# МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

# «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

**Кафедра МО ЭВМ**

# ОТЧЕТ

**по лабораторной работе №2 по дисциплине «Базы данных»**

# Тема: Реализация базы данных в СУБД PostgreSQL.

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1304 |  | Нго Тхи Йен. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург 2023

# Цель работы.

Реализация базы данных в СУБД PostgreSQL.

# Задание.

ВАРИАНТ 17

Пусть требуется создать программную систему, предназначенную для диспетчера станции техобслуживания. Такая система должна обеспечивать хранение сведений об услугах, оказываемых станцией и их стоимости, о клиентах станции, о работниках станции и об автомобилях, которые они ремонтируют в текущий момент. Клиент станции – это человек, который хотя бы раз воспользовался услугами станции. О клиенте должны хранится следующие сведения: паспортные данные, включая фамилию, имя, отчество, дату рождения, прописку, а также даты обращения на станцию техобслуживания с указанием автомобилей, которые он сдавал в ремонт. Клиент сдает в ремонт необязательно автомобиль, владельцем которого он является. Сведения об автомобилях включают в себя марку автомобиля, его цвет, год выпуска, номер государственной регистрации, перечень неисправностей и данные о владельце. Сведения о работнике – это его фамилия, имя, отчество, специальность, разряд, стаж работы. Диспетчер заносит в БД сведения об автомобиле и о клиенте, если клиент обращается на станцию впервые. После этого диспетчер определяет рабочих, которые будут устранять имеющиеся в автомобиле неисправности. Оставляя автомобиль на станции техобслуживания, клиент получает расписку, в которой указано, когда автомобиль был поставлен на ремонт, какие он имеет неисправности, когда станция обязуется возвратить отремонтированный автомобиль. После возвращения автомобиля клиенту данные о произведенном ремонте помещаются в архив, клиент получает счет, в котором содержится перечень устраненных неисправностей с указанием времени работы, стоимости работы и стоимости запчастей. Возможно увольнение и прием на работу работников станции, изменение сведений о клиенте (клиент может поменять паспорт, права, адрес, телефон), номера государственной регистрации и цвета автомобиля. Диспетчеру могут потребоваться следующие сведения:

* фамилия, имя, отчество и адрес владельца автомобиля с данным номером

государственной регистрации?

* Марка и год выпуска автомобиля данного владельца?
* Перечень устраненных неисправностей в автомобиле данного владельца?
* фамилия, имя, отчество работника станции, устранявшего данную

неисправность в автомобиле данного клиента, и время ее устранения?

* фамилия, имя, отчество клиентов, сдавших в ремонт автомобили с

указанным типом неисправности?

Необходимо развернуть PostgreSQL локально:

* Написать запросы для создания таблиц из предыдущей лабораторный работы
* Заполнить тестовыми данными: 5-10 строк на каждую таблицу, обязательно наличие связи между ними, данные приближены к реальности.
* Написать запросы к БД, отвечающие на вопросы из предыдущей лабораторной работы
* Исходный код выложить на [www.db-fiddle.com](http://www.db-fiddle.com/) для проверки работоспособности
* Исходный код в виде .sql файла запушить в виде PR в репо
* В отчете описать:
  + Цель
  + Текст задания в соответствии с вариантом
  + Скриншоты работы с СУБД PostgreSQL (psql / DBeaver / Datagrip, ...)
  + Скриншоты на каждый запрос (или группу запросов) на изменение/таблицы с выводом результатов (ответ)
  + Исходный код в приложении
  + Ссылку на исходный код [www.db-fiddle.com](http://www.db-fiddle.com/) в приложении
  + Ссылка на PR в приложении
  + Вывод

# Выполнение работы.

1. Сперва была установлена СУБД PostgreSQL и pgAdmin4 на Windows. А также локально была создана база данных library (см. рис. 1).

