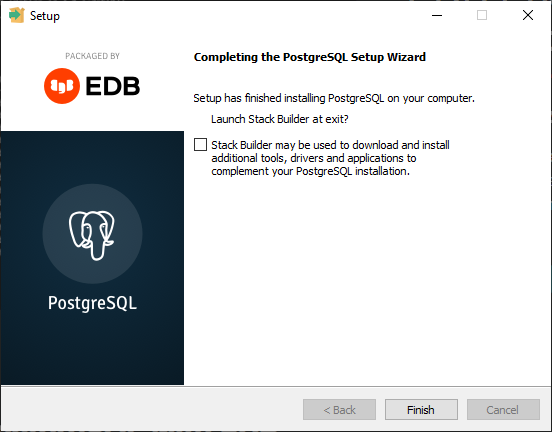
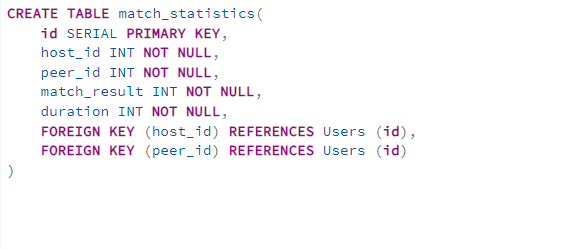
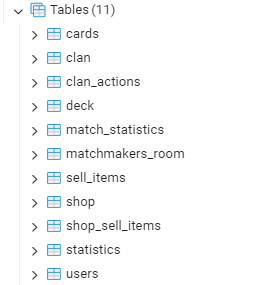
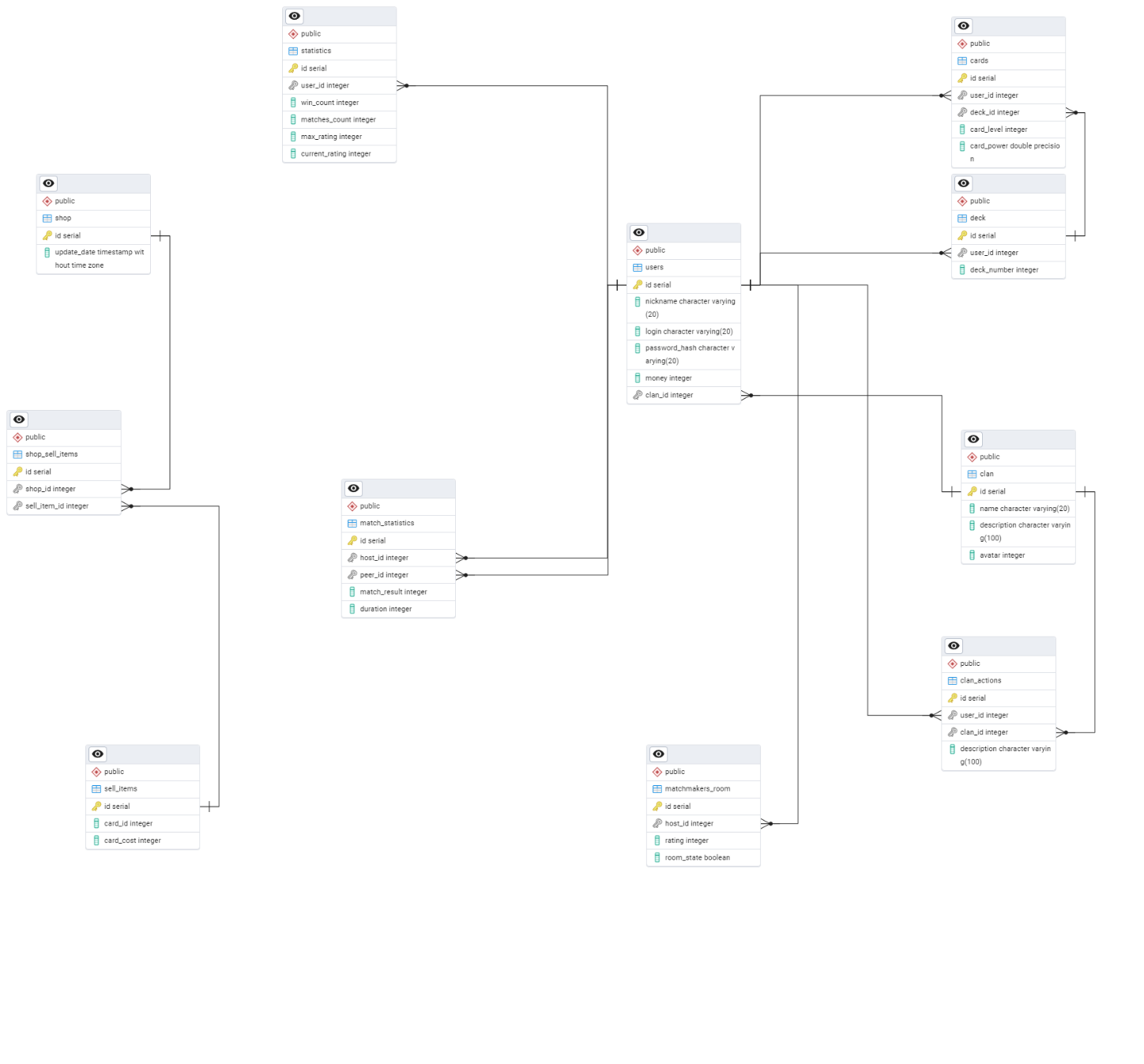
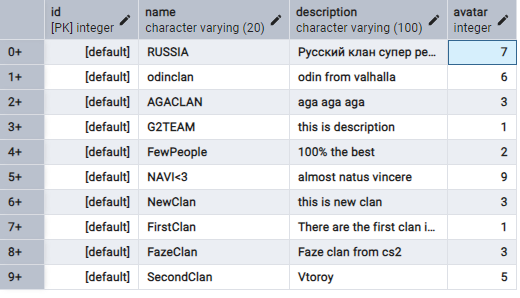
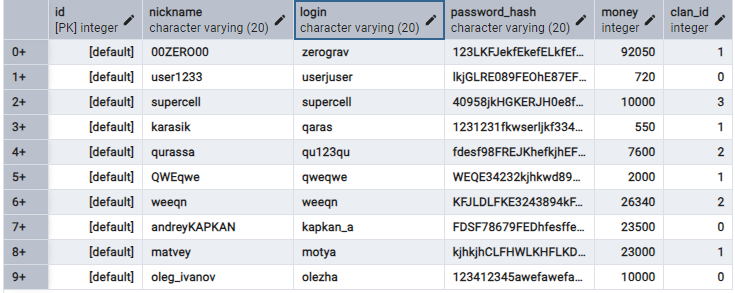
