**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: Фирсов Илья

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

**Оглавление**

Цель работы 3

Практическое задание 3

Выполнение 3

Процедуры/функции согласно индивидуальному заданию (часть 4). 3

Вывод 6

# Цель работы

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

# Практическое задание

**Вариант 1 (max - 6 баллов)**

1. Создать процедуры/функции 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.

**Вариант 2 (max - 8 баллов)**

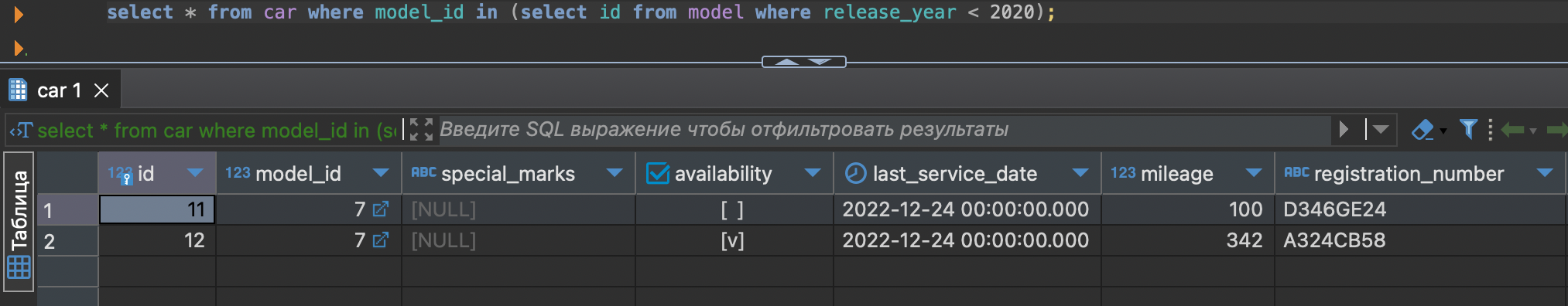
1. Создать процедуры/функции согласно индивидуальному заданию (часть 4).

*2.1.* Модифицировать триггер (триггерную функцию) на проверку корректности входа и выхода сотрудника (см. Практическое задание 1 Лабораторного практикума (Приложение)) с максимальным учетом «узких» мест некорректных данных по входу и выходу).

*2.2.* Создать авторский триггер по варианту индивидуального задания.

# Выполнение

# Процедуры/функции согласно индивидуальному заданию (часть 4).

* Выполнить списание автомобилей, выпущенных ранее заданного года.

**CREATE** **OR** **REPLACE** **PROCEDURE** remove\_cars(

**IN** **year** **INT**

)

**LANGUAGE** plpgsql

**as**

**$$**

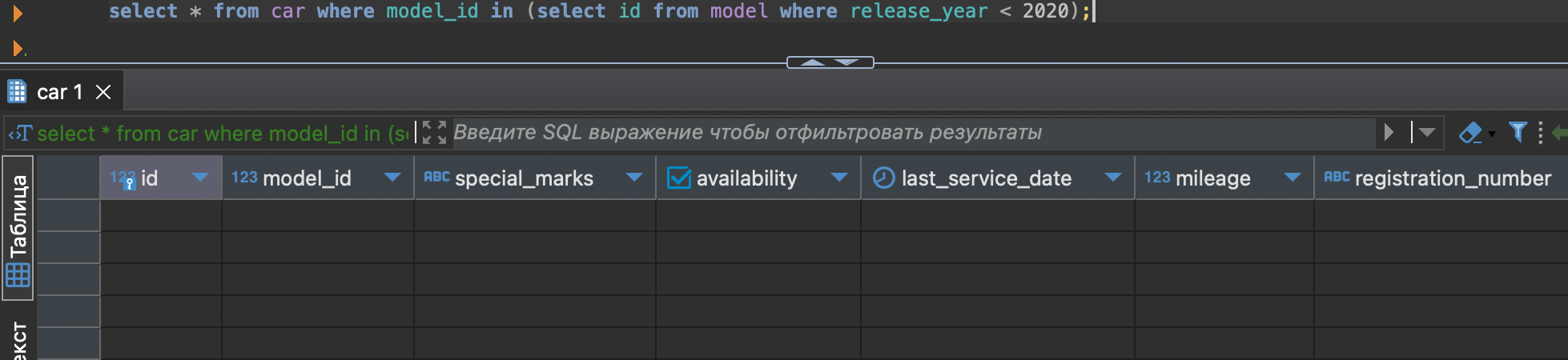
**BEGIN**

**delete** **from** car **where** model\_id **in** (**select** id **from** model **where** release\_year < **year**);

**END**;

**$$**;

**call** remove\_cars(2020);



* Выдачи автомобиля и расчета стоимости с учетом скидки постоянным клиентам.

