--Создание спец таблицы

CREATE TABLE SpecTable

(

id INT PRIMARY KEY,

TableName VARCHAR NOT NULL,

ColumnName VARCHAR NOT NULL,

CurrentMaxValue INT NOT NULL

);

--Добавление в спец. таблицу записи (1, spec, id, 1).

INSERT INTO SpecTable VALUES(1,'spec','id',1);

--Создание хранимой процедуры (ХП).

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

returns trigger

as

$$

declare new\_value INT;

begin

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

update SpecTable

set CurrentMaxValue = new\_value

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

--execute format('UPDATE SpecTable SET CurrentMaxValue=(SELECT MAX(%s) FROM NEW) WHERE (TableName=%s and ColumnName=%s and CurrentMaxValue<(SELECT MAX(%s) FROM NEW))',quote\_ident(tg\_argv[1]),quote\_ident(tg\_argv[0]),quote\_ident(tg\_argv[0]),quote\_ident(tg\_argv[0]));

return null;

--execute format('UPDATE SpecTable SET CurrentMaxValue=(SELECT MAX(%s) FROM NEW) WHERE (TableName=tg\_argv[0] and ColumnName=tg\_argv[1] and CurrentMaxValue<(SELECT MAX(%s) FROM NEW))',tg\_argv[1],tg\_argv[1]);

end

$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION Search\_Inc(IN TableNameIN varchar,IN ColumnNameIN varchar, OUT Res int) AS

$$

DECLARE name\_trig1 varchar;

DECLARE name\_trig2 varchar;

BEGIN

IF

EXISTS(SELECT \* FROM SpecTable WHERE TableName = TableNameIN AND ColumnName = ColumnNameIN)

THEN

UPDATE SpecTable SET CurrentMaxValue = CurrentMaxValue+1

WHERE TableName = TableNameIN AND ColumnName = ColumnNameIN

RETURNING CurrentMaxValue INTO Res;

ELSE

EXECUTE format('SELECT COALESCE(max(%I)+1,1) FROM %I ', ColumnNameIN,TableNameIN) INTO Res;

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

VALUES(Search\_Inc('spec', 'id'), (TableNameIN), (ColumnNameIN), Res);

name\_trig1 = concat ('insert\_', quote\_ident(TableNameIN), '\_', quote\_ident(ColumnNameIN), '\_trigger');

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 ('update\_', quote\_ident(TableNameIN ), '\_', quote\_ident(ColumnNameIN), '\_trigger');

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;

--Создание таблицы с одним полем

CREATE TABLE test

(

id INT NOT NULL

);

--Вывод таблицы

SELECT \* FROM spectable;

--Добавляем знчение 12

INSERT INTO test VALUES (12);

--Запускаем ХП от test

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

--Добавляем знчение 20

INSERT INTO test VALUES (20);

--Добавляем сразу 2 знаения

INSERT INTO test VALUES (40), (30);

--Запускаем ХП от test

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

--Вставляем знаяения, которые меньше текущих

INSERT INTO test VALUES (1), (2);

--Изменение данных при помщи Update

UPDATE test

SET id = 50;

SELECT \* FROM spectable ;

--Запускаем ХП от test

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

--Вывод значений таблицы

SELECT \* FROM spectable;

--Создаем новую таблицу с столбцами 'num\_value1', 'num\_value2'

CREATE TABLE test2

(

num\_value1 INT NOT NULL,

num\_value2 INT NOT NULL

);

--Вставка значений

INSERT INTO test2 VALUES (20, 30);

--Запус ХП от num\_value2

SELECT Search\_Inc('test2', 'num\_value2');

--Запус ХП от num\_value1

SELECT Search\_Inc('test2','num\_value1');

распечатка спец. таблицы

SELECT \* FROM test2;

--Изменение данных при помщи Update

UPDATE test2

SET num\_value2 = 30

WHERE num\_value2 = 20;

SELECT \* FROM spectable;

вызов ХП с параметрами 'test2' и 'num\_value1'

SELECT Search\_Inc('test2','num\_value1');

--Вставка значений меньше имеющихся в таблице

INSERT INTO test2 VALUES(2, 13);

SELECT \* FROM spectable;

--Удаление ХП

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

--Удаление таблиц

DROP TABLE SpecTable;

DROP TABLE test;

DROP TABLE test2;

--Удаление вспомогательной функции

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