

Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО

ОТЧЕТ

по Лабораторной работе № 3

по дисциплине «Проектирование и реализация баз данных»

Обучающийся: Иванов Виктор Сергеевич

Факультет прикладной информатики

Группа К3240

Направление подготовки 09.03.03 Прикладная информатика

Образовательная программа: Мобильные и сетевые технологии 2023

Преподаватель: Говорова Марина Михайловна

Санкт-Петербург
2024/2025

Цель работы

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

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 в формате **CUSTOM** — для восстановления базы данных
7. Восстановить базу данных.

Модель

Модель представляет процесс заключения договоров с клиентами, планирования этапов изготовления деталей на станках с использованием чертежей и листов стали.

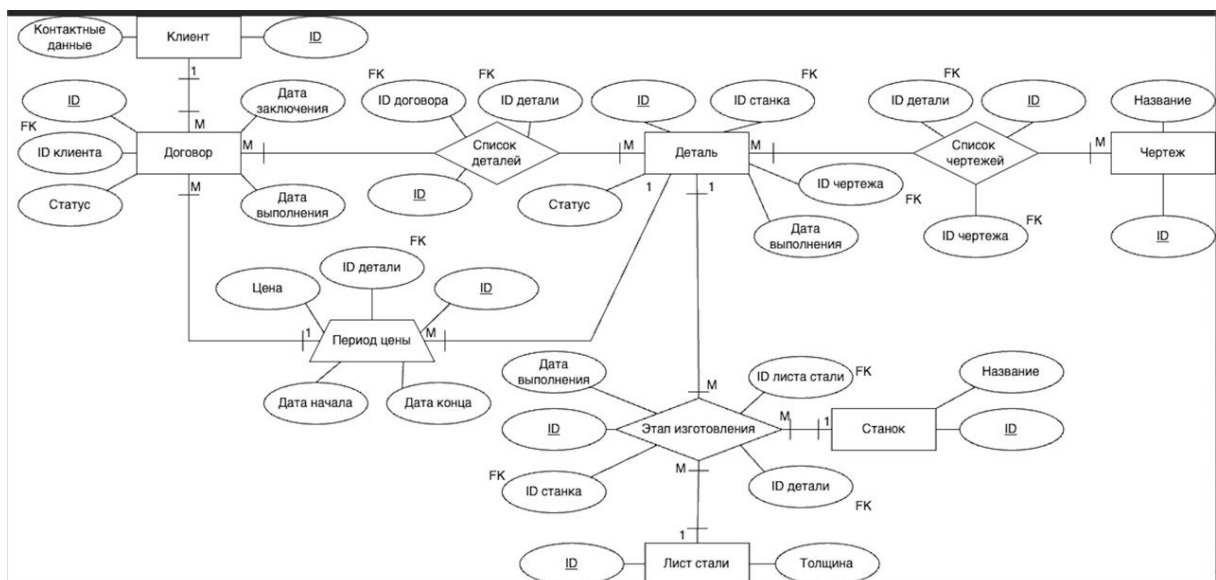


Рис. 1: Модель для создания БД

Основные сущности

- **Клиент** — содержит контактные данные.
- **Договор** — включает статус, дату заключения и выполнения, связан с клиентом.
- **Деталь** — центральная сущность, связана с договорами, чертежами, ценами, станками.
- **Чертеж** — может использоваться для изготовления множества деталей.
- **Станок** — выполняет этапы изготовления, имеет название, связан с листом стали.
- **Лист стали** — характеризуется толщиной, используется в изготовлении деталей.
- **Период цены** — задаёт стоимость детали в определённый период.
- **Этап изготовления** — отражает факт выполнения операции станком над деталью.

Связи между сущностями

- Один клиент заключает множество договоров.
- Один договор может включать множество деталей.
- Детали изготавливаются на станках (М:М через «этап изготовления»).
- Для изготовления используются чертежи (через сущность «список чертежей»).
- Детали имеют историю цен — каждый период имеет дату начала, конца и цену.
- Каждый этап изготовления связан с листом стали, станком и конкретной деталью.

