

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 №1.2

"Создание БД PostgreSQL в pgAdmin. Резервное копирование и
восстановление БД"

Работу
выполнили:
Студент II курса
Коробковский В.А.
Группа:
К3242
Преподаватель:
Говорова М.М.

Санкт-Петербург
2022

Содержание

1. Цель работы	4
2. Практическое задание	4
3. Выполнение	4
3.1. Описание предметной области	4
3.2. Описание базы данных	4
3.3. Дамп со скриптами работы базы данных	8
3.3.1. Создание базы данных и подключение к ней	8
3.3.2. Создание схемы базы данных	8
3.3.3. Создание таблицы "Филиал"	9
3.3.4. Создание таблицы "Автомобиль"	9
3.3.5. Создание таблицы "Модель"	10
3.3.6. Создание таблицы "Город"	10
3.3.7. Создание таблицы	
0 3.3.8. Создание таблицы	
0 3.3.9. Создание таблицы "Сотрудник"	12
3.3.10. Создание таблицы "Состав ремонта"	13
3.3.11. Создание таблицы	
0 3.3.12. Создание таблицы	
0 3.3.13. Создание таблицы "Вид ремонта"	15
3.3.14. Внесение данных в таблицу "Филиал"	15
3.3.15. Внесение данных в таблицу	
0 3.3.16. Внесение данных в таблицу	
0 3.3.17. Внесение данных в таблицу	
0 3.3.18. Внесение данных в таблицу	
0 3.3.19. Внесение данных в таблицу "Деталь"	17
3.3.20. Внесение данных в таблицу "Сотрудник"	17
3.3.21. Внесение данных в таблицу	
0 3.3.22. Внесение данных в таблицу	
0 3.3.23. Внесение данных в таблицу	
0 3.3.24. Внесение данных в таблицу "Вид ремонта"	19
3.3.25. Ограничения для таблицы "Филиал"	19
3.3.26. Ограничения для таблицы "Автомобиль"	19
3.3.27. Ограничения для таблицы "Модель"	19

0	3.3.28. Ограничения для таблицы	
0	3.3.29. Ограничения для таблицы	
0	3.3.30. Ограничения для таблицы	
0	3.3.31. Ограничения для таблицы	
0	3.3.32. Ограничения для таблицы	
	3.3.33. Ограничения для таблицы "Договор на ремонт"	21
	3.3.34. Ограничения для таблицы "Список деталей для ремонта"	21
	3.3.35. Ограничения для таблицы "Вид ремонта"	21
	3.3.36. Окончание работы скрипта дампа	21
	4. Выводы	22

1. Цель работы

Овладеть практическими навыками:

1. Создания таблиц базы данных PostgreSQL 1X с помощью программного обеспечения pgAdmin 4.
2. Заполнения созданных таблиц рабочими данными.
3. Резервного копирования базы данных.
4. Восстановления базы данных.

2. Практическое задание

1. Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию).
2. Создать схему в составе базы данных.
3. Создать таблицы базы данных.
4. Установить ограничения на данные: Primary Key, Unique, Check, Foreign Key.
5. Заполнить таблицы БД рабочими данными.
6. Создать резервную копию БД Указание: создать две резервные копии с расширением CUSTOM для восстановления БД и с расширением PLAIN для листинга в отчете. При создании резервных копий БД настроить параметры Dump options для Type of objects и Queries.
7. Восстановить БД.

