

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3**

**«Создание таблиц базы данных PostgreSQL. Заполнение таблиц  
рабочими данными»**

**по дисциплине «Проектирование и реализация баз данных»**

**Обучающийся:** Богданов Максим Александрович

**Факультет** прикладной информатики

**Группа** K3240

**Направление подготовки** 09.03.03 Прикладная информатика

**Образовательная программа** Мобильные и сетевые технологии 2023

**Преподаватель** Говорова Марина Михайловна

Санкт-Петербург  
2025

## **Цель работы**

Овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

## **Программное обеспечение**

СУБД PostgreSQL 1X, pgAdmin 4.

## **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;
  - 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.

## Модель для создания базы данных

Модель описывает систему управления такси или автопарком с возможностью учета сотрудников, машин, заказов и клиентов. Ознакомиться с моделью можно на рисунке 1.

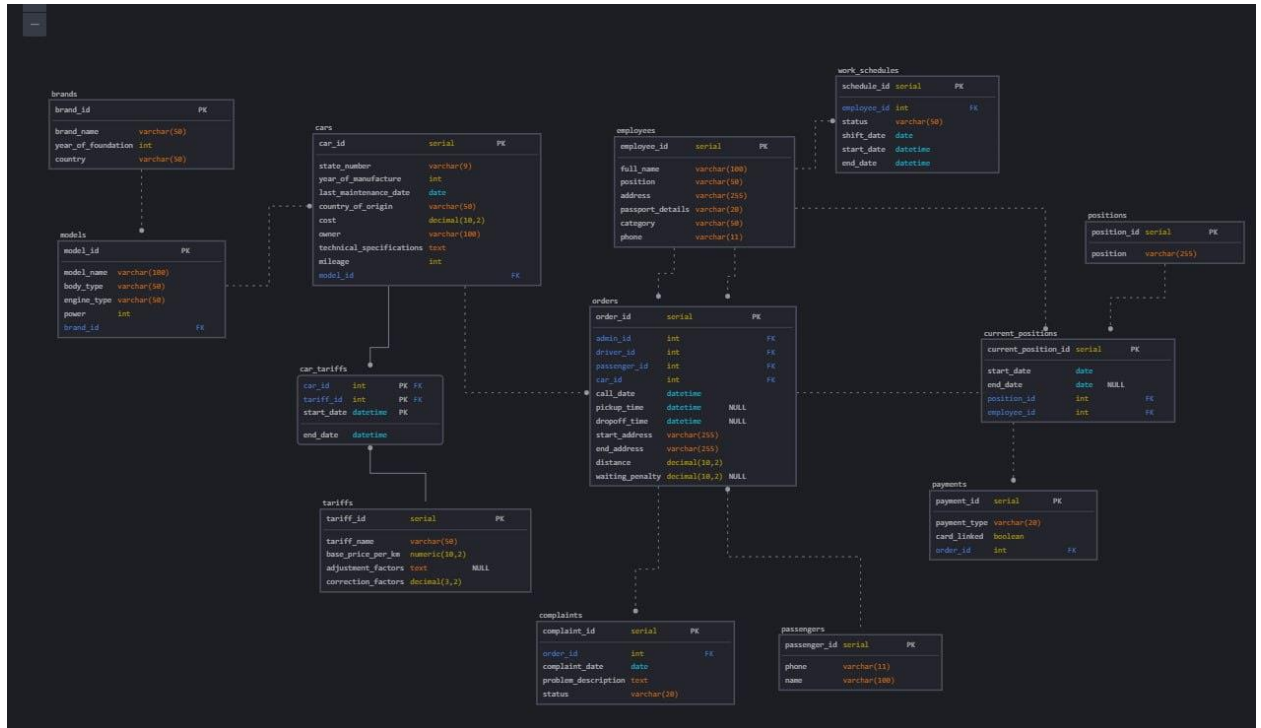


Рисунок 1 – Модель для создания базы данных

## Ход работы

### 1. Создание таблиц с помощью SQL-запросов

```
-- Таблица автомобильных брендов
CREATE TABLE brands (
  brand_id SERIAL PRIMARY KEY, -- Уникальный ID бренда
  brand_name VARCHAR(50) NOT NULL, -- Название бренда
  year_of_foundation INT NOT NULL CHECK (year_of_foundation > 1700), --
  Год основания
  country VARCHAR(50) NOT NULL -- Страна происхождения
);

-- Таблица моделей автомобилей
CREATE TABLE models (
  model_id SERIAL PRIMARY KEY, -- Уникальный ID модели
  model_name VARCHAR(100) NOT NULL UNIQUE, -- Название модели
  body_type VARCHAR(50) NOT NULL, -- Тип кузова
  engine_type VARCHAR(50) NOT NULL, -- Тип двигателя
  power INT NOT NULL CHECK (power > 0), -- Мощность двигателя
  brand_id INT REFERENCES brands (brand_id) ON DELETE CASCADE -- Связь с
  брендом
);
```

```

-- Таблица автомобилей
CREATE TABLE cars (
    car_id SERIAL PRIMARY KEY, -- Уникальный ID автомобиля
    state_number VARCHAR(9) NOT NULL, -- Госномер
    year_of_manufacture INT NOT NULL, -- Год выпуска
    last_maintenance_date DATE NOT NULL, -- Последнее ТО
