Цель работы: создать изолированную, переносимую, легко управляемую и воспроизводимую среду для запуска и работы с СУБД PostgreSQL.

**Установка Docker**

Для удобства работы мы будем устанавливать СУБД PostgreSQL в качестве Docker-контейнера. Это позволит при необходимости оперативно его переустановить, а также он не разворачивается в основной ОС.

Для того, чтобы установить Docker, перейдите по ссылке и загрузите версию для своей ОС и процессора: <https://www.docker.com/products/docker-desktop/>

Запустите скачанный установщик и следуйте инструкциям.

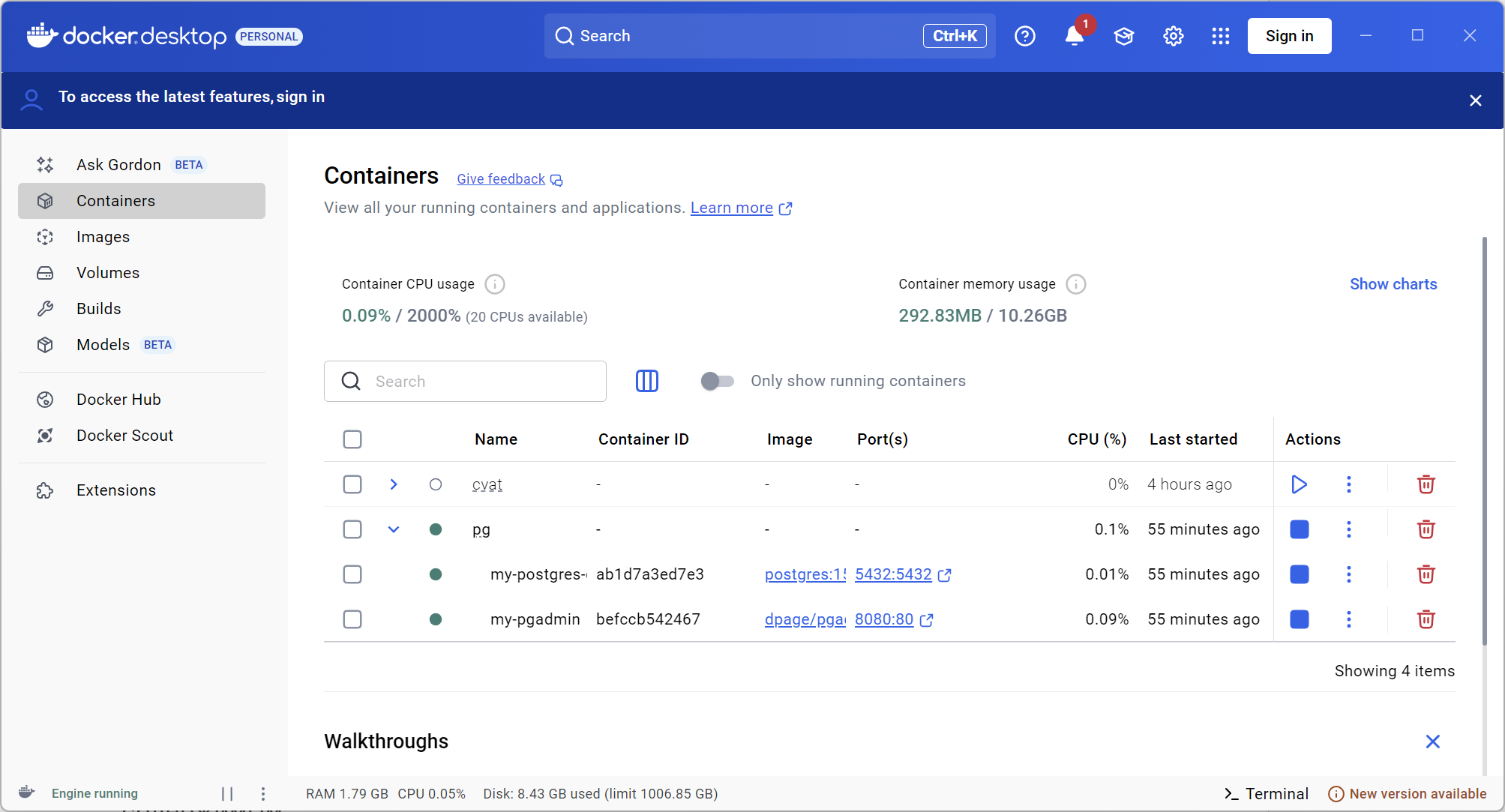


Рисунок 0. UI среды Docker Desktop. Для работы мы будем использовать его консольные средства

После установки Docker **обязательно перезагрузите компьютер**

**Установка PostreSQL**

Теперь приступим к установке контейнера с СУБД. Выберите каталог, в котором будут располагаться загружаемые файлы. Желательно для этого создать каталог в корне диска C:\

Перейдите в данный каталог. Создайте файл **docker-compose.yml** со следующим содержимым:

version: '3.8'

services:

*# Сервис для СУБД PostgreSQL*

postgres:

image: postgres:15 *# Можно указать нужную версию, например, postgres:16*

container\_name: my-postgres-db

restart: unless-stopped

environment:

POSTGRES\_USER: admin *# Имя пользователя для БД (замените на своё)*

POSTGRES\_PASSWORD: secretpassword *# Пароль пользователя (ЗАМЕНИТЕ на сложный!)*

POSTGRES\_DB: mydatabase *# Имя базы данных, создаваемой по умолчанию*

volumes:

- postgres\_data:/var/lib/postgresql/data *# Сохраняем данные БД на диск*

ports:

- "5432:5432" *# Пробрасываем порт 5432 контейнера на порт 5432 локальной машины*

networks:

- postgres-network

*# Сервис для веб-интерфейса управления pgAdmin*

pgadmin:

image: dpage/pgadmin4

container\_name: my-pgadmin

restart: unless-stopped

environment:

PGADMIN\_DEFAULT\_EMAIL: admin@example.com *# Email для входа в pgAdmin (замените на свой)*

PGADMIN\_DEFAULT\_PASSWORD: adminpassword *# Пароль для входа в pgAdmin (ЗАМЕНИТЕ!)*

ports:

- "8080:80" *# Веб-интерфейс pgAdmin будет доступен на localhost:8080*

volumes:

- pgadmin\_data:/var/lib/pgadmin *# Сохраняем настройки pgAdmin*

networks:

- postgres-network

depends\_on:

- postgres

*# Объявляем named volumes для постоянного хранения данных*

volumes:

postgres\_data:

pgadmin\_data:

*# Создаем сеть для удобного взаимодействия контейнеров*

networks:

postgres-network:

Запустите командную строку. Для этого откройте окно **«Выполнить»** (Win+R) и введите команду cmd. Или откройте служебное меню (Win+X) и выберите **«Терминал»** или **«PowerShell»**

Перейдите в каталог, в котором Вы ранее сохранили файл **docker-compose.yml**, при помощи команды **cd**

Например:

**PS C:\Users\Andrey>** cd D:\Docker\pg

И после этого запустите установку контейнера при помощи команды:

**PS C:\Users\Andrey>** docker-compose up -d

Если все прошло успешно, Docker напишет, что контейнеры успешно запущены

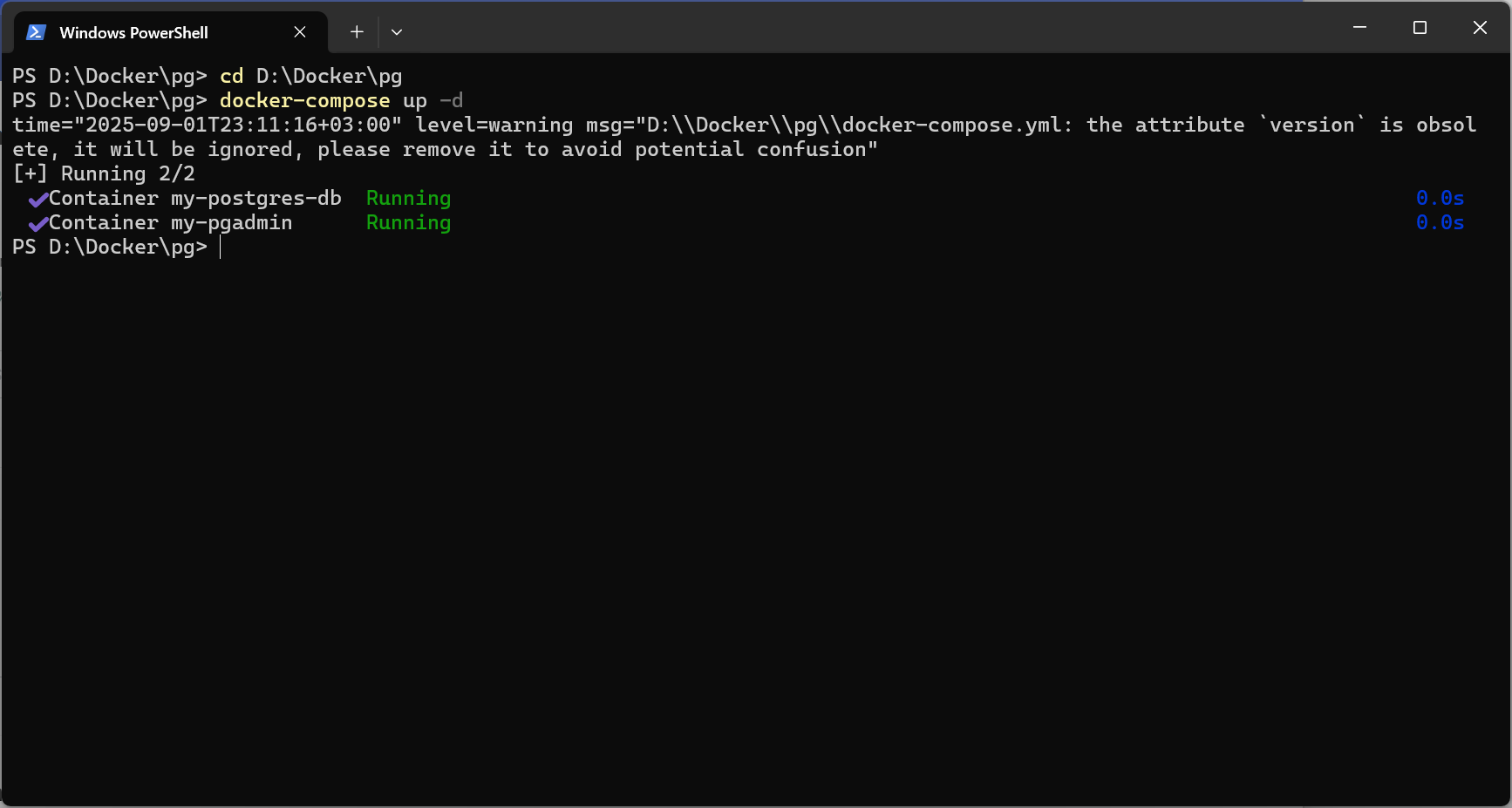


Рисунок 1. Результат запуска контейнера с PostgreSQL

После выполненных операций в СУБД PostgreSQL будет доступна по локальному порту 5432 (этот порт является значением по умолчанию, но в Docker-Compose можно указать любой другой). По умолчанию, Docker допускает работу с портом только на локальном устройстве. Из внешней сети получить доступ к этому порту не получится.