Generate ERD схемы

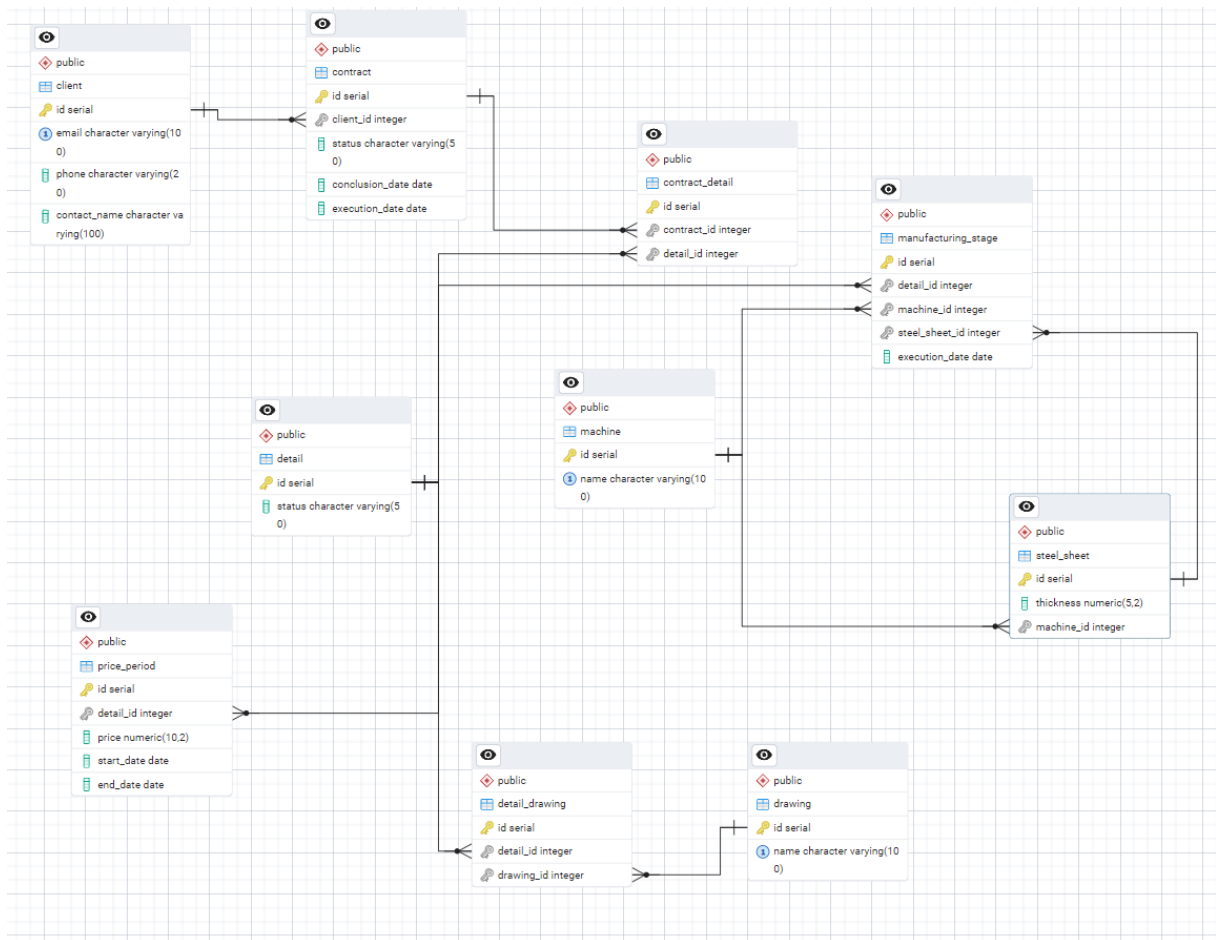


Рис. 2: Generate ERD

Листинги SQL запросов

Скрипт создания Базы Данных:

```
CREATE DATABASE postgres
WITH
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'ru-RU'
  LC_CTYPE = 'ru-RU'
  LOCALE_PROVIDER = 'libc'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1
  IS_TEMPLATE = False;
```

