

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Направление подготовки 11.03.02

Практическая работа №1

Установка и первоначальная настройка СУБД PostgreSQL. Создание и
исследование баз данных Demo и HR

Выполнил:

Доценников Никита Андреевич

Группа: К3221

Проверила:

Татьяна Евгеньевна Войтюк

Санкт-Петербург

2025

Цель работы:

Освоить базовые приёмы установки и настройки СУБД PostgreSQL, научиться использовать pgAdmin для администрирования, развернуть учебную базу данных Demo и ознакомиться с её структурой, а также закрепить навыки создания баз данных и схем на примере учебной БД HR.

Задачи, решаемые при выполнении работы.

- Установить и настроить СУБД PostgreSQL и клиент pgAdmin.
- Проверить корректность работы сервера с помощью выполнения тестового SQL-запроса.
- Развернуть учебную базу данных Demo из предоставленного SQL-дампа и убедиться в доступности её объектов.
- Ознакомиться со структурой базы данных Demo и проанализировать основные сущности.
- Создать учебную базу данных HR двумя способами: через графический интерфейс pgAdmin и с помощью SQL-оператора `CREATE DATABASE`.
- Создать в базе данных HR схемы для группировки объектов (через GUI и SQL).

Исходные данные.

- Сервер с установленными Docker и контейнерами:
 - postgres:16 — сервер СУБД PostgreSQL,
 - dpage/pgadmin4 — веб-интерфейс для администрирования.
- Учебный SQL-дамп базы данных Demo (`demo-medium-20170815.sql`), предоставленный разработчиками Postgres Pro.
- Методические материалы по практической работе №1.
- Доступ к веб-интерфейсу pgAdmin 4 для подключения и выполнения SQL-запросов.

Выполнение работы.

Задание 1. Установка необходимого ПО.

Для выполнения работы была установлена система управления базами данных PostgreSQL и клиентское приложение pgAdmin. Установка производилась в среде Docker.

- На сервере был установлен Docker и запущены контейнеры:

- ▶ `postgres:16` - сервер СУБД PostgreSQL,
- ▶ `dpape/pgadmin4` - веб-интерфейс pgAdmin для администрирования.

```
root@r980743:~# docker ps
```

CONTAINER ID	IMAGE NAMES	COMMAND	CREATED	STATUS	PORTS
3094ac4b9a23	dpape/pgadmin4:8 pgadmin	"/entrypoint.sh"	19 hours ago	Up 19 hours	443/tcp, 127.0.0.1:8888→80/tcp
ec415114f4d7	postgres:16 postgres	"docker-entrypoint.s..."	19 hours ago	Up 19 hours (healthy)	127.0.0.1:5432→5432/tcp

- Контейнеры были объединены в общую сеть Docker, чтобы pgAdmin мог напрямую подключаться к Postgres.

```
root@r980743:~# docker network inspect dbnet
[
  {
    "Name": "dbnet",
    "Id": "8d62db233b593ccfd61d93a73946d4bdf9188b9d2288e8f39285c62f3592ef82",
    "Created": "2025-09-23T17:17:25.408657713Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.23.0.0/16",
          "Gateway": "172.23.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "3094ac4b9a23e2557f2496c274d727040fc5aa4eeb8b1686e8ede9818e617fd4": {
        "Name": "pgadmin",
        "EndpointID": "1707ced2ae7b442ebdf9cccb6162a86d105305693b2deb19fc0994821840e84f",
        "MacAddress": "02:42:ac:17:00:03",
        "IPv4Address": "172.23.0.3/16",
        "IPv6Address": ""
      },
      "ec415114f4d7be443192c5ec81cc67a89f38dcb3c6dcff686f90f4ccd788b16b": {
        "Name": "postgres",
        "EndpointID": "ecaa8951afe60e34569b2dedd7d84a7e3c365283f3cf0b54fed708bf6bf5a6a8",
        "MacAddress": "02:42:ac:17:00:02",
        "IPv4Address": "172.23.0.2/16",
        "IPv6Address": ""
      }
    },
    "Options": {},
    "Labels": {}
  }
]
```