Работать с PostreSQL можно при помощи консоли, визуального редактора таблиц (например, DBeaver), через веб-интерфейс программы PgAdmin или напрямую из разрабатываемой программы, например через Entity Framework.

**Подключение с помощью DBeaver**

Для этого загрузите программу DBeaver по ссылке: <https://dbeaver.io/download/>

Запустите программу

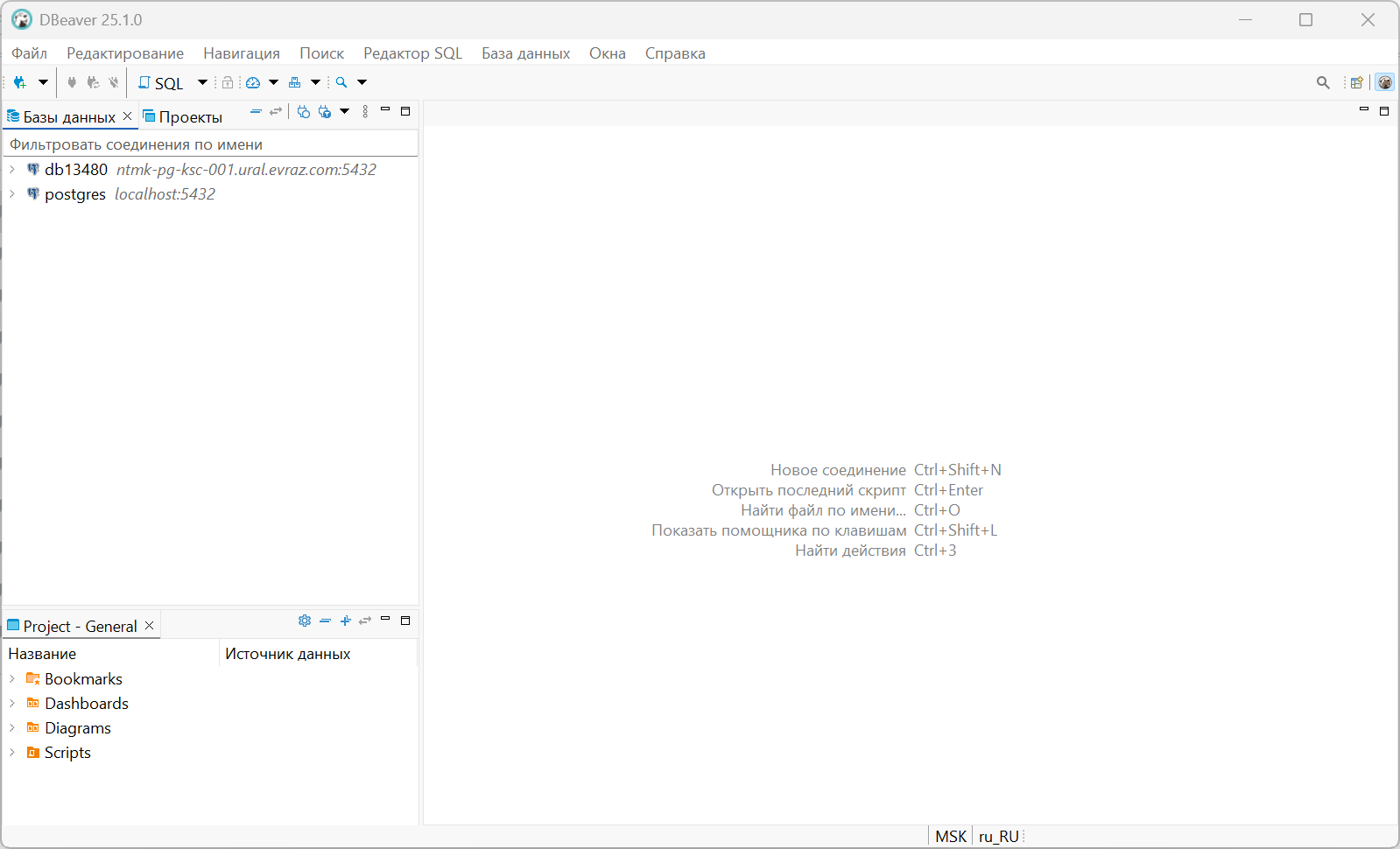


Рисунок 2. Интерфейс программы

Создайте новое подключение к СУБД. (Ctrl+Shift+N). Выберите пункт PostgreSQL

