

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

О Т Ч Е Т

по лабораторной работе
«Процедуры, функции, триггеры в PostgreSQL»
по дисциплине «Проектирование и реализация баз данных»

Автор: Анисимов Степан Дмитриевич

Факультет: ИКТ

Группа: К32422

Преподаватель: Говорова Марина Михайловна

Санкт-Петербург

2023

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

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.

Задание 1.

1. Вывести список всех звонков заданного абонента.

```
1 create or replace function user_calls(target_cot integer)
2 returns table(id_call integer, id_cot integer, id_zone integer, payment_status boolean,
3               payment_date date, act_payment_date date, payment_sum real, duration real,
4               target_number bigint, call_date date, call_time time with time zone)
5 as $$
6 select * from calls where id_cot = target_cot;
7 $$
8 language sql
```

	id_call integer	id_cot integer	id_zone integer	payment_status boolean	payment_date date	act_payment_date date	payment_sum real	duration real	target_number bigint	call_date date	call_time time with time zone
1	11	7	2	true	2023-03-04	2023-03-04	0	12.4	79999989999	2023-03-04	11:14:11+03:00
2	12	7	4	true	2023-03-05	2023-03-06	10.25	20.5	375178465829	2023-03-05	23:10:16+03:00
3	13	7	2	true	2023-03-06	2023-03-06	0	36.6	79214045002	2023-03-06	16:40:17+03:00

2. Вывести задолженность по оплате для заданного абонента.

```
1 create or replace function debt(target_cot integer)
2 returns real
3 as $$
4 select sum(duration) from calls where id_cot = target_cot and payment_status = 'false' group by id_cot
5 $$
6 language sql;
7 select * from debt(5)
```

	debt real
1	7.3

3. Рассчитать общую стоимость звонков по каждой зоне за истекшую неделю.

```
1 create or replace function sum_by_zone()
2 returns table (zone_sum real, id_zone integer)
3 as $$
4 select sum(payment_sum), id_zone from calls where call_date between current_date - 200 and current_date group by id_zone
5 $$
6 language sql;
7 select * from sum_by_zone()
```

	zone_sum real	id_zone integer
1	0	2
2	21	3
3	10.25	4
4	20	5

Задание 2.

Авторский триггер, автоматически уменьшает остаток минут соответствующего абонента при совершении звонка.

	id_cot [PK] integer	id_tariff integer	id_contract integer	gb_remainder integer	sms_remainder integer	mins_remainder integer	begin_date date	end_date date
1	6	1	2	80	1000	990	2023-03-04	2024-03-03
2	7	3	3	14	489	400	2023-03-04	2023-03-07
3	8	2	4	[null]	354	228	2023-03-05	2024-03-04
4	9	1	7	94	994	944	2023-03-05	2024-03-04
5	10	1	3	97	980	951	2023-03-07	2024-03-03
6	5	1	1	98	967	787	2023-03-03	2024-03-02

```

1 create or replace function deduct_minutes() returns trigger
2 as $$
3 begin
4 update contracts_on_tariffs
5 set mins_remainder =
6 (select mins_remainder - new.duration from contracts_on_tariffs where id_cot = new.id_cot)
7 where id_cot = new.id_cot;
8 return new;
9 end;
10 $$
11 language plpgsql;
12 create or replace trigger new_call after insert on calls for each row execute function deduct_minutes()
```

	id_cot [PK] integer	id_tariff integer	id_contract integer	gb_remainder integer	sms_remainder integer	mins_remainder integer	begin_date date	end_date date
1	7	3	3	14	489	400	2023-03-04	2023-03-07
2	8	2	4	[null]	354	228	2023-03-05	2024-03-04
3	9	1	7	94	994	944	2023-03-05	2024-03-04
4	10	1	3	97	980	951	2023-03-07	2024-03-03
5	5	1	1	98	967	787	2023-03-03	2024-03-02
6	6	1	2	80	1000	979	2023-03-04	2024-03-03

Выводы:

По результатам данной лабораторной работы были получены навыки создания функций, процедур и триггеров в PostgreSQL, созданы необходимые функции в соответствии с заданием, а также авторский триггер.