## Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

# «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

#### Отчет

по лабораторной работе №5 «Процедуры, функции, триггеры в PostgreSQL» по дисциплине «Проектирование и реализация баз данных»

Автор:Зеленин Д.С.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

# Оглавление

Цель работы:	3
Практическое задание	3
Выполнение	3
Вывод	7

**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 2 (тах - 8 баллов)

- 1. Создать процедуры/функции согласно индивидуальному заданию (часть 4).
- 2. Создать авторский триггер по варианту индивидуального задания.

#### Выполнение

### Создайте хранимые процедуры:

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

```
CREATE OR REPLACE FUNCTION passenger interval search(
      start datetime time without time zone,
      end datetime time without time zone, date date
RETURNS TABLE (
      номер заказа character varying,
      имя character varying,
      номер bigint,
      конечная точка character varying,
      время заказа time without time zone
) AS $$
BEGIN
      RETURN QUERY
      SELECT
      order .order id,
      client.name.
      client.client phone number,
      order .end point,
      order .time order
      FROM
      order
      client ON order .fk client id = client.client id
      WHERE
      order .time order BETWEEN start datetime AND end datetime and order .date = date;
END:
$$ LANGUAGE plpgsql;
```

SELECT \* FROM passenger interval search('08:00:00', '19:00:00', '2023-12-07');

## B psql

lab-3=# SELECT	* FROM pass	enger_interval	_search('08:00:00', '19:00	9:00','2023-12-07');
номер_заказа	имя	номер	конечная_точка	время_заказа
	+	<del> </del>		H
or19	Дмитрий	1234567890	Улица Чайковского, 17	08:00:00
or12	Василий	9434939204	Улица Чехова, 89	17:00:00
or20	Георгий	1987654321	Улица Гюго, 7	11:00:00
or29	Дмитрий	9876543210	Улица Гагарина, 45	15:40:00
or15	Дмитрий	9876543210	Улица Булгакова, 72	17:00:00
or11	Андрей	6543210987	Улица Гагарина, 45	13:00:00
or25	Андрей	6543210987	Улица Гагарина, 45	16:40:00
or17	Денис	5435350543	Улица Бродского, 78	08:00:00
or13	Анастасия	89817656505	Улица Достоевского, 14	11:45:00
or24	Анастасия	89817656505	Улица Булгакова, 72	17:11:00
or31	Василиса	8539298439	Улица Гагарина, 4	11:40:00
or14	Василиса	8539298439	Улица Лермонтова, 56	15:30:00
or16	Семен	4392424893	Улица Солженицына, 19	13:44:00

	номер_заказа character varying	имя character varying	номер bigint	конечная_точка character varying	время_заказа time without time zone
1	or19	Дмитрий	1234567890	Улица Чайковского, 17	08:00:00
2	or12	Василий	9434939204	Улица Чехова, 89	17:00:00
3	or20	Георгий	1987654321	Улица Гюго, 7	11:00:00
4	or29	Дмитрий	9876543210	Улица Гагарина, 45	15:40:00
5	or15	Дмитрий	9876543210	Улица Булгакова, 72	17:00:00
6	or11	Андрей	6543210987	Улица Гагарина, 45	13:00:00
7	or25	Андрей	6543210987	Улица Гагарина, 45	16:40:00

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

```
CREATE OR REPLACE FUNCTION passenger_phone_search(phone bigint)
RETURNS TABLE (номер_заказа character varying, имя character varying, номер bigint, конечная_точка character varying ) AS $$
BEGIN
```

**RETURN QUERY** 

SELECT order\_id, client.name, client.client\_phone\_number, order\_.end\_point FROM order\_

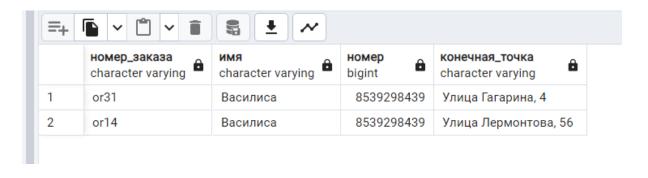
JOIN client ON order\_.fk\_client\_id = client.client\_id WHERE client.client\_phone\_number = phone;

END;

\$\$ LANGUAGE plpgsql;

SELECT \* FROM passenger\_phone\_search(8539298439);

## B psql



• Для вычисления суммарного дохода таксопарка за истекший месяц.

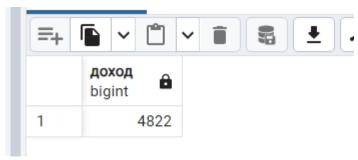
```
CREATE OR REPLACE FUNCTION Income_for_the_past_month()
RETURNS TABLE (доход bigint) AS $$
BEGIN

RETURN QUERY SELECT SUM(money)::bigint FROM order_
WHERE EXTRACT(MONTH FROM date) = EXTRACT(MONTH FROM CURRENT_DATE - INTERVAL '1 month')

AND EXTRACT(YEAR FROM date) = EXTRACT(YEAR FROM CURRENT_DATE - INTERVAL '1 month');
END;
$$ LANGUAGE plpgsql;

SELECT * FROM Income_for_the_past_month();
B psql
```

```
lab-3=# SELECT * FROM Income_for_the_past_month();
доход
-----
4822
(1 строка)
```



#### Триггер:

• Будем вычислять скидку для клиента в зависимости от суммарного количества его поездок.

```
CREATE OR REPLACE FUNCTION calculate_discount_trigger()
RETURNS TRIGGER AS $$

DECLARE

total_trip INT;

BEGIN

SELECT COALESCE(COUNT(fk_client_id), 0) INTO total_trip
FROM order_
WHERE fk_client_id = NEW.fk_client_id;

IF total_trip >= 2 AND total_trip < 20 THEN
NEW.money := 0.95 * NEW.money;
```

```
ELSIF total_trip >= 20 AND total_trip < 50 THEN

NEW.money := 0.85 * NEW.money;

ELSIF total_trip >= 50 THEN

NEW.money := 0.70 * NEW.money;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;
```

CREATE TRIGGER calculate\_discount\_trigger
BEFORE INSERT ON order\_
FOR EACH ROW
EXECUTE FUNCTION calculate\_discount\_trigger();

У данного клиента больше 2 поездок, поэтому изначальная стоимость поездки в размере 340 была уменьшена до 323.

end_point character varying	actual_boarding_time time without time zone	planing_boarding_time time without time zone	money bigint	date /	time_order time without time zone
Улица Гагарина, 46	19:39:00	19:11:00	323	2023-12-24	19:11:00

#### Вывод

В ходе лабораторной работы была освоена работа с процедурами и триггера.