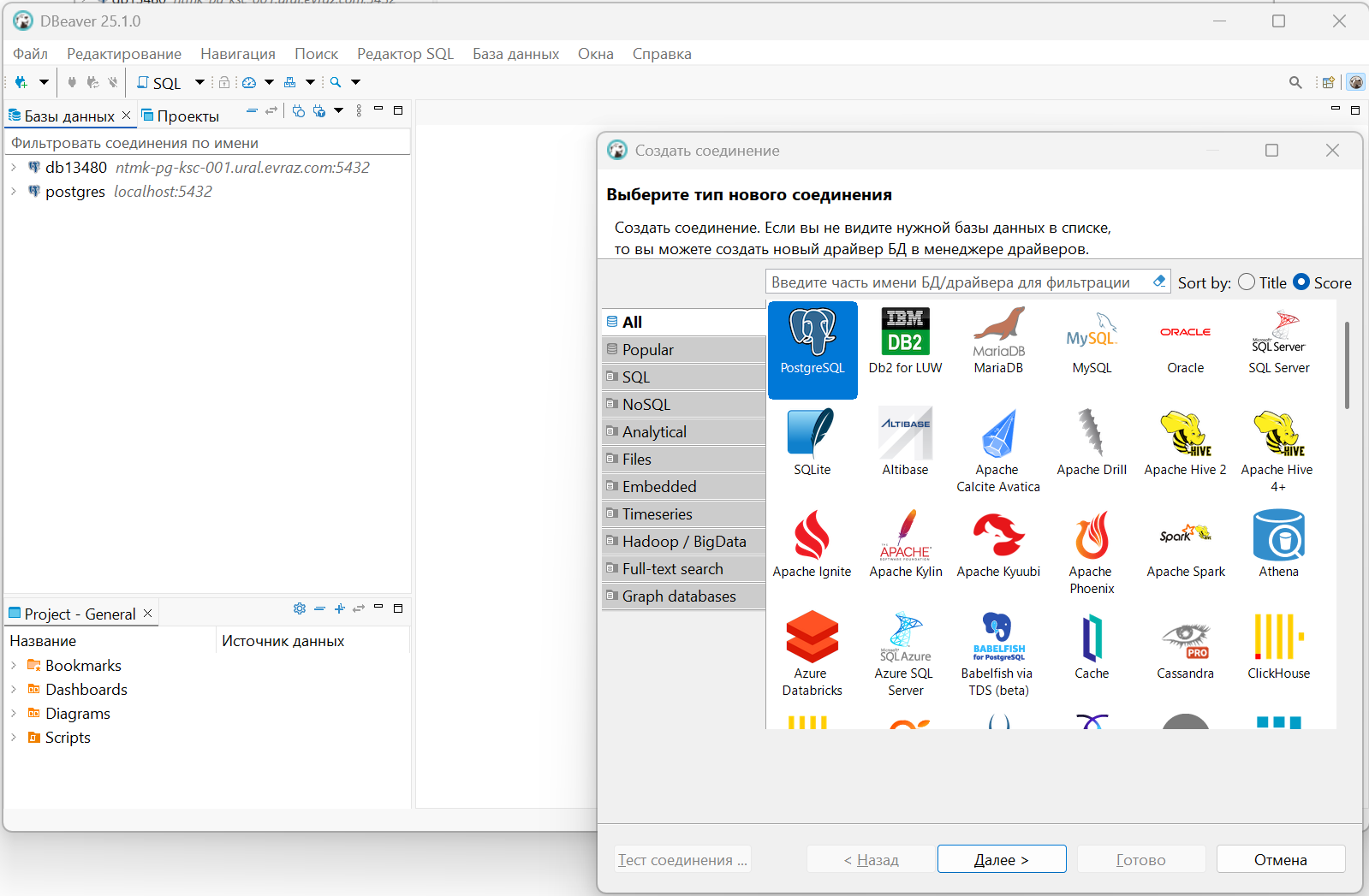


Рисунок 3. Выбор типа СУБД

После выбора типа СУБД, DBeaver автоматически скачает необходимые драйверы для данного адаптера, после чего откроется окно настройки подключения.

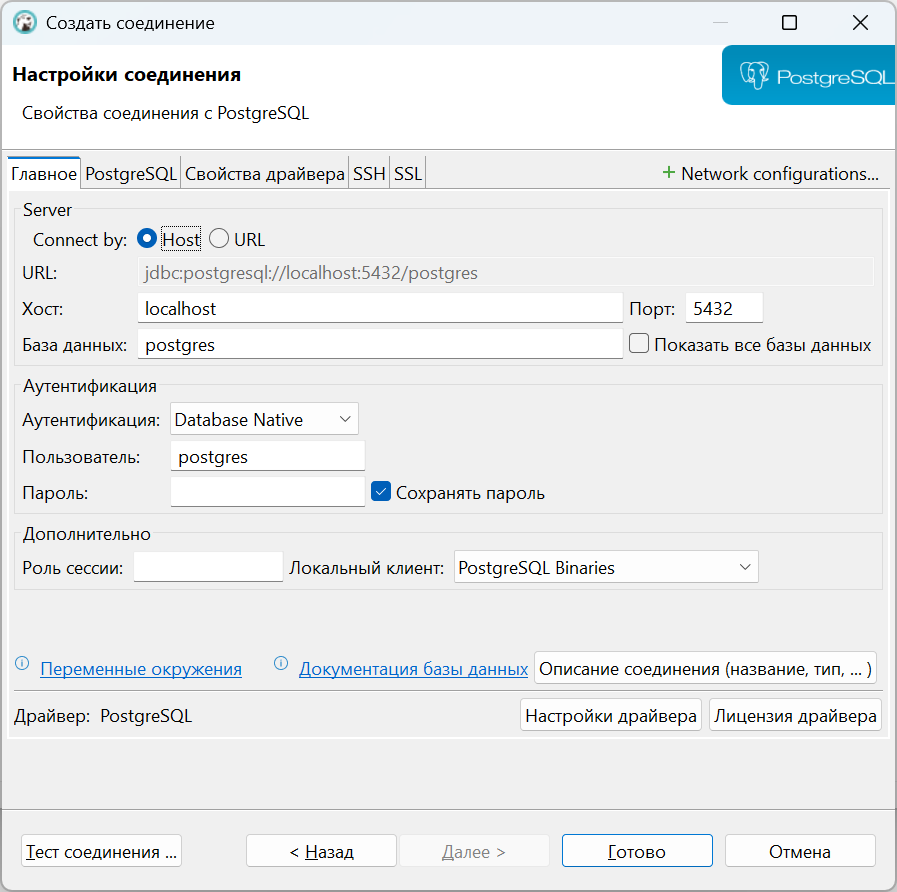


Рисунок 4. Окно настройки подключения

**Хост** – имя сетевого устройства, на котором развернут сервер СУБД. В нашем случае Docker будет имитировать развертывание СУБД на локальном устройстве, поэтому можно оставить **localhost или 127.0.0.1**