    country_of_origin VARCHAR(50) NOT NULL, -- Страна производства
    cost DECIMAL(10,2) NOT NULL CHECK (cost >= 0), -- Стоимость авто
    owner VARCHAR(100), -- Владелец (опционально)
    technical_specifications TEXT NOT NULL, -- Тех. характеристики
    mileage INT NOT NULL CHECK (mileage >= 0), -- Пробег
    model_id INT REFERENCES models(model_id) ON DELETE SET NULL -- Модель
авто
);

-- Тарифы поездок
CREATE TABLE tariffs (
    tariff_id SERIAL PRIMARY KEY, -- Уникальный ID тарифа
    tariff_name VARCHAR(50) NOT NULL, -- Название тарифа
    base_price_per_km NUMERIC(10,2) NOT NULL CHECK (base_price_per_km >= 0),
-- Базовая цена за км
    adjustment_factors TEXT, -- Корректирующие факторы (например, ночное
время)
    correction_factors DECIMAL(10,2) -- Дополнительные корректировки
);

-- История применения тарифов к автомобилям
CREATE TABLE car_tariffs (
    car_id INT NOT NULL REFERENCES cars(car_id) ON DELETE CASCADE, -- ID
автомобиля
    tariff_id INT NOT NULL REFERENCES tariffs(tariff_id) ON DELETE CASCADE,
-- ID тарифа
    start_date TIMESTAMP NOT NULL, -- Дата начала действия тарифа
    end_date TIMESTAMP, -- Дата окончания действия
    PRIMARY KEY (car_id, tariff_id, start_date) -- Композитный ключ
);

-- Пассажиры
CREATE TABLE passengers (
    passenger_id SERIAL PRIMARY KEY, -- ID пассажира
    phone VARCHAR(11) NOT NULL, -- Телефон
    name VARCHAR(100) NOT NULL -- Имя
);

-- Сотрудники (водители, администраторы и т.д.)
CREATE TABLE employees (
    employee_id SERIAL PRIMARY KEY, -- ID сотрудника
    full_name VARCHAR(100) NOT NULL, -- Полное имя
    position VARCHAR(50) NOT NULL, -- Название должности
    address VARCHAR(255), -- Адрес
    passport_details VARCHAR(20), -- Паспортные данные
    category VARCHAR(10), -- Категория прав (например, В, С)
    phone VARCHAR(11) -- Телефон
);

-- Заказы поездок
CREATE TABLE orders (
    order_id SERIAL PRIMARY KEY, -- ID заказа
    admin_id INT, -- ID администратора
    driver_id INT, -- ID водителя (employee_id)
    passenger_id INT REFERENCES passengers(passenger_id) ON DELETE SET NULL,
-- Пассажир