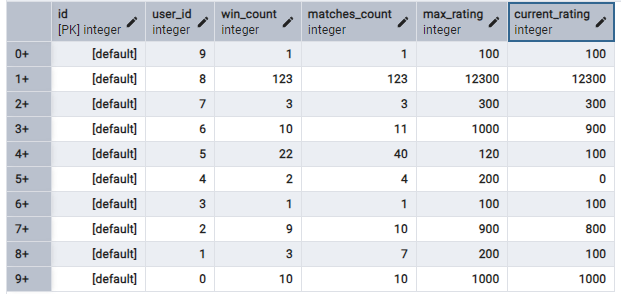
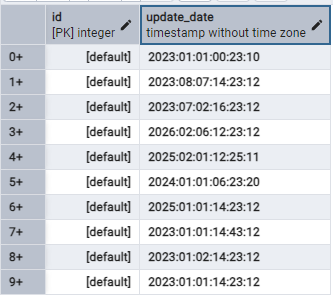
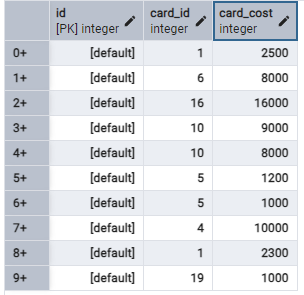
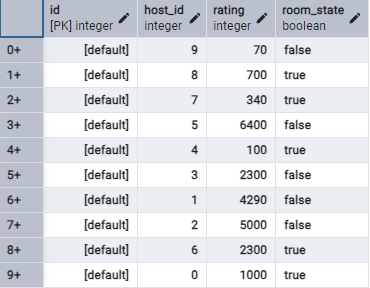
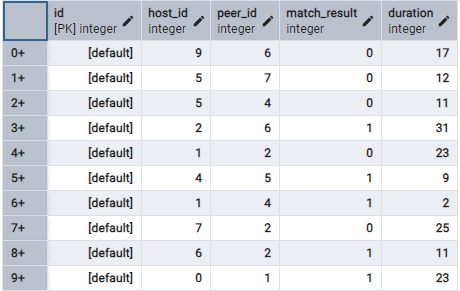
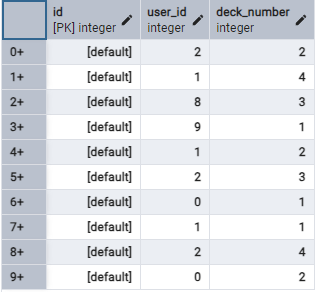
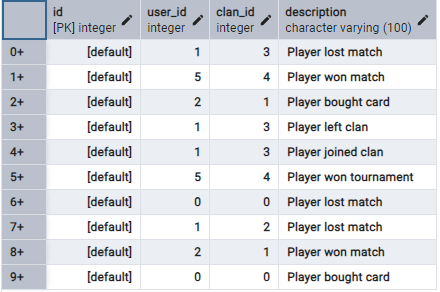