**Порт** – значение по умолчанию – **5432** или то, которые Вы указали в Docker

**База данных** – имя БД, к которой будет осуществляться подключение. Выберите пункт «Показать все базы данных», если ещё не создавали ни одной БД.

**Режим аутентификации** – Database Native.

**Пользователь и пароль** – имя пользователя и пароль из файла docker-compose.yaml

Остальные настройки оставьте по умолчанию. Нажмите кнопку «Готово» для создания соединения. После успешного соединения сервер СУБД появится в левом верхнем окне.

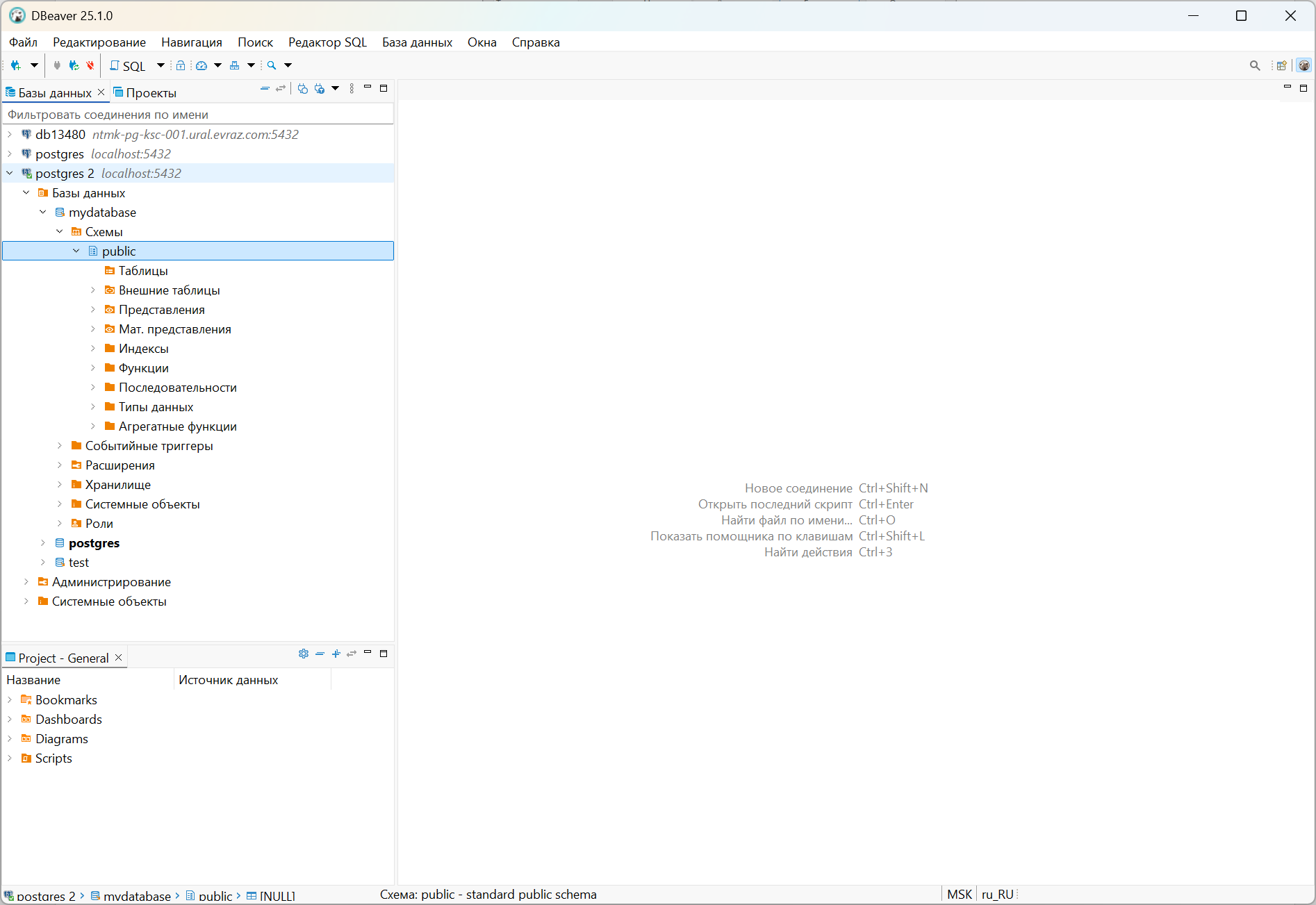


Рисунок 5. Окно просмотра компонентов БД

**Подключение с помощью веб-интерфейса**

Перейдите в **веб-браузере** по адресу <http://localhost:8080/> (или по другому, если меняли номер порта в файле **docker-compose.yaml**). Откроется страница входа в веб-интерфейс **PgAdmin**. Введите логин и пароль из файла **docker-compose.yaml**

