

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»**

**Выполнила:**

студентка II курса ИКТ  
группы К3243

Костень Анна Сергеевна

**Проверила:**

Говорова Марина Михайловна

Санкт-Петербург  
2021

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

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД PostgreSQL 1X, pgAdmin 4.

**Практическое задание:**

1. Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).
2. 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.

**Выполнение:**

Наименование БД – «БД Контроль выполнения заданий». Схема логической модели БД, сгенерированная в Generate ERD:

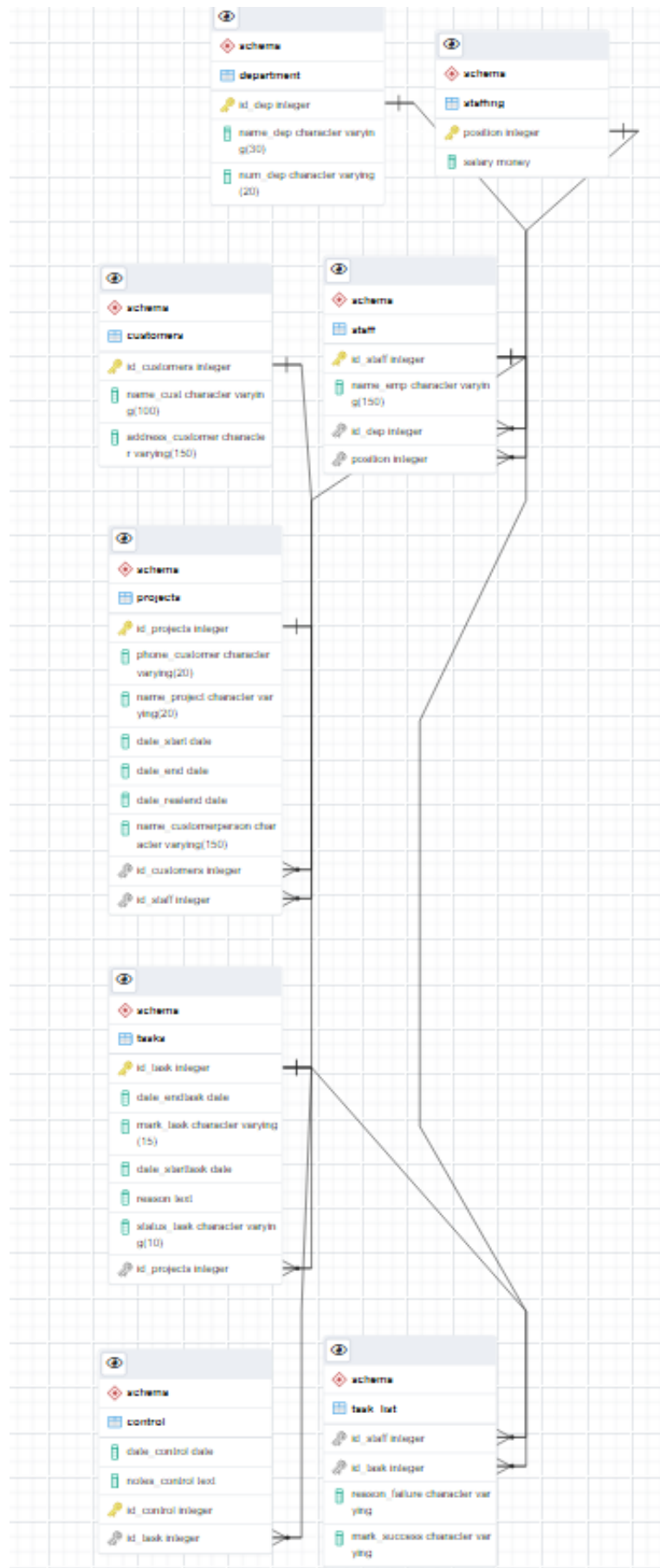


Рисунок 1 – Схема, сгенерированная в Generate ERD

## Задание 1. Создайте хранимые процедуры

1. Для повышения оклада сотрудников, выполнивших задания с трехдневным опережением графика на заданный процент.

