**ЛАБОРАТОРНАЯ РАБОТА № 5. ТРИГГЕРЫ**

Триггер − это хранимая процедура особого типа, активизируемая при каком-либо событии, связанном с изменением данных конкретной таблицы. Триггеры ассоциируются с такими операциями, как вставка (INSERT), обновление (UPDATE) и удаление (DELETE) данных.

В PostgreSQL триггеры могут вызывать функции, написанные на любом языке, но за одним исключением: функция не может быть полностью реа­лизована на SQL

Триггер создается на основе существующей функции. В определении триггера указывается, должна ли заданная функция вызывать­ся до или после выполнения некоторой операции с таблицей. Синтаксис опреде­ления триггера выглядит так:

CREATE TRIGGER триггер {BEFORE | AFTER} {событие [OR событие ... ]} ON таблица

FOR EACH { ROW | STATEMENT }

EXECUTE PROCEDURE функция (аргументы)

Приведем еще один пример использования триггеров при ведении журналов изменений. Журнал изменений в БД представляет собой таблицу БД, в которой фиксируются действия над всей БД или отдельными ее таблицами. В многопользовательских системах ведение такого журнала позволяет определить источник недостоверных или искаженных данных.

Определим в БД таблицу klient\_journal: в которую будем автоматически записывать любые изменения, добавления, удаления в таблице clienty. При этом будем фиксировать дату, операцию (INS, UPD, DEL) над таблицей clienty, а также новое и старое значения столбцов (old\_values, new\_values). Для операции удаления новое значение столбцов new\_values будет пустым. Для операции добавления пустым будет старое значение столбцов old\_values.

CREATE TRIGGER trig\_cln

AFTER INSERT OR UPDATE OR DELETE

ON bootsshop.clienty

FOR EACH ROW

EXECUTE PROCEDURE bootsshop.client\_action();

CREATE OR REPLACE FUNCTION bootsshop.client\_action()

RETURNS trigger AS

$BODY$

DECLARE

op TEXT;

new\_v TEXT;

old\_v TEXT;

BEGIN

IF(TG\_OP = 'INSERT') THEN

OP = 'INS';

new\_v = NEW.id\_cln ||', ' || NEW.cln\_family ||', ' ||NEW.cln\_name;

old\_v = '';

INSERT INTO bootsshop.klient\_journal (dat\_izm, deistvie, old\_values, new\_values)

VALUES (NOW(), op, old\_v, new\_v);

RETURN NEW;

ELSEIF (TG\_OP = 'UPDATE') THEN

OP = 'UPD';

new\_v = NEW.id\_cln ||', ' || NEW.cln\_family ||', ' ||NEW.cln\_name;

old\_v = OLD.id\_cln ||', ' || OLD.cln\_family ||', ' ||OLD.cln\_name;

INSERT INTO bootsshop.klient\_journal (dat\_izm, deistvie, old\_values, new\_values)

VALUES (NOW(), op, old\_v, new\_v);

RETURN NEW;

ELSE

OP = 'DEL';

new\_v = '';

old\_v = OLD.id\_cln ||', ' || OLD.cln\_family ||', ' ||OLD.cln\_name;

INSERT INTO bootsshop.klient\_journal (dat\_izm, deistvie, old\_values, new\_values)

VALUES (NOW(),op,old\_v,new\_v);

RETURN NEW;

END IF;

RETURN NULL;

END;

$BODY$

LANGUAGE plpgsql VOLATILE

COST 100;

ALTER FUNCTION bootsshop.client\_action()

OWNER TO postgres;