

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 5
по теме: процедуры, функции, триггеры в PostgreSQL.
по дисциплине: Проектирование и реализация баз данных

Специальность:
09.03.03 Мобильные и сетевые технологии

Проверил:
Говорова М.М. _____
Дата: «__»_____ 2021 г.
Оценка _____

Выполнил:
студентка группы К3240
Балдина Д.Д.

Санкт-Петербург 2022 г.

ЦЕЛЬ РАБОТЫ

Овладеть практическими создания и использования процедур, функций и триггеров в базе данных PostgreSQL.

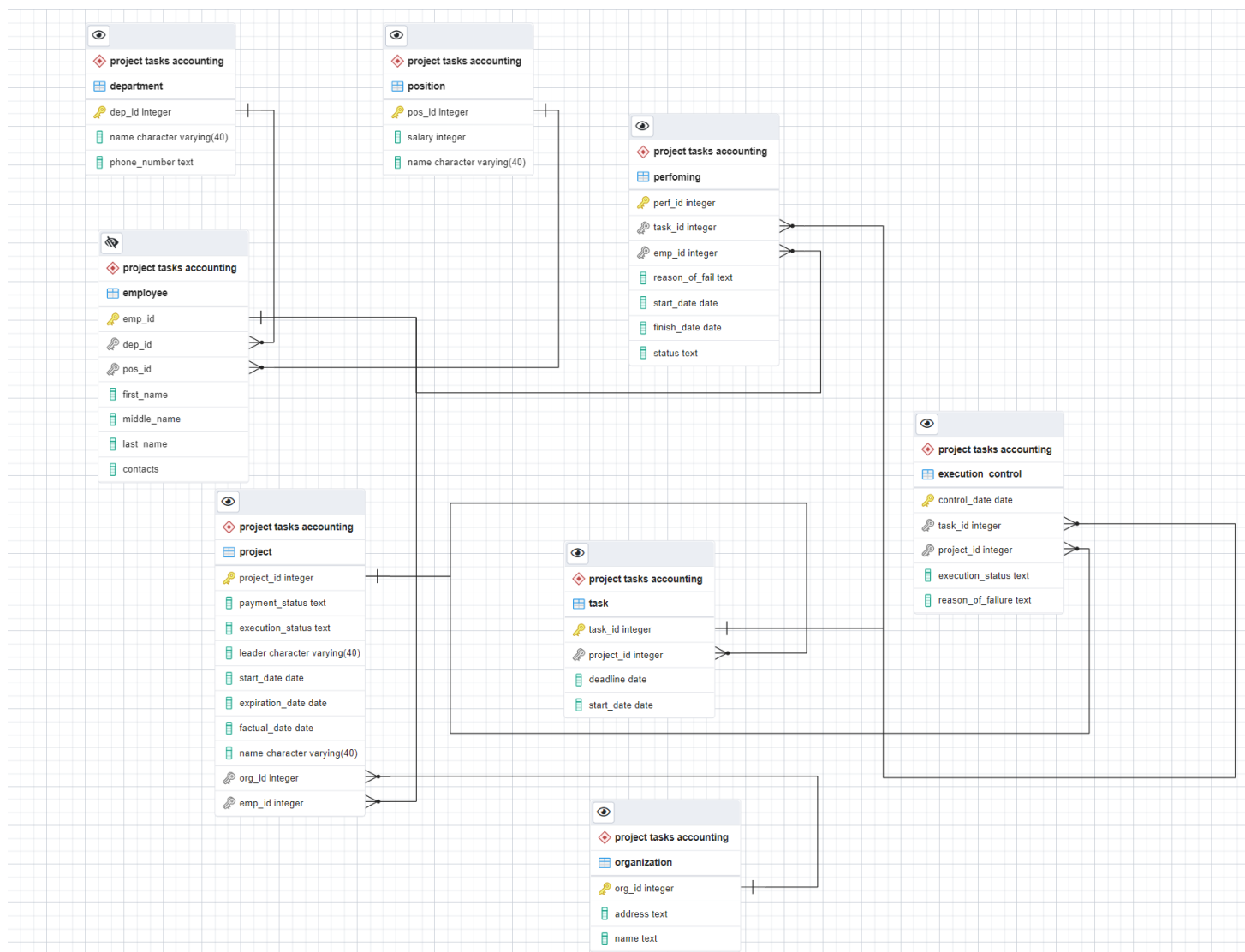
Оборудование: компьютерный класс.

