

Министерство науки и высшего образования Российской Федерации  
федеральное государственное автономное образовательное учреждение  
высшего образования  
“Национальный исследовательский университет ИТМО”

Факультет инфокоммуникационных технологий

## **ЛАБОРАТОРНАЯ РАБОТА №5**

**Процедуры, функции, триггеры в PostgreSQL**  
**по дисциплине:**  
**«Проектирование и реализация баз данных»**

**Выполнила:**

Коник Анастасия Александровна  
Группа К3241

**Проверила:**

Горова Марина Михайловна

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, SQL Shell (psql).

## Вариант 1

### Практическое задание:

- I. Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).
- II. Создать триггер для логирования событий вставки, удаления, редактирования данных в базе данных PostgreSQL (согласно индивидуальному заданию, часть 5). Допустимо создать универсальный триггер или отдельные триггеры на логирование действий.

**База данных:** “Учет выполнения заданий”



## Выполнение:

### Задание 1. Создайте хранимые процедуры.

- 1) Для повышения оклада сотрудников, выполнивших задания с трехдневным опережением графика на заданный процент (10%).

Data Output									
	emp_id [PK] integer	pos_id integer	first_name text	last_name text	middle_name text	dep_id integer	contacts text	project_id integer	salary double precision
1	1	1	Anastasia	Vorobieva	Alexandrovna	1	sbssu@mail.ru	111	85000
2	2	2	Olga	Minaeva	Igorevna	2	hjkaw@gmail.com	222	95000
3	3	3	Alex	Kross	[null]	3	dguws@ya.ru	111	60000
4	4	2	Anna	Karavaeva	Dmitrievna	1	tefas@mail.ru	111	90000
5	5	1	Lev	Korolyev	[null]	2	huhd@ya.ru	222	100000
6	6	3	Andrey	Smirnov	Alexandrovich	3	egysv@gmail.com	333	50000
7	7	2	Daniil	Mironenko	Fedorovich	1	dwuid@mail.ru	111	110000
8	8	3	Svetlana	Pupirishkina	Gennadiyevna	2	feuhi@mail.ru	222	55000
9	9	1	Marina	Konovalova	Danilovna	3	kcjae@gmail.com	222	80000
10	10	1	Igor	Sivaev	Sergeevich	2	wafcz@gmail.com	[null]	105000

```
CREATE PROCEDURE salary_raise()  
AS  
$$  
BEGIN  
UPDATE j_acc.employee  
SET salary = salary*1.1  
WHERE salary IN (SELECT employee.salary FROM j_acc.employee  
JOIN j_acc.task ON employee.emp_id = task.emp_id WHERE (task.deadline - task.factual_end_date) = 3);  
END;  
$$ language plpgsql;
```

#### Data Output Messages

CREATE PROCEDURE

Query returned successfully in 172 msec.

#### Data Output Messages

	emp_id [PK] integer	pos_id integer	first_name text	last_name text	middle_name text	dep_id integer	contacts text	project_id integer	salary integer
1	1	1	Anastasia	Vorobieva	Alexandrovna	1	sbssu@mail.ru	111	85000
2	2	2	Olga	Minaeva	Igorevna	2	hjkaw@gmail.com	222	104500
3	3	3	Alex	Kross	[null]	3	dguws@ya.ru	111	60000
4	4	2	Anna	Karavaeva	Dmitrievna	1	tefas@mail.ru	111	90000
5	5	1	Lev	Korolyev	[null]	2	huhd@ya.ru	222	100000
6	6	3	Andrey	Smirnov	Alexandrovich	3	egysv@gmail.com	333	50000
7	7	2	Daniil	Mironenko	Fedorovich	1	dwuid@mail.ru	111	110000

2) Для вычисления количества проектов, в выполнении которых участвует сотрудник.

```
CREATE function emp_pr(empl int) returns table(empl_id int, number_of_pr bigint)
AS $$
begin
return query
select empl, count(employee.project_id) from j_acc.employee where employee.emp_id = empl;
end;
$$
language plpgsql;

select * from emp_pr(3)
```

Data Output		Messages
	empl_id integer	number_of_pr bigint
1	3	1

3) Для поиска номера телефона сотрудника (телефон установлен в каждом отделе).

```
CREATE function phone_num(empl int) returns table(empl_id int, number text)
AS $$
begin
return query
select empl, department.phone_number from j_acc.department join j_acc.employee on department.dep_id = employee.dep_id
where employee.emp_id = empl;
end;
$$
language plpgsql;

select * from phone_num(2)
```

Data Output		Messages
	empl_id integer	number text
1	2	+79274137832

## Задание 2. Создать триггер для логирования событий вставки, удаления, редактирования

Создана таблица logs

```
CREATE OR REPLACE FUNCTION add_to_log() RETURNS TRIGGER AS $$
DECLARE
    mstr varchar(30);
    astr varchar(100);
    retstr varchar(254);
BEGIN
    IF TG_OP = 'INSERT' THEN
        astr = NEW;
        mstr := 'Add new data';
        retstr := mstr||astr;
        INSERT INTO j_acc.logs(text,added,table_name) values (retstr,NOW(),TG_TABLE_NAME);
        RETURN NEW;
    ELSIF TG_OP = 'UPDATE' THEN
        astr = NEW;
        mstr := 'Update data ';
        retstr := mstr||astr;
        INSERT INTO j_acc.logs(text,added,table_name) values (retstr,NOW(),TG_TABLE_NAME);
        RETURN NEW;
    ELSIF TG_OP = 'DELETE' THEN
        astr = OLD;
        mstr := 'Remove data';
        retstr := mstr || astr;
        INSERT INTO j_acc.logs(text,added,table_name) values (retstr,NOW(),TG_TABLE_NAME);
        RETURN OLD;
    END IF;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER t_emp AFTER INSERT OR UPDATE OR DELETE
ON j_acc.employee FOR EACH ROW EXECUTE PROCEDURE add_to_log ();
```

CREATE FUNCTION

Query returned successfully in 72 msec.

CREATE TRIGGER

Query returned successfully in 107 msec.

```
INSERT INTO j_acc.department(dep_id, phone_number, name) VALUES (4, '+789198342782', 'Something');
UPDATE j_acc.department SET name = 'Anything' where dep_id = 4;
DELETE from j_acc.department where dep_id = 4;

select * from j_acc.logs
```

### Data Output Messages

	text text	added date	table_name text
1	Add new data(4,+789191253612,Something)	2022-05-03	department
2	Update data (4,+789191253612,Anything)	2022-05-03	department
3	Remove data(4,+789191253612,Anything)	2022-05-03	department

**Вывод:**

В данной работе были изучены функции и процедуры, и созданы триггеры для корректного хранения данных.