**CREATE** **OR** **REPLACE** **FUNCTION** rent(

given\_client\_id **INT**,

given\_car\_id **INT**,

given\_start **TIMESTAMP** **WITH** **TIME** **zone**,

given\_end **TIMESTAMP** **WITH** **TIME** **ZONE**

)

**RETURNS** **DOUBLE** **PRECISION**

**LANGUAGE** plpgsql

**AS** **$$**

**DECLARE**

discounted\_price **DOUBLE** **PRECISION**;

**BEGIN**

**SELECT** rb.bid\_price \* (1 - c.discount\_percent / 100)

**INTO** discounted\_price

**FROM** rent\_bid rb

**JOIN** client c **ON** c.id = given\_client\_id

**WHERE** rb.model\_id = (**SELECT** model\_id **FROM** car **WHERE** id = given\_car\_id);

**INSERT** **INTO** contract (client\_id, car\_id, status, employee\_id, start\_date, end\_date)

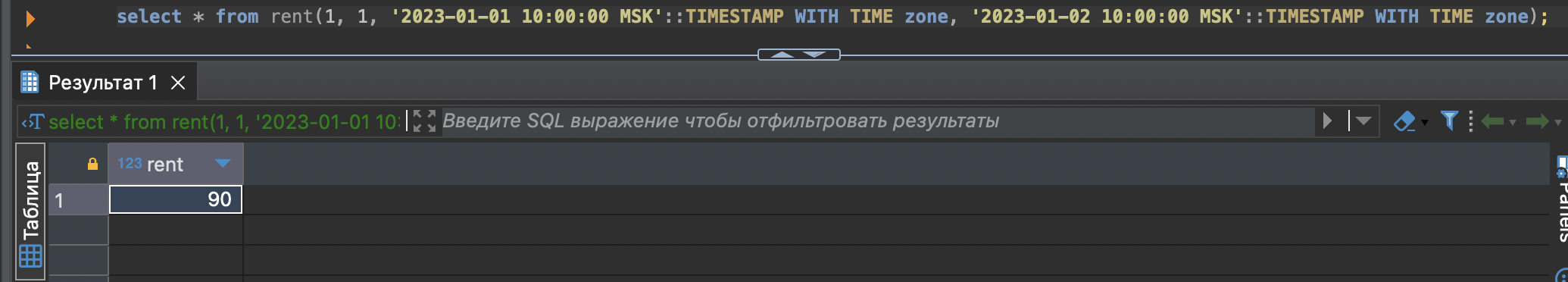
**VALUES** (given\_client\_id, given\_car\_id, **'Active'**, 1, given\_start, given\_end);

**RETURN** discounted\_price;

**END**;

**$$**;

**select** \* **from** rent(1, 1, **'2023-01-01 10:00:00 MSK'**::**TIMESTAMP** **WITH** **TIME** **zone**, **'2023-01-02 10:00:00 MSK'**::**TIMESTAMP** **WITH** **TIME** **zone**);



* Для вычисления количества автомобилей заданной марки.

**CREATE** **OR** **REPLACE** **FUNCTION** count\_cars\_by\_brand(

brand\_name **VARCHAR**

) **RETURNS** **INTEGER**

**LANGUAGE** plpgsql

**as** **$$**

**DECLARE**

car\_count **INTEGER**;