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

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

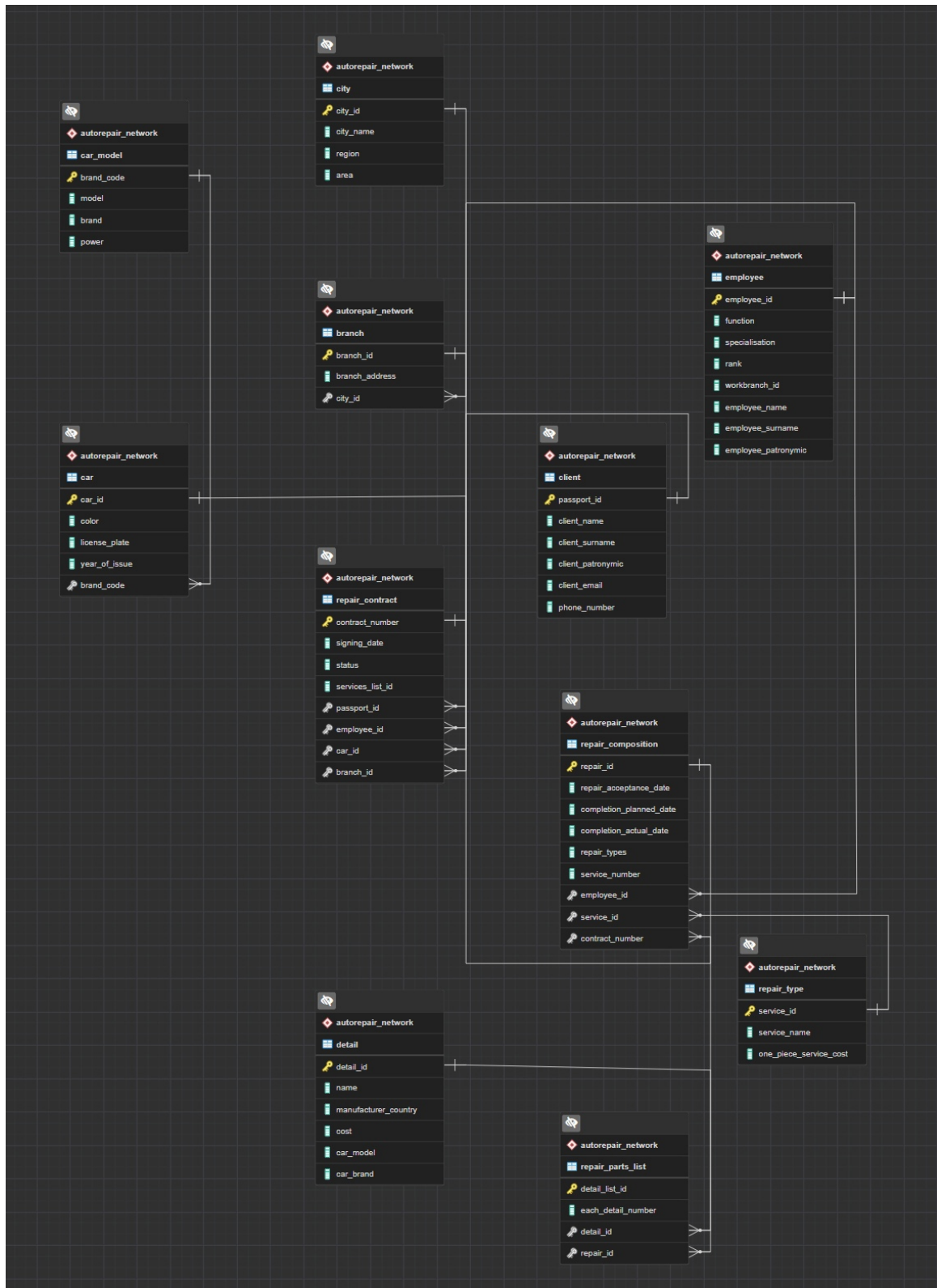


Рисунок 3.1. — Схема базы данных

3.3. Дамп со скриптами работы базы данных

3.3.1. Создание базы данных и подключение к ней

```
--  
-- PostgreSQL database dump  
--  
  
-- Dumped from database version 12.0  
-- Dumped by pg_dump version 12.0  
  
-- Started on 2022-03-04 00:20:22  
  
SET statement_timeout = 0;  
SET lock_timeout = 0;  
SET idle_in_transaction_session_timeout = 0;  
SET client_encoding = 'UTF8';  
SET standard_conforming_strings = on;  
SELECT pg_catalog.set_config('search_path', '', false);  
SET check_function_bodies = false;  
SET xmloption = content;  
SET client_min_messages = warning;  
SET row_security = off;  
  
DROP DATABASE vadem;  
  
CREATE DATABASE vadem WITH TEMPLATE = template0 ENCODING = 'UTF8'  
    LC_COLLATE = 'Russian_Russia.1251' LC_CTYPE = 'Russian_Russia.1251';  
  
ALTER DATABASE vadem OWNER TO postgres;  
  
\connect vadem  
  
SET statement_timeout = 0;  
SET lock_timeout = 0;  
SET idle_in_transaction_session_timeout = 0;  
SET client_encoding = 'UTF8';  
SET standard_conforming_strings = on;  
SELECT pg_catalog.set_config('search_path', '', false);  
SET check_function_bodies = false;  
SET xmloption = content;  
SET client_min_messages = warning;  
SET row_security = off;
```

3.3.2. Создание схемы базы данных

```
CREATE SCHEMA autorepair_network;  
ALTER SCHEMA autorepair_network OWNER TO postgres;  
SET default_tablespace = '';  
SET default_table_access_method = heap;
```


3.3.3. Создание таблицы "Филиал"

```
CREATE TABLE autorepair_network.branch (  
    branch_id integer NOT NULL,  
    branch_address text NOT NULL,  
    city_id integer NOT NULL  
);  
  
ALTER TABLE autorepair_network.branch OWNER TO postgres;  
  
COMMENT ON TABLE autorepair_network.branch IS 'Information about branches';  
  
COMMENT ON COLUMN autorepair_network.branch.branch_id IS 'Id of branch';  
  
COMMENT ON COLUMN autorepair_network.branch.branch_address IS 'Branch  
address';  
  
COMMENT ON COLUMN autorepair_network.branch.city_id IS 'Id of city';
```