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.

СХЕМА БАЗЫ ДАННЫХ



Выполнение:

Задание 1. Создайте хранимые процедуры.

1) Для повышения оклада сотрудников, выполнивших задания с трехдневным опережением графика на заданный процент(5%).

	emp_id [PK] integer	dep_id integer	pos_id integer	first_name text	middle_name text	last_name text	contacts text	project_id integer	salary double precision
1	3	333	3	Daria	[null]	Krasnova	dar@mail.ru, +283285	2	80000
2	4	333	3	Andrew	[null]	Shepel	sh@amail.ru	1	90000
3	5	111	2	Anna	Ivanovna	Kobel	ko@gmail.ru	2	100000
4	6	222	3	Lev	Victorovich	Askin	levv@mail.ru	2	110000
5	7	111	1	Zak	[null]	Newman	zakn@gmail.com	3	140000
6	8	222	2	Daria	Danilovna	Bor	borda@mail.ru	2	98800
7	2	222	2	Olga	Danilovna	Zilina	zhil@gmail.com	1	120000
8	9	333	1	Polina	Alekseevna	Mat	pol@mail.ru	1	95600
9	1	111	1	Ivan	Olegovich	Ivanov	iiii@mail.ru	1	80000
10	10	333	2	Mark	Davidovich	Morsov	mor@mail.ru	1	75000

	perf_id [PK] integer	task_id integer	emp_id integer	reason_of_fail character varying (40)	start_date date	finish_date date	project_id integer	status text	actual_finish date
1	1	1	1	[null]	2021-01-17	2021-02-01	1	finished	2021/01/29
2	2	2	2	was ill	2021-03-05	2021-04-23	2	Delayed	[null]
3	3	3	3	[null]	2021-05-15	2021-06-01	2	In work	[null]
4	4	4	4	[null]	2021-12-20	2022-01-20	3	finished	17/01/2022
5	5	5	5	[null]	2022-02-01	2022-03-02	2	finished	[null]
6	6	6	6	[null]	2021-04-15	2022-03-15	2	finished	12/03/2022

```

1 CREATE PROCEDURE salary_increase_5_procent()
2 AS
3 $$
4 BEGIN
5 UPDATE ps_tasks.employee
6 SET salary = salary*1.05
7 WHERE salary IN (SELECT employee.salary FROM ps_tasks.employee
8 JOIN ps_tasks.performing ON performing.emp_id = employee.emp_id WHERE (performing.finish_date - performing.actual_finish) = 3);
9 END;
10 $$ LANGUAGE plpgsql;
11

```

Результат План выполнения Сообщения Notifications

CREATE PROCEDURE

Запрос завершён успешно, время выполнения: 51 мсек.

1	1	111	1	Ivan	Olegovich	Ivanov	iiiiiv@mail.ru	1	88200
2	2	222	2	Olga	Danilovna	Zilina	zhil@gmail.com	1	120000
3	3	333	3	Daria	[null]	Krasnova	dar@mail.ru, +283285	2	88200
4	4	333	3	Andrew	[null]	Shepel	sh@amail.ru	1	94500
5	5	111	2	Anna	Ivanovna	Kobel	ko@gmail.ru	2	100000
6	6	222	3	Lev	Victorovich	Askin	levv@mail.ru	2	115500
7	7	111	1	Zak	[null]	Newman	zakn@gmail.com	3	140000
8	8	222	2	Daria	Danilovna	Bor	borda@mail.ru	2	98800
9	9	333	1	Polina	Alekseevna	Mat	pol@mail.ru	1	95600
10	10	333	2	Mark	Davidivich	Morsov	mor@mail.ru	1	75000