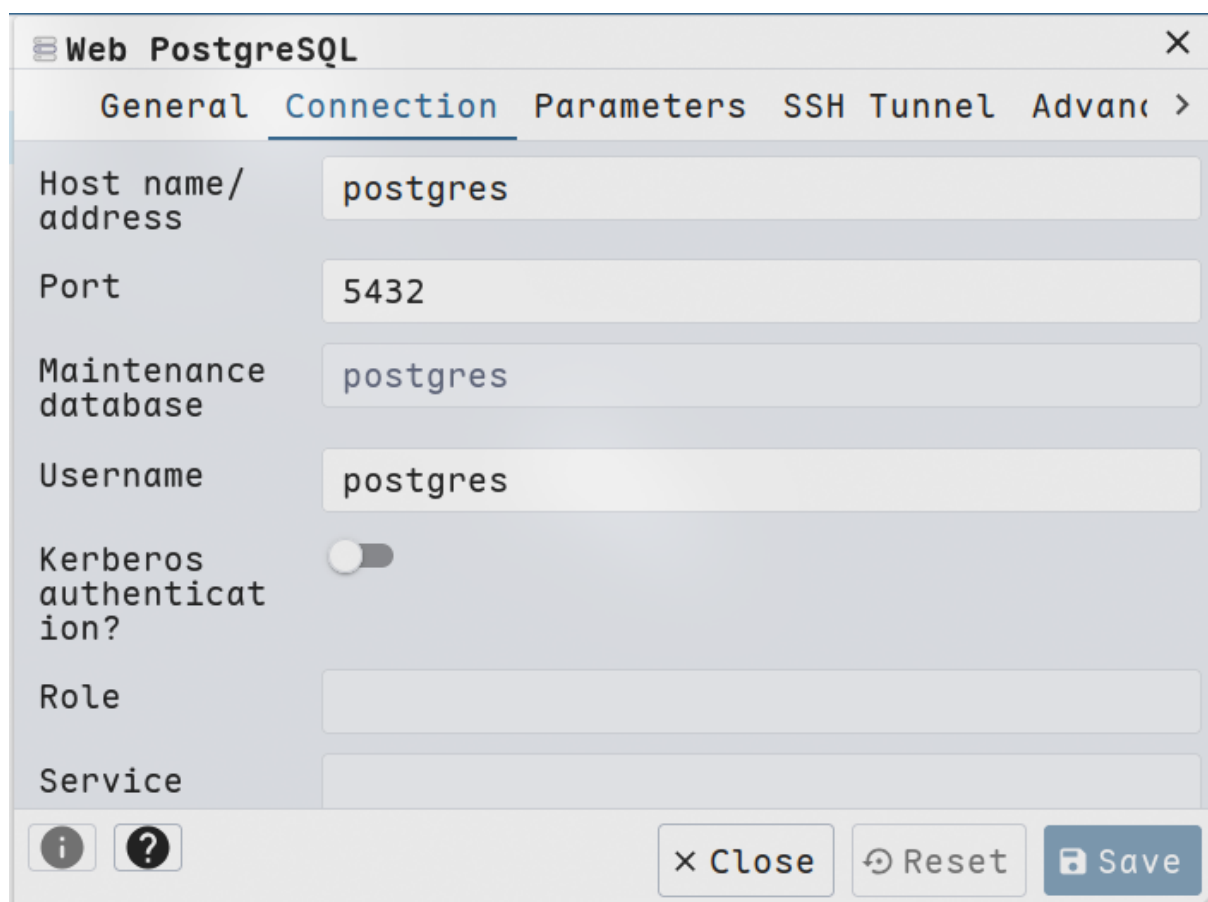
- pgAdmin 4 стал доступен по адресу db.fymio.us

```
db.fymio.us {
  reverse_proxy 127.0.0.1:8888
  encode gzip
  header {
    Strict-Transport-Security max-age=31536000;
    X-Content-Type-Options nosniff
    X-Frame-Options SAMEORIGIN
  }
  tls {
    protocols tls1.2 tls1.3
  }
}
root@r980743:~#
```

