**Клиенты (Clients)**

client\_id - INT (PK, AI) - Уникальный идентификатор клиента

client\_type - ENUM('0', '1') - Тип клиента (0 для физ лиц, 1 для юр лиц)

full\_name - VARCHAR(100) - ФИО

contact\_phone - VARCHAR(20) - Контактный номер телефона

email - VARCHAR(100) - Контактная почта

organization\_name - VARCHAR(100) - Название организации для юр лиц, NULL для физ лиц

client\_notes - TEXT - Примечание к клиенту

**Устройства (Devices)**  
device\_id - INT (PK, AI) - Уникальный идентификатор устройства

device\_type - VARCHAR(50) - Тип устройства (принтер, компьютер)

manufacturer - VARCHAR(50) - Производитель устройства

model\_number - VARCHAR(50) - Номер модели устройства

serial\_number - VARCHAR(50) - Серийный номер устройства

completeness - TEXT – Комплектность

fault\_description - TEXT - Характер неисправности, конкретная поломка, с которой устройство сдали

device\_notes - TEXT - Примечание к устройству (например, "Корпус поцарапан, батарея старая")

**Ремонты (Repairs)**

repair\_id - INT (PK, AI) - Уникальный ID ремонта

device\_id - INT (FK) - Ссылка на устройство

receipt\_id - INT (FK) - Ссылка на квитанцию

work\_performed - TEXT - Проведённые работы

acceptance\_date - DATE - Дата принятия в ремонт

completion\_date - DATE - Фактическая дата выдачи клиенту, если в работе NULL

status - repair\_status\_enum AS ENUM ('принят', 'выполнен', 'выдан');

repair\_notes - TEXT - Доп. примечания к ремонту

**Receipts (Квитанции)**

receipt\_id - INT (PK, AI) - Уникальный ID заказа

client\_id - INT (FK) - Ссылка на клиента

doc\_path - VARCHAR(255) - Путь к файлу с квитанцией

**Запчасть (Sparepart)**

sparepart\_id - INT (PK, AI) - Уникальный ID запчасти

repair\_id - INT (FK) - Ссылка на ремонт (NULL, если запчасть на складе)

name - VARCHAR(100) - Название запчасти

price - DECIMAL(10, 2) – Стоимость запчасти

quantity - INT – Количество (по умолчанию 1)

**-- RepairLogDB**

-- Таблица клиентов (с измененным client\_type)

CREATE TABLE Clients (

client\_id SERIAL PRIMARY KEY,

client\_type INT NOT NULL CHECK (client\_type IN (0, 1)), -- 0 для физ лиц, 1 для юр лиц

full\_name VARCHAR(100) NOT NULL,

contact\_phone VARCHAR(20) NOT NULL,

email VARCHAR(100),

organization\_name VARCHAR(100),

client\_notes TEXT

);

-- Таблица устройств

CREATE TABLE Devices (

device\_id SERIAL PRIMARY KEY,

device\_type VARCHAR(50) NOT NULL,

manufacturer VARCHAR(50) NOT NULL,

model\_number VARCHAR(50) NOT NULL,

serial\_number VARCHAR(50) NOT NULL,

completeness TEXT NOT NULL,

fault\_description TEXT NOT NULL,

device\_notes TEXT

);

-- Таблица квитанций

CREATE TABLE Receipts (

receipt\_id SERIAL PRIMARY KEY,

client\_id INTEGER NOT NULL,

doc\_path VARCHAR(255) NOT NULL,

FOREIGN KEY (client\_id) REFERENCES Clients(client\_id) ON DELETE CASCADE

);

-- Создаем перечисление для статуса ремонта

CREATE TYPE repair\_status\_enum AS ENUM ('принят', 'выполнен', 'выдан');

-- Таблица ремонтов

CREATE TABLE Repairs (

repair\_id SERIAL PRIMARY KEY,

device\_id INTEGER NOT NULL,

receipt\_id INTEGER NOT NULL,

work\_performed TEXT NOT NULL,

acceptance\_date DATE NOT NULL,

completion\_date DATE,

status repair\_status\_enum NOT NULL DEFAULT 'принят',

repair\_notes TEXT,

FOREIGN KEY (device\_id) REFERENCES Devices(device\_id) ON DELETE CASCADE,

FOREIGN KEY (receipt\_id) REFERENCES Receipts(receipt\_id) ON DELETE CASCADE

);

