

ALTER TRIGGER

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

Description

Alters an existing [trigger](#). Relation triggers cannot be changed into [database triggers](#) or vice versa. The associated [table](#) or [view](#) of a relation trigger cannot be changed.

Syntax

```
ALTER TRIGGER name
  [ACTIVE | INACTIVE]
  [{BEFORE | AFTER} <mutation_list> | ON db_event]
  [POSITION number]
  [AS
    [<declarations>]
  BEGIN
    [<statements>]
  END ]
```

See [CREATE TRIGGER](#) for the meaning of <mutation_list> etc.

[back to top of page](#)

Database triggers

Added in: 2.1

Description

The `ALTER TRIGGER` syntax (see above) has been extended to support database triggers. For a full discussion of this feature, see [CREATE TRIGGER :: Database triggers](#).

TYPE OF COLUMN in variable declarations

Added in: 2.5

Description

Analogous to the “`TYPE OF domain`” syntax supported since version 2.1, it is now also possible to declare variables as having the type of an existing table or view column. See [PSQL::DECLARE](#) for syntax and details.

Domains instead of datatypes

Changed in: 2.1

Description

Firebird 2.1 and up allow the use of [domains](#) instead of SQL [data types](#) when declaring local [trigger variables](#). See [PSQL::DECLARE](#) for the exact syntax and details.

[back to top of page](#)

COLLATE in variable declarations

Changed in: 2.1

Description

Firebird 2.1 and up allow `COLLATE` clauses in local [variable declarations](#). See [PSQL::DECLARE](#) for syntax and details.

NOT NULL in variable declarations

Changed in: 2.1

Description

Firebird 2.1 and up allow [NOT NULL](#) constraints in [local variable](#) declarations. See [PSQL::DECLARE](#) for syntax and details.

[back to top of page](#)

Multi-action triggers

Added in: 1.5

Description

The `ALTER TRIGGER` syntax (see above) has been extended to support multi-action triggers. For a full discussion of this feature, see [CREATE TRIGGER :: Multi-action triggers](#).

Restriction on altering used triggers

Changed in: 2.0, 2.0.1

Description

In Firebird 2.0 only, a restriction is in place which prevents anyone from dropping, altering or recreating a [trigger](#) or [stored procedure](#) if it has been used since the [database](#) was opened. This restriction has been removed again in version 2.0.1.

Still, performing these operations on a live database is potentially risky and should only be done with the utmost care.

[back to top of page](#)

PLAN allowed in trigger code

Changed in: 1.5

Description

Before Firebird 1.5, a trigger containing a `PLAN` statement would be rejected by the compiler. Now a valid plan can be included and will be used.

ALTER TRIGGER no longer increments table change count

Changed in: 1.0

Description

Each time you use `CREATE`, `ALTER` or `DROP TRIGGER`, InterBase increments the [metadata](#) change counter of the associated [table](#).

Once that counter reaches 255, [no more metadata changes](#) are possible on the table (you can still work with the data though). A [backup-restore](#) cycle is needed to reset the counter and perform metadata operations again.

While this obligatory cleanup after many metadata changes is in itself a useful feature, it also means that users who regularly use `ALTER TRIGGER` to deactivate triggers during e.g. bulk import operations are forced to backup and restore much more often than needed.

Since changes to triggers don't imply structural changes to the table itself, Firebird no longer increments the table change counter when `CREATE`, `ALTER` or `DROP TRIGGER` is used. One thing has

remained though: once the counter is at 255, you can no longer create, alter or drop triggers for that table.

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

Permanent link:
<http://ibexpert.com/docu/doku.php?id=01-documentation:01-09-sql-language-references:firebird2.5-language-reference-update:ddl-statements:trigger:alter-trigger>

Last update: **2023/08/02 15:39**

