



PostgreSQL
le SGBD Open Source
de référence

Accueil
Support

Actualités

Documentation

Forums

Association

Développeurs

Planète

Rechercher

[Version anglaise](#)

36.4. Un exemple complet de trigger

Voici un exemple très simple de fonction déclencheur écrite en C (les exemples de déclencheurs écrits avec différents langages de procédures se trouvent dans la documentation de ceux-ci).

La fonction `trigf` indique le nombre de lignes de la table `ttest` et saute l'opération si la commande tente d'insérer une valeur NULL dans la colonne `x` (ainsi le déclencheur agit comme une contrainte non NULL mais n'annule pas la transaction).

Tout d'abord, la définition des tables :

```
CREATE TABLE ttest (
  x integer
);
```

Voici le code source de la fonction trigger :

```
#include "postgres.h"
#include "executor/spi.h"      /* nécessaire pour fonctionner avec SPI */
#include "commands/trigger.h" /* ... et les déclencheurs */

#ifdef PG_MODULE_MAGIC
PG_MODULE_MAGIC;
#endif

extern Datum trigf(PG_FUNCTION_ARGS);

PG_FUNCTION_INFO_V1(trigf);

Datum
trigf(PG_FUNCTION_ARGS)
{
    TriggerData *trigdata = (TriggerData *) fcinfo->context;
    TupleDesc   tupdesc;
    HeapTuple   rettup;
    char        *when;
    bool        checkNULL = false;
    bool        isNULL;
    int         ret, i;

    /* on s'assure que la fonction est appelée en tant que déclencheur */
    if (!CALLED_AS_TRIGGER(fcinfo))
        elog(ERROR, "trigf: not called by trigger manager");

    /* nuplet à retourner à l'exécuteur */
    if (TRIGGER_FIRED_BY_UPDATE(trigdata->tg_event))
        rettup = trigdata->tg_newtuple;
    else
        rettup = trigdata->tg_trigtuple;

    /* vérification des valeurs NULL */
    if (!TRIGGER_FIRED_BY_DELETE(trigdata->tg_event)
        && TRIGGER_FIRED_BEFORE(trigdata->tg_event))
        checkNULL = true;

    if (TRIGGER_FIRED_BEFORE(trigdata->tg_event))
        when = "before";
    else
        when = "after ";

    tupdesc = trigdata->tg_relation->rd_att;

    /* connexion au gestionnaire SPI */
```

```

if ((ret = SPI_connect()) < 0)
    elog(ERROR, "trigf (fired %s): SPI_connect returned %d", when, ret);

/* obtient le nombre de lignes dans la table */
ret = SPI_exec("SELECT count(*) FROM ttest", 0);

if (ret < 0)
    elog(ERROR, "trigf (fired %s): SPI_exec returned %d", when, ret);

/* count(*) renvoie int8, prenez garde à bien convertir */
i = DatumGetInt64(SPI_getbinval(SPI_tuptable->vals[0],
                                SPI_tuptable->tupdesc,
                                1,
                                &isNULL));

elog (INFO, "trigf (fired %s): there are %d rows in ttest", when, i);

SPI_finish();

if (checkNULL)
{
    SPI_getbinval(rettuple, tupdesc, 1, &isNULL);
    if (isNULL)
        rettuple = NULL;
}

return PointerGetDatum(rettuple);
}

```

Après avoir compilé le code source (voir [Section 35.9.6, « Compiler et lier des fonctions chargées dynamiquement »](#)), déclarez la fonction et les déclencheurs :

```

CREATE FUNCTION trigf() RETURNS trigger
AS 'nomfichier'
LANGUAGE C;

CREATE TRIGGER tbefore BEFORE INSERT OR UPDATE OR DELETE ON ttest
FOR EACH ROW EXECUTE PROCEDURE trigf();

CREATE TRIGGER tafter AFTER INSERT OR UPDATE OR DELETE ON ttest
FOR EACH ROW EXECUTE PROCEDURE trigf();

```

À présent, testez le fonctionnement du déclencheur :

```

=> INSERT INTO ttest VALUES (NULL);
INFO:  trigf (fired before): there are 0 rows in ttest
INSERT 0 0

-- Insertion supprimée et déclencheur APRES non exécuté

=> SELECT * FROM ttest;
 x
---
(0 rows)

=> INSERT INTO ttest VALUES (1);
INFO:  trigf (fired before): there are 0 rows in ttest
INFO:  trigf (fired after ): there are 1 rows in ttest
                                ^^^^^^^
                                souvenez-vous de ce que nous avons dit sur la visib

INSERT 167793 1
vac=> SELECT * FROM ttest;
 x
---
 1
(1 row)

=> INSERT INTO ttest SELECT x * 2 FROM ttest;
INFO:  trigf (fired before): there are 1 rows in ttest
INFO:  trigf (fired after ): there are 2 rows in ttest
                                ^^^^^^
                                souvenez-vous de ce que nous avons dit sur la visib

INSERT 167794 1
=> SELECT * FROM ttest;
 x
---
 1
 2
(2 rows)

```

```

=> UPDATE ttest SET x = NULL WHERE x = 2;
INFO:  trigf (fired before): there are 2 rows in ttest
UPDATE 0
=> UPDATE ttest SET x = 4 WHERE x = 2;
INFO:  trigf (fired before): there are 2 rows in ttest
INFO:  trigf (fired after ): there are 2 rows in ttest
UPDATE 1
vac=> SELECT * FROM ttest;
   x
---
   1
   4
(2 rows)

=> DELETE FROM ttest;
INFO:  trigf (fired before): there are 2 rows in ttest
INFO:  trigf (fired before): there are 1 rows in ttest
INFO:  trigf (fired after ): there are 0 rows in ttest
INFO:  trigf (fired after ): there are 0 rows in ttest
                                ^^^^^^
                                souvenez-vous de ce que nous avons dit sur la visib

DELETE 2
=> SELECT * FROM ttest;
   x
---
(0 rows)

```

Vous trouverez des exemples plus complexes dans `src/test/regress/regress.c` et dans [spi](#).