden.

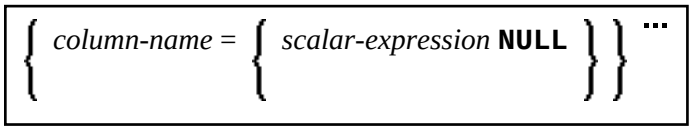


Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Syntax-Elementbeschreibung – Syntax 1:

view-name	**View-Name:** *view-name* ist jeweils der Name eines im `DEFINE DATA`-Statement definierten Natural-Views. Siehe *view-name* im Abschnitt *Grundlegende Syntaxbestandteile*.
SET	**SET-Klausel:** Wenn sich die Änderungen auf einen View beziehen, müssen Sie in der `SET`-Klausel einen Stern (*) angeben, da alle Spalten des Views geändert werden müssen. Wenn sich die Änderungen auf eine Tabelle beziehen, können Sie in der `SET`-Klausel entweder eine *assignment-list* angeben oder den Namen eines Views, der die zu ändernden Spalten enthält.
assignment-list	Siehe *Assignment List* weiter unten.
WHERE ***search-condition***	**WHERE-Klausel:** In der `WHERE`-Klausel geben Sie eine Suchbedingung (*search-condition*) an, die bestimmt, welche Reihen geändert werden sollen. Wenn Sie keine `WHERE`-Klausel angeben, wird die gesamte Tabelle geändert.

assignment-list



In einer *assignment-list* können Sie einer oder mehreren Spalten Werte zuweisen. Ein Wert kann entweder eine *scalar-expression* oder `NULL` sein. Weitere Informationen siehe *Scalar Expressions*.

Wenn Sie `NULL` zuweisen, bedeutet dies, dass das betreffende Feld keinen Wert enthalten soll (auch nicht den Wert 0 oder Leerzeichen).

SQL Extended Set

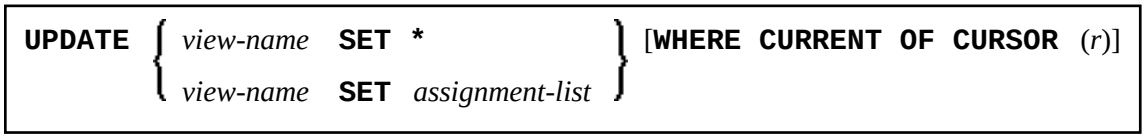
Die folgenden Syntax-Elemente gehören zum SQL Extended Set:

correlation-name	Das Element *correlation-name* ist ein Alias-Name für *table-name*. Weitere Informationen siehe *correlation-name* (im Abschnitt *Grundlegende Syntaxbestandteile*).
WITH	**WITH Isolation Level-Klausel:** Diese Klausel ermöglicht die explizite Angabe des beim Suchen der zu ändernden Reihen benutzten Isolation Level. Diese Klausel gilt nur für DB2-Datenbanken. Bei anderen Datenbanken verursacht sie einen Laufzeitfehler.
QUERYNO ***integer***	**QUERYNO-Klausel:** Diese Klausel wird zurzeit nicht unterstützt und wird ignoriert.

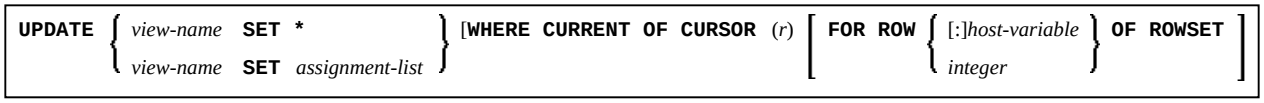
Syntax 2 — Positioned UPDATE

Ein Positioned UPDATE bezieht sich auf einen Cursor innerhalb einer Datenbankschleife. Es muss daher dieselbe Tabelle bzw. denselben View referenzieren wie das entsprechende SELECT-Statement, sonst erfolgt eine Fehlermeldung. Ein Positioned UPDATE kann nur bei cursor-orientierter Selektion verwendet werden.