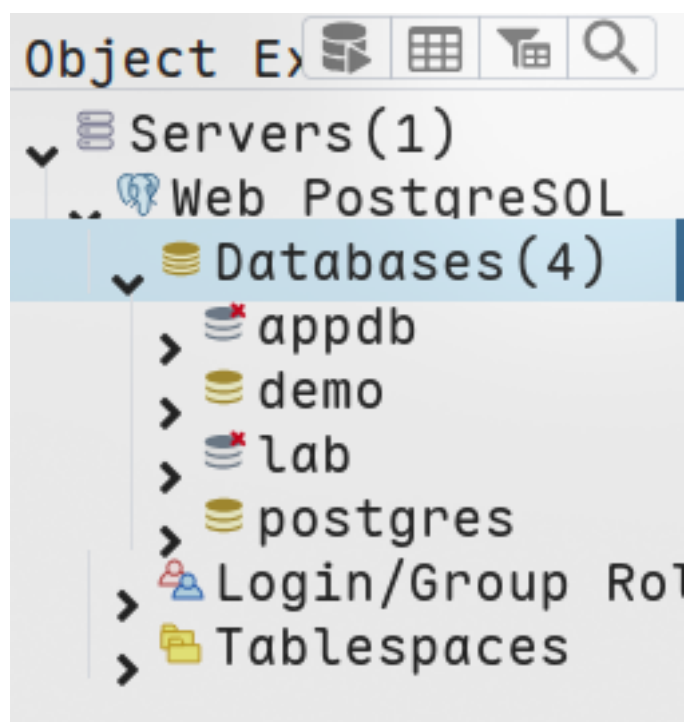
- В pgAdmin был зарегистрирован новый сервер. В настройках подключения были указаны следующие параметры:

The screenshot shows the 'Web PostgreSQL' configuration window in pgAdmin. The 'General' tab is selected, showing fields for Name, Server group, Background, Foreground, Shared?, Shared Username, and Comments. The Name field contains 'Web PostgreSQL', the Server group is set to 'Servers', and the Shared? toggle is turned off. At the bottom, there are buttons for 'Close', 'Reset', and 'Save'.

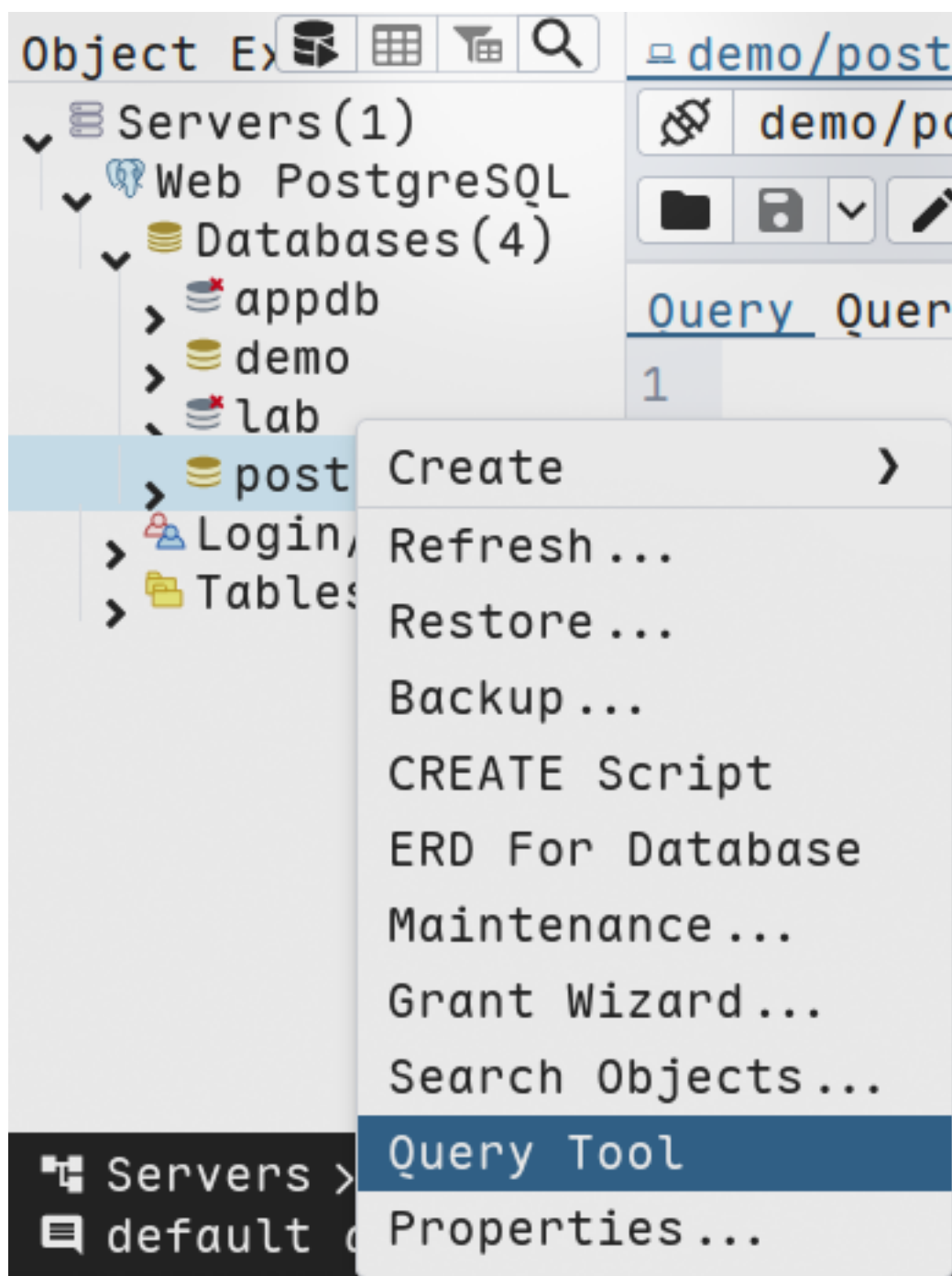
Field	Value
Name	Web PostgreSQL
Server group	Servers
Background	☐
Foreground	☐
Shared?	☐
Shared Username	
Comments	