Листинг 1: Создание БД-шки

Скрипт создания всех таблиц:

```
--- client
CREATE TABLE client (
  id SERIAL PRIMARY KEY,
```

```

    email VARCHAR(100) NOT NULL UNIQUE,
    phone VARCHAR(20) NOT NULL,
    contact_name VARCHAR(100)
);

--- contract
CREATE TABLE contract (
    id SERIAL PRIMARY KEY,
    client_id INT NOT NULL,
    status VARCHAR(50) NOT NULL,
    conclusion_date DATE NOT NULL,
    execution_date DATE,
    FOREIGN KEY (client_id) REFERENCES client(id) ON DELETE CASCADE
);

--- detail
CREATE TABLE detail (
    id SERIAL PRIMARY KEY,
    status VARCHAR(50) NOT NULL
);

--- contract_detail(contract-detail)
CREATE TABLE contract_detail (
    id SERIAL PRIMARY KEY,
    contract_id INT NOT NULL,
    detail_id INT NOT NULL,
    FOREIGN KEY (contract_id) REFERENCES contract(id) ON DELETE CASCADE,
    FOREIGN KEY (detail_id) REFERENCES detail(id) ON DELETE CASCADE
);

--- drawing
CREATE TABLE drawing (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL UNIQUE
);

--- detail_drawing
CREATE TABLE detail_drawing (
    id SERIAL PRIMARY KEY,
    detail_id INT NOT NULL,
    drawing_id INT NOT NULL,
    FOREIGN KEY (detail_id) REFERENCES detail(id) ON DELETE CASCADE,
    FOREIGN KEY (drawing_id) REFERENCES drawing(id) ON DELETE CASCADE
);

--- machine
CREATE TABLE machine (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL UNIQUE
);

--- steel_sheet
CREATE TABLE steel_sheet (

```

```

        id SERIAL PRIMARY KEY,
        thickness NUMERIC(5,2) NOT NULL CHECK (thickness > 0),
        machine_id INT NOT NULL,
        FOREIGN KEY (machine_id) REFERENCES machine(id) ON DELETE CASCADE
    );

--- manufacturing_stage
CREATE TABLE manufacturing_stage (
    id SERIAL PRIMARY KEY,
    detail_id INT NOT NULL,
    machine_id INT NOT NULL,
    steel_sheet_id INT NOT NULL,
    execution_date DATE NOT NULL,
    FOREIGN KEY (detail_id) REFERENCES detail(id) ON DELETE CASCADE,
    FOREIGN KEY (machine_id) REFERENCES machine(id) ON DELETE CASCADE,
    FOREIGN KEY (steel_sheet_id) REFERENCES steel_sheet(id) ON DELETE CASCADE
);

--- price_period
CREATE TABLE price_period (
    id SERIAL PRIMARY KEY,
    detail_id INT NOT NULL,
    price NUMERIC(10,2) NOT NULL CHECK (price > 0),
    start_date DATE NOT NULL,
    end_date DATE NOT NULL CHECK (end_date >= start_date),
    FOREIGN KEY (detail_id) REFERENCES detail(id) ON DELETE CASCADE
);

```

Листинг 2: Скрипт создания таблиц

Теперь добавим чеки для наших свойств, используя ALTER TABLE:

```

ALTER TABLE contract
ADD CONSTRAINT check_contract_status
CHECK (status IN ('в работе', 'завершен', 'отменен'));

ALTER TABLE contract
ADD CONSTRAINT check_execution_after_conclusion
CHECK (execution_date IS NULL OR execution_date >= conclusion_date);

---

ALTER TABLE detail
ADD CONSTRAINT check_detail_status
CHECK (status IN ('проект', 'в производстве', 'выполнена'));

---

ALTER TABLE steel_sheet
ADD CONSTRAINT check_thickness
CHECK (thickness >= 0.1 AND thickness <= 100);

---

ALTER TABLE manufacturing_stage

```

```

ADD CONSTRAINT check_execution_date
CHECK (execution_date <= CURRENT_DATE);

---

ALTER TABLE price_period
ADD CONSTRAINT check_price_positive
CHECK (price > 0);

ALTER TABLE price_period
ADD CONSTRAINT check_price_period_dates
CHECK (end_date >= start_date);

