Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Кафедра информационных систем и технологий

**Основы работы с Docker и PostgreSQL**

Отчет по лабораторной работе №1

по дисциплине «Разработка приложений»

Студент 24.09.2025 А.С. Смирнов

Дата Роспись студ.

Группа РИМ - 150950

Преподаватель 24.09.2025 Д.И. Кузьмин

Дата Роспись препод.

Екатеринбург 2025

## Цели работы

Освоить фундаментальные концепции и базовые операции Docker: создание образов, запуск контейнеров, управление ими, работа с сетями и томами. На практике закрепить навыки, запустив изолированную базу данных PostgreSQL и подключившись к ней извне.

## Часть 0: Установка и проверка Docker

### 1 Постановка задачи

При подготовке к лабораторной работе необходимо сделать ряд действий:

1. Установить Docker Desktop;

2. Проверить через терминал.

### 2 Проверка Docker

Была проведена проверка установки Docker с помощью ряда команд, что показано на рисунках 1 и 2.



Рисунок 1 - Проверка установки Docker

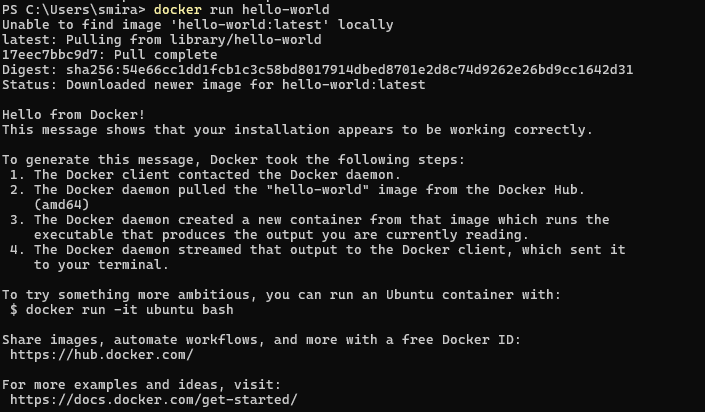


Рисунок 2 - Проверка установки Docker

## Часть 1: Базовые команды Docker. Работа с образами и контейнерами

### 1 Постановка задачи

В ходе части 1 лабораторной работы необходимо выполнить:

1. Просмотр информации;

2. Запуск простого контейнера (на примере Nginx);

3. Проверка работоспособности контейнера;

4. Остановка и удаление контейнера.

### 2 Выполнение базовых команд

Были выполнены базовые команды, что показано на рисунках 3.

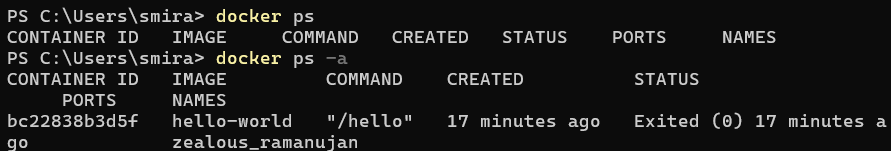
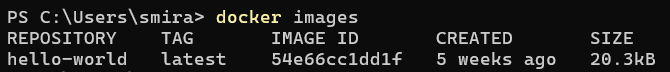


Рисунок 3 - Выполнение базовых команд

### 3 Запуск простого контейнера

Запуск простого контейнера показан на рисунках 4. Проверка работы контейнера показана на рисунке 5. Остановка и удаление контейнера показана на рисунке 6.