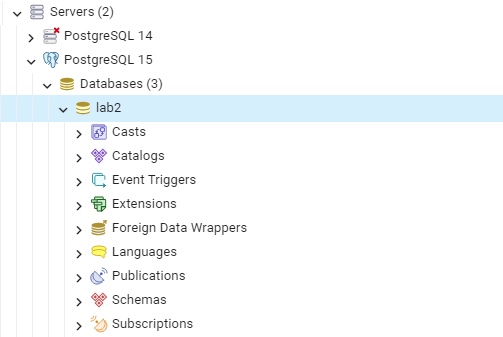


Рисунок 1 – База данных lab2.

1. Затем были созданы таблицы по структуре, которая была описана в предыдущей лабораторной работе. Запросы по созданию таблиц находятся в приложении А. Вид таблиц смотреть рисунки 2 – 10.

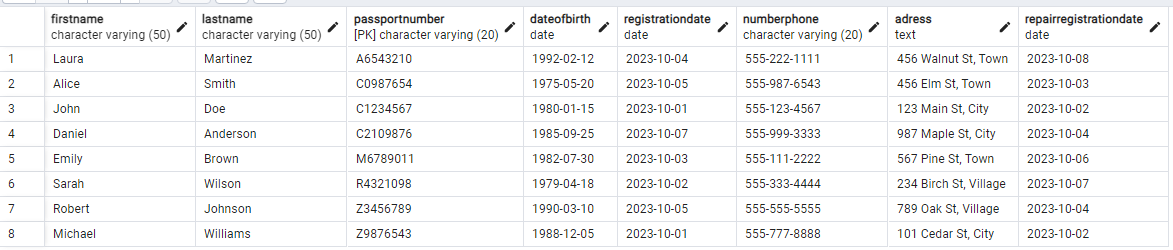


Рисунок 2 – Таблица customers.

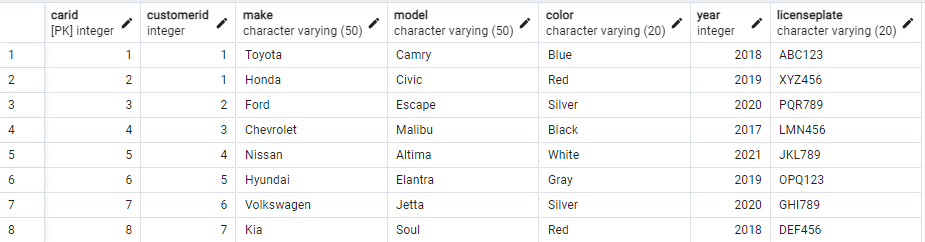


Рисунок 3 – Таблица cars.

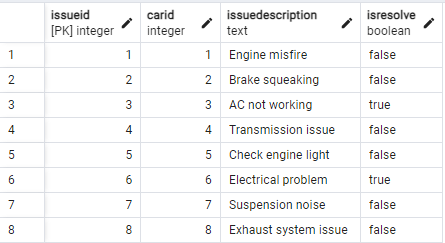


Рисунок 4 – Таблица carsissues

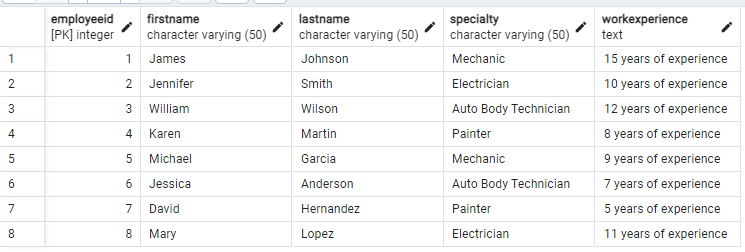


Рисунок 5 – Таблица employees .

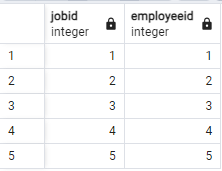


Рисунок 6 – Таблица repairsworker.

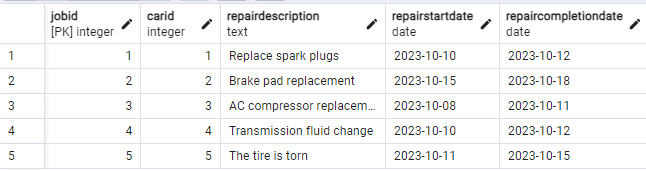


Рисунок 7 – Таблица repairjobs.