```

```

car_id INT REFERENCES cars(car_id) ON DELETE SET NULL, -- Автомобиль
call_date TIMESTAMP, -- Время вызова
pickup_time TIMESTAMP, -- Время посадки
dropoff_time TIMESTAMP, -- Время окончания поездки
start_address VARCHAR(255), -- Адрес отправления
end_address VARCHAR(255), -- Адрес назначения
distance DECIMAL(10,2) CHECK (distance >= 0), -- Расстояние
waiting_penalty DECIMAL(10,2) -- Штраф за ожидание
);

-- Жалобы по заказам
CREATE TABLE complaints (
    complaint_id SERIAL PRIMARY KEY, -- ID жалобы
    order_id INT REFERENCES orders(order_id) ON DELETE CASCADE, -- К какому
заказу относится
    complaint_date DATE NOT NULL, -- Дата жалобы
    problem_description TEXT NOT NULL, -- Описание проблемы
    status VARCHAR(20) NOT NULL -- Статус жалобы (например: Open, Closed)
);

-- Платежи за поездки
CREATE TABLE payments (
    payment_id SERIAL PRIMARY KEY, -- ID платежа
    payment_type VARCHAR(20) NOT NULL, -- Тип оплаты (наличные, карта)
    card_linked BOOLEAN NOT NULL, -- Привязана ли карта
    order_id INT REFERENCES orders(order_id) ON DELETE CASCADE -- К какому
заказу относится
);

-- График работы сотрудников
CREATE TABLE work_schedules (
    schedule_id SERIAL PRIMARY KEY, -- ID смены
    employee_id INT REFERENCES employees(employee_id) ON DELETE CASCADE, --
Сотрудник
    status VARCHAR(50), -- Статус смены
    shift_date DATE, -- Дата смены
    start_date TIMESTAMP, -- Начало смены
    end_date TIMESTAMP -- Конец смены
);

-- Доступные должности в компании
CREATE TABLE positions (
    position_id SERIAL PRIMARY KEY, -- ID должности
    position VARCHAR(255) NOT NULL -- Название должности
);

-- Текущая позиция сотрудника
CREATE TABLE current_positions (
    current_position_id SERIAL PRIMARY KEY, -- ID записи
    start_date DATE NOT NULL, -- Дата назначения
    end_date DATE, -- Дата окончания (может быть NULL)
    position_id INT REFERENCES positions(position_id) ON DELETE CASCADE, --
Должность
    employee_id INT REFERENCES employees(employee_id) ON DELETE CASCADE --
Сотрудник
);

