Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»
Факультет инфокоммуникационных технологий
Мегафакультет информационных и трансляционных технологий

# Проектирование и реализация баз данных

Лабораторная работа №3
"Запросы на выборку и модификацию данных, представления и индексы в PostgreSQL"

**Работу выполнил:**
Студент II курса
Коробковский В.А.
**Группа:**
K3242
**Преподаватель:**
Говорова М.М.

Санкт-Петербург
2022

# Содержание

# 1. Цель работы

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

# 2. Практическое задание

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.

# 3. Выполнение

## 3.1. Описание предметной области

Сеть автомастерских осуществляет ремонт автомашин, используя для этих целей штат мастеров и свои мастерские. Стоимость ремонта включает цену деталей и стоимость работы. Заработная плата мастеров составляет 50% стоимости работы.

## 3.2. Описание базы данных

База данных "Автомастерская" состоит из следующих сущностей и реквизитов:

1) Сущность "Сотрудник"
Реквизиты:
    1. Табельный номер
    2. Id филиала, в котором работает
    3. Должность
    4. Разряд
    5. Специализация
    6. Имя сотрудника
    7. Фамилия сотрудника
    8. Отчество сотрудника

2) Сущность "Клиент"
Реквизиты:
    1. Серия и номер паспорта
    2. E-mail
    3. Телефон
    4. Имя клиента
    5. Фамилия клиента
    6. Отчество клиента

3) Сущность "Автомобиль"
Реквизиты:

1. Id автомобиля
2. Цвет
3. Год выпуска
4. Госномер
5. Код марки

4) Сущность "Модель"
Реквизиты:

1. Код марки
2. Модель
3. Марка
4. Мощность

5) Сущность "Договор на ремонт"
Реквизиты:

1. Номер договора
2. Дата заключения
3. Статус
4. Id списка услуг
5. Id автомобиля
6. Id филиала
7. Серия и номер паспорта
8. Табельный номер сотрудника

6) Сущность "Филиал"
Реквизиты:

1. Id филиала
2. Адрес
3. Id города

7) Сущность "Город"
Реквизиты:

1. Id города
2. Регион
3. Область
4. Город

8) Сущность "Состав ремонта"
Реквизиты:

1. Id ремонта

2. Номер договора
3. Дата принятия на ремонт
4. Плановая дата окончания
5. Фактическая дата окончания
6. Количество услуги
7. Виды ремонта
8. Табельный номер сотрудника
9. Id услуги

9) Сущность "Вид ремонта"
Реквизиты:

1. Id услуги
2. Название услуги
3. Стоимость услуги для одной детали

10) Сущность "Список деталей для ремонта"
Реквизиты:

1. Id списка деталей
2. Id ремонта
3. Количество каждой детали
4. Id детали

11) Сущность "Деталь"
Реквизиты:

1. Id детали
2. Название
3. Страна производителя
4. Стоимость
5. Модель автомобиля
6. Марка автомобиля

## 3.3. Схема базы данных

База данных получила название "Autorepair network" (с английского "Сеть автомастерских"). Схема логической модели базы данных, сгенерированная в Generate ERD, представлена на рисунке 3.1.