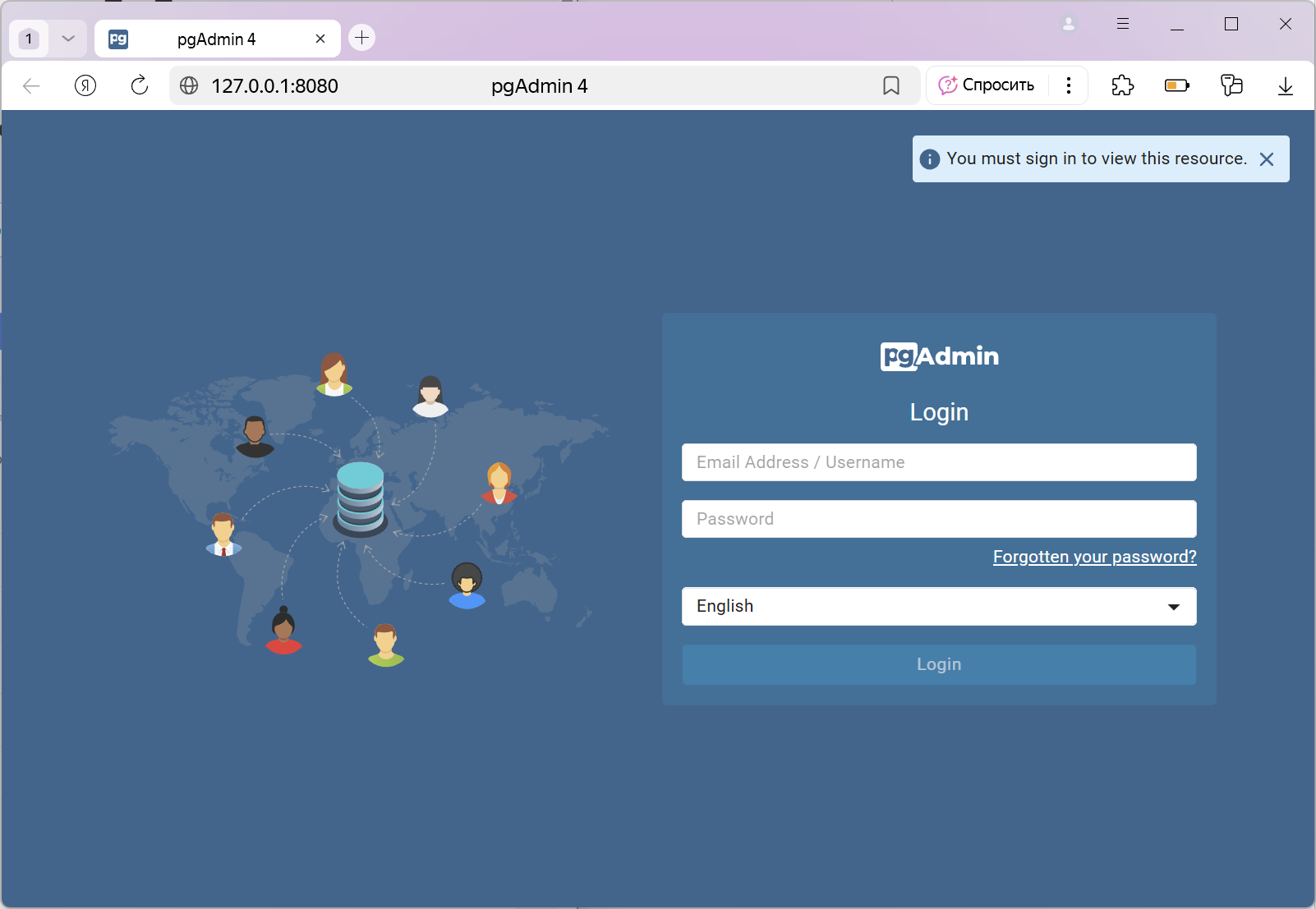


Рисунок 6. Окно входа в систему

После успешного входа откроется панель управления серверами. Выберите пункт **«Добавить новый сервер»**