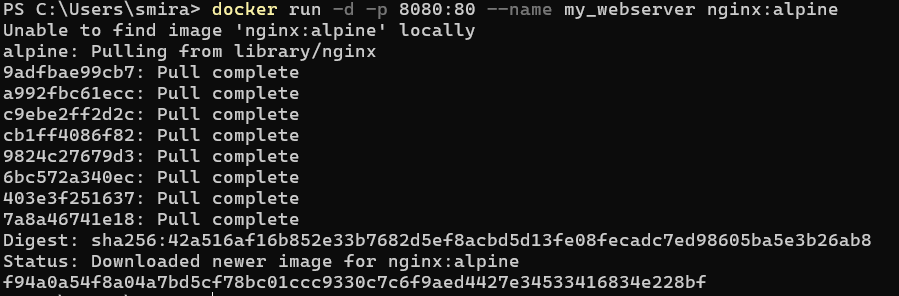


Рисунок 4 - Запуск простого контейнера



Рисунок 5 - Проверка работы контейнера

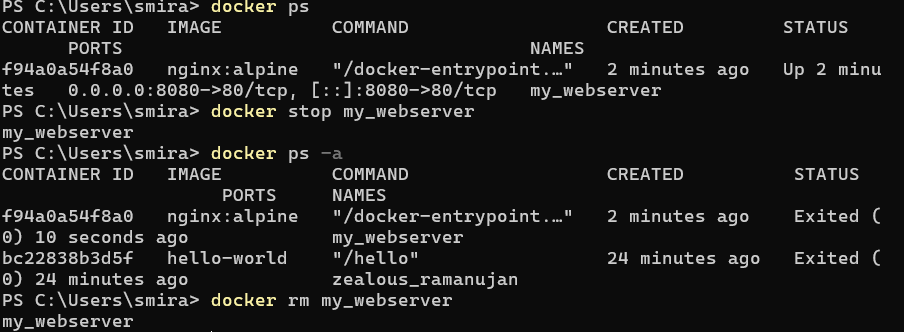


Рисунок 6 - Остановка и удаление контейнера

## Часть 2: Запуск PostgreSQL в контейнере

### 1 Постановка задачи

В ходе части 2 лабораторной работы необходимо выполнить:

1. Запустить контейнер с PostgreSQL;

2. Проверить, что контейнер запущен и слушает порт;

3. Подключиться к БД прямо из контейнера (через psql);

4. Выполнение тестовых запросов.

### 2 Запуск PostgreSQL

Был выполнен запуск контейнера с PostgreSQL, что показано на рисунках 7. Проверка, что контейнер запущен и слушает порт показана на рисунке 8. Подключение к БД прямо из контейнера (через psql) показано на рисунке 9. На рисунках 10-13 показано выполнение тестовых запросов.

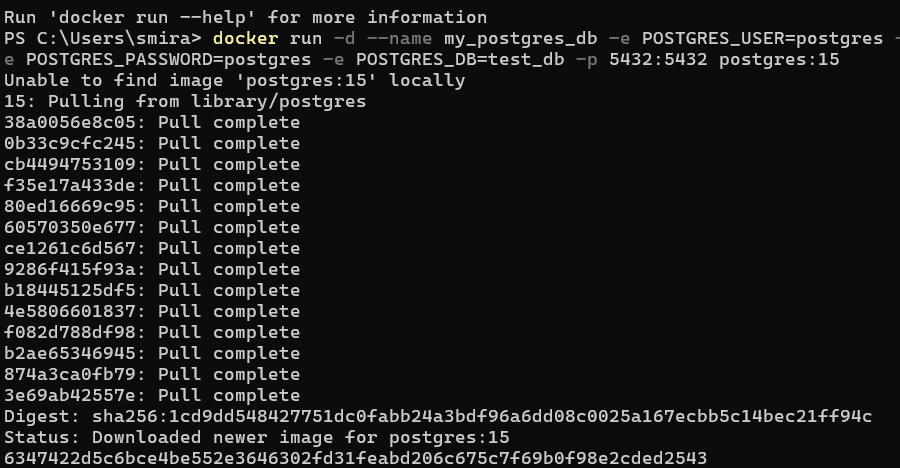


Рисунок 7 - Запуск контейнера с PostgreSQL с помощью команды

docker exec -it my\_postgres\_db psql -U postgres -d test\_db



Рисунок 8 - Проверка, что контейнер запущен и слушает порт

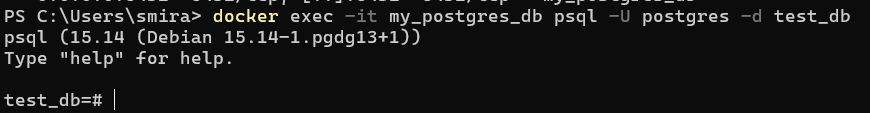


Рисунок 9 - Подключение к БД прямо из контейнера (через psql)