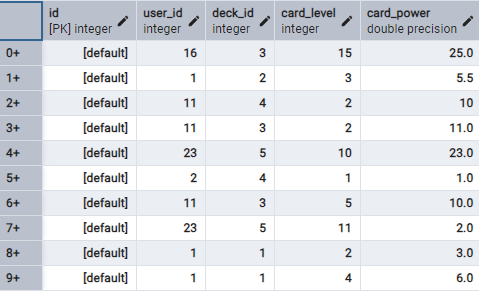
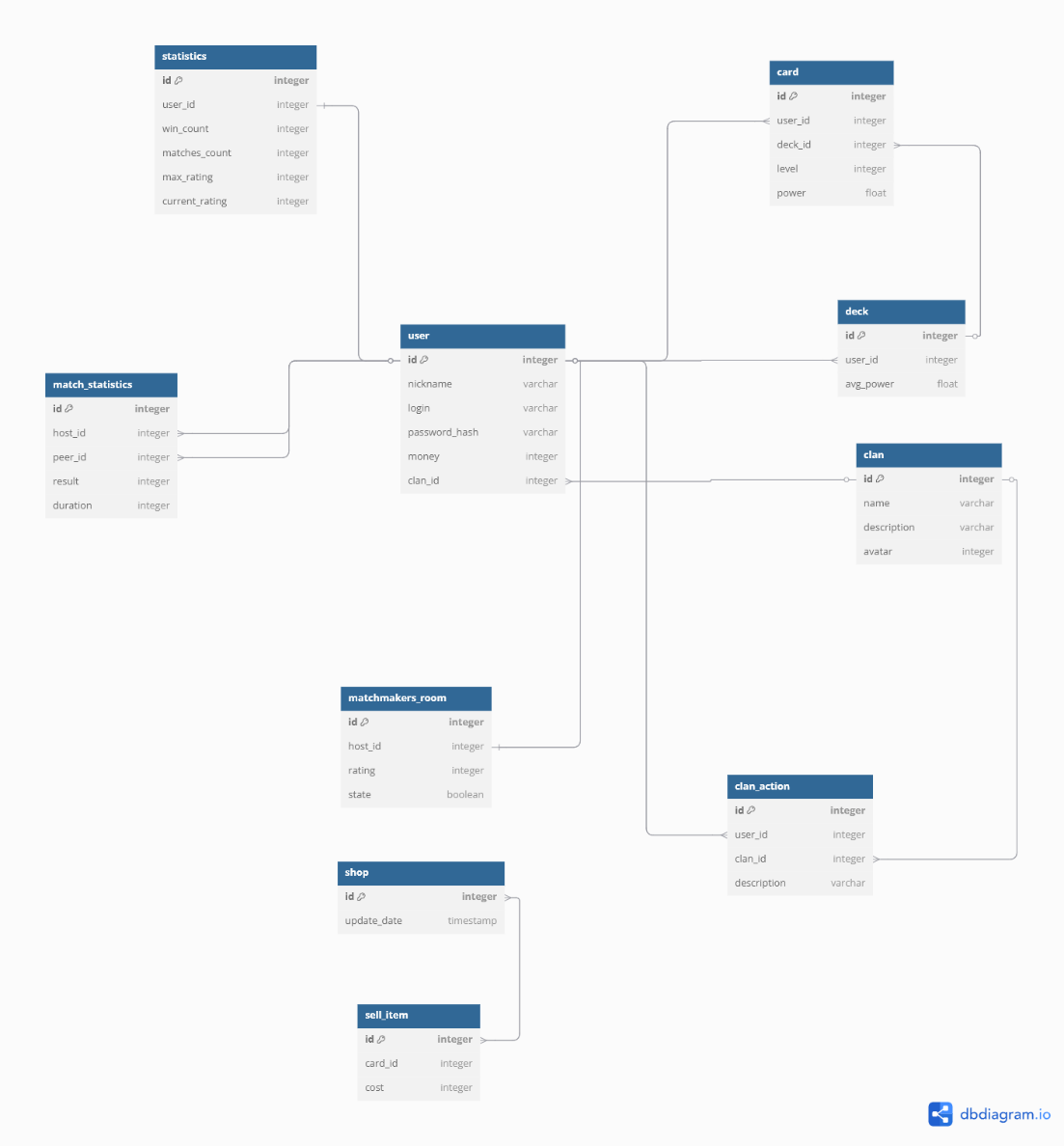
|  |  |  |
| --- | --- | --- |
| Лабораторная работа №3  Физическое проектирование базы данных | Ф.И.О. | Попов А.В. |
| Группа | ПрИн 166 |
| Преподаватель |  |
| Дата сдачи |  |