Результат		План выполнения		Сообщения		Notifications	
	id_staff integer	 name_emp character varying (150)	 id_dep integer	 position integer	 position integer	 salary numeric	
1	2	Титенко Елена	1	13	13	500000.00	
2	3	Махнева Анастасия	5	11	11	67000.00	
3	4	Смирнов Тимур Олегович	2	2	2	340000.00	
4	5	Асонов Николай Павлович	3	2	2	340000.00	
5	6	Фатин Леша)	4	13	13	500000.00	

Рисунок 2 – Таблица с зарплатами до выполнения процедуры

```
1 create or replace function raise_salary(percent integer) returns integer
2 as $$
3 begin
4 update schema.staffing
5 set salary = salary * ((percent/100) + 1)
6 where staffing.position in (select position from schema.staff, schema.projects
7                             where staff.id_staff = projects.id_staff and (projects.date_end - projects.date_realend >= 3));
8 return 1;
9 end;
10 $$ language plpgsql;
```

Результат   Сообщения   Notifications

CREATE FUNCTION

Рисунок 3 – Первая функция

В моей базе данных не было сотрудников, работающих на опережение, поэтому после вызова данной функции зарплаты не изменились.

2. Для вычисления кол-ва проектов, в выполнении которых участвует сотрудник

```
1 create or replace function staff_proj(staff integer) returns table (st_id integer, nb_proj integer)
2 as
3 $$
4 begin
5 return query
6 select projects.id_staff, count(projects.id_projects)
7 from schema.projects where id_staff = staff group by projects.id_staff;
8 end;
9 $$ language plpgsql;
```

Результат   Сообщения   Notifications

CREATE FUNCTION

Рисунок 4 – Вторая функция

```
1 select * from staff_proj(6)
```

Результат   Сообщения   Notifications			
	st_id integer	nb_proj bigint	
1	6	5	

Рисунок 5 – Результат выполнения второй функции

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

```
1 create function staff_phone(st integer) returns table (st_id integer, phone_st text)
2 as
3 $$
4 begin
5 return query
6 select staff.id_staff, department.num_dep
7 from schema.staff, schema.department where staff.id_dep = department.id_dep
8 and staff.id_staff = st;
9 end;
10 $$ language plpgsql;
```

Результат   Сообщения   Notifications

CREATE FUNCTION

Рисунок 6 – Третья функция

1	<code>select * from staff_phone(5)</code>			
Результат		Сообщения	Notifications	
	st_id integer		phone_st text	
1	5	766655221		

Рисунок 7 – Результат выполнения третьей функции

Задание 2. Создание триггера для логирования вставки, обновления и удаления.

```

1 CREATE OR REPLACE FUNCTION add_to_log() RETURNS TRIGGER AS $$
2 DECLARE
3     mstr varchar(30);
4     astr varchar(100);
5     retstr varchar(254);
6 BEGIN
7     IF TG_OP = 'INSERT' THEN
8         astr = NEW.id;
9         mstr := 'Add new user ';
10        retstr := mstr||astr;
11        INSERT INTO logs(text,added) values (retstr,NOW());
12        RETURN NEW;
13    ELSIF TG_OP = 'UPDATE' THEN
14        astr = NEW.id;
15        mstr := 'Update user ';
16        retstr := mstr||astr;
17        INSERT INTO logs(text,added) values (retstr,NOW());
18        RETURN NEW;
19    ELSIF TG_OP = 'DELETE' THEN
20        astr = OLD.id;
21        mstr := 'Remove user ';
22        retstr := mstr || astr;
23        INSERT INTO logs(text,added) values (retstr,NOW());
24        RETURN OLD;
25    END IF;
26 END;

```

Результат   Сообщения   Notifications

CREATE FUNCTION

Рисунок 8 – Функция логирования

```

1 CREATE TRIGGER staff_moment AFTER INSERT OR UPDATE OR DELETE ON schema.staff FOR EACH ROW EXECUTE PROCEDURE add_to_log ();

```

Результат   Сообщения   Notifications

CREATE TRIGGER

Рисунок 9 – Триггер логирования таблицы работников



	text 	added 
1	Add new employee 5	2022-05-09 15:04:05.808367
2	Update employee 6	2022-05-09 15:05:51.164084

Рисунок 10 – Таблица логирования изменений в таблице сотрудников

### **Вывод.**

Я овладела навыками создания функций и триггеров в базе данных PostgreSQL, а также узнала что моя версия PostgreSQL не поддерживает создание процедур, что значительно осложнило выполнение лабораторной работы.