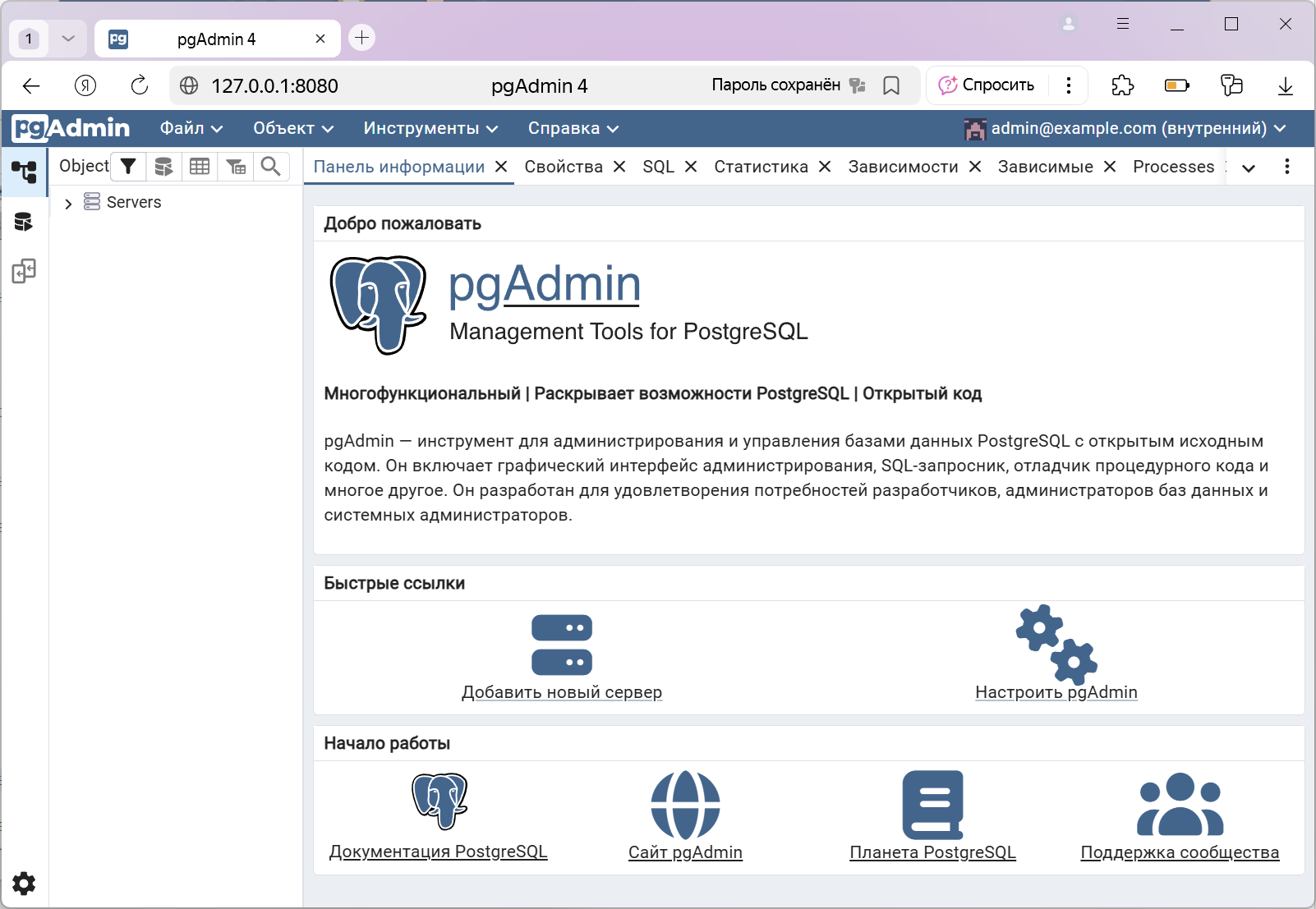


Рисунок 7. Окно управления серверами

Потребуется заполнить поля во вкладках Общие и Соединение.

В разделе **Общие – Имя** укажите любое отображаемое имя для сервера.

В разделе **Соединение – Имя сервера** укажите **my-postgres-db** (название контейнера из docker-compose.yaml).

**ВАЖНО!** PostreSQL и PgAdmin физически располагаются в разных контейнерах, поэтому они имеют разные внутренние IP-адреса и логически разделены между собой. Поэтому для получения доступа к контейнеру с СУБД необходимо прописать или его внутренний IP-адрес или имя контейнера (в нашем случае мы используем второй путь).

**Логин и пароль** – также из настроек, которые Вы указали в файле **docker-compose.yaml**

Остальные поля заполняются значениями по умолчанию. Нажмите **Сохранить**. Если все заполнено правильно, в окне слева появится подключение к БД.