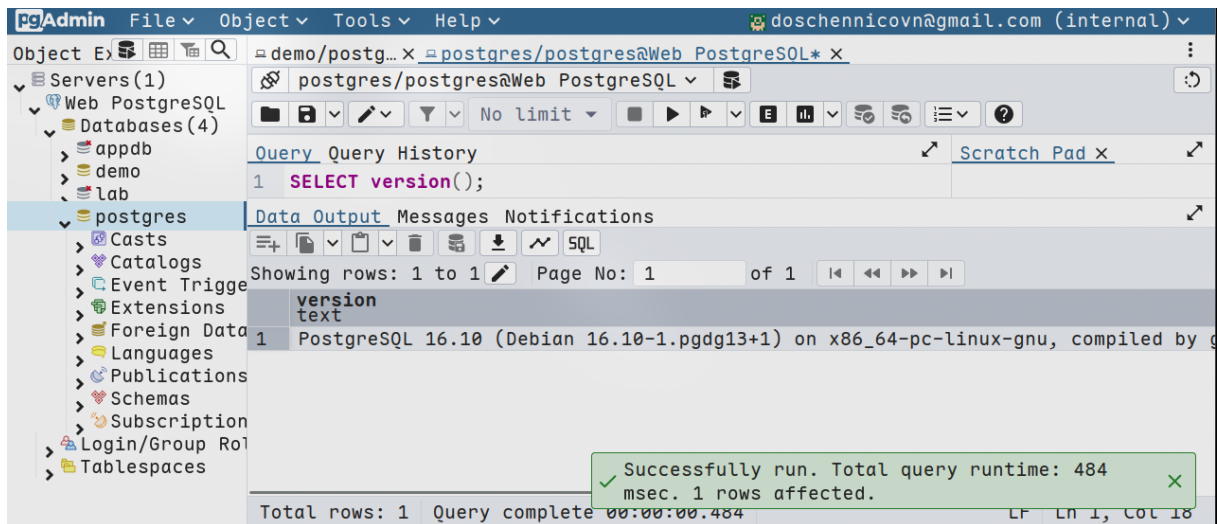
В “Object Explorer” слева я раскрыл узел “Databases”:



Затем я нажал ПКМ по базе “postgres” и выбрал инструмент “Query Tool”:



В открывшемся окне написал запрос проверки версии сервера `SELECT version();`. После этого нажал кнопку “Execute script”.



В поле “Data Output” я увидел версию PostgreSQL 16.10.