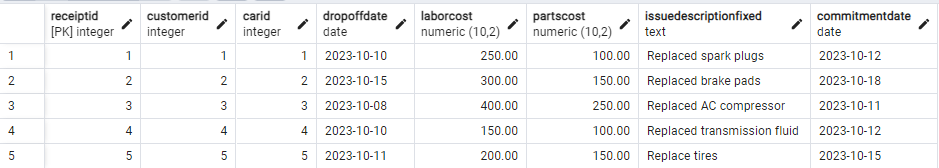


Рисунок 8 – Таблица receipts.

1. Были написаны запросы к таблицам, отвечающие вопросам в задании.

* Фамилия, имя, отчество и адрес владельца автомобиля с данным номером государственной регистрации?

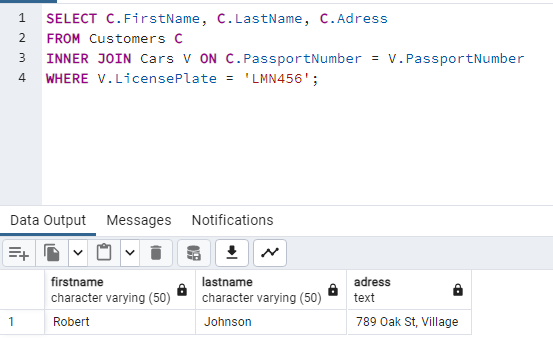


Рисунок 9 - Таблица Фамилия, имя, отчество и адрес

* Марка и год выпуска автомобиля данного владельца?

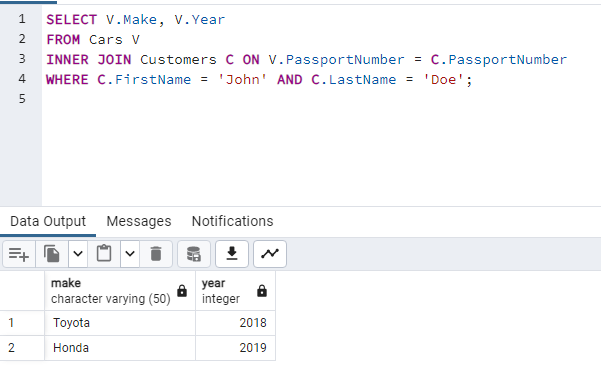


Рисунок 10 - Таблица Марка и год выпуска автомобиля данного

* Перечень устраненных неисправностей в автомобиле данного владельца?

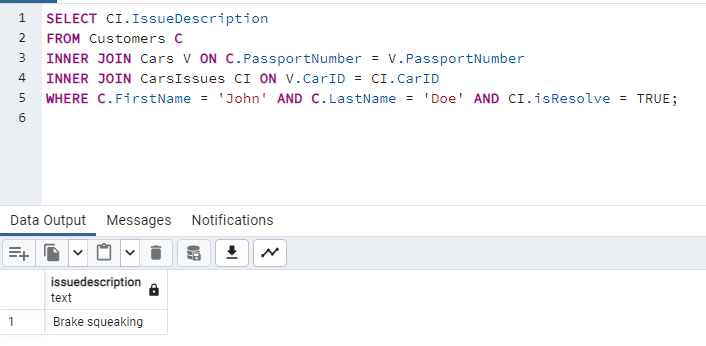


Рисунок 11- Таблица Перечень устраненных неисправностей в автомобиле

* фамилия, имя, отчество работника станции, устранявшего данную неисправность в автомобиле данного клиента, и время ее устранения?