```

Листинг 1 – Создание таблиц с помощью SQL запросов

## 2. Вставка рабочих данных в таблицу

```
-- Вставка данных в brands
INSERT INTO brands (brand_name, year_of_foundation, country) VALUES
('Toyota', 1937, 'Japan'),
('BMW', 1916, 'Germany'),
('Ford', 1903, 'USA');

-- Вставка данных в models
INSERT INTO models (model_name, body_type, engine_type, power, brand_id)
VALUES
('Camry', 'Sedan', 'Petrol', 200, 1),
('X5', 'SUV', 'Diesel', 300, 2),
('Mustang', 'Coupe', 'Petrol', 450, 3);

-- Вставка данных в cars
INSERT INTO cars (state_number, year_of_manufacture, last_maintenance_date,
country_of_origin, cost, owner, technical_specifications, mileage, model_id)
VALUES
('A123BC77', 2020, '2024-01-15', 'Japan', 25000.00, 'Ivan Petrov', '2.5L
engine, automatic', 45000, 1),
('B456CD88', 2019, '2023-12-20', 'Germany', 40000.00, 'John Smith', '3.0L
diesel, AWD', 60000, 2),
('C789DE99', 2021, '2024-02-10', 'USA', 55000.00, 'Anna Ivanova', '5.0L V8',
30000, 3);

-- Вставка данных в tariffs
INSERT INTO tariffs (tariff_name, base_price_per_km, adjustment_factors,
correction_factors) VALUES
('Standard', 10.00, 'time_of_day:night=1.2', 0.95),
('Business', 15.00, 'distance>10=0.9', 1.1);

-- Вставка данных в car_tariffs
INSERT INTO car_tariffs (car_id, tariff_id, start_date, end_date) VALUES
(1, 1, '2024-01-01 00:00:00', NULL),
(2, 2, '2024-01-01 00:00:00', NULL);

-- Вставка данных в passengers
INSERT INTO passengers (phone, name) VALUES
('89161234567', 'Sergey Ivanov'),
('89261234567', 'Elena Petrova');

-- Вставка данных в employees
INSERT INTO employees (full_name, position, address, passport_details,
category, phone) VALUES
('Mikhail Sidorov', 'Driver', 'Lenina St, 15', '1234 567890', 'B',
'89051234567'),
('Olga Smirnova', 'Admin', 'Pushkina St, 20', '2345 678901', 'A',
'89161234568');

-- Вставка данных в orders
INSERT INTO orders (admin_id, driver_id, passenger_id, car_id, call_date,
pickup_time, dropoff_time, start_address, end_address, distance,
waiting_penalty) VALUES
(2, 1, 1, 1, '2025-03-01 08:00:00', '2025-03-01 08:10:00', '2025-03-01
08:30:00', 'Central Park', 'Airport', 15.5, 50.00),
(2, 1, 2, 2, '2025-03-02 10:00:00', '2025-03-02 10:15:00', '2025-03-02
10:45:00', 'Downtown', 'University', 20.0, 0.00);

-- Вставка данных в complaints
INSERT INTO complaints (order_id, complaint_date, problem_description,
status) VALUES
(1, '2025-03-01', 'Driver was late', 'Open'),
```

```

(2, '2025-03-02', 'Car was not clean', 'Resolved');

-- Вставка данных в payments
INSERT INTO payments (payment_type, card_linked, order_id) VALUES
('Card', TRUE, 1),
('Cash', FALSE, 2);

-- Вставка данных в work_schedules
INSERT INTO work_schedules (employee_id, status, shift_date, start_date,
end_date) VALUES
(1, 'Active', '2025-03-01', '2025-03-01 08:00:00', '2025-03-01 20:00:00'),
(2, 'Active', '2025-03-01', '2025-03-01 08:00:00', '2025-03-01 18:00:00');

-- Вставка данных в positions
INSERT INTO positions (position) VALUES
('Driver'),
('Admin');

-- Вставка данных в current_positions
INSERT INTO current_positions (start_date, end_date, position_id,
employee_id) VALUES
('2023-01-01', NULL, 1, 1),
('2023-01-01', NULL, 2, 2);

```

Листинг 2 – Вставка данных в таблицы

### 3. Получение таблиц с заполненными данными

Примеры заполненных таблиц БД представлены на рисунках 2–5. Данные таблиц выводились с помощью оператора SELECT.

	model_id [PK] integer	model_name character varying (100)	body_type character varying (50)	engine_type character varying (50)	power integer	brand_id integer
1	1	Camry	Sedan	Petrol	200	1
2	2	X5	SUV	Diesel	300	2
3	3	Mustang	Coupe	Petrol	450	3

