

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИТМО»**

**Факультет инфокоммуникационных технологий**

**Дисциплина:**

**«Проектирование и реализация баз данных»**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1  
«СОЗДАНИЕ БД POSTGRESQL В PGADMIN. РЕЗЕРВНОЕ  
КОПИРОВАНИЕ И ВОССТАНОВЛЕНИЕ БД»**

**Выполнил:**

студент группы К32392  
Тишалович Леонид  
Михайлович

---

(подпись)

**Проверил(а):**

Говорова Марина Михайловна

---

(отметка о выполнении)

---

(подпись)

Санкт-Петербург  
2023 г.

**Цель работы 1.1:** овладеть практическими навыками установки СУБД PostgreSQL и создания базы данных в pgadmin 4.

**Практическое задание 1.1:**

1. Установить СУБД PostgreSQL 1X.
2. Создать базу данных с использованием pgadmin 4.

**Цель работы 1.2:** овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

**Практическое задание 1.2:**

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.

## Выполнение

Наименование БД: Opt Base

ERD диаграмма:



Рисунок 1 - ERD диаграмма

**Dump БД:** dump базы данных в двух вариантах, сгенерированный в pgAdmin, приложен к отчету. Файлы: backup\_custom.sql, backup\_plain.sql

**Скрипты работы с БД для создания таблиц:**

CREATE TABLE provider (

provider\_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

name\_of\_company VARCHAR(50) NOT NULL,

```
full_name VARCHAR(40) NOT NULL,  
address VARCHAR(40) NOT NULL  
);
```

```
CREATE TABLE manager (  
    manager_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    full_name VARCHAR(40) NOT NULL,  
    position VARCHAR(25) NOT NULL,  
    contact_information VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE shipment (  
    shipment_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    provider_id BIGINT,  
    manager_id BIGINT,  
    invoice_creation_date DATE,  
    invoice_fulfillment_date DATE,  
    status VARCHAR(40) CHECK (status IN ('completed', 'not completed')),  
    CHECK (invoice_creation_date <= invoice_fulfillment_date),  
    FOREIGN KEY (provider_id) REFERENCES provider(provider_id),  
    FOREIGN KEY (manager_id) REFERENCES manager(manager_id),  
);
```

```
CREATE TABLE shipment_invoice (  
    shipment_invoice_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
```

```
shipment_id BIGINT,  
  
invoice_number INTEGER NOT NULL,  
  
invoice_date DATE NOT NULL,  
  
payment_date DATE NOT NULL,  
  
FOREIGN KEY (shipment_id) REFERENCES shipment(shipment_id),  
  
CHECK (payment_date > invoice_date)  
  
);
```

```
CREATE TABLE unit_of_measurement (  
  
    unit_of_measurement_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY  
    KEY,  
  
    name VARCHAR(20) NOT NULL  
  
);
```

```
CREATE TABLE product (  
  
    product_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  
    product_type VARCHAR(30) NOT NULL,  
  
    product_name VARCHAR(50) NOT NULL,  
  
    unit_of_measurement_id BIGINT,  
  
    FOREIGN KEY (unit_of_measurement_id) REFERENCES  
    unit_of_measurement(unit_of_measurement_id)  
  
);
```

```
CREATE TABLE shipment_content (  
  
    shipment_content_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  
    shipment_id BIGINT,
```

```
product_id BIGINT,  
  
remains INTEGER CHECK (remains > 0) NOT NULL,  
  
amount_of_goods INTEGER CHECK (amount_of_goods > 0) NOT NULL,  
  
expiration_date DATE NOT NULL,  
  
price INTEGER CHECK (price >= 0) NOT NULL,  
  
FOREIGN KEY (shipment_id) REFERENCES shipment(shipment_id),  
  
FOREIGN KEY (product_id) REFERENCES product(product_id)  
  
);
```

```
CREATE TABLE client (  
  
    client_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  
    full_name VARCHAR(40) NOT NULL,  
  
    organization_name VARCHAR(50) NOT NULL,  
  
    address VARCHAR(50) NOT NULL  
  
);
```

```
CREATE TABLE purchase (  
  
    purchase_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  
    client_id BIGINT,  
  
    manager_id BIGINT,  
  
    invoice_creation_date DATE,  
  
    invoice_fulfillment_date DATE,  
  
    status VARCHAR(40),  
  
    CHECK (invoice_creation_date <= invoice_fulfillment_date),  
  
    FOREIGN KEY (client_id) REFERENCES client(client_id),
```

```
FOREIGN KEY (manager_id) REFERENCES manager(manager_id)

);

CREATE TABLE purchase_invoice (

    purchase_invoice_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    purchase_id BIGINT,

    invoice_number INTEGER NOT NULL,

    invoice_date DATE NOT NULL,

    payment_date DATE NOT NULL,

    FOREIGN KEY (purchase_id) REFERENCES purchase(purchase_id),

    CHECK (payment_date > invoice_date)

);
```

```
CREATE TABLE purchase_content (

    purchase_content_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    purchase_id BIGINT,

    product_id BIGINT,

    amount_of_goods INTEGER CHECK (amount_of_goods > 0) NOT NULL,

    expiration_date DATE NOT NULL,

    price INTEGER CHECK (price > 0) NOT NULL,

    FOREIGN KEY (purchase_id) REFERENCES purchase(purchase_id),

    FOREIGN KEY (product_id) REFERENCES product(product_id)

);
```

**Скрипты работы с БД для вставки данных:**

```
INSERT INTO provider (name_of_company, full_name, address) VALUES  
  
('ABC Supplies', 'John Smith', '123 Main St'),  
  
('XYZ Inc.', 'Jane Doe', '456 Oak Ave'),  
  
('Acme Corp.', 'Bob Johnson', '789 Maple Ln');
```

```
INSERT INTO manager (full_name, position, contact_information) VALUES  
  
('John Doe', 'CEO', '555-1234'),  
  
('Jane Smith', 'CFO', '555-5678'),  
  
('Bob Johnson', 'COO', '555-9101'),  
  
('Sue Wilson', 'Marketing Director', '555-1212');
```

```
INSERT INTO unit_of_measurement (name) VALUES  
  
('kg'),  
  
('g'),  
  
('lb'),  
  
('oz'),  
  
('L'),  
  
('mL'),  
  
('gal'),  
  
('fl oz');
```

```
INSERT INTO client (full_name, organization_name, address) VALUES  
  
('John Smith', 'ABC Corp', '123 Main St'),  
  
('Jane Johnson', 'XYZ Inc', '456 Maple Ave'),
```



('Bob Brown', 'Brown Enterprises', '789 Oak St'),

('Sue Wilson', 'Wilson Marketing', '321 Elm St');

INSERT INTO client (full\_name, organization\_name, address) VALUES

('John Smith', 'ABC Company', '123 Main St'),

('Jane Doe', 'XYZ Corporation', '456 Second St'),

('Bob Johnson', 'DEF Enterprises', '789 Third St'),

('Sarah Williams', 'GHI Industries', '321 Fourth St');

### **Выводы**

В процессе выполнения лабораторной работы удалось более детально ознакомиться с работой в pgAdmin 4, получить практические навыки создания таблиц, установки ограничений на таблицы, создания и восстановления резервных копий баз данных.