UPDATE

Structured Mode Syntax

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

Reporting Mode Syntax

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

          [ SET  ]
          [ WITH ]  { SAME [RECORD]                }
          [ USING ] { {operand1=operand2} ... }
```

This chapter covers the following topics:

- Function
- Restrictions
- Database-Specific Considerations
- Syntax Description
- Example

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

Related 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

Belongs to Function Group: *Database Access and Update*

Function

The UPDATE statement is used to update one or more fields of a record in a database. The record to be updated must have been previously selected with a FIND, GET or READ statement (or, for Adabas only, with a STORE statement).

Hold Status

The use of the UPDATE statement causes each record read for processing in the corresponding FIND or READ statement to be placed in hold status.

For further information, see *Record Hold Logic* (in the *Programming Guide*).

Restrictions

- The UPDATE statement must not be entered on the same line as the statement used to select the record to be updated.
- The UPDATE statement cannot be applied to Entire System Server views.

Database-Specific Considerations

DL/I	The UPDATE statement can be used to update a segment in a DL/I database. If necessary, the segment length is increased to accommodate all fields specified with the UPDATE statement. If a multiple-value field or a periodic group is defined as variable in length, only the occurrences as specified in the UPDATE statement are written to the segment. The DL/I AIX field name cannot be used in an UPDATE statement. AIX fields, however, may be updated by referring to the source field which comprises the AIX field. DL/I sequence fields cannot be updated because of DL/I restrictions. Due to GSAM restrictions, the UPDATE statement cannot be used for GSAM databases.
VSAM	VSAM primary keys cannot be updated because of VSAM restrictions. The DL/I AIX field name cannot be used in an UPDATE statement. AIX fields, however, may be updated by referring to the source field which comprises the AIX field.
SQL	The UPDATE statement can be used to update a row in a database table. It corresponds with the SQL statement UPDATE WHERE CURRENT OF CURSOR (positioned UPDATE), which means that only the row which was read last can be updated. Only columns (fields) that have been modified within the program, as well as columns that might have been (but not necessarily actually have been) modified outside the program (for example, as input fields in maps), are updated. On all other platforms, all columns are updated. With most SQL databases, a row that was read with a FIND SORTED BY or with a READ LOGICAL statement cannot be updated.

Syntax Description

Operand Definition Table:

Operand	Possible Structure					Possible Formats													Referencing Permitted	Dynamic Definition
operand1		S	A			A		N	P	I	F	B	D	T	L				no	no
operand2	C	S	A			A		N	P	I	F	B	D	T	L				yes	no

Syntax Element Description:

(*r*)	**Statement Reference:** The notation `(r)` is used to indicate the statement in which the record to be modified was read. *r* may be specified as a source-code line number or as a statement label. If no reference is specified, the `UPDATE` statement will reference the innermost active `READ` or `FIND` processing loop. If no `READ` or `FIND` loop is active, it will reference the last preceding `GET` (or `STORE`) statement. **Note:** The `UPDATE` statement must be placed within the `READ` or `FIND` loop it references.
USING SAME	This clause is not permitted if a `DEFINE DATA` statement is used, because in that case the `UPDATE` statement always refers to the entire view as defined in the `DEFINE DATA` statement. The layout of the record buffer or format buffer may be declared using the `OBTAIN` statement. `USING SAME` can be used in reporting mode to indicate that the same fields as read in the statement referenced by the `UPDATE` statement are to be used for the update function. In this case, the most recent value assigned to each database field will be used to update the field. If no new value has been assigned, the old value will be used. If the field to be updated is an array range of a multiple-value field or periodic group and you use a variable index for this array range, the latest range will be updated. This means that if the index variable is modified after the record has been read and before the `UPDATE USING SAME` (reporting mode) or `UPDATE` (structured mode) statement respectively is executed, the range updated will not be the same as the range read.
SET/WITH *operand1=operand2*	This clause can be used in reporting mode to specify the fields to be updated and the values to be used. This clause is not permitted if a `DEFINE DATA` statement is used, because in that case the `UPDATE` statement always refers to the entire view as defined in the `DEFINE DATA` statement. **Note:** For DL/I databases: If the `SET/WITH` clause is used, only I/O (sensitive) fields can be provided. A segment sequence field cannot be updated (`DELETE` and `STORE` must be used instead).

Example

```
** 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
```

Output of Program SUBEX1S

```
ENTER A NAME: BROWN
```

After entering and confirming name:

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

Equivalent reporting-mode example: UPDEX1R.