Common Set-Syntax



Extended Set-Syntax



Syntax-Elementbeschreibung – Syntax 2:

view-name	**View-Name:** Bezieht sich auf den Namen eines im `DEFINE DATA`-Statement definierten Natural-Views. Weitere Informationen siehe *view-name* unter *Grundlegende Syntaxbestandteile*.
SET * **SET *assignment-list***	**SET-Klausel:** Wenn ein View zum Ändern spezifiziert wurde, muss ein Stern (*) in der `SET`-Klausel angegeben werden, weil alle Spalten des Views aktualisiert werden müssen. Wurde eine Tabelle zum Ändern spezifiziert, muss die `SET`-Klausel entweder eine *assignment-list* oder den Namen des Views enthalten, der die zu ändernden Spalten enthält.
WHERE CURRENT OF CURSOR (*r*)	**Statement-Referenz:** Die Notation `(r)` dient zur Referenzierung des Statements, das zur Selektion der zu ändernden Reihe verwendet wurde. Wird keine Statement-Referenz angegeben, bezieht sich das `UPDATE`-Statement auf die jeweils innerste aktive Verarbeitungsschleife, mit der ein Datensatz ausgewählt wurde.
FOR ROW … OF ROWSET	**FOR ROW … OF ROWSET-Klausel:** Diese Klausel gehört zum SQL Extended Set. Die optionale `FOR ROW ... OF ROWSET`-Klausel für Positioned `SQL UPDATE`-Statements gibt an, welche Reihe des aktuellen Rowset aktualisiert werden muss. Sie sollte nur angegeben werden, wenn das `UPDATE`-Statement sich auf ein `SELECT`-Statement bezieht, das die Rowset-Positionierung verwendet und das Spalten-Arrays in der *INTO-Klausel* hat. Wenn diese Klausel weggelassen wird, werden alle Reihen des aktuellen Rowset durch die Werte in der *assignment-list* überschrieben. Diese Klausel kann nicht angegeben werden, wenn *view-name* `SET *` angegeben wird.

Beispiele

- Beispiel 1 - Searched UPDATE
- Beispiel 2 - Searched UPDATE mit assignment-list

- Beispiel 3 - Positioned UPDATE
- Beispiel 4 - Positioned UPDATE mit assignment-list

Beispiel 1 - Searched UPDATE

```
DEFINE DATA LOCAL
1 PERS VIEW OF SQL-PERSONNEL
2 NAME
2 AGE
...
END-DEFINE
...
ASSIGN AGE = 45
ASSIGN NAME = 'SCHMIDT'
UPDATE PERS SET * WHERE NAME = 'SCHMIDT'
...
```

Beispiel 2 - Searched UPDATE mit *assignment-list*

```
DEFINE DATA LOCAL
1 PERS VIEW OF SQL-PERSONNEL
2 NAME
2 AGE
...
END-DEFINE
...
UPDATE SQL-PERSONNEL SET AGE = AGE + 1 WHERE NAME = 'SCHMIDT'
...
```

Beispiel 3 - Positioned UPDATE

```
DEFINE DATA LOCAL
1 PERS VIEW OF SQL-PERSONNEL
2 NAME
2 AGE
...
END-DEFINE
...
SELECT * INTO PERS FROM SQL_PERSONNEL WHERE NAME = 'SCHMIDT'
COMPUTE AGE = AGE + 1
UPDATE PERS SET * WHERE CURRENT OF CURSOR
END-SELECT
...
```

Beispiel 4 - Positioned UPDATE mit *assignment-list*

```
DEFINE DATA LOCAL
1 PERS VIEW OF SQL-PERSONNEL
2 NAME
2 AGE
...
END-DEFINE
...
SELECT * INTO PERS FROM SQL-PERSONNEL WHERE NAME = 'SCHMIDT'
UPDATE SQL-PERSONNEL SET AGE = AGE + 1 WHERE CURRENT OF CURSOR
END-SELECT
...
```