3.3.4. Создание таблицы "Автомобиль"

```
CREATE TABLE autorepair_network.car (  
    car_id integer NOT NULL,  
    color text NOT NULL,  
    license_plate character varying(10) NOT NULL,  
    year_of_issue integer NOT NULL,  
    brand_code integer NOT NULL,  
    CONSTRAINT year_of_issue_check CHECK ((year_of_issue <= 2022))  
);  
  
ALTER TABLE autorepair_network.car OWNER TO postgres;  
  
COMMENT ON TABLE autorepair_network.car IS 'Information about cars';  
  
COMMENT ON COLUMN autorepair_network.car.car_id IS 'Id of car';  
  
COMMENT ON COLUMN autorepair_network.car.color IS 'Car colors';  
  
COMMENT ON COLUMN autorepair_network.car.license_plate IS 'License plate of the  
car';  
  
COMMENT ON COLUMN autorepair_network.car.year_of_issue IS 'Year of car  
manufacture';  
  
COMMENT ON COLUMN autorepair_network.car.brand_code IS 'Code of brand';
```

3.3.5. Создание таблицы "Модель"

```
CREATE TABLE autorepair_network.car_model (  
    brand_code integer NOT NULL,  
    model text NOT NULL,  
    brand text NOT NULL,  
    power integer NOT NULL,  
    CONSTRAINT power_check CHECK ((power <= 10000))  
);  
  
ALTER TABLE autorepair_network.car_model OWNER TO postgres;  
  
COMMENT ON TABLE autorepair_network.car_model IS 'Information about car models  
';  
  
COMMENT ON COLUMN autorepair_network.car_model.brand_code IS 'Code of the  
brand of the car';  
  
COMMENT ON COLUMN autorepair_network.car_model.model IS 'The model of car';  
  
COMMENT ON COLUMN autorepair_network.car_model.brand IS 'The brand of car';  
  
COMMENT ON COLUMN autorepair_network.car_model.power IS 'The power of car';
```

3.3.6. Создание таблицы "Город"

```
CREATE TABLE autorepair_network.city (  
    city_id integer NOT NULL,  
    city_name text NOT NULL,  
    region text,  
    area text  
);  
  
ALTER TABLE autorepair_network.city OWNER TO postgres;  
  
COMMENT ON TABLE autorepair_network.city IS 'Information about city';  
  
COMMENT ON COLUMN autorepair_network.city.city_id IS 'Id of city';  
  
COMMENT ON COLUMN autorepair_network.city.city_name IS 'City name';  
  
COMMENT ON COLUMN autorepair_network.city.region IS 'The region in which the  
city is located';  
  
COMMENT ON COLUMN autorepair_network.city.area IS 'The area in which the city is  
located';
```

3.3.7. Создание таблицы "Клиент"

```
CREATE TABLE autorepair_network.client (  
    passport_id bigint NOT NULL,  
    client_name character varying(35) NOT NULL,  
    client_surname character varying(50) NOT NULL,  
    client_patronymic character varying(40),  
    client_email character varying(100) NOT NULL,  
    phone_number character varying(20) NOT NULL  
);  
  
ALTER TABLE autorepair_network.client OWNER TO postgres;  
  
COMMENT ON TABLE autorepair_network.client IS 'Information about clients';  
  
COMMENT ON COLUMN autorepair_network.client.passport_id IS 'Series and passport  
number ';  
  
COMMENT ON COLUMN autorepair_network.client.client_name IS 'Client name ';  
  
COMMENT ON COLUMN autorepair_network.client.client_surname IS 'Client surname';  
  
COMMENT ON COLUMN autorepair_network.client.client_patronymic IS 'Middle name  
of the client ';  
  
COMMENT ON COLUMN autorepair_network.client.client_email IS 'Client email ';  
  
COMMENT ON COLUMN autorepair_network.client.phone_number IS 'Client phone  
number';
```

3.3.8. Создание таблицы "Деталь"

```
CREATE TABLE autorepair_network.detail (  
    detail_id integer NOT NULL,  
    name text NOT NULL,  
    manufacturer_country text NOT NULL,  
    cost numeric NOT NULL,  
    car_model character varying(50) NOT NULL,  
    car_brand character varying(25) NOT NULL  
);  
  
ALTER TABLE autorepair_network.detail OWNER TO postgres;  
  
COMMENT ON TABLE autorepair_network.detail IS 'Information about details';  
  
COMMENT ON COLUMN autorepair_network.detail.detail_id IS 'Id of detail';  
  
COMMENT ON COLUMN autorepair_network.detail.name IS 'Detail name';
```

```
COMMENT ON COLUMN autorepair_network.detail.manufacturer_country IS '
    Manufacturer's country name';
```

```
COMMENT ON COLUMN autorepair_network.detail.cost IS 'Detail price';
```

```
COMMENT ON COLUMN autorepair_network.detail.car_model IS 'The model of car';
```

```
COMMENT ON COLUMN autorepair_network.detail.car_brand IS 'The brand of car';
```