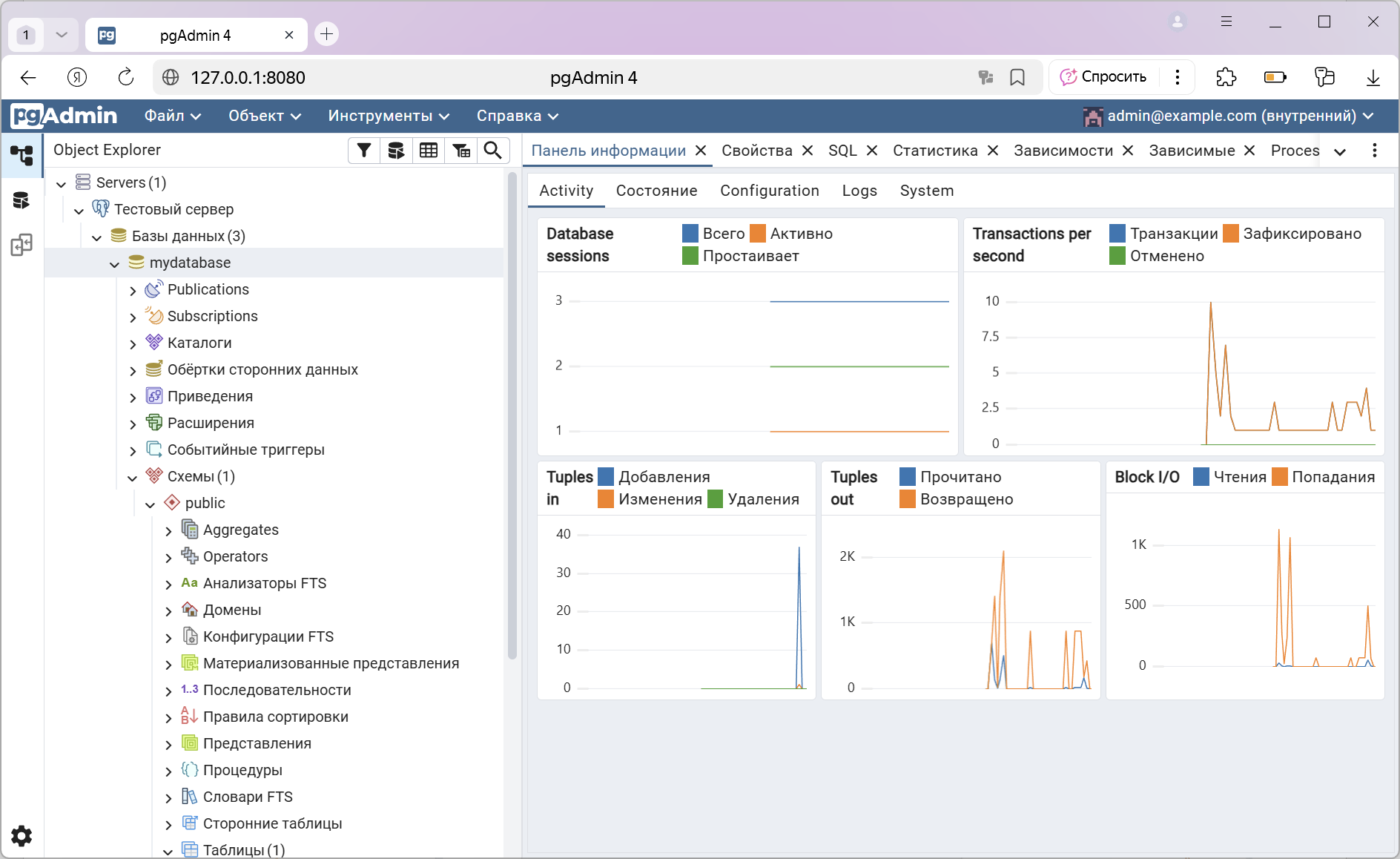


Рисунок 8. Окно мониторинга работы СУБД

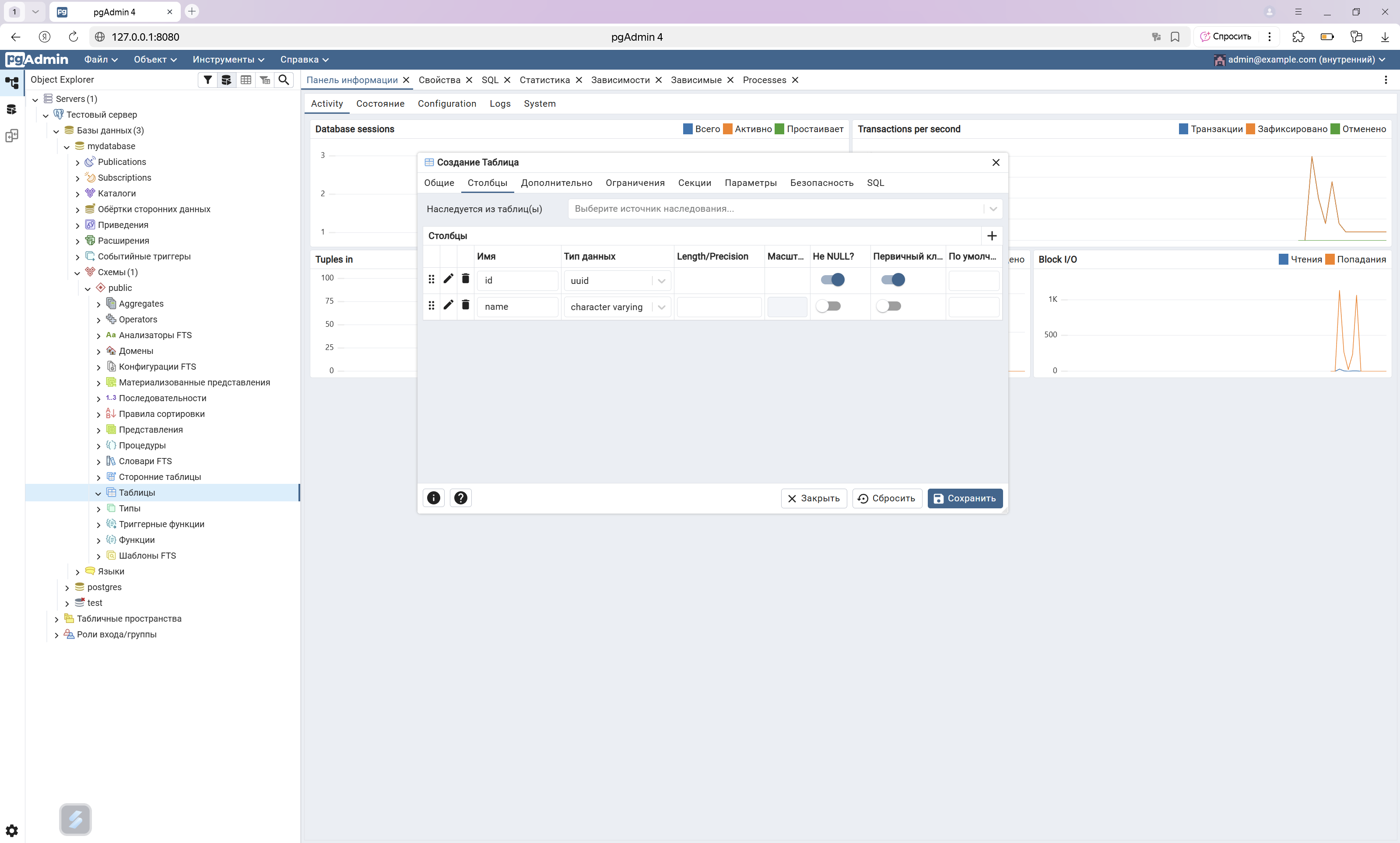


Рисунок 9. Пример создания таблицы в PgAdmin