

RDB\$SET_CONTEXT()

Note: `RDB$GET_CONTEXT` and its counterpart `RDB$SET_CONTEXT` are actually predeclared UDFs. They are listed here as internal functions because they are always present – the user doesn't have to do anything to make them available.

Available in: [DSQL](#), [PSQL](#), [ESQL](#)

Added in: 2.0

Description

Creates, sets or unsets a [variable](#) in one of the user-writable namespaces `USER_SESSION` and `USER_TRANSACTION`.

Result type: [INTEGER](#)

Syntax

```
RDB$SET_CONTEXT ('<namespace>', '<varname>', <value> | NULL)

<namespace> ::= USER_SESSION | USER_TRANSACTION
<varname>    ::= A case-sensitive string of max. 80 characters
<value>      ::= A value of any type, as long as it's castable to a
                VARCHAR(255)
```

The namespaces

The `USER_SESSION` and `USER_TRANSACTION` namespaces are initially empty. The user can create and set [variables](#) in them with `RDB$SET_CONTEXT()` and retrieve them with `RDB$GET_CONTEXT()`. The `USER_SESSION` context is bound to the current connection. Variables in `USER_TRANSACTION` only exist in the [transaction](#) in which they have been set. When the transaction ends, the context and all the variables defined in it are destroyed.

Return values and error behaviour

The function returns 1 if the variable already existed before the call and 0 if it didn't. To remove a variable from a context, set it to NULL. If the given namespace doesn't exist, an error is raised. Both namespace and variable names must be entered as single-quoted, case-sensitive, non-NULL strings.

Examples

```
select rdb$set_context('USER_SESSION', 'MyVar', 493) from rdb$database

rdb$set_context('USER_SESSION', 'RecordsFound', RecCounter);

select rdb$set_context('USER_TRANSACTION', 'Savepoints', 'Yes')
       from rdb$database
```

Notes

- The maximum number of variables in any single context is 1000.
- All `USER_TRANSACTION` variables will survive a `ROLLBACK RETAIN` or `ROLLBACK TO SAVEPOINT` unaltered, no matter when in the transaction they were set.
- Due to its UDF-like nature, `RDB$SET_CONTEXT` can – in `PSQL` only – be called like a void function, without assigning the result, as in the second example above. Regular internal functions don't allow this type of use.

From:
<http://ibexpert.com/docu/> - **IBExpert**

Permanent link:
http://ibexpert.com/docu/doku.php?id=01-documentation:01-09-sql-language-references:firebird2.1-language-reference:internal-functions:rdb_set_context

Last update: **2023/07/21 10:42**