3.3.9. Создание таблицы "Сотрудник"

```
CREATE TABLE autorepair_network.employee (
    employee_id integer NOT NULL,
    function text NOT NULL,
    specialisation text NOT NULL,
    rank text NOT NULL,
    workbranch_id integer NOT NULL,
    employee_name character varying(35) NOT NULL,
    employee_surname character varying(50) NOT NULL,
    employee_patronymic character varying(40)
);
```

```
ALTER TABLE autorepair_network.employee OWNER TO postgres;
```

```
COMMENT ON TABLE autorepair_network.employee IS 'Information about employees';
```

```
COMMENT ON COLUMN autorepair_network.employee.employee_id IS 'Employee ID ';
```

```
COMMENT ON COLUMN autorepair_network.employee.function IS 'Working position ';
```

```
COMMENT ON COLUMN autorepair_network.employee.specialisation IS 'Working
specialisation';
```

```
COMMENT ON COLUMN autorepair_network.employee.rank IS 'Working rank';
```

```
COMMENT ON COLUMN autorepair_network.employee.workbranch_id IS 'Workbranch
ID ';
```

```
COMMENT ON COLUMN autorepair_network.employee.employee_name IS 'Employee
name';
```

```
COMMENT ON COLUMN autorepair_network.employee.employee_surname IS 'Employee
surname';
```

```
COMMENT ON COLUMN autorepair_network.employee.employee_patronymic IS '
    Middle name of employee';
```

3.3.10. Создание таблицы "Состав ремонта"

```
CREATE TABLE autorepair_network.repair_composition (  
    repair_id integer NOT NULL,  
    repair_acceptance_date date NOT NULL,  
    completion_planned_date date NOT NULL,  
    completion_actual_date date,  
    repair_types text NOT NULL,  
    service_number integer NOT NULL,  
    employee_id integer NOT NULL,  
    service_id integer NOT NULL,  
    contract_number integer NOT NULL,  
    CONSTRAINT date2 CHECK ((completion_planned_date >=  
        repair_acceptance_date)),  
    CONSTRAINT date3 CHECK ((completion_actual_date >= repair_acceptance_date  
        ))  
);  
  
ALTER TABLE autorepair_network.repair_composition OWNER TO postgres;  
  
COMMENT ON TABLE autorepair_network.repair_composition IS 'Information about  
    repair compositions';  
  
COMMENT ON COLUMN autorepair_network.repair_composition.repair_id IS 'Id of  
    repair';  
  
COMMENT ON COLUMN autorepair_network.repair_composition.  
    repair_acceptance_date IS 'Date of acceptance for repair';  
  
COMMENT ON COLUMN autorepair_network.repair_composition.  
    completion_planned_date IS 'Planned completion date for repair';  
  
COMMENT ON COLUMN autorepair_network.repair_composition.  
    completion_actual_date IS 'Actual completion date for repair';  
  
COMMENT ON COLUMN autorepair_network.repair_composition.repair_types IS '  
    Types of repair';  
  
COMMENT ON COLUMN autorepair_network.repair_composition.service_number IS '  
    Service number';  
  
COMMENT ON COLUMN autorepair_network.repair_composition.employee_id IS 'Id of  
    employee';  
  
COMMENT ON COLUMN autorepair_network.repair_composition.service_id IS 'Id of  
    service';  
  
COMMENT ON COLUMN autorepair_network.repair_composition.contract_number IS '  
    Number of contract';
```

3.3.11. Создание таблицы "Договор на ремонт"

```
CREATE TABLE autorepair_network.repair_contract (  
    contract_number integer NOT NULL,  
    signing_date date NOT NULL,  
    status text NOT NULL,  
    services_list_id integer NOT NULL,  
    passport_id bigint NOT NULL,  
    employee_id integer NOT NULL,  
    car_id integer NOT NULL,  
    branch_id integer NOT NULL  
);  
  
ALTER TABLE autorepair_network.repair_contract OWNER TO postgres;  
  
COMMENT ON TABLE autorepair_network.repair_contract IS 'Information about repair  
contracts';  
  
COMMENT ON COLUMN autorepair_network.repair_contract.contract_number IS '  
Contract number';  
  
COMMENT ON COLUMN autorepair_network.repair_contract.signing_date IS 'Date of  
sign';  
  
COMMENT ON COLUMN autorepair_network.repair_contract.status IS 'Status of repair  
';  
  
COMMENT ON COLUMN autorepair_network.repair_contract.services_list_id IS 'Id of  
services list';  
  
COMMENT ON COLUMN autorepair_network.repair_contract.passport_id IS 'Id of  
passport';  
  
COMMENT ON COLUMN autorepair_network.repair_contract.employee_id IS 'Id of  
employee';  
  
COMMENT ON COLUMN autorepair_network.repair_contract.car_id IS 'Id of car';  
  
COMMENT ON COLUMN autorepair_network.repair_contract.branch_id IS 'Id of  
branch';
```

