# Скелет функции в PostgreSQL

Язык PL/pgSQL

1. CREATE OR REPLACE FUNCTION \_calc\_repair\_time(\_id smallint)
2. RETURNS boolean
3. LANGUAGE plpgsql
4. as $$
5. <тело\_функции>
6. $$;

# Вызов функции

Простой вызов функции на SQL работает через SELECT:

1. SELECT concat\_lower\_or\_upper('Hello', 'World', true);

# Механизм работы триггеров

Триггер является указанием, что база данных должна автоматически выполнить заданную функцию, всякий раз когда выполнен определённый тип операции. Триггеры можно использовать с таблицами, с представлениями и с внешними таблицами.

# Создание триггера

CREATE TRIGGER создаёт новый триггер. Триггер будет связан с указанной таблицей, представлением или сторонней таблицей и будет выполнять заданную функцию *имя\_функции* при определённых событиях.

Виды триггеров:

1. Триггеры уровня строк (вызывается один раз для каждой строки)
2. Триггеры уровня оператора (вызываются один раз после при выполнении оператора)

# Триггерная функция

Триггерная функция должна быть создана до триггера. Она должна быть объявлена без аргументов и возвращать тип trigger. (Триггерная функция получает данные на вход посредством специально переданной структуры TriggerData, а не в форме обычных аргументов.)

После создания триггерной функции создаётся триггер с помощью CREATE TRIGGER. Одна и та же триггерная функция может быть использована для нескольких триггеров.

NEW – Тип данных RECORD. Переменная содержит новую строку базы данных для команд INSERT/UPDATE в триггерах уровня строки. В триггерах уровня оператора и для команды DELETE эта переменная имеет значение null.. Триггер

Если триггер срабатывает до или вместо события, он может пропустить операцию с текущей строкой, либо изменить добавляемую строку (только для операций INSERT и UPDATE). Если триггер срабатывает после события, он «видит» все изменения, включая результат действия других триггеров

# Рекурсивный вызов триггеров

Если триггерная функция выполняет команды SQL, эти команды могут заново запускать триггеры. Это известно как каскадные триггеры.

**Ограничение readOnly для поля с помощью триггера**

Чтобы сделать ограничение readonly для поля таблицы в триггере можно создать условие:

1. if(OLD.имя\_поля IS DISTINCT FROM NEW.имя\_поля) then
2. RAISE EXCEPTION 'field имя\_поля IS read only';
3. end if;

**Про NEW и OLD**

Это псевдоотношения (то есть таблица или строка), которые заполняются данными команд INSERT, UPDATE и DELETE. NEW соответствует новой строке добавляемой в таблицу, а OLD старой, которая храниться в таблице по первичному ключу. Обращение к полям NEW/OLD происходит через точку NEW.имя\_параметра. Таблица значений при различных операциях:

|  |  |
| --- | --- |
| INSERT | |
| OLD | пустое значение |
| NEW | содержит новую строку |
| UPDATE | |
| OLD | старая строка из таблицы |
| NEW | содержит новую строку |
| DELETE | |
| OLD | старая строка из таблицы |
| NEW | пустое значение |