**BEGIN**

**SELECT** **COUNT**(\*)

**INTO** car\_count

**FROM** car c

**JOIN** model m **ON** c.model\_id = m.id

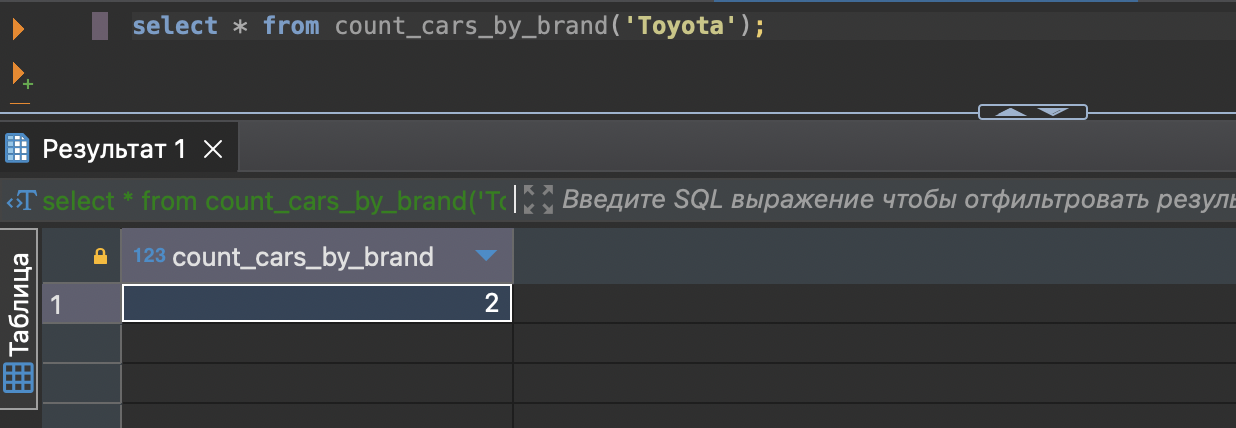
**WHERE** m.mark = brand\_name;

**RETURN** car\_count;

**END**;

**$$**;

**select** \* **from** count\_cars\_by\_brand(**'Toyota'**);



**Авторский триггер по варианту индивидуального задания.**

**CREATE** **OR** **REPLACE** **FUNCTION** update\_car\_availability()

**RETURNS** **TRIGGER** **AS** **$$**

**BEGIN**

**IF** TG\_OP = **'INSERT'** **OR** TG\_OP = **'UPDATE'** **THEN**

**UPDATE** car

**SET** availability = **FALSE**

**FROM** contract

**WHERE** car.id = **NEW**.car\_id

**AND** contract.status = **'Active'**;

**END** **IF**;

**RETURN** **NEW**;

**END**;

**$$** **LANGUAGE** plpgsql;

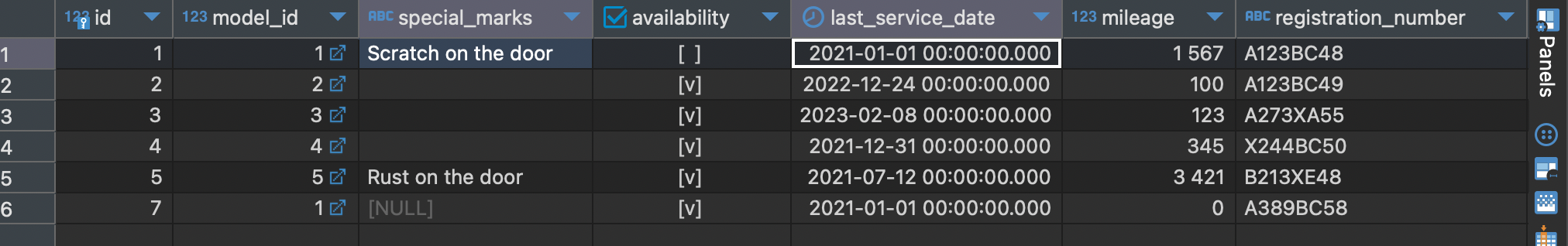
**CREATE** **TRIGGER** trigger\_update\_car\_availability

**AFTER** **INSERT** **OR** **UPDATE** **OR** **DELETE** **ON** contract

**FOR** **EACH** **ROW**

**EXECUTE** **FUNCTION** update\_car\_availability();

**select** \* **from** rent(1, 1, **'2023-01-01 10:00:00 MSK'**::**TIMESTAMP** **WITH** **TIME** **zone**, **'2023-01-02 10:00:00 MSK'**::**TIMESTAMP** **WITH** **TIME** **zone**);



# Вывод

Мы овладели практическими навыками создания и использования процедур, функций и триггеров в базе данных PostgreSQL, научились писать и тестировать их в консоли SQL Shell.