3.3.12. Создание таблицы "Список деталей для ремонта"

```
CREATE TABLE autorepair_network.repair_parts_list (  
    detail_list_id integer NOT NULL,  
    detail_number integer,  
    detail_id integer,  
    repair_id integer NOT NULL  
);
```

```

ALTER TABLE autorepair_network.repair_parts_list OWNER TO postgres;

COMMENT ON TABLE autorepair_network.repair_parts_list IS 'Information about
repair parts lists';

COMMENT ON COLUMN autorepair_network.repair_parts_list.detail_list_id IS 'Id of
detail list';

COMMENT ON COLUMN autorepair_network.repair_parts_list.detail_number IS '
Number of each detail';

COMMENT ON COLUMN autorepair_network.repair_parts_list.detail_id IS 'Id of detail
';

COMMENT ON COLUMN autorepair_network.repair_parts_list.repair_id IS 'Id of
repair';

```

3.3.13. Создание таблицы "Вид ремонта"

```

CREATE TABLE autorepair_network.repair_type (
    service_id integer NOT NULL,
    service_name text NOT NULL,
    one_piece_service_cost numeric NOT NULL
);

ALTER TABLE autorepair_network.repair_type OWNER TO postgres;

COMMENT ON TABLE autorepair_network.repair_type IS 'Information about type of
repair';

COMMENT ON COLUMN autorepair_network.repair_type.service_id IS 'Id of service';

COMMENT ON COLUMN autorepair_network.repair_type.service_name IS 'Service
name';

COMMENT ON COLUMN autorepair_network.repair_type.one_piece_service_cost IS '
Price of each service (one piece, if it is about detail)';

```

3.3.14. Внесение данных в таблицу "Филиал"

```

INSERT INTO autorepair_network.branch (branch_id, branch_address, city_id)
VALUES (1, 'Улица' Типанова, дом24', 1);
INSERT INTO autorepair_network.branch (branch_id, branch_address, city_id)
VALUES (2, 'Бережковская' набережная, дом36', 2);
INSERT INTO autorepair_network.branch (branch_id, branch_address, city_id)
VALUES (3, 'Улица' Хохрякова, дом10', 3);
INSERT INTO autorepair_network.branch (branch_id, branch_address, city_id)
VALUES (4, 'Улица' Капитанская, дом27', 4);
INSERT INTO autorepair_network.branch (branch_id, branch_address, city_id)
VALUES (5, 'Улица' Бондарева, дом4', 5);

```

3.3.15. Внесение данных в таблицу "Автомобиль"

```
INSERT INTO autorepair_network.car (car_id, color, license_plate, year_of_issue,
brand_code) VALUES (2, Белый'', ПСЙ'130|178', 2016, 2);
INSERT INTO autorepair_network.car (car_id, color, license_plate, year_of_issue,
brand_code) VALUES (4, Белый'', РХТ'444|178', 2019, 4);
INSERT INTO autorepair_network.car (car_id, color, license_plate, year_of_issue,
brand_code) VALUES (5, Синий'', СБР'056|78', 2020, 5);
INSERT INTO autorepair_network.car (car_id, color, license_plate, year_of_issue,
brand_code) VALUES (1, Черный'', НТТ'468|78', 2017, 1);
INSERT INTO autorepair_network.car (car_id, color, license_plate, year_of_issue,
brand_code) VALUES (3, Желтый'', ФРА'246|799', 2021, 3);
```

3.3.16. Внесение данных в таблицу "Модель"

```
INSERT INTO autorepair_network.car_model (brand_code, model, brand, power)
VALUES (1, 'X7 M50d', 'BMW', 249);
INSERT INTO autorepair_network.car_model (brand_code, model, brand, power)
VALUES (2, 'X5 xDrive30d', 'BMW', 218);
INSERT INTO autorepair_network.car_model (brand_code, model, brand, power)
VALUES (3, '820 Competizione A', 'Ferrari', 830);
INSERT INTO autorepair_network.car_model (brand_code, model, brand, power)
VALUES (4, 'CR-V', 'Honda', 150);
INSERT INTO autorepair_network.car_model (brand_code, model, brand, power)
VALUES (5, 'Impreza Sport', 'Subaru', 250);
```

3.3.17. Внесение данных в таблицу "Город"

```
INSERT INTO autorepair_network.city (city_id, city_name, region, area) VALUES (1,
СанктПетербург'—', СанктПетербург'—', Ленинградская' область');
INSERT INTO autorepair_network.city (city_id, city_name, region, area) VALUES (2,
Москва'', Москва'', Московская' область');
INSERT INTO autorepair_network.city (city_id, city_name, region, area) VALUES (3,
Екатеринбург'', Свердловская' область', Свердловская' область');
INSERT INTO autorepair_network.city (city_id, city_name, region, area) VALUES (4,
Севастополь'', Севастополь'', Таврическая' область');
INSERT INTO autorepair_network.city (city_id, city_name, region, area) VALUES (5,
Сортавала'', Республика' Карелия', '');
```

