CREATE TABLE public.spec

(

id integer,

TableName varchar,

ColumnName varchar,

CurrentMaxValue integer

);

INSERT INTO spec(id, TableName, ColumnName, CurrentMaxValue )

VALUES (1, 'spec' , 'id' , 1);

create or replace function update\_max\_in\_test\_table()

returns trigger

as

$$

declare new\_value integer;

begin

EXECUTE format('SELECT MAX(%s) FROM NEW ', tg\_argv[1]) INTO new\_value;

update spec

set CurrentMaxValue = new\_value

where (TableName = tg\_argv[0] and ColumnName = tg\_argv[1] and CurrentMaxValue < new\_value);

return null;

end

$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION Search\_Inc(IN TableNameIN spec.TableName%type,IN ColumnNameIN spec.ColumnName%type, OUT Res integer)

AS

$$

DECLARE count\_trig integer;

DECLARE new\_id integer;

DECLARE name\_trig1 varchar;

DECLARE name\_trig2 varchar;

BEGIN

ASSERT EXISTS(SELECT \* FROM information\_schema.tables

WHERE table\_name = TableNameIN), 'Такой таблицы не существует';

ASSERT EXISTS(SELECT \* FROM information\_schema.columns

WHERE table\_name = TableNameIN AND column\_name = ColumnNameIN), 'Такого столбца не существует';

ASSERT EXISTS(SELECT \* FROM information\_schema.columns

WHERE table\_name = TableNameIN AND column\_name = ColumnNameIN AND data\_type = 'integer'), 'Не целочисленный тип данных';

UPDATE spec

SET CurrentMaxValue = CurrentMaxValue+1

WHERE spec.TableName = TableNameIN and spec.ColumnName = ColumnNameIN

RETURNING CurrentMaxValue INTO Res;

IF Res IS NULL THEN

new\_id = Search\_Inc('spec','id');

EXECUTE FORMAT('SELECT COALESCE(MAX(%s) + 1, 1) FROM %s ', quote\_ident(ColumnNameIN), quote\_ident(TableNameIN))

INTO Res;

INSERT INTO spec (id, TableName, ColumnName, CurrentMaxValue )

VALUES (new\_id, TableNameIN, ColumnNameIN, Res);

SELECT count(\*) + 1 FROM information\_schema.triggers

WHERE event\_object\_table = TableNameIN

INTO count\_trig;

name\_trig1 = concat ((TableNameIN), '\_', (ColumnNameIN), '\_', CAST(count\_trig AS varchar));

IF EXISTS (SELECT \* FROM information\_schema.triggers

WHERE trigger\_name =quote\_ident(name\_trig1)) THEN

name\_trig1 = concat(name\_trig1, '\_', CAST((SELECT CURRENT\_TIMESTAMP) AS varchar));

END IF;

EXECUTE FORMAT('CREATE TRIGGER %s

AFTER INSERT ON %s

REFERENCING NEW TABLE AS NEW

FOR EACH STATEMENT

EXECUTE FUNCTION update\_max\_in\_test\_table(%s,%s);', quote\_ident(name\_trig1), quote\_ident(TableNameIN),

quote\_ident(TableNameIN), quote\_ident(ColumnNameIN));

name\_trig2 = concat ((TableNameIN), '\_', (ColumnNameIN), '\_', CAST((count\_trig + 1) AS varchar));

IF EXISTS (SELECT \* FROM information\_schema.triggers

WHERE trigger\_name = quote\_ident(name\_trig2)) THEN

name\_trig2 = concat(name\_trig2, '\_', CAST((SELECT CURRENT\_TIMESTAMP) AS varchar));

END IF;

EXECUTE FORMAT('CREATE TRIGGER %s

AFTER UPDATE ON %s

REFERENCING NEW TABLE AS NEW

FOR EACH STATEMENT

EXECUTE FUNCTION update\_max\_in\_test\_table(%s,%s);', quote\_ident(name\_trig2),

quote\_ident(TableNameIN),

quote\_ident(TableNameIN), quote\_ident(ColumnNameIN));

END IF;

END

$$

LANGUAGE plpgsql;

-- ОШИБКА: Такой таблицы не существует

SELECT Search\_Inc('test', 'id');

-- ОШИБКА: Такого столбца не существует

SELECT Search\_Inc('spec', 'id1');

-- ОШИБКА: Не целочисленный тип данных

SELECT Search\_Inc('spec', 'tablename');

-- Создание таблиц и триггеров

CREATE TABLE test(

id integer,

num\_value1 integer

);

CREATE OR REPLACE FUNCTION trig\_function()

RETURNS TRIGGER

AS

$$

BEGIN

RETURN NULL;

END

$$

LANGUAGE plpgsql;

CREATE TRIGGER trigger1

AFTER DELETE ON test

EXECUTE FUNCTION trig\_function();

SELECT Search\_Inc('test', 'id');

SELECT trigger\_name FROM information\_schema.triggers

WHERE event\_object\_table = 'test';

--новое имя

CREATE TABLE test2(

id integer,

num\_value1 integer

);

--Создаем триггер с уже имеющимся названием

CREATE TRIGGER test2\_id\_2

AFTER DELETE ON test2

EXECUTE FUNCTION trig\_function();

SELECT Search\_Inc('test2', 'id');

SELECT trigger\_name FROM information\_schema.triggers

WHERE event\_object\_table = 'test2';

DROP TABLE spec;

DROP TABLE test;

DROP TABLE test2;

DROP FUNCTION trig\_function();

DROP FUNCTION Search\_Inc(TableNameIN varchar, ColumnNameIN varchar);

drop function update\_max\_in\_test\_table();