**Цель работы:** научиться физическому проектированию баз данных используя выбранную СУБД.

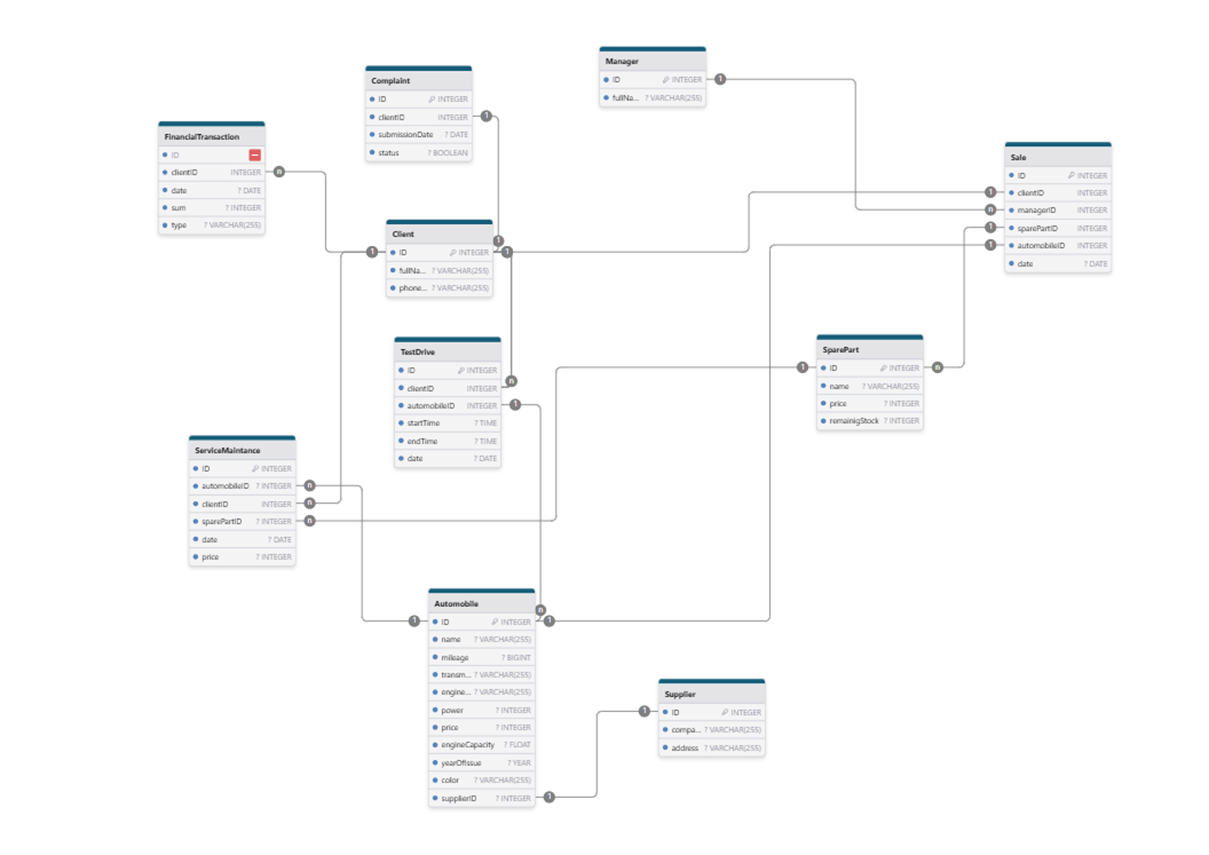
**Задачи работы:** выполнить физическое проектирование для собственной базы данных; выполнить физическое проектирования для базы данных одногруппника.

**Ход выполнения лабораторной работы:**

**Часть 1.**

1. Была выбрана СУБД PostgreSQL.
2. Была выбрана графическая оболочка для СУБД: pgAdmin.
3. Была установлена СУБД PostgreSQL на компьютер:  
   
4. Были созданы таблицы в БД используя скрипты.  
   Пример создания таблицы: 
5. Были созданы 11 таблиц (10 – сущностей, 1 для many-to-many отношения):  
   
6. Была создана реляционная модель средствами pgAdmin, она совпадает с той моделью, что была сделана в лаб. работе №2 (кроме таблицы many-to-many)  
     
   
7. Заполнил данные в таблицы:  
   Таблица clan:   
   Таблица users:  
   Таблица statistics:   
     
     
   Таблица shop:  
     
   Таблица sell\_items:  
     
   Таблица matchmakers\_room:  
     
     
   Таблица match\_statistics:  
     
   Таблица deck:  
     
   Таблица clan\_actions:  
     
     
   Таблица cards:  
   
8. Реляционная модель, созданная в лабораторной работе №2.  
   

