INSERT

Available in: DSQL, ESQL, PSQL

Description

Adds rows to a database table, or to one or more tables underlying a view. Field values can be given in the VALUES clause, they can be totally absent (in both cases, exactly one row is inserted), or they can come from a SELECT statement (0 to many rows inserted).

Syntax

```
INSERT [TRANSACTION name]
  INTO {tablename | viewname}
  {DEFAULT VALUES | [(<column_list>)] <value_source>}
  [RETURNING <value_list> [INTO <var_list>]]

<column_list>   ::= colname [, colname ...]
<value_source>  ::= VALUES (<value_list>) | <select_stmt>
<value_list>    ::= value [, value ...]
<var_list>      ::= :varname [, :varname ...]
<select_stmt>   ::= a SELECT whose result set fits the target columns
```

Restrictions:

- The TRANSACTION directive is only available in ESQL.
- The RETURNING clause is not available in ESQL.
- The INTO <variables> subclause is only available in PSQL.
- When returning values into the context variable NEW, this name must not be preceded by a colon (":").
- Since v. 2.0, no column may appear more than once in the insert list.

back to top of page

INSERT ... DEFAULT VALUES

Available in: DSQL, PSQL

Added in: 2.1

Description

The DEFAULT VALUES clause allows insertion of a record without providing any values at all, neither directly nor from a SELECT statement. This is only possible if every NOT NULL or CHECKed column in the table either has a valid default declared or gets such a value from a BEFORE INSERT trigger. Furthermore, triggers providing required field values must not depend on the presence of input values.

Example

```
insert into journal default values
  returning entry_id
```

back to top of page

RETURNING clause

Available in: DSQL, PSQL

Added in: 2.0

Changed in: 2.1

Description

An INSERT statement adding at most one row may optionally include a RETURNING clause in order to return values from the inserted row. The clause, if present, need not contain all of the insert columns and may also contain other columns or expressions. The returned values reflect any changes that may have been made in BEFORE triggers, but not those in AFTER triggers.

Examples

```
insert into Scholars (firstname, lastname, address, phone, email)
  values ('Henry', 'Higgins', '27A Wimpole Street', '3231212', null)
  returning lastname, fullname, id

insert into Dumbbells (firstname, lastname, iq)
  select fname, lname, iq from Friends order by iq rows 1
  returning id, firstname, iq into :id, :fname, :iq;
```

Notes:

- RETURNING is only supported for VALUES inserts and – since version 2.1 – singleton SELECT inserts.
- In DSQL, a statement with a RETURNING clause always returns exactly one row. If no record was actually inserted, the fields in this row are all NULL. This behaviour may change in a later version of Firebird. In PSQL, if no row was inserted, nothing is returned, and the receiving variables keep their existing values.

back to top of page

UNION allowed in feeding SELECT

Changed in: 2.0

Description

A SELECT query used in an INSERT statement may now be a UNION.

Example

```
insert into Members (number, name)
  select number, name from NewMembers where Accepted = 1
    union
  select number, name from SuspendedMembers where Vindicated = 1
```