

MERGE - SQL

```

MERGE INTO table-name [[AS] correlation-name]
  [include-columns] USING source-table
  ON search-condition
  WHEN matching-condition THEN modification-operation ...
  [NOT ATOMIC CONTINUE ON SQLEXCEPTION]
  [QUERYNO integer]

```

This chapter covers the following topics:

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

Belongs to Function Group: *Database Access and Update*

See also *MERGE - SQL* in the *Natural for DB2* part of the *Database Management System Interfaces* documentation.

Function

The MERGE statement updates a table using the specified input data. Rows in the target table that match the input data are updated as specified, and rows that do not exist in the target table are inserted.

The MERGE statement belongs to the SQL Extended Set.

Restriction

This statement is available only with Natural for DB2.

Syntax Description

Syntax Element	Description
MERGE INTO	MERGE INTO Clause: MERGE initiates an SQL MERGE statement, which is a combination of an SQL INSERT and an SQL Searched UPDATE statement.
<i>table-name</i>	Table Name: Identifies the target of the INSERT or UPDATE operation of the MERGE statement. See <i>table-name</i> specification.

Syntax Element	Description
[AS] <i>correlation-name</i>	[AS] <i>correlation-name</i> Clause: Specifies an alternate name for the target table. The alternate name can be used as qualifier when referencing columns of the intermediate result table.
<i>include-columns</i>	Include Columns Clause: Specifies a set of columns that are include, along with the columns of the target table, in the result table of the MERGE statement if it is nested in the FROM clause in a SELECT statement. The included columns are appended to end of the column list identified by the target table. See <i>include-columns</i> .
USING <i>source-table</i>	USING <i>source-table</i> Clause: Specifies the values for the row data to merge into the target table. See <i>source-table</i> .
ON <i>search-condition</i>	ON <i>search-condition</i> Clause: Specifies join conditions between the <i>source-table</i> and the target table. Each column name in the search condition must name a column of the target table or <i>source-table</i> .
WHEN <i>matching-condition</i>	WHEN <i>matching-condition</i> Clause: Specifies the condition under which the <i>modification-operation</i> is run.
THEN <i>modification-operation</i>	THEN <i>modification-operation</i> Clause: Specifies the operation to run when the <i>matching-condition</i> is evaluates to true.
NOT ATOMIC CONTINUE ON SQLEXCEPTION	NOT ATOMIC CONTINUE ON SQLEXCEPTION Clause: Specifies whether merge processing continues in case an error occurred during processing one row of a set of source rows.
QUERYNO <i>integer</i>	QUERYNO <i>integer</i> Clause: Specifies the number for this SQL statement that is used in EXPLAIN output and DB2 trace records.

source-table

(VALUES { <i>values-single-row</i> <i>values-multiple-row</i> }) [AS] <i>correlation-name</i> (<i>column-name</i>),...

Syntax Element	Description
VALUES	VALUES introduces the specification of values for the row data to merge into the target table.
<i>values-single-row</i>	Specifies a single row of source data.
<i>values-multiple-row</i>	Specifies multiple rows of source data.
[AS] <i>correlation-name</i>	Specifies a correlation name for the source-table.
<i>column-name</i>	Specifies a column name to associate the input data to the SET assignment-clause clause for an UPDATE operation or the VALUES clause for an INSERT operation.

values-single-row

$\left[\begin{array}{l} \left\{ \begin{array}{l} \textit{expression} \\ \text{NULL} \end{array} \right\} \\ \left\{ \begin{array}{l} \textit{expression},\dots \\ \text{NULL} \end{array} \right\} \end{array} \right]$
--

Syntax Element	Description
<i>expression</i>	Specifies a scalar expression as described in scalar expressions.
NULL	Specifies the null value.

values-multiple-row

$\left[\begin{array}{l} \left\{ \begin{array}{l} \textit{expression} \\ \textit{host-variable-array} \end{array} \right\} \\ \text{NULL} \\ \left\{ \begin{array}{l} \textit{expression},\dots \\ \textit{host-variable-array} \end{array} \right\} \\ \text{NULL} \end{array} \right]$
FOR $\left\{ \begin{array}{l} \textit{host-variable} \\ \textit{integer-constant} \end{array} \right\}$ ROWS

Syntax Element	Description
<i>expression</i>	Specifies a scalar expression as described in scalar expressions.
<i>host-variable-array</i>	Specifies a host variable array or an index range of an array. If <i>host-variable-array</i> are specified, the compiler option DB2ARRY has to be set to ON. An optional indicator array can be specified for each host-variable-array by the keyword INDICATOR, that is, <i>host-variable-array</i> INDICATOR [:] <i>indicator-array</i> .
NULL	Specifies the null value.
FOR ... ROWS	Specifies the number of rows to merge. <i>host-variable</i> or <i>integer-constant</i> is assigned to a value <i>k</i> . <i>k</i> must be in the range of 0 to 32767 and must be lower than or equal to the minimum size of all specified host-variable-arrays.