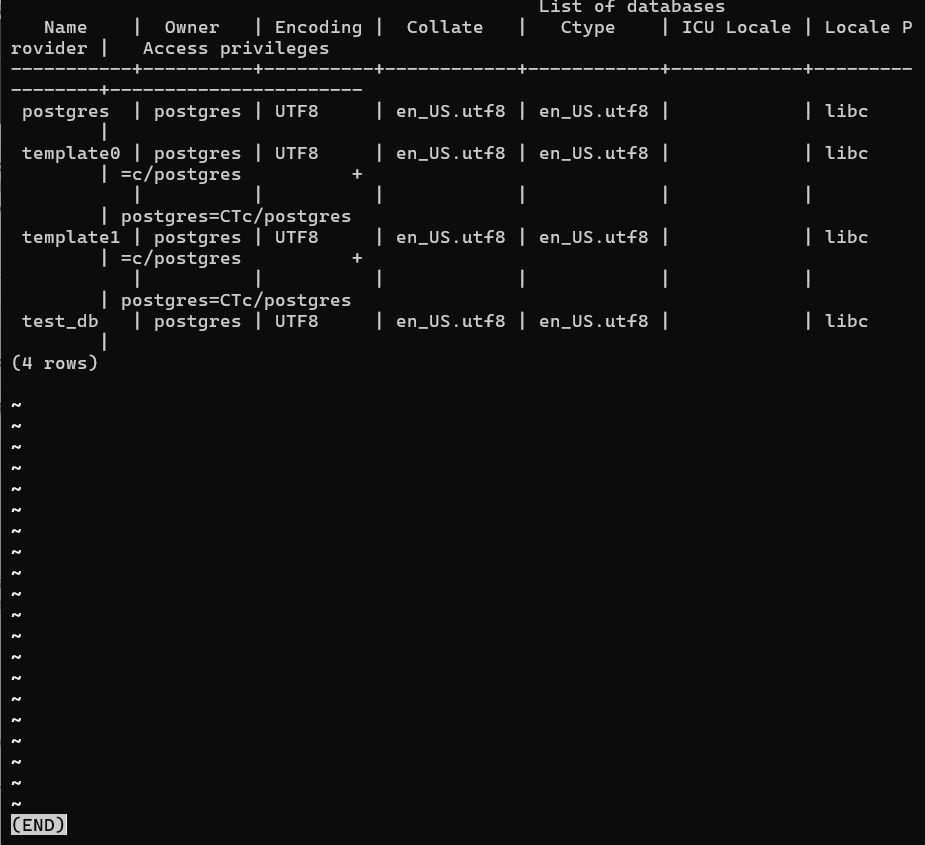


Рисунок 10 - \l список всех баз данных



Рисунок 11 - список таблиц в текущей базе (сейчас пусто)

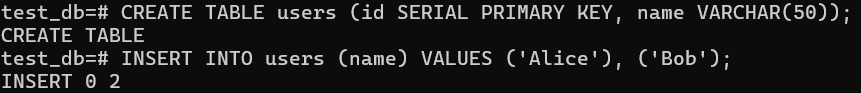


Рисунок 12 - Создание таблицы users, вставка данных

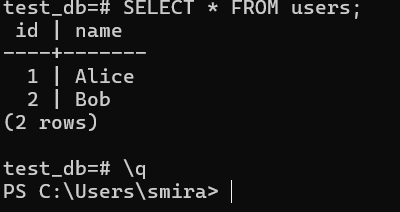


Рисунок 13 - Выбор всех строк из users и выход из консоли psql

## Часть 3: Подключение к БД через pgAdmin из второго контейнера

### 1 Постановка задачи

В ходе части 3 лабораторной работы необходимо выполнить:

1. Создание сети Docker;

2. Подключение контейнера с PostgreSQL к сети;

3. Запуск pgAdmin в той же сети;

4. Настройка подключения в pgAdmin с добавлением нового сервера и выполнение запроса через Query Tool в pgAdmin.

### 2 Подключение к БД через pgAdmin из второго контейнера

Создание сети Docker показано на рисунках 14. Подключение контейнера с PostgreSQL к сети показано на рисунке 15. Запуск pgAdmin в той же сети показан на рисунке 16. Настройка подключения в pgAdmin с добавлением нового сервера показана на рисунке 17. Настройка подключения с выполнением запроса через Query Tool в pgAdmin показана на рисунке 18.