3.3.18. Внесение данных в таблицу "Клиент"

```
INSERT INTO autorepair_network.client (passport_id, client_name, client_surname,
client_patronymic, client_email, phone_number) VALUES (4014755996, Андрей'',
Коробковский'', Геннадьевич'', 'korobkovskiy@gmail.com', '+79211234567');
INSERT INTO autorepair_network.client (passport_id, client_name, client_surname,
client_patronymic, client_email, phone_number) VALUES (4016655112, Вадим'',
Коробковский'', Андреевич'', 'vkorobkovskiy@gmail.com', '+79312039247');
```



```

INSERT INTO autorepair_network.client (passport_id, client_name, client_surname,
    client_patronymic, client_email, phone_number) VALUES (4445560528, Вячеслав'',
    Вершинин'', Давидович'', 'vvershinin@mail.ru', '+79219600732');
INSERT INTO autorepair_network.client (passport_id, client_name, client_surname,
    client_patronymic, client_email, phone_number) VALUES (4444246842, Алексей'',
    Поздняков'', Иванович'', 'alexpoz51@gmail.com', '+79219002367');
INSERT INTO autorepair_network.client (passport_id, client_name, client_surname,
    client_patronymic, client_email, phone_number) VALUES (4015678234, Елена'',
    Большакова'', Ивановна'', 'elenabolshakova@mail.ru', '+79215699955');

```

3.3.19. Внесение данных в таблицу "Деталь"

```

INSERT INTO autorepair_network.detail (detail_id, name, manufacturer_country, cost,
    car_model, car_brand) VALUES (1, Аккумулятор'', Япония'', 35000, 'Impreza Sport
    ', 'Subaru');
INSERT INTO autorepair_network.detail (detail_id, name, manufacturer_country, cost,
    car_model, car_brand) VALUES (2, Двигатель'', Великобритания'', 75000, 'Senna', '
    McLaren');
INSERT INTO autorepair_network.detail (detail_id, name, manufacturer_country, cost,
    car_model, car_brand) VALUES (3, Тормозная' колодка', Великобритания'', 50000,
    'Valkyrie', 'Aston Martin');
INSERT INTO autorepair_network.detail (detail_id, name, manufacturer_country, cost,
    car_model, car_brand) VALUES (4, Фара'', Швеция'', 15000, 'V90 Cross Country', '
    Volvo');
INSERT INTO autorepair_network.detail (detail_id, name, manufacturer_country, cost,
    car_model, car_brand) VALUES (5, Зеркало' заднеговида', Великобритания'',
    18000, 'I—Pace', 'Jaguar');

```

3.3.20. Внесение данных в таблицу "Сотрудник"

```

INSERT INTO autorepair_network.employee (employee_id, function, specialisation, rank,
    workbranch_id, employee_name, employee_surname, employee_patronymic) VALUES
    (1, Инженер'', Составление' планаработ', '3', 2, Филипп'', Киркоров'',
    Бедросович'');
INSERT INTO autorepair_network.employee (employee_id, function, specialisation, rank,
    workbranch_id, employee_name, employee_surname, employee_patronymic) VALUES
    (2, Мойщик'', Мойщик' салона', '7', 1, Николай'', Басков'', Викторovich'');
INSERT INTO autorepair_network.employee (employee_id, function, specialisation, rank,
    workbranch_id, employee_name, employee_surname, employee_patronymic) VALUES
    (3, Парковщик'', Парковщик' автомобилейсреднейценовойкатегории', '6', 3,
    Евгений'', Ковалев'', Сергеевич'');
INSERT INTO autorepair_network.employee (employee_id, function, specialisation, rank,
    workbranch_id, employee_name, employee_surname, employee_patronymic) VALUES
    (4, Автоэлектрик'', Настройка' иремонтэлектроники', '5', 4, Даниил'', Милохин'',
    Вячеславovich'');
INSERT INTO autorepair_network.employee (employee_id, function, specialisation, rank,
    workbranch_id, employee_name, employee_surname, employee_patronymic) VALUES
    (5, Менеджер'', Закупка' запчастей', '7', 5, Марина'', Говорова'', Михайловна'');

```

3.3.21. Внесение данных в таблицу "Состав ремонта"

```
INSERT INTO autorepair_network.repair_composition (repair_id,  
    repair_acceptance_date, completion_planned_date, completion_actual_date,  
    repair_types, service_number, employee_id, service_id, contract_number) VALUES  
    (2, '2020-04-01', '2020-04-15', '2020-04-16', 'Текущий', 1, 4, 2, 2);  
INSERT INTO autorepair_network.repair_composition (repair_id,  
    repair_acceptance_date, completion_planned_date, completion_actual_date,  
    repair_types, service_number, employee_id, service_id, contract_number) VALUES  
    (3, '2022-02-25', '2022-03-11', NULL, 'Средний', 1, 5, 4, 3);  
INSERT INTO autorepair_network.repair_composition (repair_id,  
    repair_acceptance_date, completion_planned_date, completion_actual_date,  
    repair_types, service_number, employee_id, service_id, contract_number) VALUES  
    (4, '2022-02-25', '2022-02-26', NULL, 'Мойка' салона', 1, 2, 1, 4);  
INSERT INTO autorepair_network.repair_composition (repair_id,  
    repair_acceptance_date, completion_planned_date, completion_actual_date,  
    repair_types, service_number, employee_id, service_id, contract_number) VALUES  
    (5, '2021-05-05', '2021-05-05', '2021-05-05', 'Парковка', 1, 3, 6, 5);  
INSERT INTO autorepair_network.repair_composition (repair_id,  
    repair_acceptance_date, completion_planned_date, completion_actual_date,  
    repair_types, service_number, employee_id, service_id, contract_number) VALUES  
    (1, '2021-02-25', '2021-03-04', '2021-03-04', 'Составление' плана работ', 1, 1, 5, 1);
```