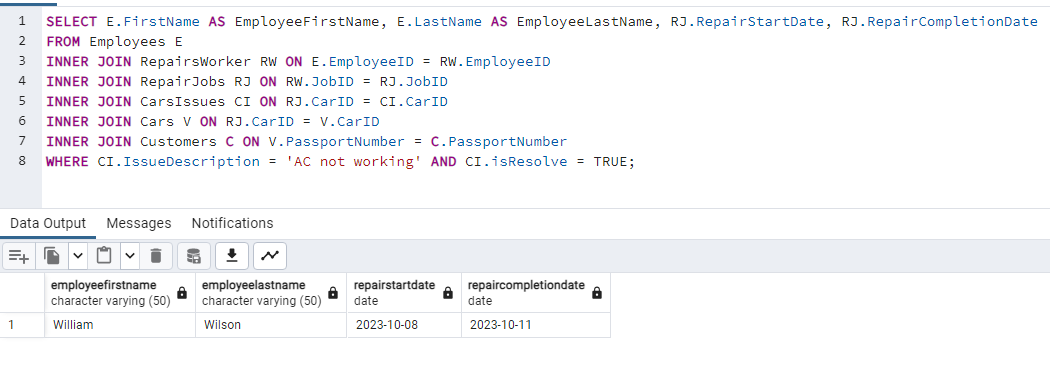


Рисунок 12 - Таблица Фамилия, имя, отчество работника станции, устранявшего данную неисправность и время ее устранения

* Фамилия, имя, отчество клиентов, сдавших в ремонт автомобили с указанным типом неисправности?

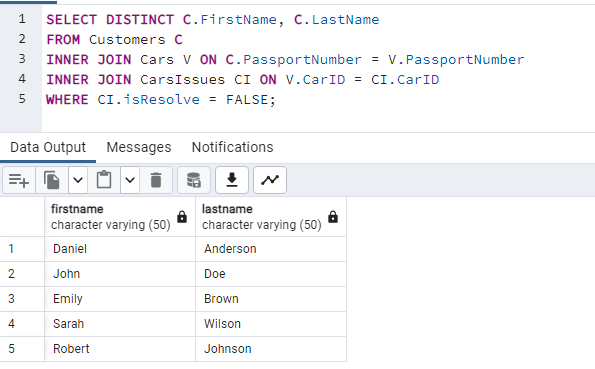


Рисунок 13 - Фамилия, имя, отчество клиентов, сдавших в ремонт автомобили с указанным типом неисправност

Исходный код находиться в приложения A, файле lab2.sql, гитхаб репозитории и на сайте [www.db-fiddle.com](http://www.db-fiddle.com)

# Выводы.

В ходе работы при работе с СУБД PostgreSQL и pgAdmin4 были приобретены навыки работы, а также одновременно реализована спроектированная в предыдущей работе база данных.

# ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД

Название файла: lab2.sql

-- Создание таблицы Customers

CREATE TABLE Customers (

FirstName VARCHAR(50),

LastName VARCHAR(50),

PassportNumber VARCHAR(20) PRIMARY KEY,

DateOfBirth DATE,

RegistrationDate DATE,

NumberPhone VARCHAR(20),

Adress TEXT,

RepairRegistrationDate DATE

);

-- Вставка тестовых данных

INSERT INTO Customers (FirstName, LastName, PassportNumber, DateOfBirth, RegistrationDate, NumberPhone, Adress, RepairRegistrationDate)

VALUES

('John', 'Doe', 'C1234567', '1980-01-15', '2023-10-01', '555-123-4567', '123 Main St, City', '2023-10-02'),

('Alice', 'Smith', 'C0987654', '1975-05-20', '2023-10-05', '555-987-6543', '456 Elm St, Town', '2023-10-03'),

('Robert', 'Johnson', 'Z3456789', '1990-03-10', '2023-10-05', '555-555-5555', '789 Oak St, Village', '2023-10-04'),

('Emily', 'Brown', 'M6789011', '1982-07-30', '2023-10-03', '555-111-2222', '567 Pine St, Town', '2023-10-06'),

('Michael', 'Williams', 'Z9876543', '1988-12-05', '2023-10-01', '555-777-8888', '101 Cedar St, City', '2023-10-02'),

('Sarah', 'Wilson', 'R4321098', '1979-04-18', '2023-10-02', '555-333-4444', '234 Birch St, Village', '2023-10-07'),

