

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»**

Отчет
По Лабораторной работе 3
Процедуры, функции, триггеры в PostgreSQL
по дисциплине **«Базы данных»**

Автор: Горбатов Дмитрий Алексеевич

Факультет: ИКТ

Группа: K32402

Преподаватель: Говорова Марина Михайловна

Санкт-Петербург 2023

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

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, SQL Shell (psql)..

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:

Задание 1. Создайте хранимые процедуры.

1) Для повышения оклада сотрудников, выполнивших задания с трехдневным опережением графика на заданный процент (3%).

Код запроса:

```
CREATE OR REPLACE PROCEDURE "Uchet".increase_salary()
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE "Uchet".employee_in_department eid
    SET salary = salary * 1.03
    FROM "Uchet".task t
    WHERE eid.id_employee_in_department = t.id_employee_in_department
    AND (t.deadline_of_task - t.date_of_finish) = 3;
END;
$$;
```

Результат до:

	id_employee_in_department [PK] integer	hours_of_work character (20)	salary integer	id_department integer	id_employee integer	id_positions integer
1	11	30	40000	1	123	1
2	12	35	30000	1	234	2
3	13	36	40000	2	345	1
4	14	37	30000	2	456	2
5	15	34	40000	3	567	1
6	16	33	30000	3	678	2

Результат после:

	id_employee_in_department [PK] integer	hours_of_work character (20)	salary integer	id_department integer	id_employee integer	id_positions integer
1	14	37	30000	2	456	2
2	16	33	30000	3	678	2
3	15	34	40000	3	567	1
4	12	35	30000	1	234	2
5	11	30	40000	1	123	1
6	13	36	41200	2	345	1

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

Код запроса:

```
CREATE FUNCTION "Uchet".count_employee_projects (emp int) returns table
(id_employee_in_department int, id_project bigint)
LANGUAGE plpgsql
AS $$
BEGIN
    return query
    SELECT project.id_employee_in_department, count (project.id_project) from
    "Uchet".project where project.id_employee_in_department = emp
    GROUP BY project.id_employee_in_department;
END;
$$;
```

Результат запроса:

1 select * from "Uchet".count_employee_projects (14)

	id_employee_in_department integer	id_project bigint
1	14	2

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

Код запроса:

```
CREATE FUNCTION "Uchet".employee_phone_num (emp int) returns table
(emp_id int, phone_emp character)
language plpgsql
as $$
begin
return query
select
employee_in_department.id_employee_in_department,
employee.contacts_of_employee
from
"Uchet".employee,
"Uchet".employee_in_department
where employee_in_department.id_employee = employee.id_employee and emp =
employee_in_department.id_employee_in_department;
end;
$$
```

Результат запроса:

```
1 select * from "Uchet".employee_phone_num (13);|
```

Data Output			Messages	Notifications
	emp_id integer	phone_emp character		
1	13	89345345345		

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

Код запроса:

```
create or replace function "Uchet".add_to_new() returns trigger
language plpgsql
as
$$
declare
```

```

        mstr varchar(30);
        astr varchar(100);
        restr varchar (255);
begin
if TG_OP = 'INSERT' then
    astr = new;
    mstr := 'Add data';
    restr := mstr||astr;
    insert into "Uchet".logs (text, added, table_name) values (restr, now(), TG_TABLE_NAME);
    return new;
elsif TG_OP = 'UPDATE' then
    astr = new;
    mstr := 'Update data';
    restr := mstr||astr;
    insert into "Uchet".logs (text, added, table_name) values (restr, now(), TG_TABLE_NAME);
    return new;
elsif TG_OP = 'DELETE' then
    astr = old;
    mstr := 'Remove data';
    restr := mstr||astr;
    insert into "Uchet".logs (text, added, table_name) values (restr, now(), TG_TABLE_NAME);
    return old;
end if;
end;
$$

```

```

create trigger divv after insert or delete or update on "Uchet".employee for each row execute procedure
"Uchet".add_to_new()

```

1

```

insert into "Uchet".employee (id_employee, name_of_employee, contacts_of_employee) values (901,
'NIKITA', '88887776655')

```

2

```

update "Uchet".employee set contacts_of_employee ='88883213213' where id_employee = 901

```

3

```

delete from "Uchet".employee where id_employee = 901

```

Результат запроса:

```
1 select * from "Uchet".logs
```

Data Output Messages Notifications

	id [PK] integer	text character varying (255)	added timestamp without time zone	table_name character varying (50)
1	1	Add data(901,"NIKITA ",88887776655)	2023-06-07 21:26:58.509347	employee
2	2	Update data(901,"NIKITA ",88883213213)	2023-06-07 21:28:54.965674	employee
3	3	Remove data(901,"NIKITA ",88883213213)	2023-06-07 21:29:27.967701	employee

Вывод:

В данной работе я изучил функции и процедуры, создал их под разные задачи и условия. Также я узнал про триггеры и использовал их для хранения информации об изменениях в данных своей базы.