-- Таблица запчастей

CREATE TABLE Sparepart (

sparepart\_id SERIAL PRIMARY KEY,

repair\_id INTEGER,

name VARCHAR(100) NOT NULL,

price DECIMAL(10, 2) NOT NULL,

quantity INTEGER DEFAULT 1,

FOREIGN KEY (repair\_id) REFERENCES Repairs(repair\_id) ON DELETE SET NULL

);

-- Создаем индексы для улучшения производительности

CREATE INDEX idx\_clients\_phone ON Clients(contact\_phone);

CREATE INDEX idx\_devices\_serial ON Devices(serial\_number);

CREATE INDEX idx\_repairs\_dates ON Repairs(acceptance\_date, completion\_date);

CREATE INDEX idx\_receipts\_client ON Receipts(client\_id);

CREATE INDEX idx\_sparepart\_repair ON Sparepart(repair\_id);

**-- ТРИГЕР**  
-- Создаем функцию-обработчик для триггера

CREATE OR REPLACE FUNCTION update\_repair\_status\_on\_completion()

RETURNS TRIGGER AS $$

BEGIN

-- Если completion\_date установлена (не NULL) и статус еще не "выдан"

IF NEW.completion\_date IS NOT NULL AND NEW.status <> 'выдан' THEN

NEW.status := 'выдан';

END IF;

-- Если completion\_date сбрасывается в NULL и статус "выдан"

IF NEW.completion\_date IS NULL AND OLD.completion\_date IS NOT NULL AND NEW.status = 'выдан' THEN

NEW.status := 'принят'; -- или 'выполнен', в зависимости от бизнес-логики

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

-- Создаем триггер для таблицы Repairs

CREATE TRIGGER trigger\_update\_repair\_status

BEFORE UPDATE OF completion\_date, status ON Repairs

FOR EACH ROW

EXECUTE FUNCTION update\_repair\_status\_on\_completion();

добавь в dataGridViewRepairs такую информацию:

первый столбец с названием "Устройство" содержащий device\_type + manufacturer + model\_number,

второй столбец с названием "Заказчик" содержащий full\_name если client\_type=0, organization\_name если client\_type=1,

третий столбец с названием "Ремонта" содержащий если repair\_status\_enum = “принят”, то repair\_status\_enum + acceptance\_date, если repair\_status\_enum = “выполнен”, то repair\_status\_enum, если repair\_status\_enum = “выдан”, то repair\_status\_enum + completion\_date,  
четвёртый столбец с названием "Запчасти" содержащий кнопку для нажатия,  
пятый столбец с названием "Квитанция " содержащий receipt\_id

-- **УДАЛЕНИЕ**-- Удаляем таблицы в правильном порядке, учитывая зависимости по внешним ключам

DROP TABLE IF EXISTS Repairs;

DROP TABLE IF EXISTS Devices;

DROP TABLE IF EXISTS Clients;

-- **ЗАПОЛНЕНИЕ**

-- Отключаем проверку внешних ключей временно

SET session\_replication\_role = replica;

-- Заполнение таблицы Clients

INSERT INTO Clients (client\_id, client\_type, full\_name, contact\_phone, email, organization\_name, client\_notes) VALUES

-- Физические лица (client\_type = 0)

(1, 0, 'Иванов Петр Сергеевич', '+7-912-345-67-89', 'ivanov@mail.ru', NULL, 'Постоянный клиент, аккуратный'),

(2, 0, 'Смирнова Анна Владимировна', '+7-923-456-78-90', 'smirnova.anna@gmail.com', NULL, 'Предпочитает утренние визиты'),

(3, 0, 'Козлов Дмитрий Игоревич', '+7-934-567-89-01', NULL, NULL, 'Требует детальный отчет о работе'),

-- Юридические лица (client\_type = 1)

(4, 1, 'ООО "ВекторТех"', '+7-495-123-45-67', 'info@vectortech.ru', 'ООО "ВекторТех"', 'Корпоративный клиент, договор №123'),

(5, 1, 'ИП Сорокин А.В.', '+7-800-555-35-35', 'ip@sorokin.ru', 'ИП Сорокин Александр Владимирович', 'Мелкий опт, скидка 10%');