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

```

1 CREATE FUNCTION emp_project(emp int) RETURNS TABLE (empl_id int , number_of_pr bigint)
2 AS
3 $$
4 BEGIN
5 RETURN QUERY
6 SELECT employee.emp_id, COUNT(employee.project_id) FROM ps_tasks.employee where employee.emp_id = emp GROUP BY employee.emp_id;
7 END;
8 $$ language plpgsql;

```

Результат План выполнения Сообщения Notifications

CREATE FUNCTION

Запрос завершён успешно, время выполнения: 57 msec.

```

1 SELECT * FROM emp_project(1)

```

Результат		План выполнения		Сообщения	Notifications
	empl_id integer		number_of_pr bigint		
1	1		1		

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

```

1 CREATE FUNCTION emp_phone(emp int) RETURNS TABLE (empl_id int , phone_of_emp text)
2 AS
3 $$
4 BEGIN
5 RETURN QUERY
6 SELECT employee.emp_id, department.phone_number FROM ps_tasks.employee JOIN ps_tasks.department
7 ON department.dep_id = employee.dep_id WHERE employee.emp_id = emp;
8 END;
9 $$ language plpgsql;

```

Результат План выполнения Сообщения Notifications

CREATE FUNCTION

Запрос завершён успешно, время выполнения: 45 msec.

```

1 SELECT * FROM emp_phone(1)

```

Результат План выполнения Сообщения Notifications

	empl_id integer	phone_of_emp text	
1	1	+79617573850	

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

```

1 CREATE OR REPLACE FUNCTION add_to_log() RETURNS TRIGGER AS $$
2 DECLARE
3     mstr varchar(30);
4     astr varchar(100);
5     retstr varchar(254);
6 BEGIN
7     IF TG_OP = 'INSERT' THEN
8         astr = NEW;
9         mstr := 'Add data ';
10        retstr := mstr || astr;
11        INSERT INTO ps_tasks.logs(text,added, table_name) values (retstr,NOW(),TG_TABLE_NAME);
12        RETURN NEW;
13    ELSIF TG_OP = 'UPDATE' THEN
14        astr = NEW;
15        mstr := 'Update data ';
16        retstr := mstr || astr;
17        INSERT INTO ps_tasks.logs(text,added, table_name) values (retstr,NOW(),TG_TABLE_NAME);
18        RETURN NEW;
19    ELSIF TG_OP = 'DELETE' THEN
20        astr = OLD.id;
21        mstr := 'Remove data ';
22        retstr := mstr || astr;
23        INSERT INTO ps_tasks.logs(text,added, table_name) values (retstr,NOW(),TG_TABLE_NAME);
24        RETURN OLD;
25    END IF;
26 END;
27 $$ LANGUAGE plpgsql;
28

```

```

1 CREATE TRIGGER t_dep AFTER INSERT OR UPDATE OR DELETE ON department FOR EACH ROW EXECUTE PROCEDURE add_to_log ()

```

Результат План выполнения Сообщения Notifications

CREATE TRIGGER

Запрос завершён успешно, время выполнения: 68 msec.

```

1 INSERT INTO ps_tasks.department (dep_id, phone_number, name) VALUES (4, '+79635458791', 'Innovation')

```

Результат План выполнения Сообщения Notifications

INSERT 0 1

Запрос завершён успешно, время выполнения: 44 msec.

```
1 UPDATE ps_tasks.department SET name = 'Rules' where dep_id = 4
```

Результат План выполнения Сообщения Notifications

UPDATE 1

Запрос завершён успешно, время выполнения: 67 msec.

```
1 DELETE FROM ps_tasks.department WHERE dep_id = 4
```




Результат План выполнения Сообщения Notifications

DELETE 1

Запрос завершён успешно, время выполнения: 33 msec.

```
1 SELECT * FROM ps_tasks.logs
```

Результат План выполнения Сообщения Notifications

	 text	 added timestamp without time zone	 table_name text
1	Add data (4,Innovation,+79635458791)	2022-05-03 18:20:45.960683	department
2	Update data (4,Rules,+79635458791)	2022-05-03 18:24:08.029873	department
3	Remove data (4,Rules,+79635458791)	2022-05-03 18:26:00.068528	department

ВЫВОДЫ

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