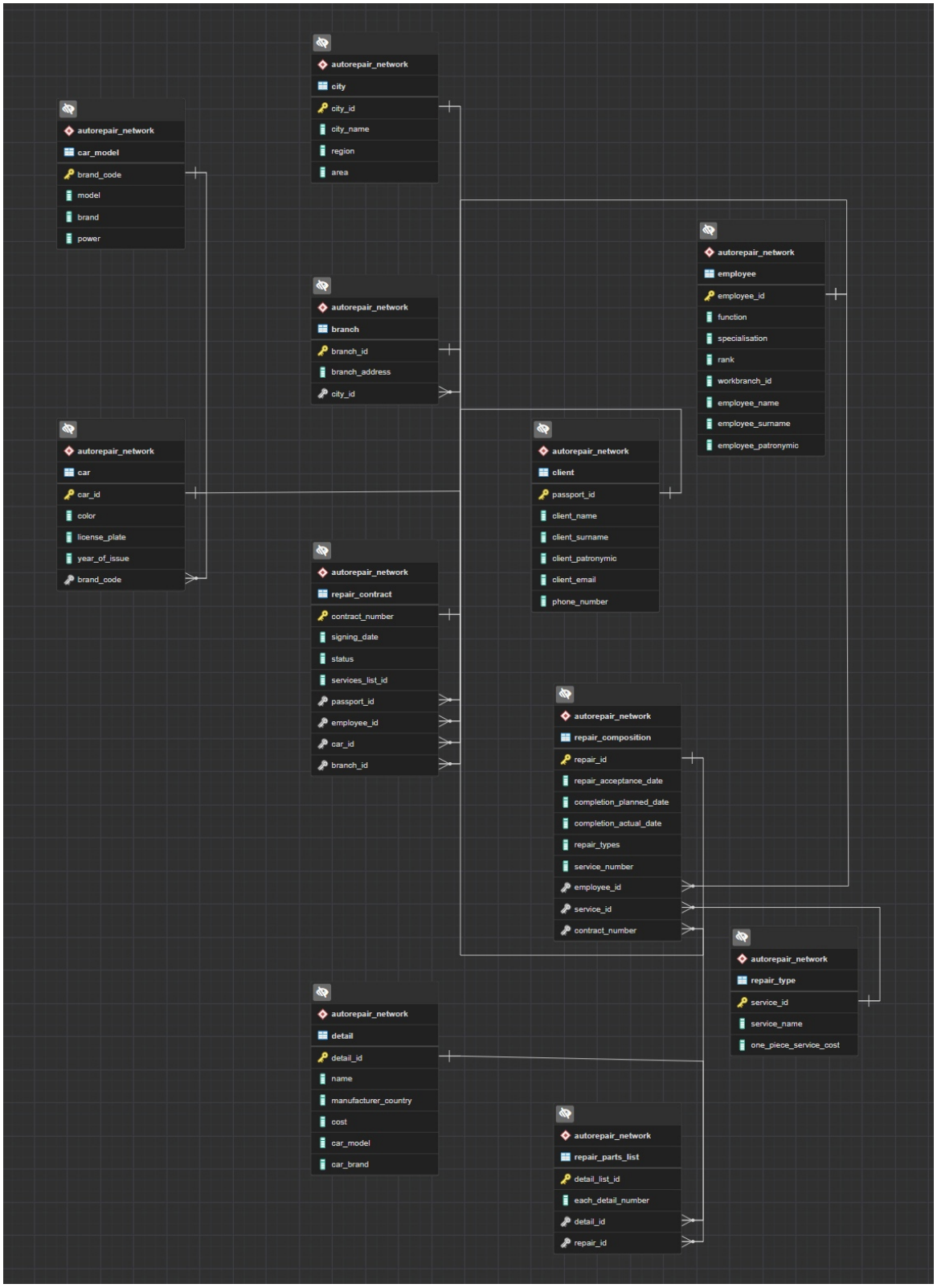


Рисунок 3.1. — Схема базы данных

## 3.4. Процедуры/функции

### 3.4.1. Задание №1

Суть функции: повышения цены деталей для автомобиля “Ford” на 10%. Результат представлен на рисунке 3.3.

```
lab1=# select * from autorepair_network.detail;
 detail_id |                   name                   | manufacturer_country |  cost  |   car_model   | car_brand
-----------+------------------------------------------+----------------------+--------+---------------+-----------
         1 | Аккумулятор                              | Япония               |  35000 | Impreza Sport | Subaru
         6 | Комплект петель для дверей               | Япония               |   2500 | RAV4          | Toyota
         2 | Двигатель Ford Explorer 4 4,6 2007 T46usem | США                | 140000 | Explorer      | Ford
(3 строки)


lab1=# create or replace function price_increase_10percent() returns numeric
lab1-# as
lab1-# $$
lab1$# begin
lab1$# UPDATE autorepair_network.detail SET cost = cost * 1.1 where car_brand = 'Ford';
lab1$# RAISE NOTICE 'Prices for parts for Ford cars updated successfully';
lab1$# return 1;
lab1$# end;
lab1$# $$ language plpgsql volatile;
CREATE FUNCTION
lab1=# select * from price_increase_10percent();
ЗАМЕЧАНИЕ:  Prices for parts for Ford cars updated successfully
 price_increase_10percent
--------------------------
                        1
(1 строка)


lab1=# select * from autorepair_network.detail;
 detail_id |                   name                   | manufacturer_country |   cost   |   car_model   | car_brand
-----------+------------------------------------------+----------------------+----------+---------------+-----------
         1 | Аккумулятор                              | Япония               |    35000 | Impreza Sport | Subaru
         6 | Комплект петель для дверей               | Япония               |     2500 | RAV4          | Toyota
         2 | Двигатель Ford Explorer 4 4,6 2007 T46usem | США                | 154000.0 | Explorer      | Ford
(3 строки)


lab1=# select * from price_increase_10percent();
ЗАМЕЧАНИЕ:  Prices for parts for Ford cars updated successfully
 price_increase_10percent
--------------------------
                        1
(1 строка)


lab1=# select * from autorepair_network.detail;
 detail_id |                   name                   | manufacturer_country |   cost    |   car_model   | car_brand
-----------+------------------------------------------+----------------------+-----------+---------------+-----------
         1 | Аккумулятор                              | Япония               |     35000 | Impreza Sport | Subaru
         6 | Комплект петель для дверей               | Япония               |      2500 | RAV4          | Toyota
         2 | Двигатель Ford Explorer 4 4,6 2007 T46usem | США                | 169400.00 | Explorer      | Ford
(3 строки)
```