```

Листинг 3: Скрипт создания чеков

```

INSERT INTO client (email, phone, contact_name) VALUES
('ivanov@example.com', '+79210000001', 'Иванов И.И.'),
('petrova@example.com', '+79210000002', 'Петрова А.А.'),
('smith@example.com', '+79210000003', 'Smith Industries');

INSERT INTO contract (client_id, status, conclusion_date, execution_date) VALUES
(1, 'в работе', '2025-03-01', NULL),
(2, 'завершен', '2024-11-10', '2025-01-20'),
(3, 'отменен', '2025-01-05', NULL);

INSERT INTO detail (status) VALUES
('проект'),
('в производстве'),
('выполнена');

INSERT INTO contract_detail (contract_id, detail_id) VALUES
(1, 1),
(1, 2),
(2, 3);

INSERT INTO drawing (name) VALUES
('Чертѣж A1'),
('Чертѣж B2');

INSERT INTO detail_drawing (detail_id, drawing_id) VALUES
(1, 1),
(2, 2),
(3, 1);

INSERT INTO machine (name) VALUES
('Станок 01'),
('Станок 02');

INSERT INTO steel_sheet (thickness, machine_id) VALUES
(5.0, 1),
(2.5, 2);

INSERT INTO manufacturing_stage (detail_id, machine_id, steel_sheet_id,
execution_date) VALUES

```

```
(1, 1, 1, '2025-03-20'),  
(2, 2, 2, '2025-03-21');  
  
INSERT INTO price_period (detail_id, price, start_date, end_date) VALUES  
(1, 1200.00, '2025-01-01', '2025-03-31'),  
(2, 1500.50, '2025-02-01', '2025-04-30');
```

Листинг 4: Скрипт заполнения моковыми данными

Выводы

В ходе лабораторной работы была реализована база данных производственного процесса в PostgreSQL. Созданы таблицы, установлены связи и ограничения (ПК, FK, CHECK). Составные ключи были заменены на суррогатные. Таблицы заполнены тестовыми данными, выполнено резервное копирование в формате CUSTOM.

Ну и в конце проверим, что таблички действительно создались через ui pgAdmin:

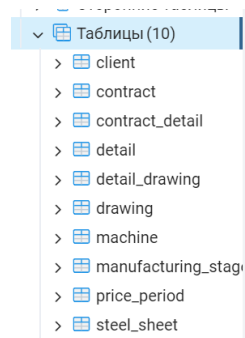


Рис. 3: drawing insert out

Скриншоты вывода моковых данных через *SELECT * FROM <tableName>*:

	id [PK] integer	email character varying (100)	phone character varying (20)	contact_name character varying (100)
1	1	ivanov@example.com	+79210000001	Иванов И.И.
2	2	petrova@example.com	+79210000002	Петрова А.А.
3	3	smith@example.com	+79210000003	Smith Industries

Рис. 4: client insert out

	id [PK] integer	client_id integer	status character varying (50)	conclusion_date date	execution_date date
1	1	1	в работе	2025-03-01	[null]
2	2	2	завершен	2024-11-10	2025-01-20
3	3	3	отменен	2025-01-05	[null]

Рис. 5: contract insert out

	id [PK] integer	status character varying (50)
1	1	проект
2	2	в производстве
3	3	выполнена

Рис. 6: detail insert out

	id [PK] integer	contract_id integer	detail_id integer
1	1	1	1
2	2	1	2
3	3	2	3

Рис. 7: contract detail out

	id [PK] integer	name character varying (100)
1	1	Чертёж А1
2	2	Чертёж В2

Рис. 8: drawing insert out

	id [PK] integer	detail_id integer	drawing_id integer
1	1	1	1
2	2	2	2
3	3	3	1

Рис. 9: detail drawing insert out

	id [PK] integer	name character varying (100)
1	1	Станок 01
2	2	Станок 02

Рис. 10: machine insert out

	id [PK] integer	thickness numeric (5,2)	machine_id integer
1	1	5.00	1
2	2	2.50	2

Рис. 11: steel sheet insert out

	id [PK] integer	detail_id integer	machine_id integer	steel_sheet_id integer	execution_date date
1	1	1	1	1	2025-03-20
2	2	2	2	2	2025-03-21