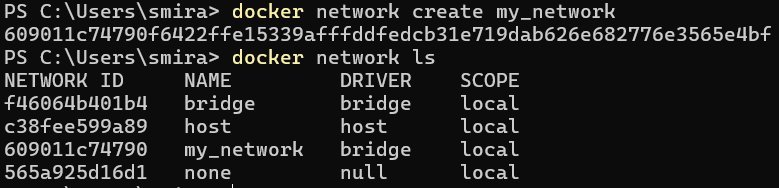


Рисунок 14 - Создание сети Docker



Рисунок 15 - Подключение контейнера с PostgreSQL к сети

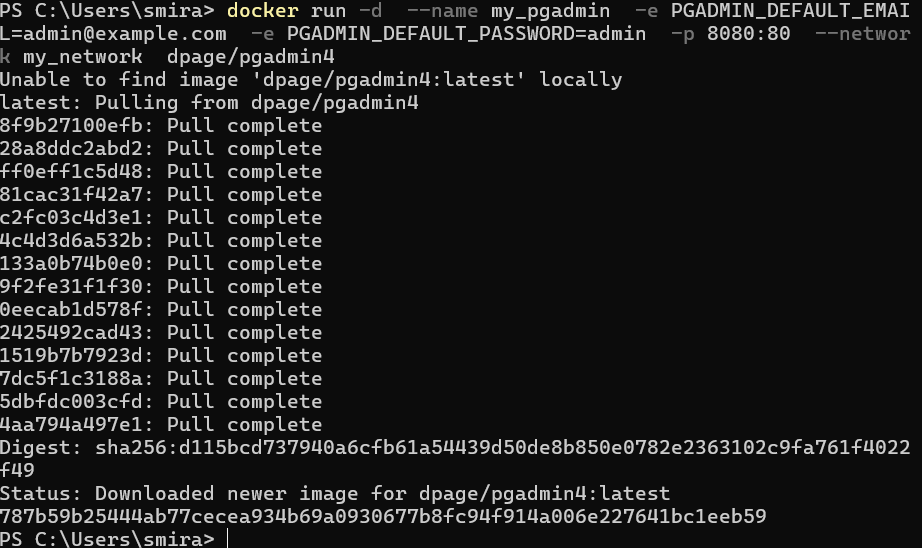


Рисунок 16 - Запуск pgAdmin в той же сети с помощью команды

docker run -d --name my\_pgadmin -e PGADMIN\_DEFAULT\_EMAIL=admin@example.com -e PGADMIN\_DEFAULT\_PASSWORD=admin -p 8080:80 --network my\_network dpage/pgadmin4