Рисунок 3.2. — Результат работы функции №1

Команду можно посмотреть по этой ссылке.

### 3.4.2. Задание №2

Суть функции: повышение разряда тех мастеров, которые отремонтировали больше 3 автомобилей. Результат представлен на рисунке 3.3.

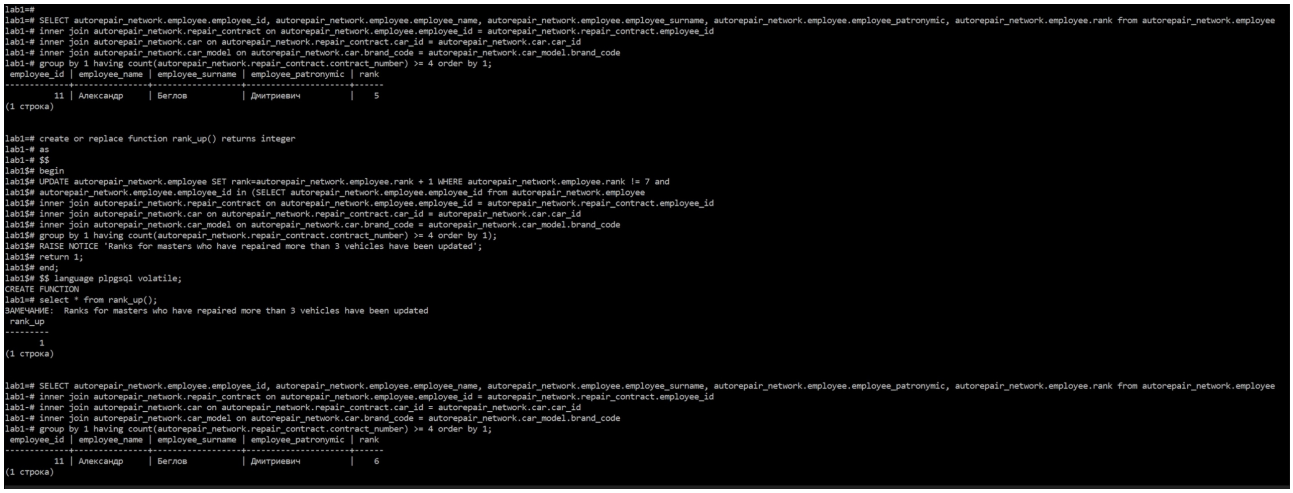


Рисунок 3.3. — Результат работы функции №2

Команду можно посмотреть по этой ссылке.

### 3.4.3. Задание №3

Суть функции: вывод количества автомобилей, отремонтированных каждым механиком за истекший квартал. Результат представлен на рисунке 3.4.

