The Firebird FAQ

Google Search

○ Web ● FirebirdFAQ.org

Categories
Newbies
SQL
Installation and setup
Backup and restore
Performance
Security
Connectivity and API
HOWTOs
Errors and error codes
Miscellaneous

How to create an autoincrement column?

Firebird does support autoincrement columns via BEFORE INSERT TRIGGERs and GENERATORs. Generators are also named SEQUENCES in Firebird 2.0 and above - and are compliant to the SQL standard.

For example, let's suppose you have a table named T1, with the following columns:

```
create table t1
(
id integer not null,
field1 varchar(20) not null
);
```

To make column ID autoincrement, we don't have to do anything special to the column itself, but we need to create a BEFORE INSERT trigger that will put a new value each time a new record is inserted. To make sure we get unique values consistently we use a GENERATOR (a.k.a. SEQUENCE):

```
CREATE GENERATOR gen_t1_id;
SET GENERATOR gen_t1_id TO 0;
```

SET GENERATOR statement is used to set the starting value of a sequence. Since we set it to zero, the first record will get a value of one.

Now we create a trigger:

```
set term !! ;
CREATE TRIGGER T1_BI FOR T1
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
if (NEW.ID is NULL) then NEW.ID = GEN_ID(GEN_T1_ID, 1);
END!!
set term ; !!
```

Checking whether NEW.ID is NULL is a common way to prevent getting incremented values when you copy data from some other database or import from some other data source. It can be done in many different ways, but this is the most simple approach: if value for column is not supplied - autoincrement it.

If you wonder why are those SET TERM statements needed: it's because many tools use semi-colon to separate statements and trigger code contains a semi-colon so statement terminator character needs to be changed. Please refer to FAQ #78 for more details.

All this might seem awkward, but it's a proper way to do it in a multiuser environment. Most GUI administration tools have options to generate this code for you automatically, so it is not a problem.

If you wish to get the new ID from an INSERT statement, use the RETURNING clause (only available in Firebird 2.1 and above). Here's an example that would return the newly inserted ID from an insert statement:

```
INSERT INTO t1(field1) VALUES('my stuff') RETURNING id;
```

If you execute such statement in your admin tool, you would get a single-row resultset with column ID containing newly inserted record ID.

If you use an older version of Firebird, you should first get the value using GEN_ID and then use it in INSERT statement. To read in the generator value, you can use some single-record table like RDB$DATABASE:

```
select GEN_ID(GEN_T1_ID, 1) from RDB$DATABASE;
```

If you are using Delphi and Zeos (or a similar library), you can use ZSequence component to have GEN_ID read automatically for you. In ZTable, mark the autoincrement field as SequenceField and set the appropriate ZSequence object and it will all work automatically.

Do you find this FAQ incorrect or incomplete? Please e-mail us what needs to be changed. To ensure quality, each change is checked by our editors (and often tested on live Firebird databases), before it enters the main FAQ database. If you desire so, the changes will be credited to your name. To learn more, visit our add content page.