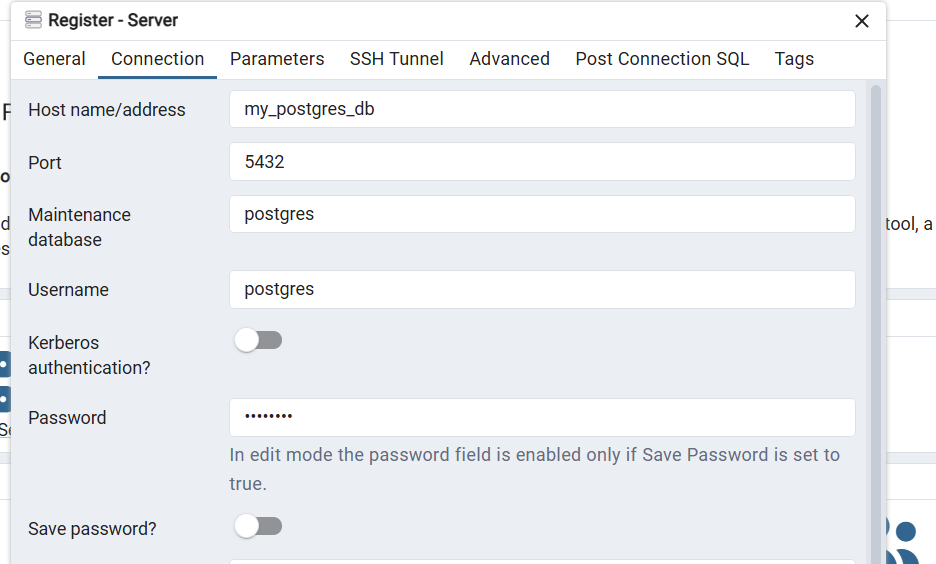


Рисунок 17 - Настройка подключения в pgAdmin с добавлением нового сервера

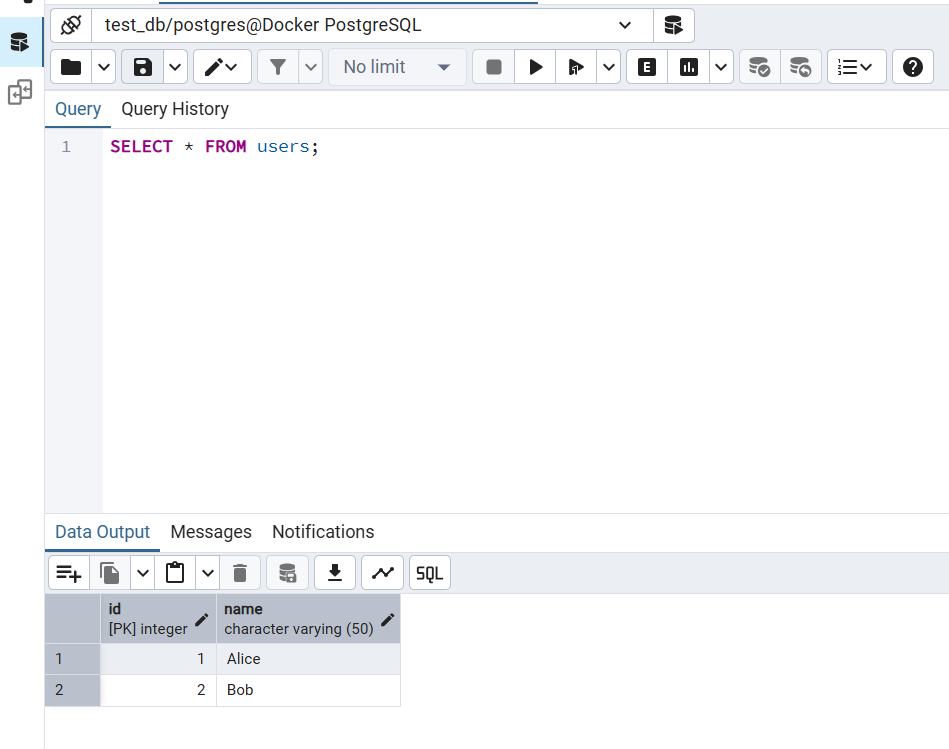


Рисунок 18 - Настройка подключения с выполнением запроса через Query Tool в pgAdmin

## Часть 4: Сохранение данных с помощью Томов (Volumes)

### 1 Постановка задачи

В ходе части 4 лабораторной работы необходимо выполнить:

1. Остановка и удаление текущего контейнера с БД;

2. Создание тома для хранения данных БД;

3. Запуск нового контейнера с PostgreSQL, подключив том;

4. Проверка сохранности данных.

### 2 Сохранение данных с помощью томов

Остановка и удаление текущего контейнера с БД показано на рисунках 19. Создание тома для хранения данных БД показано на рисунке 20. Запуск нового контейнера с PostgreSQL, подключив том показан на рисунке 21. Проверка сохранности данных показана на рисунке 22. Добавление таблицы и данных показана на рисунке 23. Проверка таблицы users показана на рисунке 24.