**Часть 2.**

Реляционная модель одногруппника:  


1. Создание базы данных:  
CREATE DATABASE db3\_popov\_potapovich

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'Russian\_Russia.1251'

LC\_CTYPE = 'Russian\_Russia.1251'

LOCALE\_PROVIDER = 'libc'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1

IS\_TEMPLATE = False;  
  
2. Создание таблиц:

-- Таблица поставщиков

CREATE TABLE Supplier (

supplierID INT PRIMARY KEY,

companyName VARCHAR(255),

companyAddress VARCHAR(255)

);

-- Таблица клиентов

CREATE TABLE Client (

clientID INT PRIMARY KEY,

fullName VARCHAR(255),

phoneNumber VARCHAR(255)

);

-- Таблица менеджеров

CREATE TABLE Manager (

managerID INT PRIMARY KEY,

fullName VARCHAR(255)

);

-- Таблица запчастей

CREATE TABLE SparePart (

sparePartID INT PRIMARY KEY,

name VARCHAR(255),

price INT,

remainingStock INT

);

-- Таблица автомобилей

CREATE TABLE Automobile (

automobileID INT PRIMARY KEY,

supplierID INT,

name VARCHAR(255),

mileage INT,

transmissionType VARCHAR(255),

engineType VARCHAR(255),

powerAutomobile INT,

price INT,

engineCapacity INT,

yearOfIssue INT,

color VARCHAR(255),

FOREIGN KEY (supplierID) REFERENCES Supplier(supplierID)

);

-- Таблица финансовых транзакций

CREATE TABLE FinancialTransaction (

financialTransactionID INT PRIMARY KEY,

clientID INT,

dateFinancialTransaction DATE,

amount INT,

typeFinancialTransaction VARCHAR(255),

FOREIGN KEY (clientID) REFERENCES Client(clientID)

);

-- Таблица обслуживания автомобилей

CREATE TABLE ServiceMaintenance (

serviceMaintenanceID INT PRIMARY KEY,

automobileID INT,

clientID INT,

sparePartID INT,

dateServiceMaintenance DATE,

price INT,

FOREIGN KEY (automobileID) REFERENCES Automobile(automobileID),

FOREIGN KEY (clientID) REFERENCES Client(clientID),

FOREIGN KEY (sparePartID) REFERENCES SparePart(sparePartID)

);

-- Таблица продаж

CREATE TABLE Sale (

saleID INT PRIMARY KEY,

clientID INT,

managerID INT,

dateSale DATE,

FOREIGN KEY (clientID) REFERENCES Client(clientID),

FOREIGN KEY (managerID) REFERENCES Manager(managerID)

);

-- Таблица проданных автомобилей

CREATE TABLE SaleAutomobile (

saleAutomobileID INT PRIMARY KEY,

saleID INT,

automobileID INT,

FOREIGN KEY (saleID) REFERENCES Sale(saleID),

FOREIGN KEY (automobileID) REFERENCES Automobile(automobileID)

);

-- Таблица проданных запчастей

CREATE TABLE SaleSparePart (

saleSparePartID INT PRIMARY KEY,

saleID INT,

sparePartID INT,

quantity INT,

FOREIGN KEY (saleID) REFERENCES Sale(saleID),

FOREIGN KEY (sparePartID) REFERENCES SparePart(sparePartID)

);

-- Таблица жалоб

CREATE TABLE Complaint (

complaintID INT PRIMARY KEY,

clientID INT,

submissionDate DATE,

status BOOLEAN,

FOREIGN KEY (clientID) REFERENCES Client(clientID)

);

-- Таблица тест-драйвов

CREATE TABLE TestDrive (

testDriveID INT PRIMARY KEY,

clientID INT,

automobileID INT,

startTime TIME,

endTime TIME,

dateTestDrive DATE,

FOREIGN KEY (clientID) REFERENCES Client(clientID),

FOREIGN KEY (automobileID) REFERENCES Automobile(automobileID)

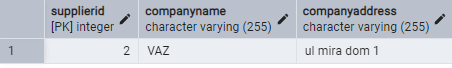
);  
  
3. Заполнение таблиц данными:

INSERT INTO Supplier (companyName, companyAddress) VALUES(

'VAZ',

'ul mira dom 1'

);

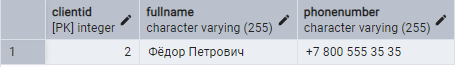


INSERT INTO Client (fullName, phoneNumber) VALUES(

'Фёдор Петрович',

'+7 800 555 35 35'

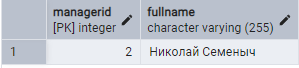
);



INSERT INTO Manager (fullName) VALUES(

'Николай Семеныч'

);



INSERT INTO SparePart (name, price, remainingStock) VALUES(

'Fuze',

23000,

12

);



INSERT INTO Automobile (supplierID, name, mileage, transmissionType, engineType, powerAutomobile, price, engineCapacity, yearOfIssue, color) VALUES(