-- Заполнение таблицы Devices

INSERT INTO Devices (device\_id, device\_type, manufacturer, model\_number, serial\_number, completeness, fault\_description, device\_notes) VALUES

(1, 'Ноутбук', 'Lenovo', 'ThinkPad T480', 'SN12345678', 'Ноутбук, блок питания', 'Не включается, нет реакции на кнопку питания', 'На корпусе царапины'),

(2, 'Смартфон', 'Samsung', 'Galaxy S21', 'SN87654321', 'Телефон, кабель USB-C', 'Разбит экран, не работает тачскрин', 'Требуется замена дисплейного модуля'),

(3, 'Принтер', 'Canon', 'PIXMA MG3640', 'SN11223344', 'Принтер, блок питания, кабели', 'Не печатает, зажевывает бумагу', 'Последняя заправка 2 месяца назад'),

(4, 'Монитор', 'Dell', 'U2419H', 'SN55667788', 'Монитор, кабель питания, HDMI', 'Мерцание экрана, полосы по краям', 'На гарантии до 2024 года'),

(5, 'Планшет', 'Apple', 'iPad Air 4', 'SN99001122', 'Планшет, стилус', 'Не заряжается, разъем поврежден', 'Водные повреждения, срочный ремонт');

-- Заполнение таблицы Receipts

INSERT INTO Receipts (receipt\_id, client\_id, doc\_path) VALUES

(1, 1, '/docs/receipts/2023/10/001.pdf'),

(2, 2, '/docs/receipts/2023/10/002.pdf'),

(3, 3, '/docs/receipts/2023/10/003.pdf'),

(4, 4, '/docs/receipts/2023/10/004.pdf'),

(5, 5, '/docs/receipts/2023/10/005.pdf');

-- Заполнение таблицы Repairs

INSERT INTO Repairs (repair\_id, device\_id, receipt\_id, work\_performed, acceptance\_date, completion\_date, status, repair\_notes) VALUES

(1, 1, 1, 'Диагностика, замена материнской платы', '2023-10-01', '2023-10-05', 'выдан', 'Клиент проинформирован о замене комплектующих'),

(2, 2, 2, 'Замена дисплейного модуля, калибровка', '2023-10-02', '2023-10-04', 'выдан', 'Установлено защитное стекло'),

(3, 3, 3, 'Чистка механизма подачи бумаги, замена картриджей', '2023-10-03', NULL, 'выполнен', 'Ожидает выдачи клиентом'),

(4, 4, 4, 'Диагностика, ремонт матрицы', '2023-10-04', NULL, 'принят', 'Требуется дополнительная диагностика'),

(5, 5, 5, 'Замена разъема charging port, чистка от коррозии', '2023-10-05', NULL, 'принят', 'Высокий риск повреждения платы');

-- Заполнение таблицы Sparepart

INSERT INTO Sparepart (sparepart\_id, repair\_id, name, price, quantity) VALUES

(1, 1, 'Материнская плата Lenovo T480', 8500.00, 1),

(2, 1, 'Термопаста', 200.00, 1),

(3, 2, 'Дисплейный модуль Samsung S21', 12000.00, 1),

(4, 2, 'Защитное стекло', 1500.00, 1),

(5, 3, 'Картридж черный Canon PIXMA MG3640', 1800.00, 1),

(6, 3, 'Картридж цветной Canon PIXMA MG3640', 2100.00, 3),

(7, 4, 'Матрица Dell U2419H', 7500.00, 1),

(8, 5, 'Разъем charging port iPad Air 4', 3200.00, 1);

-- Включаем проверку внешних ключей обратно

SET session\_replication\_role = DEFAULT;

-- Обновляем последовательности чтобы они продолжались с правильного номера

SELECT setval('clients\_client\_id\_seq', (SELECT MAX(client\_id) FROM Clients));

SELECT setval('devices\_device\_id\_seq', (SELECT MAX(device\_id) FROM Devices));

SELECT setval('receipts\_receipt\_id\_seq', (SELECT MAX(receipt\_id) FROM Receipts));

SELECT setval('repairs\_repair\_id\_seq', (SELECT MAX(repair\_id) FROM Repairs));

SELECT setval('sparepart\_sparepart\_id\_seq', (SELECT MAX(sparepart\_id) FROM Sparepart));