

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

Факультет инфокоммуникационных технологий

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5
тема: Процедуры, функции, триггеры в PostgreSQL.
дисциплина: «ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗ ДАННЫХ»

Выполнил:
Костенников Данил Вячеславович
Группа К3241
Проверила:
Говорова 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).

Практическое задание:

Вариант 1

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.

Наименование БД: «Служба заказа такси»

Задание 4. Создать хранимые процедуры:

- Для вывода данных о пассажирах, которые заказывали такси в заданном, как параметр, временном интервале.
- Вывести сведения о том, куда был доставлен пассажир по заданному номеру телефона пассажира.
- Для вычисления суммарного дохода таксопарка за истекший месяц.

Задание 5. Создать необходимые триггеры

Выполнение:

1. Хранимые процедуры

- Для вывода данных о пассажирах, которые заказывали такси в заданном, как параметр, временном интервале.

```

1 create or replace function order_time_interval(start_time date, end_time date)
2 returns table (id bigint, name varchar, phone_number varchar)
3 as
4 $$
5 begin
6     return query
7     select "Client"."Id", "Client"."Name", "Client"."Phone_number"
8     from public."Client", public."Order"
9     where "Client"."Id" = "Order"."Client_Id"
10    and "Order"."Date_of_call" between start_time and end_time;
11 end;
12 $$ language plpgsql;
13 SELECT * from order_time_interval('2022-04-17', '2022-04-18')

```

Data Output	Explain	Messages	Notifications	Query History
	id bigint	name character varying	phone_number character varying	
1	1	Danil	+7904000100	
2	1	Danil	+7904000100	
3	3	Fedor	+7904000102	
4	3	Fedor	+7904000102	
5	5	Ilya	+7904000104	
6	5	Ilya	+7904000104	

- Вывести сведения о том, куда был доставлен пассажир по заданному номеру телефона пассажира.

```

1 create or replace function destination(phone_number varchar)
2 returns table (phone varchar, drop_off_address varchar)
3 as
4 $$
5 begin
6     return query
7     select "Client"."Phone_number", "Order"."Drop_off_address"
8     from public."Client"
9     join public."Order" on "Client"."Id" = "Order"."Client_Id"
10    where "Client"."Phone_number" = phone_number;
11 end;
12 $$ language plpgsql;
13 SELECT * from destination('+7904000100')

```

Data Output	Explain	Messages	Notifications	Query History
	phone character varying	drop_off_address character varying		
1	+7904000100	Alp str		
2	+7904000100	Lomo str 9		

- Для вычисления суммарного дохода таксопарка за истекающий месяц.

```

1 create or replace function income()
2 returns table (income double precision)
3 as
4 $$
5 begin
6     return query
7     select sum ("Order"."Total_distance" * "Rate"."Killometr_price" + "Order"."Waiting_time" + "Rate"."Waiting_fine" )
8     from public."Order", public."Rate"
9     where "Order"."Rate_Id" = "Rate"."Id"
10    and extract(month from "Order"."Date_of_call") = extract (month from current_date) -1;
11 end;
12 $$ language plpgsql;
13 SELECT * from income()

```

Data Output Explain Messages Notifications Query History

	income	
	double precision	
1	2144	

2. Необходимые триггеры

- 1) Создание таблицы logs
- 2) Создание триггерной функции
- 3) Создание триггера
- 4) Проверка

```

1 INSERT INTO public."Employee"(
2     "Id", "Adress", "Code", "Name", "Passport number", "Phone", "Position", "Category")
3     VALUES (5, 'Spb', '005', 'Demirkan', '0005', '+7904000004', 'Driver', 'B');

```

Data Output Explain Messages Notifications Query History

INSERT 0 1

Query returned successfully in 57 msec.

```
1 UPDATE public."Employee"  
2   SET "Name"='Feador'  
3   WHERE "Id" = 5;
```

Data Output Explain Messages Notifications Query History

UPDATE 1

Query returned successfully in 63 msec.

```
1 DELETE FROM public."Employee"  
2   WHERE "Id" = 5;
```

Data Output Explain Messages Notifications Query History

DELETE 1

Query returned successfully in 78 msec.

Query Editor

1

2

SELECT * FROM public." logs"

Data Output

Explain

Messages

Notifications

	text text	added time without time zone	
1	Add ne...	02:06:21.900581	
2	Update...	02:10:09.64579	
3	Remov...	02:10:52.588488	

Вывод

В ходе проделанной работы были изучены и созданы процедуры и функции, создан триггер для логирования событий.