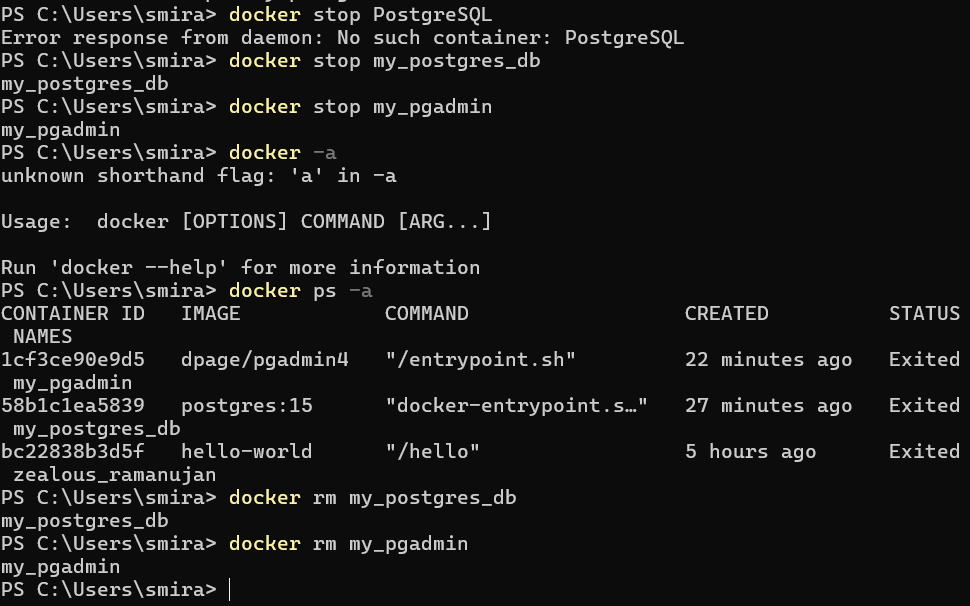


Рисунок 19 - Остановка и удаление текущего контейнера с БД

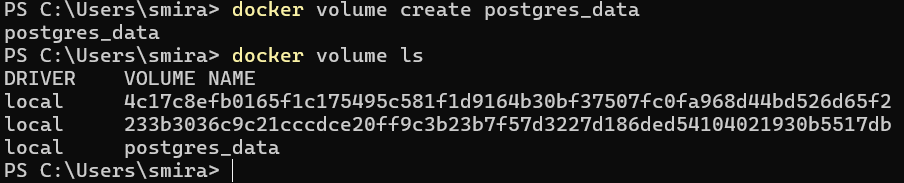


Рисунок 20 - Остановка и удаление текущего контейнера с БД

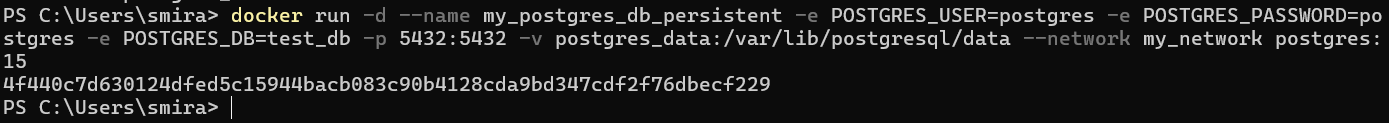


Рисунок 21 - Запуск нового контейнера с PostgreSQL, подключив том

docker run -d --name my\_postgres\_db\_persistent -e POSTGRES\_USER=postgres -e POSTGRES\_PASSWORD=postgres -e POSTGRES\_DB=test\_db -p 5432:5432 -v postgres\_data:/var/lib/postgresql/data --network my\_network postgres:15