Рисунок 2 – Таблица models

	car_id [PK] integer	state_number character varying (9)	year_of_manufacture integer	last_maintenance_date date	country_of_origin character varying (50)	cost numeric (10,2)	owner character varying (100)	technical_specifications text	mileage integer	model_id integer
1	1	A123BC77	2020	2024-01-15	Japan	25000.00	Ivan Petrov	2.5L engine, automatic	45000	1
2	2	B456CD88	2019	2023-12-20	Germany	40000.00	John Smith	3.0L diesel, AWD	60000	2
3	3	C789DE99	2021	2024-02-10	USA	55000.00	Anna Ivanova	5.0L V8	30000	3

Рисунок 3 – Таблица cars

	payment_id [PK] integer	payment_type character varying (20)	card_linked boolean	order_id integer
1	1	Card	true	1
2	2	Cash	false	2

Рисунок 4 – Таблица payments

	current_position_id [PK] integer	start_date date	end_date date	position_id integer	employee_id integer
1	1	2023-01-01	[null]	1	1
2	2	2023-01-01	[null]	2	2

Рисунок 5 – Таблица current\_positions

#### 4. Генерация схемы модели базы данных

Сгенерированная при помощи GUI pgAdmin 4 модель БД представлена на рисунке 6.

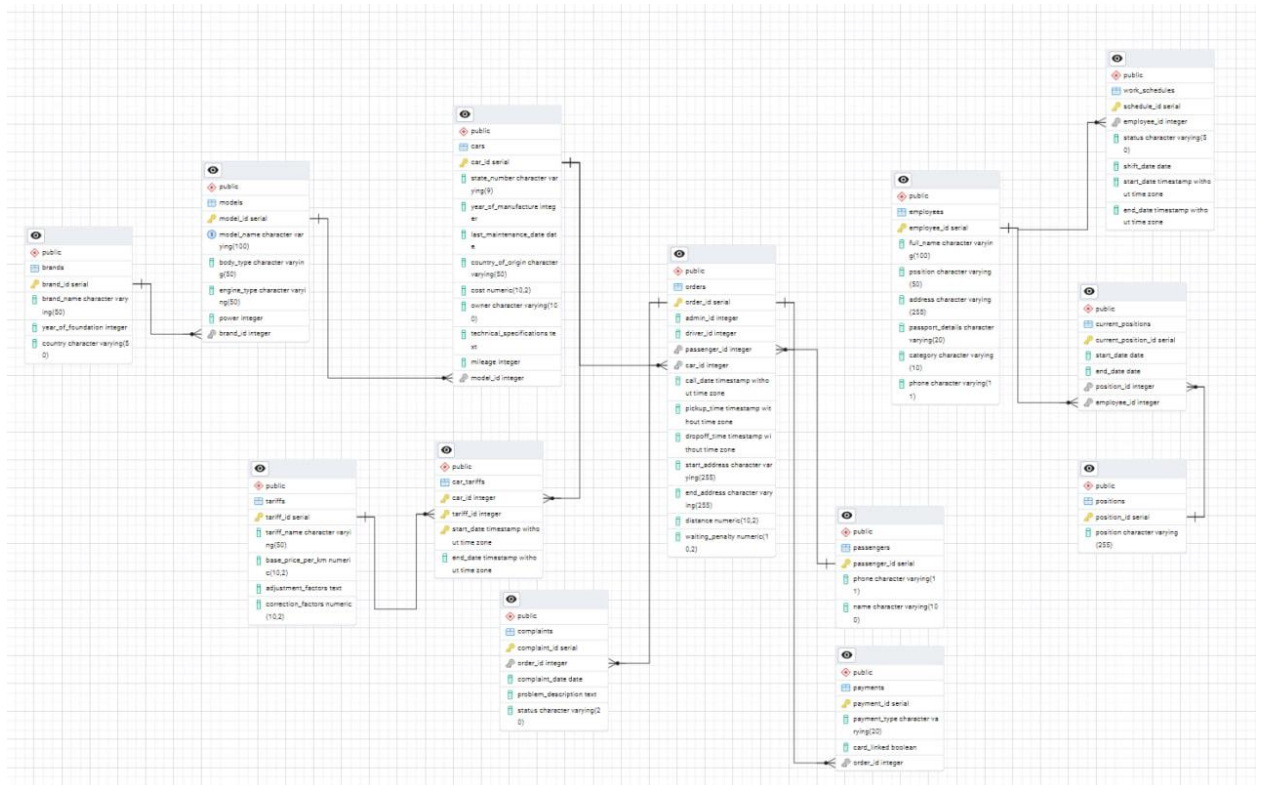


Рисунок 6 – Схема модели БД

#### 5. Создание резервной копии БД

Было создано две резервные копии БД согласно указаниям лабораторной работы. Бэкап файл представлен на рисунке 7.



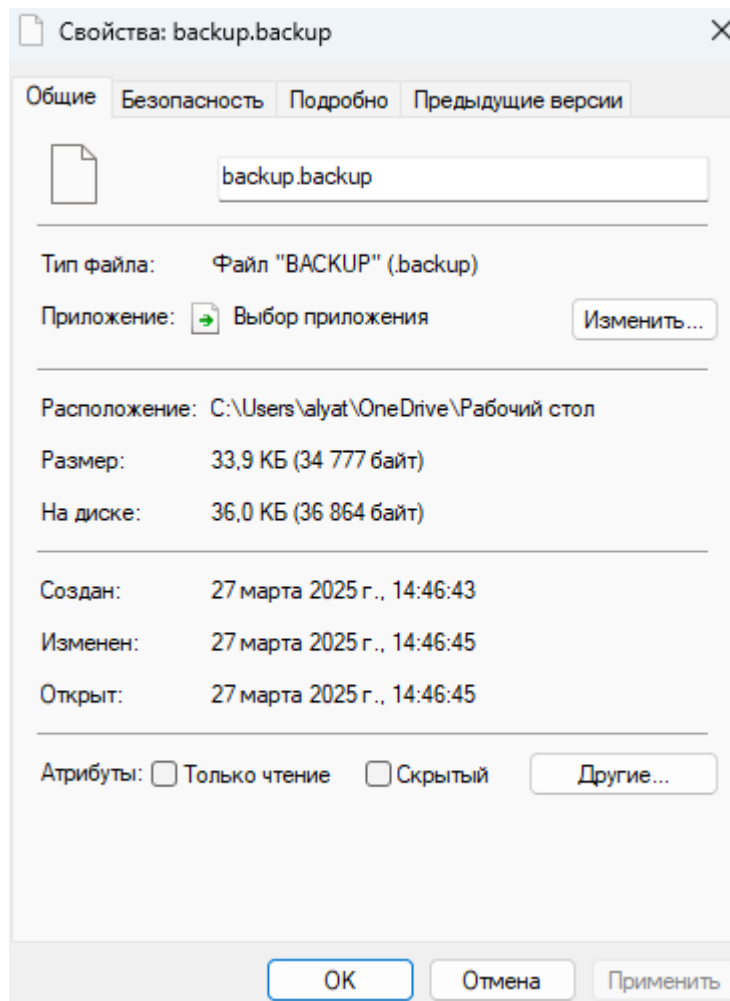


Рисунок 7 – Бэкап файл

## 6. Восстановление БД

С помощью диалога Restore была восстановлена база данных с файла .backup. Успешный рестор представлен на рисунке 8.

	PID	Тип	Сервер	Объект	Start Time	Состояние	Time Taken (sec)
<input type="checkbox"/>	17852	Восстановить	PostgreSQL 17 (localhost:5432)	Lab3_bogdanov	27.03.2025, 15:14:25	Finished	0.33

Рисунок 8 – Восстановление БД

**Вывод:** в ходе выполнения лабораторной работы были получены навыки создания таблиц баз данных PostgreSQL, заполнения их рабочими данными, создания бэкапов и восстановления БД.