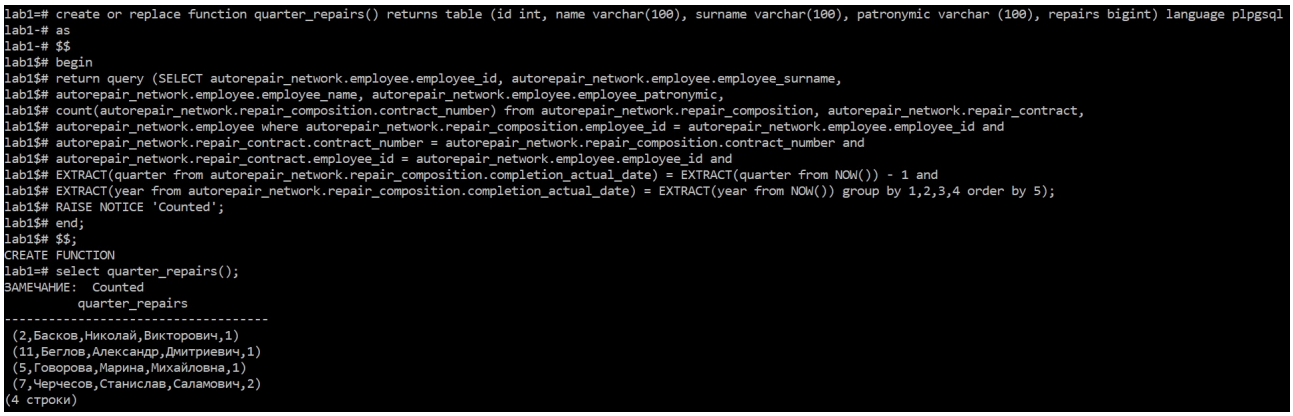


Рисунок 3.4. — Результат работы функции №3

Команду можно посмотреть по этой ссылке.

## 3.5. Триггеры

### 3.5.1. Задание №4

Суть триггера: логирование событий вставки, удаления, редактирования данных в базе данных PostgreSQL (согласно индивидуальному заданию, часть 5). Допустимо создать универсальный триггер или отдельные триггеры на логирование действий. Результат представлен на рисунках 3.5 и 3.6.

```
lab1=# create table logs (text text, added timestamp without time zone, table_name text);
CREATE TABLE
lab1=# CREATE OR REPLACE FUNCTION log_function() RETURNS TRIGGER AS $$
lab1$# DECLARE
lab1$#     mstr varchar(50);
lab1$#     astr varchar(100);
lab1$#     retstr varchar(254);
lab1$# BEGIN
lab1$#     IF TG_OP = 'INSERT' THEN
lab1$#         astr = NEW;
lab1$#         mstr := 'Add new information: ';
lab1$#         retstr := mstr || astr;
lab1$#         INSERT INTO logs(text, added, table_name) values (retstr, NOW(), TG_TABLE_NAME);
lab1$#         RETURN NEW;
lab1$#     ELSIF TG_OP = 'UPDATE' THEN
lab1$#         astr = UPDATED;
lab1$#         mstr := 'Update some information: ';
lab1$#         retstr := mstr || astr;
lab1$#         INSERT INTO logs(text, added, table_name) values (retstr, NOW(), TG_TABLE_NAME);
lab1$#         RETURN UPDATED;
lab1$#     ELSIF TG_OP = 'DELETE' THEN
lab1$#         astr = OLD;
lab1$#         mstr := 'Remove some information: ';
lab1$#         retstr := mstr || astr;
lab1$#         INSERT INTO logs(text, added, table_name) values (retstr, NOW(), TG_TABLE_NAME);
lab1$#         RETURN OLD;
lab1$#     END IF;
lab1$# END;
lab1$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
lab1=# create trigger t_client after insert or update or delete on autorepair_network.client for each row execute procedure log_function();
CREATE TRIGGER
lab1=# create trigger t_car after insert or update or delete on autorepair_network.car for each row execute procedure log_function();
CREATE TRIGGER
lab1=# create trigger t_car_model after insert or update or delete on autorepair_network.car_model for each row execute procedure log_function();
CREATE TRIGGER
lab1=# create trigger t_branch after insert or update or delete on autorepair_network.branch for each row execute procedure log_function();
CREATE TRIGGER
lab1=# create trigger t_city after insert or update or delete on autorepair_network.city for each row execute procedure log_function();
CREATE TRIGGER
lab1=# create trigger t_employee after insert or update or delete on autorepair_network.employee for each row execute procedure log_function();
CREATE TRIGGER
lab1=# create trigger t_detail after insert or update or delete on autorepair_network.detail for each row execute procedure log_function();
CREATE TRIGGER
lab1=# create trigger t_repair_composition after insert or update or delete on autorepair_network.repair_composition for each row execute procedure log_function();
CREATE TRIGGER
lab1=# create trigger t_repair_contract after insert or update or delete on autorepair_network.repair_contract for each row execute procedure log_function();
CREATE TRIGGER
lab1=# create trigger t_repair_parts_list after insert or update or delete on autorepair_network.repair_parts_list for each row execute procedure log_function();
CREATE TRIGGER
lab1=# create trigger t_repair_type after insert or update or delete on autorepair_network.repair_type for each row execute procedure log_function();
CREATE TRIGGER
lab1=#
```