2,

'VAZ 2112',

2002,

'MG',

'16V',

200,

500000,

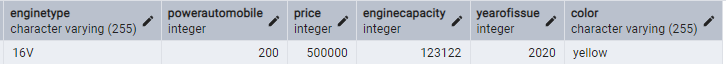
123122,

2020,

'yellow'

);





INSERT INTO FinancialTransaction (clientID, dateFinancialTransaction, amount, typeFinancialTransaction) VALUES(

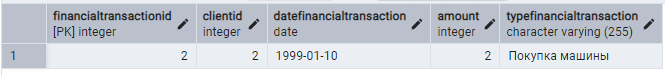
2,

'1999-01-10',

2,

'Покупка машины'

);



INSERT INTO ServiceMaintenance (automobileID, clientID, sparePartID, dateServiceMaintenance, price) VALUES(

3,

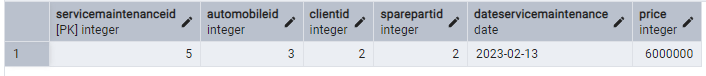
2,

2,

'2023-02-13',

6000000

);



INSERT INTO Sale (clientID, managerID, dateSale) VALUES(

0,

0,

'2025-02-20'

);



INSERT INTO SaleSparePart (saleID, sparePartID, quantity) VALUES(

1,

2,

12

);



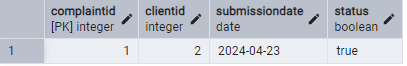
INSERT INTO Complaint (clientID, submissionDate, status) VALUES(

2,

'2024-04-23',

true

);



INSERT INTO TestDrive (clientID, automobileID, startTime, endTime, dateTestDrive) VALUES(

2,

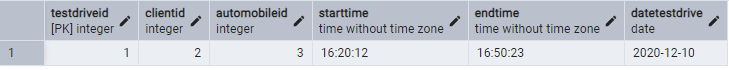
3,

'16:20:12',

'16:50:23',

'2020-12-10'

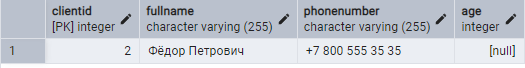
);



4. Изменение таблиц:

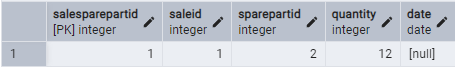
ALTER TABLE Client

ADD age INT;



ALTER TABLE SaleSparePart

ADD COLUMN date DATE;



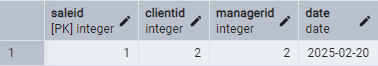
ALTER TABLE SaleSparePart

RENAME COLUMN saleSparePartID TO id;



ALTER TABLE Sale

RENAME COLUMN datesale TO date;



ALTER TABLE automobile

ADD COLUMN ordered\_by INT;

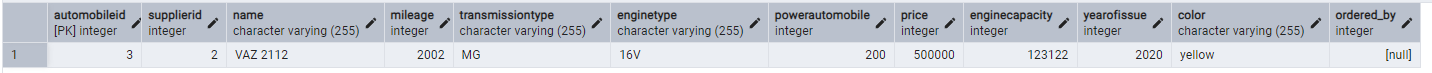
ALTER TABLE automobile

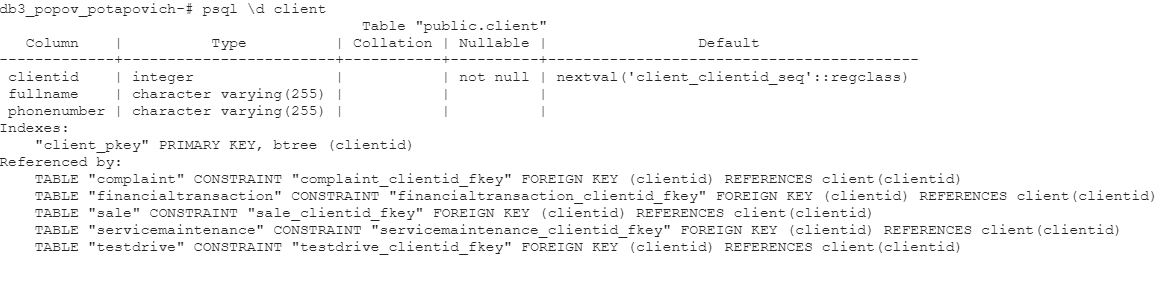
ADD CONSTRAINT ordered\_by FOREIGN KEY (ordered\_by) REFERENCES Client(clientID)



ALTER TABLE automobile

DROP CONSTRAINT ordered\_by



Для удаления PRIMARY KEY из таблицы, для начала стоит узнать название ограничения, для этого воспользуемся командой psql \d [table\_name]  


Название ограничения – client\_pkey

Теперь исполним SQL запрос:

