

DBS_DELETE

DB_DELETE and DBS_DELETE actions

Function

Deleting of one or several rows of the structure.

Declaration

```
DB_DELETE handleIdent_Int, rowIdent,retCodeIdent_Int [ORAHINT hintIdent_Str]

DBS_DELETE dbObjIdent, rowIdent, retCodeIdent_Int [TRANS transHandle_Int] [ORAHINT hintIdent_Str]
```

or

```
DB_DELETE handleIdent_Int, retCodeIdent_Int WHERE strExpression_Str [BINDIN varIdent1, varIdent2, ... ] [ORAHINT hintIdent_Str]

DB_DELETE handleIdent_Int, retCodeIdent_Int WHERE strExpression_Str [BINDIN structRowIdent] [ORAHINT hintIdent_Str]

DBS_DELETE dbObjIdent, retCodeIdent_Int WHERE strExpression_Str [BINDIN varIdent1, varIdent2, ... ] [TRANS transHandle_Int] [ORAHINT hintIdent_Str]

DBS_DELETE dbObjIdent, retCodeIdent_Int WHERE strExpression_Str [BINDIN structRowIdent] [TRANS transHandle_Int] [ORAHINT hintIdent_Str]
```

Parameters

handl eIdent _Int	in	Identifier (handle) of Int type of the connection with a table (DB_CONNECT).
dbObj Ident	in	Reference to an object of Database table type.
retCo delde nt_Int	o ut	Return value of Int type- action success.
rowId ent	in	One structure row identifier .
strEx pressi on_Str	in	Expression of <i>String</i> type, which identifies rows to delete. If the expression is parameterized , the keyword BINDIN and the values of parameters (<i>structRowIdent</i> or <i>varIdent1</i> , <i>varIdent2</i> , ...) are mandatory.
varId ent1, varId ent2, ...	in	List of objects, constants or local variables , which will specify the values of parameters of parameterized SQL expression <i>strExpression_Str</i> .
struct RowI dent	in	Reference to a row of local variable of <i>Record</i> type or to a row of structured variable . The row's values will specify the values of parameters of parameterized SQL expression <i>strExpression_Str</i> .
trans Handl e_Int	in	Identifier of the Connection to the database.
hintId ent_S tr	in	Expression of <i>String</i> type that defines Oracle SQL hint. It is used as an instruction for the performance optimizer of SQL command. The value is used without the opening and terminating characters <i>/*+ <orahint> */</i> . The example is mentioned here .

Return code

The value of the parameter *transHandle_Int*. See the table of [error codes](#). It is possible to get [extended error information](#).

Description

Table must be opened with the access **_DB_MODIFY**.

In the first case (versions with the `rowIdent` parameter), the deleted row is determined by a value of the key items in a value of Structure type parameter `rowIdent` . The particular row is to be found and deleted according to the key items. The values of all key items in `rowIdent` must be valid, otherwise the deleting will not take place, it will end with an error and `%GetLastExtErrorCode()` will return error 667.

In the second case, the expression of String type (`strExpression_Str`), which result value is used as

```
WHERE
```

clause for the SQL command

```
DELETE ,
```

is the condition to delete. In this case, a value of the key item is not used and may not be adjusted. If there is more rows in a table which meet the condition (`strExpression_Str`), they all are to be deleted.

The advantage of the action **DBS_DELETE** at work with a table is the possibility to leave out its closing and opening (shorter code).

For D2000 v5.00: an disadvantage of the action **DBS_DELETE** is in speed. Each **DBS_DELETE** call results in necessity to open and close the database in DBManager - it can be a time-consuming operation and it is a comparatively non-standard method in term of databases. The need to open and close the database may be avoided in the scope of transaction processing so that the command is followed by the parameter

```
TRANS
```

.

For D2000 v6.00 and higher: DBManager [optimization](#) (connection recycling, predefined connections) causes the action **DBS_DELETE** to execute as quick as the action **DB_DELETE** and as moreover there is saved a time required for execution of the action **DB_CONNECT** to open the database.

The need to open and close the database may be avoided in the scope of transaction processing so that the command is followed by the parameter

```
TRANS
```

.

Warning

A variant of the command

```
DB_DELETE WHERE
```

allows to delete the entire contents of a table.

If the condition (`strExpression_Str`) is met for all rows of a table (e.g. "`1=1`"), its entire contents is to be deleted.

Example

[Work with a database table \(actions DB_ ...\)](#).

Related topics

- [DB_CONNECT](#)
- [DB_DISCONNECT](#)
- [DB_INSERT](#)
- [DB_INSUPD](#)
- [DB_READ](#)
- [DB_READ_BLOB](#)
- [DB_UPDATE](#)
- [DB_UPDATE_BLOB](#)

- [DB_TRANS_OPEN](#)
- [DB_TRANS_COMMIT](#)
- [DB_TRANS_ROLLBACK](#)
- [DB_TRANS_CLOSE](#)

[All database related actions](#)



Related pages:

[Script actions](#)