

# UPDATE - SQL

This chapter covers the following topics:

- Function
- Syntax 1 - Searched UPDATE
- Syntax 2 - Positioned UPDATE
- Examples

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Belongs to Function Group: *Database Access and Update*

See also the following sections in the *Database Management System Interfaces* documentation:

- *UPDATE - SQL* in the *Natural for DB2* part.
  - *UPDATE - SQL* in the *Natural SQL Gateway* part.
  - *UPDATE - SQL* in the *Natural for SQL/DS* part.
- 

## Function

The SQL UPDATE statement is used to perform an UPDATE operation on either rows in a table without using a cursor ("searched" UPDATE) or columns in a row to which a cursor is positioned ("positioned" UPDATE).

Two different structures are possible.

## Syntax 1 - Searched UPDATE

The "Searched" UPDATE statement is a stand-alone statement not related to any SELECT statement. With a single statement you can update zero, one, multiple or all rows of a table. The rows to be updated are determined by a *search-condition* that is applied to the table. Optionally, view and table names can be assigned a *correlation-name*.

### Note:

The number of rows that have actually been updated with a "searched" UPDATE can be ascertained by using the system variable \*ROWCOUNT.

<pre> UPDATE {   view-name [correlation-name] SET *   table-name [correlation-name] [include-columns] SET assignment-list }  [WHERE search-condition] [ WITH { RR } ] [SKIP LOCKED DATA] [QUERY NO integer]                                 { RS }                                 CS </pre>
--

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Syntax Element Description - Syntax 1:

Syntax Element	Description
<i>view-name</i>	<p><b>View Name:</b></p> <p>Refers to the name of a Natural view as defined in the DEFINE DATA statement. For further information, see <i>view-name</i> (in the section <i>Basic Syntactical Items</i>).</p>
<i>correlation-name</i>	<p><b>Correlation Name:</b></p> <p>The item <i>correlation-name</i> represents an alias name for a <i>table-name</i>.</p> <p>For further information, see <i>correlation-name</i> (in the section <i>Basic Syntactical Items</i>).</p>
<i>include-columns</i>	<p><b>Include Columns Clause:</b></p> <p>Specifies a set of columns that are included, along with the columns of <i>table-name</i> in the result table of the UPDATE statement, when it is nested in the FROM clause of a SELECT statement. The included columns are appended to the end of the list of columns. For further details, see <i>include-columns</i>.</p>
SET	<p><b>SET Clause:</b></p> <p>If a view has been specified for updating, an asterisk (*) has to be specified in the SET clause, because all columns of the view must be updated.</p> <p>If a table has been specified for updating, the SET clause must contain either an <i>assignment-list</i> or the name of the view which contains the columns to be updated.</p>
<i>assignment-list</i>	<p><b>Assignment List:</b></p> <p>See <i>Assignment List</i> below.</p>

Syntax Element	Description						
WHERE <i>search-condition</i>	<p><b>WHERE Clause:</b></p> <p>This clause is used to specify the selection criteria for the rows to be updated.</p> <p>If no WHERE clause is specified, the entire table is updated.</p>						
WITH	<p><b>WITH Clause:</b></p> <p>This clause allows the explicit specification of the isolation level used when locating the row to be updated. For detailed information, see <i>WITH Clause - Isolation Level</i> in the description of the SELECT statement.</p> <table border="1"> <tr> <td>CS</td> <td>Cursor Stability</td> </tr> <tr> <td>RR</td> <td>Repeatable Read</td> </tr> <tr> <td>RS</td> <td>Read Stability</td> </tr> </table>	CS	Cursor Stability	RR	Repeatable Read	RS	Read Stability
CS	Cursor Stability						
RR	Repeatable Read						
RS	Read Stability						
SKIP LOCKED DATA	<p><b>SKIP LOCKED DATA Clause:</b></p> <p>Specifies that rows are skipped when incompatible locks are held on the row by other transactions.</p>						
QUERYNO <i>integer</i>	<p><b>QUERYNO Clause:</b></p> <p>This clause allows you to explicitly specify the number to be used in EXPLAIN output and trace records for this statement. The number is used as QUERYNO column in the PLAN_TABLE for the rows that contain information on this statement.</p>						

## Assignment List

$\left\{ \begin{array}{l} \text{column-name} = \left\{ \begin{array}{l} \text{scalar-expression} \\ \text{DEFAULT} \\ \text{NULL} \end{array} \right\} \end{array} \right\}, \dots$
---

In an *assignment-list*, you can assign values to one or more columns. A value can be a *scalar-expression*, DEFAULT or NULL. For further information, see *Scalar Expressions*.