ALTER TABLE client

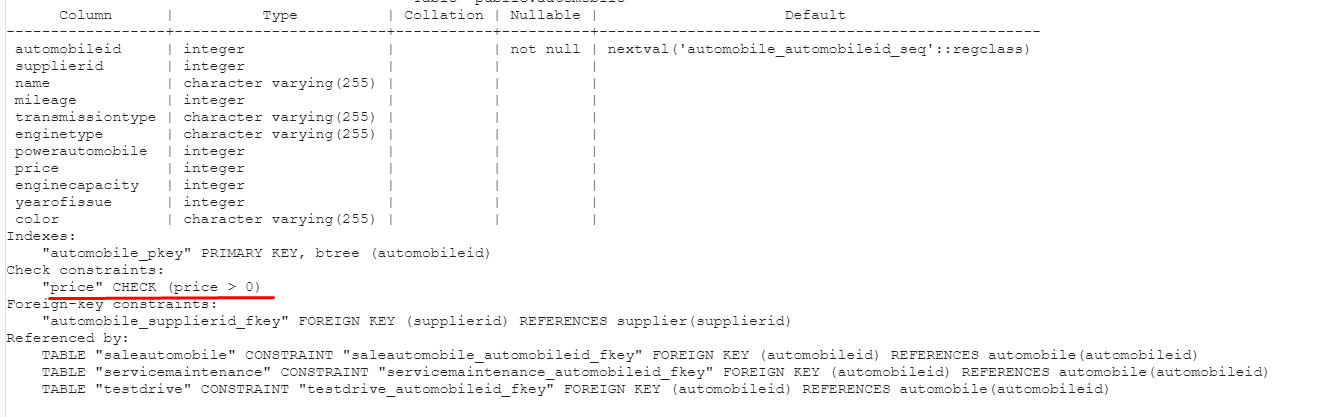
DROP CONSTRAINT client\_pkey;

Для добавления первичного ключа выполним SQL запрос:  
ALTER TABLE client

ADD PRIMARY KEY (clientID)

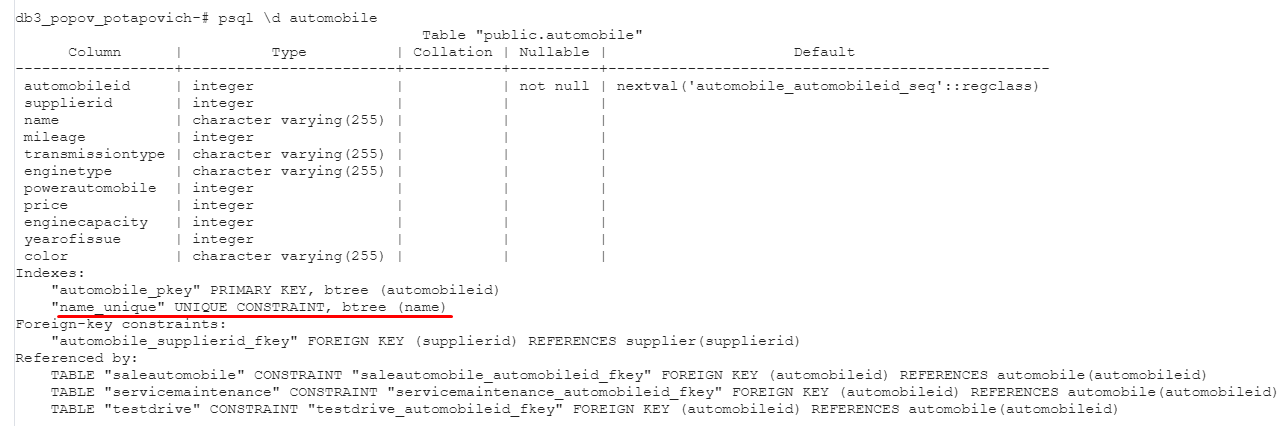
ALTER TABLE Automobile

ADD CONSTRAINT price CHECK (price > 0)