Рис. 12: manufacturing stage insert out

	id [PK] integer	detail_id integer	price numeric (10,2)	start_date date	end_date date
1	1	1	1200.00	2025-01-01	2025-03-31
2	2	2	1500.50	2025-02-01	2025-04-30

Рис. 13: price period insert out

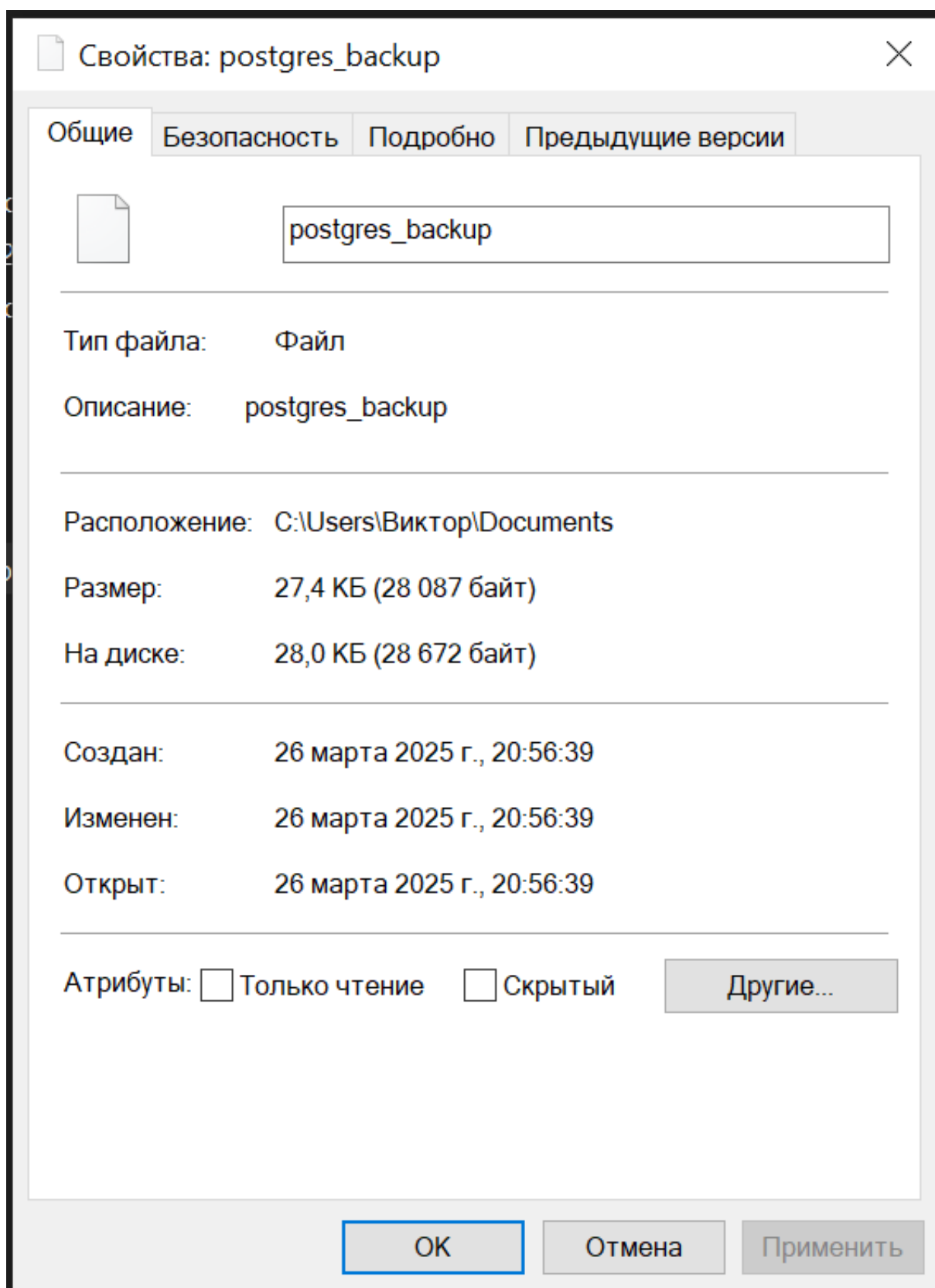


Рис. 14: Красавчик-бэкап