If the value NULL has been assigned, it means that the addressed field is to contain no value (not even the value "0" or "blank").

Alternative:

$$(column-name,...) = ( \left\{ \begin{array}{l} scalar-expression \\ \text{DEFAULT} \\ \text{NULL} \\ row-fullselect \end{array} \right\} , \dots )$$

Syntax Element Description:

Syntax Element	Description
<i>column-name</i>	<p><b>Column Name:</b></p> <p>Specifies the name of a column of the result table of the MERGE statement that is not the same name as another include column or a column in the target table.</p>
DEFAULT	<p><b>DEFAULT Option:</b></p> <p>Specifies that the default value is used based on how the corresponding column is defined in the table.</p>
NULL	<p><b>NULL Option:</b></p> <p>Specifies the null value as the new value of the column.</p> <p>If the value NULL has been assigned, it means that the addressed field is to contain no value (not even the value 0 or "blank").</p>
<i>row-fullselect</i>	<p><b>Row Full Select Option:</b></p> <p>Specifies a full select that returns a single row. The column values are assigned to the corresponding column-names.</p>

## Syntax 2 - Positioned UPDATE

The "positioned" UPDATE statement always refers to a cursor within a database loop. Thus, the table or view referenced by a positioned UPDATE statement must be the same as the one referenced by the corresponding SELECT statement; otherwise an error message is returned. A positioned UPDATE cannot be used with a non-cursor selection.

Common Set Syntax:

$$\text{UPDATE} \left\{ \begin{array}{l} \text{view-name SET *} \\ \text{view-name SET assignment-list} \end{array} \right\} [\text{WHERE CURRENT OF CURSOR } (r)]$$

Extended Set Syntax:

$$\text{UPDATE} \left\{ \begin{array}{l} \text{view-name SET *} \\ \text{view-name SET assignment-list} \end{array} \right\} [\text{WHERE CURRENT OF CURSOR } (r) \left[ \text{FOR ROW} \left\{ \begin{array}{l} [:] \text{host-variable} \\ \text{integer} \end{array} \right\} \text{OF ROWSET} \right]]$$

## Syntax Element Description - Syntax 2:

Syntax Element	Description
<i>view-name</i>	<p><b>Natural View:</b></p> <p>Refers to the name of a Natural view as defined in the DEFINE DATA statement; see also <i>view-name</i> (in the section <i>Basic Syntactical Items</i>).</p>
SET *  SET <i>assignment-list</i>	<p><b>SET Clause:</b></p> <p>If a Natural view has been specified for updating, an asterisk (*) has to be specified in the SET clause, because all columns of the view must be updated.</p> <p>If a table has been specified for updating, the SET clause must contain either an <i>assignment-list</i> or the name of the view which contains the columns to be updated.</p>
WHERE CURRENT OF CURSOR ( <i>r</i> )	<p><b>Statement Reference:</b></p> <p>The (<i>r</i>) notation is used to reference the statement which was used to select the row to be updated. If no statement reference is specified, the UPDATE statement is related to the innermost active processing loop in which a database record was selected.</p>
FOR ROW ... OF ROWSET	<p><b>FOR ROW ... OF ROWSET Clause:</b></p> <p>This clause belongs to the SQL Extended Set.</p> <p>The optional FOR ROW ... OF ROWSET clause for positioned SQL UPDATE statements specifies which row of the current rowset has to be updated. It should only be specified if the UPDATE statement is related to a SELECT statement which uses rowset positioning and which has column arrays in the <i>INTO Clause</i>.</p> <p>If this clause is omitted, all rows of the current rowset are updated by the values in the <i>assignment-list</i>.</p> <p>This clause cannot be specified if <i>view-name</i> SET * is specified.</p>

## Examples

- Example 1 - Searched UPDATE
- Example 2 - Searched UPDATE with assignment-list

- Example 3 - Positioned UPDATE
- Example 4 - Positioned UPDATE with assignment-list

## Example 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'
...

```

## Example 2 - Searched UPDATE with *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'
...

```

## Example 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
...

```

## Example 4 - Positioned UPDATE with *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
...

```