3.3.22. Внесение данных в таблицу "Договор на ремонт"

```
INSERT INTO autorepair_network.repair_contract (contract_number, signing_date,  
    status, services_list_id, passport_id, employee_id, car_id, branch_id) VALUES (1,  
    '2021-02-25', 'Выполнен', 1, 4014755996, 1, 1, 2);  
INSERT INTO autorepair_network.repair_contract (contract_number, signing_date,  
    status, services_list_id, passport_id, employee_id, car_id, branch_id) VALUES (3,  
    '2022-02-25', 'Выполняется', 3, 4016655112, 5, 2, 5);  
INSERT INTO autorepair_network.repair_contract (contract_number, signing_date,  
    status, services_list_id, passport_id, employee_id, car_id, branch_id) VALUES (4,  
    '2022-02-25', 'Выполняется', 4, 4015678234, 2, 3, 1);  
INSERT INTO autorepair_network.repair_contract (contract_number, signing_date,  
    status, services_list_id, passport_id, employee_id, car_id, branch_id) VALUES (2,  
    '2020-04-01', 'Отменен', 2, 4444246842, 4, 4, 4);  
INSERT INTO autorepair_network.repair_contract (contract_number, signing_date,  
    status, services_list_id, passport_id, employee_id, car_id, branch_id) VALUES (5,  
    '2021-05-05', 'Выполнен', 5, 4445560528, 3, 5, 3);
```

3.3.23. Внесение данных в таблицу "Список деталей для ремонта"

```
INSERT INTO autorepair_network.repair_parts_list (detail_list_id, detail_number,  
    detail_id, repair_id) VALUES (1, 0, NULL, 1);  
INSERT INTO autorepair_network.repair_parts_list (detail_list_id, detail_number,  
    detail_id, repair_id) VALUES (2, 0, NULL, 2);  
INSERT INTO autorepair_network.repair_parts_list (detail_list_id, detail_number,  
    detail_id, repair_id) VALUES (3, 0, NULL, 3);
```

```

INSERT INTO autorepair_network.repair_parts_list (detail_list_id, detail_number,
    detail_id, repair_id) VALUES (4, 0, NULL, 4);
INSERT INTO autorepair_network.repair_parts_list (detail_list_id, detail_number,
    detail_id, repair_id) VALUES (5, 0, NULL, 5);

```

3.3.24. Внесение данных в таблицу "Вид ремонта"

```

INSERT INTO autorepair_network.repair_type (service_id, service_name,
    one_piece_service_cost) VALUES (1, 'Мойка салона', 1500);
INSERT INTO autorepair_network.repair_type (service_id, service_name,
    one_piece_service_cost) VALUES (5, 'Составление плана ремонта', 1000);
INSERT INTO autorepair_network.repair_type (service_id, service_name,
    one_piece_service_cost) VALUES (4, 'Закупка запчастей', 2500);
INSERT INTO autorepair_network.repair_type (service_id, service_name,
    one_piece_service_cost) VALUES (2, 'Ремонт электроники', 3500);
INSERT INTO autorepair_network.repair_type (service_id, service_name,
    one_piece_service_cost) VALUES (3, 'Установка электроники', 5000);
INSERT INTO autorepair_network.repair_type (service_id, service_name,
    one_piece_service_cost) VALUES (6, 'Парковка', 750);

```

3.3.25. Ограничения для таблицы "Филиал"

```

ALTER TABLE ONLY autorepair_network.branch
    ADD CONSTRAINT branch_pkey PRIMARY KEY (branch_id);

ALTER TABLE ONLY autorepair_network.branch
    ADD CONSTRAINT city_id FOREIGN KEY (city_id) REFERENCES
        autorepair_network.city(city_id) NOT VALID;

```

3.3.26. Ограничения для таблицы "Автомобиль"

```

ALTER TABLE ONLY autorepair_network.car
    ADD CONSTRAINT car_pkey PRIMARY KEY (car_id);

ALTER TABLE ONLY autorepair_network.car
    ADD CONSTRAINT brand_code FOREIGN KEY (brand_code) REFERENCES
        autorepair_network.car_model(brand_code) NOT VALID;