Рисунок 22 - Проверка сохранности данных

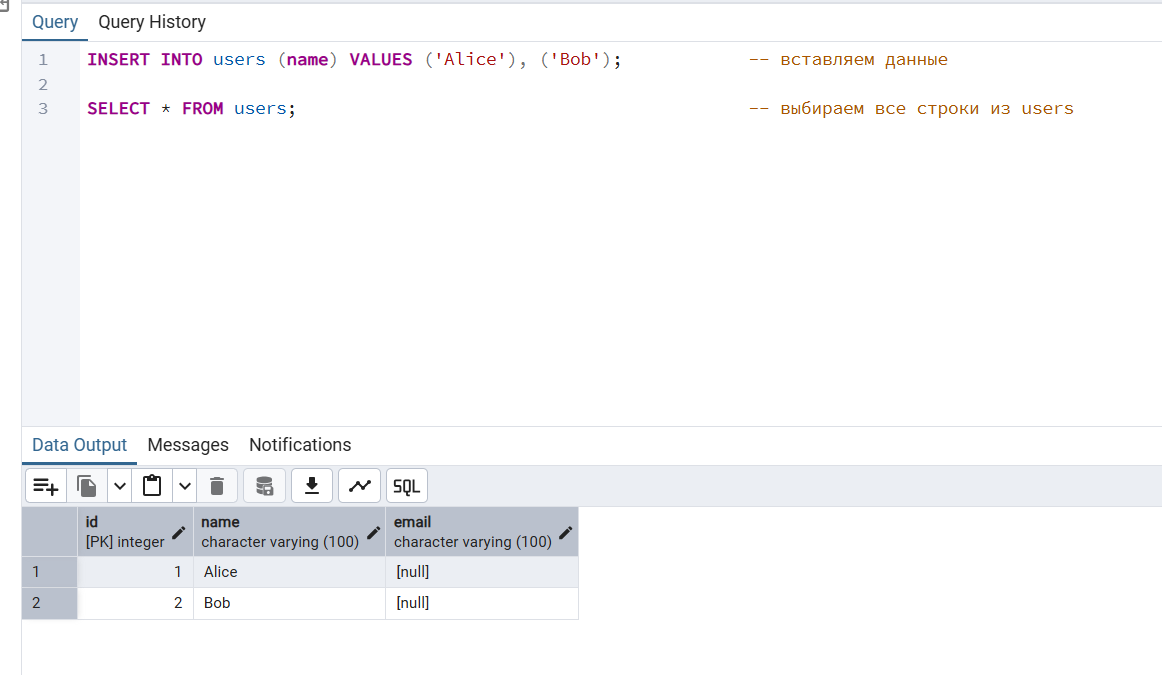


Рисунок 23 - Добавление таблицы и данных

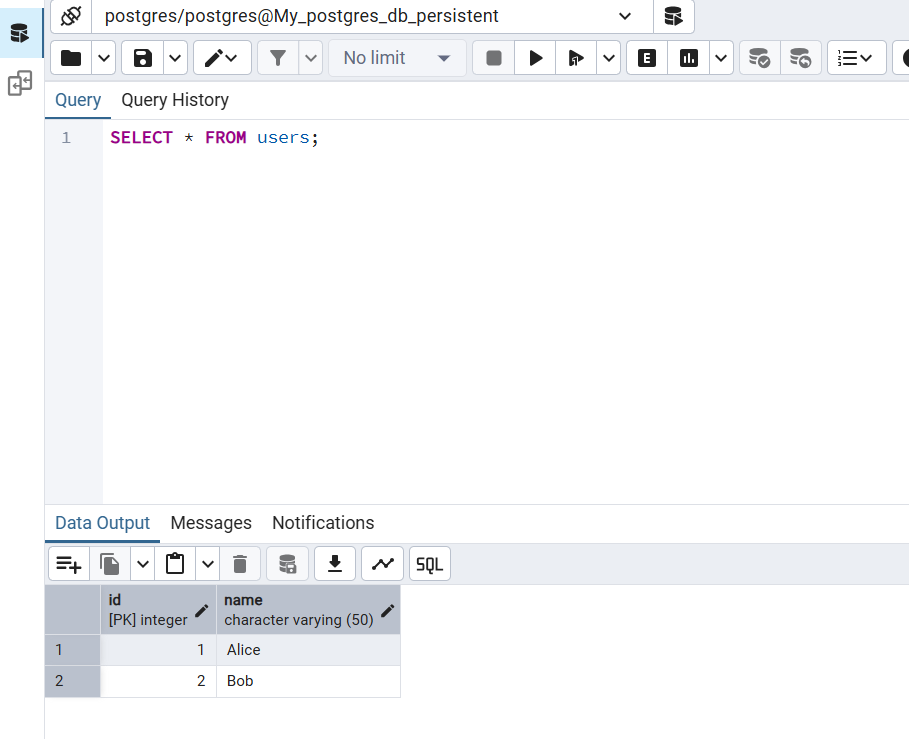


Рисунок 24 – Проверка таблицы users

## Часть 5: Перенос конфигурации контейнеров в docker-compose.yaml

### 1 Постановка задачи

В ходе части 5 лабораторной работы необходимо выполнить:

1. Настройка docker-compose.yaml.

### 2 Перенос конфигурации контейнеров

Был создан файл docker-compose.yaml. Далее была проверка работоспособности данных из файла, показано на рисунке 25.