Рисунок 3.5. — Создание триггеров

```
lab1=# select trigger_schema, trigger_name, action_statement
lab1-# from information_schema.triggers;
   trigger_schema    |     trigger_name     |        action_statement
---------------------+----------------------+--------------------------------
 autorepair_network | t_client             | EXECUTE FUNCTION log_function()
 autorepair_network | t_client             | EXECUTE FUNCTION log_function()
 autorepair_network | t_client             | EXECUTE FUNCTION log_function()
 autorepair_network | t_car                | EXECUTE FUNCTION log_function()
 autorepair_network | t_car                | EXECUTE FUNCTION log_function()
 autorepair_network | t_car                | EXECUTE FUNCTION log_function()
 autorepair_network | t_car_model          | EXECUTE FUNCTION log_function()
 autorepair_network | t_car_model          | EXECUTE FUNCTION log_function()
 autorepair_network | t_car_model          | EXECUTE FUNCTION log_function()
 autorepair_network | t_branch             | EXECUTE FUNCTION log_function()
 autorepair_network | t_branch             | EXECUTE FUNCTION log_function()
 autorepair_network | t_branch             | EXECUTE FUNCTION log_function()
 autorepair_network | t_city               | EXECUTE FUNCTION log_function()
 autorepair_network | t_city               | EXECUTE FUNCTION log_function()
 autorepair_network | t_city               | EXECUTE FUNCTION log_function()
 autorepair_network | t_employee           | EXECUTE FUNCTION log_function()
 autorepair_network | t_employee           | EXECUTE FUNCTION log_function()
 autorepair_network | t_employee           | EXECUTE FUNCTION log_function()
 autorepair_network | t_detail             | EXECUTE FUNCTION log_function()
 autorepair_network | t_detail             | EXECUTE FUNCTION log_function()
 autorepair_network | t_detail             | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_composition | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_composition | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_composition | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_contract    | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_contract    | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_contract    | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_parts_list  | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_parts_list  | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_parts_list  | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_type        | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_type        | EXECUTE FUNCTION log_function()
 autorepair_network | t_repair_type        | EXECUTE FUNCTION log_function()
(33 строки)
```

Рисунок 3.6. — Список созданных триггеров

Команды можно посмотреть по этой ссылке.

### 3.5.2. Проверка триггеров из задания №4

Результат представлен на рисунке 3.7.

```
lab1=# INSERT INTO autorepair_network.client(
lab1(# passport_id, client_name, client_surname, client_patronymic, client_email, phone_number)
lab1-# VALUES (4020682480, 'Александр', 'Капитонов', 'Александрович', 'kapitonov@niuitmo.ru', '+89214002020');
INSERT 0 1
lab1=#
lab1=# UPDATE autorepair_network.client set phone_number = '89314006080' where autorepair_network.client.client_surname = 'Капитонов';
UPDATE 1
lab1=#
lab1=# DELETE FROM autorepair_network.client where autorepair_network.client.client_surname = 'Капитонов';
DELETE 1
lab1=# select  * from logs;
                                          text                                                 |           added            | table_name
-----------------------------------------------------------------------------------------------+----------------------------+------------
 Add new information: (4020682480,Александр,Капитонов,Александрович,kapitonov@niuitmo.ru,+89214002020)    | 2022-05-02 18:43:12.155131 | client
 Update some information: (4020682480,Александр,Капитонов,Александрович,kapitonov@niuitmo.ru,89314006080) | 2022-05-02 18:43:12.161605 | client
 Remove some information: (4020682480,Александр,Капитонов,Александрович,kapitonov@niuitmo.ru,89314006080) | 2022-05-02 18:43:14.364818 | client
(3 строки)
```

Рисунок 3.7. — Проверка триггеров

Команды можно посмотреть по этой ссылке.

# 4. Вывод

При выполнении данной лабораторной работы я познакомился с понятиями "функция" и "триггер" и понял их различия. Также я создал функции и универсальные триггеры для всех таблиц моей базы данных в соответствии с заданием.