

Министерство науки и высшего образования Российской Федерации  
федеральное государственное автономное образовательное учреждение  
высшего образования  
«Национальный исследовательский университет ИТМО»  
Факультет инфокоммуникационных технологий

**Лабораторная работа №3**  
**«Процедуры, функции, триггеры в PostgreSQL»**  
**по дисциплине «Проектирование и**  
**реализация баз данных»**

Выполнили:  
студенты II курса ИКТ  
группы К3241  
Конев А. и Хайрнасов А.

Проверил:  
Говорова М.М.

Санкт-Петербург  
2022

**Цель работы:** овладеть практическими создания и использования процедур, функций и триггеров в базе данных PostgreSQL.

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД PostgreSQL, JetBrains DataGrip.

**Практическое задание:**

Модифицировать триггер (триггерную функцию) на проверку корректности входа и выхода сотрудника (см. Практическое задание 1 Лабораторного практикума (Приложение)) с максимальным учетом «узких» мест некорректных данных по входу и выходу.

## Выполнение:

1. Создание таблицы trigger\_test

```
-- auto-generated definition
create database trigger_test
  with owner lichesscool;
```

Создание таблиц employee и time\_punch

```
-- auto-generated definition
create table employee
(
  id          serial
  |          primary key,
  username    varchar
);
```

```
-- auto-generated definition
create table time_punch
(
  id          serial
  |          primary key,
  employee_id integer
  |          references employee,
  is_out_punch boolean    default false,
  punch_time   timestamp default now()
);
```

## 2. Создание триггерной функции

```
create function fn_check_time_punch() returns trigger
    language plpgsql
as
$$
begin
    if new.punch_time <= (select tps.punch_time
                        from time_punch tps
                        where tps.employee_id = new.employee_id
                        order by tps.id desc
                        limit 1)
        or new.is_out_punch = (select tps.is_out_punch
                        from time_punch tps
                        where tps.employee_id = new.employee_id
                        order by tps.id desc
                        limit 1)
        or (new.is_out_punch = true and COUNT(
                                (select tps.punch_time
                                from time_punch tps
                                where tps.employee_id = new.employee_id
                                order by tps.id desc
                                limit 1)
                                ) = 0
    then
        return null;
    end if;
    return new;
end;
$;
```

Привязка функции к таблице time\_punch

```
-- auto-generated definition
create trigger check_time_punch
    before insert
    on time_punch
    for each row
execute procedure fn_check_time_punch();
```

### 3. Заполнение таблиц данными для тестирования

Заполнение таблицы employee

```
trigger_test> INSERT INTO public.employee (id, username)
                VALUES (DEFAULT, 'Егорыс')
[2022-05-06 19:00:57] 1 row affected in 21 ms
trigger_test> INSERT INTO public.employee (id, username)
                VALUES (DEFAULT, 'Поповыс')
[2022-05-06 19:00:57] 1 row affected in 20 ms
```

Заполнение таблицы time\_punch корректными данными

```
trigger_test> INSERT INTO public.time_punch (id, employee_id, is_out_punch, punch_time)
                VALUES (DEFAULT, 3::integer, false::boolean, '2022-05-05 10:29:11.000000'::timestamp)
[2022-05-06 19:04:17] 1 row affected in 20 ms
trigger_test> INSERT INTO public.time_punch (id, employee_id, is_out_punch, punch_time)
                VALUES (DEFAULT, 4::integer, false::boolean, '2022-05-05 14:29:42.000000'::timestamp)
[2022-05-06 19:04:17] 1 row affected in 20 ms
trigger_test> INSERT INTO public.time_punch (id, employee_id, is_out_punch, punch_time)
                VALUES (DEFAULT, 3::integer, true::boolean, '2022-05-05 14:30:25.000000'::timestamp)
[2022-05-06 19:04:17] 1 row affected in 20 ms
trigger_test> INSERT INTO public.time_punch (id, employee_id, is_out_punch, punch_time)
                VALUES (DEFAULT, 4::integer, true::boolean, '2022-05-05 16:31:15.000000'::timestamp)
[2022-05-06 19:04:18] 1 row affected in 20 ms
```

Вывод данных

	id	employee_id	is_out_punch	punch_time
1	15	3	false	2022-05-05 10:29:11.000000
2	16	4	false	2022-05-05 14:29:42.000000
3	17	3	• true	2022-05-05 14:30:25.000000
4	18	4	• true	2022-05-05 16:31:15.000000

### 4. Тестирование на некорректных данных

- 1) Попытка вставить выход внутри уже существующего стэка входов/выходов

```
trigger_test.public> INSERT INTO public.time_punch (id, employee_id, is_out_punch, punch_time)
                     VALUES (DEFAULT, 3::integer, true::boolean, '2022-05-05 11:29:11.000000'::timestamp)
[2022-05-06 19:18:33] completed in 21 ms
```

Транзакция не была выполнена

	id	employee_id	is_out_punch	punch_time
1	15	3	false	2022-05-05 10:29:11.000000
2	16	4	false	2022-05-05 14:29:42.000000
3	17	3	• true	2022-05-05 14:30:25.000000
4	18	4	• true	2022-05-05 16:31:15.000000

2) Попытка выйти уже после совершенного выхода

```
trigger_test.public> INSERT INTO public.time_punch (id, employee_id, is_out_punch, punch_time)
                      VALUES (DEFAULT, 3::integer, true::boolean, '2022-05-05 15:30:25.000000'::timestamp)
[2022-05-06 19:33:15] completed in 23 ms
```

Транзакция не была выполнена

	id	employee_id	is_out_punch	punch_time
1	15	3	false	2022-05-05 10:29:11.000000
2	16	4	false	2022-05-05 14:29:42.000000
3	17	3	• true	2022-05-05 14:30:25.000000
4	18	4	• true	2022-05-05 16:31:15.000000

3) Введем нового пользователя

```
trigger_test.public> INSERT INTO public.employee (id, username)
                      VALUES (DEFAULT, 'Новенький')
[2022-05-06 19:23:01] 1 row affected in 19 ms
```

Пусть он попытается сразу выйти

```
INSERT INTO public.time_punch (id, employee_id, is_out_punch, punch_time)
VALUES (DEFAULT, 5::integer, true::boolean, '2022-05-05 11:29:11.000000'::timestamp);
```

Транзакция не была выполнена

	id	employee_id	is_out_punch	punch_time
1	15	3	false	2022-05-05 10:29:11.000000
2	16	4	false	2022-05-05 14:29:42.000000
3	17	3	• true	2022-05-05 14:30:25.000000
4	18	4	• true	2022-05-05 16:31:15.000000

**Вывод:** Была модифицирована и протестирована триггерная функция для проверки корректности входа и выхода сотрудника с учетом “узких” мест.