('Daniel', 'Anderson', 'C2109876', '1985-09-25', '2023-10-07', '555-999-3333', '987 Maple St, City', '2023-10-04'),

('Laura', 'Martinez', 'A6543210', '1992-02-12', '2023-10-04', '555-222-1111', '456 Walnut St, Town', '2023-10-08');

-- Создание таблицы Cars

CREATE TABLE Cars (

CarID INT PRIMARY KEY,

PassportNumber VARCHAR(20),

Make VARCHAR(50),

Model VARCHAR(50),

Color VARCHAR(20),

Year INT,

LicensePlate VARCHAR(20),

FOREIGN KEY (PassportNumber) REFERENCES Customers(PassportNumber)

);

-- Вставка тестовых данных

INSERT INTO Cars (CarID, PassportNumber, Make, Model, Color, Year, LicensePlate)

VALUES

(1, 'C1234567', 'Toyota', 'Camry', 'Blue', 2018, 'ABC123'),

(2, 'C1234567', 'Honda', 'Civic', 'Red', 2019, 'XYZ456'),

(3, 'C0987654', 'Ford', 'Escape', 'Silver', 2020, 'PQR789'),

(4, 'Z3456789', 'Chevrolet', 'Malibu', 'Black', 2017, 'LMN456'),

(5, 'M6789011', 'Nissan', 'Altima', 'White', 2021, 'JKL789'),

(6, 'Z9876543', 'Hyundai', 'Elantra', 'Gray', 2019, 'OPQ123'),

(7, 'R4321098', 'Volkswagen', 'Jetta', 'Silver', 2020, 'GHI789'),

(8, 'C2109876', 'Kia', 'Soul', 'Red', 2018, 'DEF456');

-- Создание таблицы CarsIssues

CREATE TABLE CarsIssues (

IssueID INT PRIMARY KEY,

CarID INT,

IssueDescription TEXT,

isResolve BOOLEAN,

FOREIGN KEY (CarID) REFERENCES Cars(CarID)

);

-- Вставка тестовых данных

INSERT INTO CarsIssues (IssueID, CarID, IssueDescription, isResolve)

VALUES

(1, 1, 'Engine misfire', FALSE),

(2, 2, 'Brake squeaking', TRUE),

(3, 3, 'AC not working', TRUE),

(4, 4, 'Transmission issue', FALSE),

(5, 5, 'Check engine light', FALSE),

(6, 6, 'Electrical problem', TRUE),

(7, 7, 'Suspension noise', FALSE),

(8, 8, 'Exhaust system issue', FALSE);

-- Создание таблицы Employees

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Specialty VARCHAR(50),

WorkExperience TEXT

);

-- Вставка тестовых данных

INSERT INTO Employees (EmployeeID, FirstName, LastName, Specialty, WorkExperience)

VALUES

(1, 'James', 'Johnson', 'Mechanic', '15 years of experience'),

(2, 'Jennifer', 'Smith', 'Electrician', '10 years of experience'),

(3, 'William', 'Wilson', 'Auto Body Technician', '12 years of experience'),

(4, 'Karen', 'Martin', 'Painter', '8 years of experience'),

(5, 'Michael', 'Garcia', 'Mechanic', '9 years of experience'),

(6, 'Jessica', 'Anderson', 'Auto Body Technician', '7 years of experience'),

(7, 'David', 'Hernandez', 'Painter', '5 years of experience'),

(8, 'Mary', 'Lopez', 'Electrician', '11 years of experience');

CREATE TABLE RepairJobs (

JobID INT PRIMARY KEY,

CarID INT,

RepairDescription TEXT,

RepairStartDate DATE,

RepairCompletionDate DATE,

FOREIGN KEY (CarID) REFERENCES Cars(CarID)

);

-- Вставка тестовых данных

INSERT INTO RepairJobs (JobID, CarID, RepairDescription, RepairStartDate, RepairCompletionDate)