matching-condition

<p>{ MATCHED NOT MATCHED }</p>

Syntax Element	Description
MATCHED	Specifies the operation to perform on the rows where the ON <i>search-condition</i> is true. Only UPDATE can be specified after the THEN keyword.
NOT MATCHED	Specifies the operation to perform on the rows where the ON <i>search-condition</i> is false. Only INSERT can be specified after the THEN keyword

modification-operation

<p>{ UPDATE SET <i>assignment-clause</i> <i>insert-operation</i> }</p>
--

Syntax Element	Description
UPDATE SET	Specifies the UPDATE operation to run when the matching-condition evaluates to true. See <i>assignment-clause</i> below.
<i>insert-operation</i>	Specifies the <i>insert-operation</i> to run for the rows where the matching condition evaluates to false.

assignment-clause

{	<i>column-name</i> =	{	<i>expression</i>	}	...
			DEFAULT		
}			NULL		
	{	(<i>column-name</i> ,...)= ({	<i>expression</i> ,...	}
				DEFAULT	
}			NULL		

Syntax Element	Description
<i>column-name</i>	Specifies the column for which an insert value is provided.
<i>expression</i>	Specifies the new value for the column. An <i>expression</i> can contain references to columns of the <i>source-table</i> or target table. <i>expression</i> cannot contain references to included columns.
DEFAULT	Specifies the default value for the column. The value that is assigned depends on how the column is defined.
NULL	Specifies the null value as new value for the column.

insert-operation

INSERT [(<i>column-name</i> ,...)] VALUES ({	<i>expression</i>	}	...
		DEFAULT		
		NULL		

Syntax Element	Description
INSERT	Specifies the INSERT operation to run for the rows where the matching condition evaluates to false.
<i>column-name</i>	Specifies the column for which a insert value is provided.
VALUES	Introduces one or more column values to insert.
<i>expression</i>	Specifies the new value for the column. An <i>expression</i> can contain references to columns of the source-table. <i>expression</i> cannot contain references to columns of the target table.
DEFAULT	Specifies the default value for the column. The value that is assigned depends on how the column is defined.
NULL	Specifies the null value as new value for the column.

Examples

Example 1:

Update the inventory at a car dealership. Add new car model to the inventory or update information about existing car model that is already in the inventory.

```

DEFINE DATA LOCAL
01 #MODEL (A20)
01 #DELTA (I4)
END-DEFINE
* Setup input host variables
ASSIGN #MODEL = 'Grand Turbo'
ASSIGN #DELTA := 5
* Insert/Update into INVENTORY table
MERGE INTO CDS-INVENTORY T
  USING (VALUES (:#MODEL, :#DELTA)) AS S(MODEL, DELTA)
  ON T.MODEL = S.MODEL
WHEN MATCHED THEN UPDATE SET T.QUANTITY = T.QUANTITY + S.DELTA
WHEN NOT MATCHED THEN INSERT VALUES (S.MODEL, S.DELTA)
END TRANSACTION
END

```

Example 2:

Update the inventory at a car dealership. Add new car models to the inventory and update information about car models that are already in the inventory. Input comes from Natural arrays. Array specific code is marked with red color.

```

OPTIONS DB2ARRY ON
DEFINE DATA LOCAL
01 #MODEL_ARR (A20/1:20)
01 #DELTA_ARR (I4/1:20))
01 #ROW-COUNT (I4)
01 #NUM-ERRORS (I4)
01 #SQLCODE (I4)
01 #SQLSTATE (A5)
01 #ROW-NUM (I4)
END-DEFINE

```

```

* Setup input host variables
ASSIGN #MODEL_ARR(1) = 'Grand Turbo'
ASSIGN #DELTA_ARR(1) := 5
ASSIGN #MODEL_ARR(2) = 'Blue Car'
ASSIGN #DELTA_ARR(2) := 3
. . .
* Insert/Update into INVENTORY table
CALLNAT 'NDBNOERR'
MERGE INTO CDS-INVENTORY T
  USING (VALUES (:#MODEL_ARR(*), :#DELTA_ARR(*))
  FOR 20 ROWS)
  AS S(MODEL, DELTA)
ON T.MODEL = S.MODEL
WHEN MATCHED THEN UPDATE SET T.QUANTITY = T.QUANTITY + S.DELTA
WHEN NOT MATCHED THEN INSERT VALUES (S.MODEL, S.DELTA)
NOT ATOMIC CONTINUE ON SQLEXCEPTION
* Check outcome of MERGE
PROCESS SQL SYSIBM-SYSDUMMY1
  <<GET DIAGNOSTICS
  :#ROW-COUNT = ROW_COUNT
  ,:#NUM-ERRORS = NUMBER>>
WRITE 'Number of rows merged           :' #ROW-COUNT /
      'NUMBER OF ERRORS                 :' #NUM-ERRORS
IF #NUM-ERRORS > 0
  FOR #I = 1 TO #NUM-ERRORS
    PROCESS SQL SYSIBM-SYSDUMMY1
      <<GET DIAGNOSTICS CONDITION :#I
      :#SQLCODE = DB2_RETURNED_SQLCODE,
      :#SQLSTATE = RETURNED_SQLSTATE,
      :#ROW_NUM = DB2_ROW_NUMBER>>
    PRINT 'DB2_RETURNED_SQLCODE:' #SQLCODE
          'RETURNED_SQLSTATE:' #SQLSTATE
          'DB2_ROW_NUMBER:' #ROW_NUM (EM=99Z)
  END-FOR
END-IF
END TRANSACTION
END

```