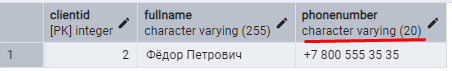
ALTER TABLE Automobile

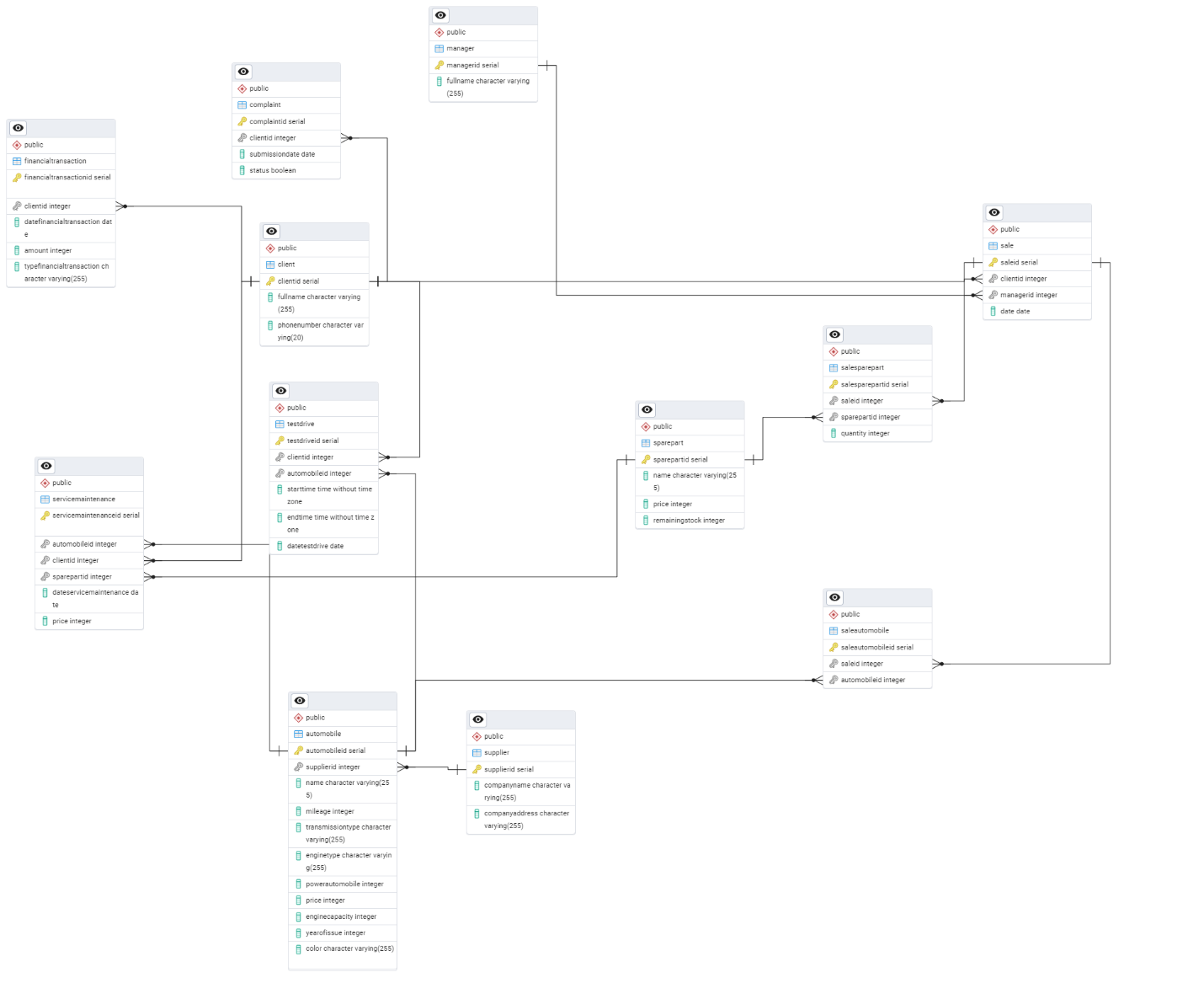
ADD CONSTRAINT name\_unique UNIQUE (name)



ALTER TABLE client

ALTER COLUMN phoneNumber TYPE VARCHAR(20)



5. Сгенерированная реляционная модель:  


Модель совпадает с изначально данной, за исключением таблиц many-to-many отношений.

6. Удаление таблиц и базы данных:  
DROP TABLE IF EXISTS Supplier CASCADE;

DROP TABLE IF EXISTS Client CASCADE;

DROP TABLE IF EXISTS Manager CASCADE;

DROP TABLE IF EXISTS SparePart CASCADE;

DROP TABLE IF EXISTS Automobile CASCADE;

DROP TABLE IF EXISTS FinancialTransaction CASCADE;

DROP TABLE IF EXISTS ServiceMaintenance CASCADE;

DROP TABLE IF EXISTS Sale CASCADE;

DROP TABLE IF EXISTS SaleAutomobile CASCADE;

DROP TABLE IF EXISTS SaleSparePart CASCADE;

DROP TABLE IF EXISTS Complaint CASCADE;

DROP TABLE IF EXISTS TestDrive CASCADE;

DROP DATABASE db3\_popov\_potapovich  
  
Также к команде удаления базы данных можно добавить параметр FORCE,  
для удаления без доп. Проверок  
  
7. Все скрипты можно найти в репозитории по ссылке из лабораторной работы №2. Файлы находятся в папке docs/PostgreSQL

**Вывод:** выполнив лабораторную работу №3 я научился физическому проектированию баз данных. Создал свою базу данных в СУБД pgAdmin для своего проекта, а также по реляционной модели одногруппника.