```

3.3.27. Ограничения для таблицы "Модель"

```

ALTER TABLE ONLY autorepair_network.car_model
    ADD CONSTRAINT car_model_pkey PRIMARY KEY (brand_code);

ALTER TABLE ONLY autorepair_network.car_model
    ADD CONSTRAINT model_unique_values UNIQUE (brand_code);

```

3.3.28. Ограничения для таблицы "Город"

```
ALTER TABLE ONLY autorepair_network.city  
ADD CONSTRAINT city_pkey PRIMARY KEY (city_id);
```

3.3.29. Ограничения для таблицы "Клиент"

```
ALTER TABLE ONLY autorepair_network.client  
ADD CONSTRAINT client_pkey PRIMARY KEY (passport_id);  
  
ALTER TABLE ONLY autorepair_network.client  
ADD CONSTRAINT client_unique_values UNIQUE (passport_id, client_email,  
phone_number);
```

3.3.30. Ограничения для таблицы "Деталь"

```
ALTER TABLE ONLY autorepair_network.detail  
ADD CONSTRAINT detail_pkey PRIMARY KEY (detail_id);
```

3.3.31. Ограничения для таблицы "Сотрудник"

```
ALTER TABLE ONLY autorepair_network.employee  
ADD CONSTRAINT employee_pkey PRIMARY KEY (employee_id);  
  
ALTER TABLE ONLY autorepair_network.employee  
ADD CONSTRAINT employee_unique_values UNIQUE (employee_id);
```

3.3.32. Ограничения для таблицы "Состав ремонта"

```
ALTER TABLE ONLY autorepair_network.repair_composition  
ADD CONSTRAINT repair_composition_pkey PRIMARY KEY (repair_id);  
  
ALTER TABLE ONLY autorepair_network.repair_composition  
ADD CONSTRAINT contract_number FOREIGN KEY (contract_number)  
REFERENCES autorepair_network.repair_contract(contract_number) NOT  
VALID;  
  
ALTER TABLE ONLY autorepair_network.repair_composition  
ADD CONSTRAINT employee_id FOREIGN KEY (employee_id) REFERENCES  
autorepair_network.employee(employee_id) NOT VALID;  
  
ALTER TABLE ONLY autorepair_network.repair_composition  
ADD CONSTRAINT service_id FOREIGN KEY (service_id) REFERENCES  
autorepair_network.repair_type(service_id) NOT VALID;
```

3.3.33. Ограничения для таблицы "Договор на ремонт"

```
ALTER TABLE ONLY autorepair_network.repair_contract
    ADD CONSTRAINT repair_contract_pkey PRIMARY KEY (contract_number);
```

```
ALTER TABLE ONLY autorepair_network.repair_contract
    ADD CONSTRAINT branch_id FOREIGN KEY (branch_id) REFERENCES
        autorepair_network.branch(branch_id) NOT VALID;
```

```
ALTER TABLE ONLY autorepair_network.repair_contract
    ADD CONSTRAINT car_id FOREIGN KEY (car_id) REFERENCES
        autorepair_network.car(car_id) NOT VALID;
```

```
ALTER TABLE ONLY autorepair_network.repair_contract
    ADD CONSTRAINT employee_id FOREIGN KEY (employee_id) REFERENCES
        autorepair_network.employee(employee_id) NOT VALID;
```

```
ALTER TABLE ONLY autorepair_network.repair_contract
    ADD CONSTRAINT passport_id FOREIGN KEY (passport_id) REFERENCES
        autorepair_network.client(passport_id) NOT VALID;
```

3.3.34. Ограничения для таблицы "Список деталей для ремонта"

```
ALTER TABLE ONLY autorepair_network.repair_parts_list
    ADD CONSTRAINT repair_parts_list_pkey PRIMARY KEY (detail_list_id);
```

```
ALTER TABLE ONLY autorepair_network.repair_parts_list
    ADD CONSTRAINT detail_id FOREIGN KEY (detail_id) REFERENCES
        autorepair_network.detail(detail_id) NOT VALID;
```

```
ALTER TABLE ONLY autorepair_network.repair_parts_list
    ADD CONSTRAINT repair_id FOREIGN KEY (repair_id) REFERENCES
        autorepair_network.repair_composition(repair_id) NOT VALID;
```

3.3.35. Ограничения для таблицы "Вид ремонта"

```
ALTER TABLE ONLY autorepair_network.repair_type
    ADD CONSTRAINT repair_type_pkey PRIMARY KEY (service_id);
```

3.3.36. Окончание работы скрипта дампа

```
-- Completed on 2022-03-04 00:20:22
```

```
--
```

```
-- PostgreSQL database dump complete
```

```
--
```

4. Выводы

В процессе выполнения лабораторной работы я научился:

1. Работать с программным обеспечением pgAdmin 4.
2. Создавать базу данных и её схему.
3. Ставить ограничения на значения каждого столбца.
4. Выбирать правильный тип данных для каждого столбца.
5. Создавать резервную копию базы данных и восстанавливать её.