VALUES

(1, 1, 'Replace spark plugs', '2023-10-10', '2023-10-12'),

(2, 2, 'Brake pad replacement', '2023-10-15', '2023-10-18'),

(3, 3, 'AC compressor replacement', '2023-10-08', '2023-10-11'),

(4, 4, 'Transmission fluid change', '2023-10-10', '2023-10-12'),

(5, 5, 'The tire is torn', '2023-10-11', '2023-10-15');

-- Создание таблицы RepairsWorker

CREATE TABLE RepairsWorker (

JobID INT,

EmployeeID INT,

FOREIGN KEY (JobID) REFERENCES RepairJobs(JobID),

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

-- Вставка тестовых данных

INSERT INTO RepairsWorker (JobID, EmployeeID)

VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5);

-- Создание таблицы Receipts

CREATE TABLE Receipts (

ReceiptID INT PRIMARY KEY,

PassportNumber VARCHAR(20),

CarID INT,

DropOffDate DATE,

LaborCost DECIMAL(10, 2),

PartsCost DECIMAL(10, 2),

IssueDescriptionFixed TEXT,

CommitmentDate DATE,

FOREIGN KEY (PassportNumber) REFERENCES Customers(PassportNumber),

FOREIGN KEY (CarID) REFERENCES Cars(CarID)

);

-- Вставка тестовых данных

INSERT INTO Receipts (ReceiptID, PassportNumber, CarID, DropOffDate, LaborCost, PartsCost, IssueDescriptionFixed, CommitmentDate)

VALUES

(1, 'C1234567', 1, '2023-10-10', 250.00, 100.00, 'Replaced spark plugs', '2023-10-12'),

(2, 'C1234567', 2, '2023-10-15', 300.00, 150.00, 'Replaced brake pads', '2023-10-18'),

(3, 'Z3456789', 3, '2023-10-08', 400.00, 250.00, 'Replaced AC compressor', '2023-10-11'),

(4, 'M6789011', 4, '2023-10-10', 150.00, 100.00, 'Replaced transmission fluid', '2023-10-12'),

(5, 'Z9876543', 5, '2023-10-11', 200.00, 150.00, 'Replace tires', '2023-10-15');

-- 1

SELECT C.FirstName, C.LastName, C.Adress

FROM Customers C

INNER JOIN Cars V ON C.PassportNumber = V.PassportNumber

WHERE V.LicensePlate = 'LMN456';

-- 2

SELECT V.Make, V.Year

FROM Cars V

INNER JOIN Customers C ON V.PassportNumber = C.PassportNumber

WHERE C.FirstName = 'John' AND C.LastName = 'Doe';

-- 3

SELECT CI.IssueDescription

FROM Customers C

INNER JOIN Cars V ON C.PassportNumber = V.PassportNumber

INNER JOIN CarsIssues CI ON V.CarID = CI.CarID

WHERE C.FirstName = 'John' AND C.LastName = 'Doe' AND CI.isResolve = TRUE;

-- 4

SELECT E.FirstName AS EmployeeFirstName, E.LastName AS EmployeeLastName, RJ.RepairStartDate, RJ.RepairCompletionDate

FROM Employees E

INNER JOIN RepairsWorker RW ON E.EmployeeID = RW.EmployeeID

INNER JOIN RepairJobs RJ ON RW.JobID = RJ.JobID

INNER JOIN CarsIssues CI ON RJ.CarID = CI.CarID

INNER JOIN Cars V ON RJ.CarID = V.CarID

INNER JOIN Customers C ON V.PassportNumber = C.PassportNumber

WHERE CI.IssueDescription = 'AC not working' AND CI.isResolve = TRUE;

-- 5

SELECT DISTINCT C.FirstName, C.LastName

FROM Customers C

INNER JOIN Cars V ON C.PassportNumber = V.PassportNumber

INNER JOIN CarsIssues CI ON V.CarID = CI.CarID

WHERE CI.isResolve = FALSE;

# ПРИЛОЖЕНИЕ Б ССЫЛКИ

https://github.com/moevm/sql-2023-1304/pull/36

<https://www.db-fiddle.com/f/82tHwZfyskUD6wFiUnpgdj/0>