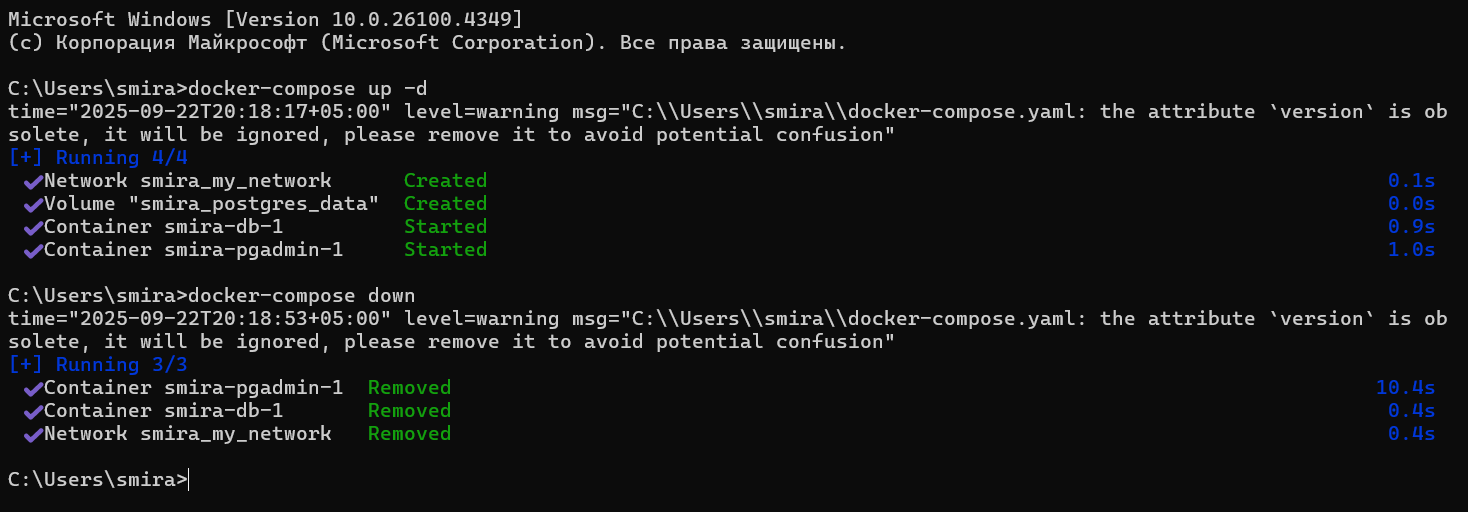


Рисунок 25 - Проверка работоспособности данных из файла

## ВЫВОД

В ходе работы была проведена работа по установке и проверке работоспособности системы контейнеризации Docker. Были выполнены базовые команды в системе Docker.

Был запущен PostgreSQL в контейнере, а также было проведено подключение к базе данных через pgAdmin из второго контейнера.

Данные были сохранены с помощью томов (Volumes) и далее был произведен перенос конфигурации контейнеров в docker-compose.yaml.

**1. Что такое Docker**

Docker – это платформа для создания, распространения и запуска приложений внутри контейнеров. Контейнеры позволяют упаковать программу со всеми её зависимостями в изолированную среду, чтобы запускать её одинаково на разных компьютерах и серверах.

**2. Для чего нужны тома и сети Docker**

Тома (volumes) – это специально выделенные области для постоянного хранения данных контейнеров вне их файловой системы. Это позволяет сохранять данные даже при удалении контейнера. Используются для базы данных, файлов и других важных данных.

Сети (networks) – обеспечивают связь между контейнерами в изолированной виртуальной сети. Помогают контейнерам "видеть" друг друга и обмениваться данными безопасно и эффективно.

**3. Как подключиться к контейнеру и выполнить в нём команды**

Для подключения к контейнеру используется команда docker network connect my\_network my\_postgres\_db или docker exec -it my\_postgres\_db psql -U postgres -d test\_db.

Далее можно можно выполнить следующие команды:

- \l – выводит список всех доступных баз данных.

- \dt – показывает таблицы, которые есть в текущей базе данных (если база пустая, таблиц нет).

- CREATE TABLE users (id SERIAL PRIMARY KEY, name VARCHAR(50)); – создаёт таблицу users с полями:

- id – уникальный идентификатор с автоматическим увеличением (SERIAL), первичный ключ.

- name – поле для имени, текст до 50 символов.

- INSERT INTO users (name) VALUES ('Alice'), ('Bob'); – вставляет две записи в таблицу с именами Alice и Bob.

- SELECT \* FROM users; – выводит все записи из таблицы users.

- \q – выход из интерактивного режима psql.

**4. Для чего нужен pgAdmin**

pgAdmin – это графический веб-интерфейс для управления и администрирования PostgreSQL. Он позволяет создавать базы данных, выполнять SQL-запросы, просматривать структуру и содержимое таблиц удобным и наглядным способом без использования командной строки.