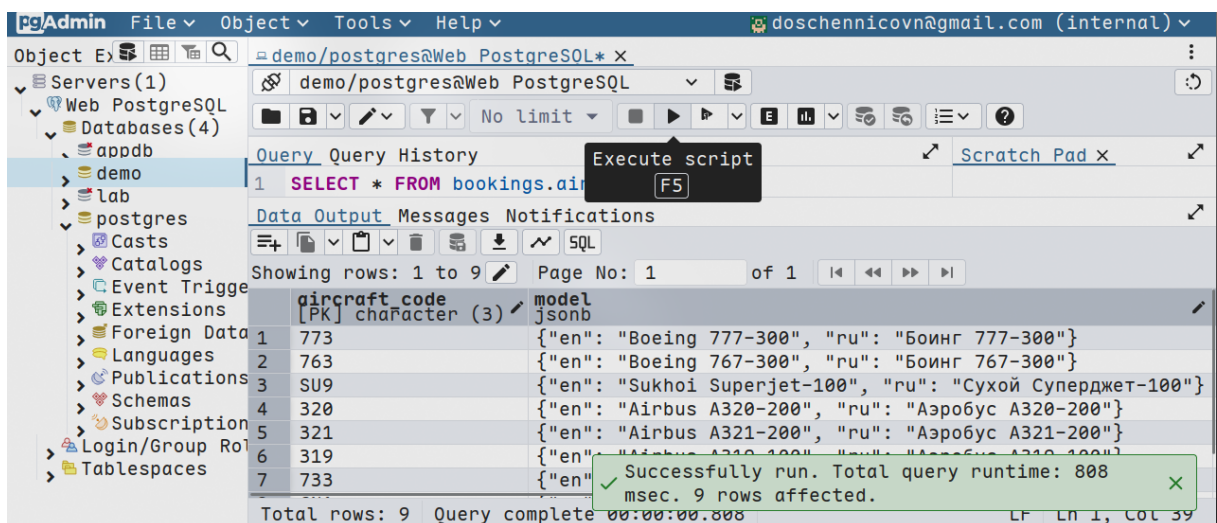
Задание 2. Подключение учебной базы данных.

После выполнения следующих команд на сервере учебная база данных появилась в “Object Explorer”.

```
wget https://edu.postgrespro.ru/demo-medium.zip
unzip demo-medium.zip
docker exec -i postgres psql -U postgres -d postgres < demo-medium-20170815.sql
```

После появления базы данных “demo”, в “Query Tool” я выполнил следующий скрипт:

```
SELECT * FROM bookings.aircrafts_data;
```



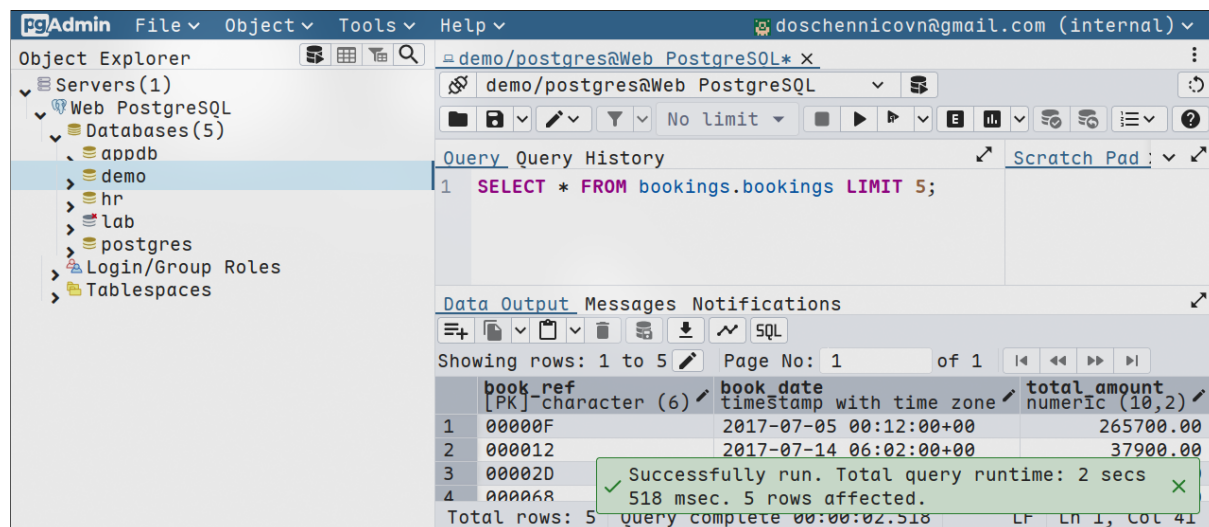
Задание 3. Ознакомление с учебной БД.

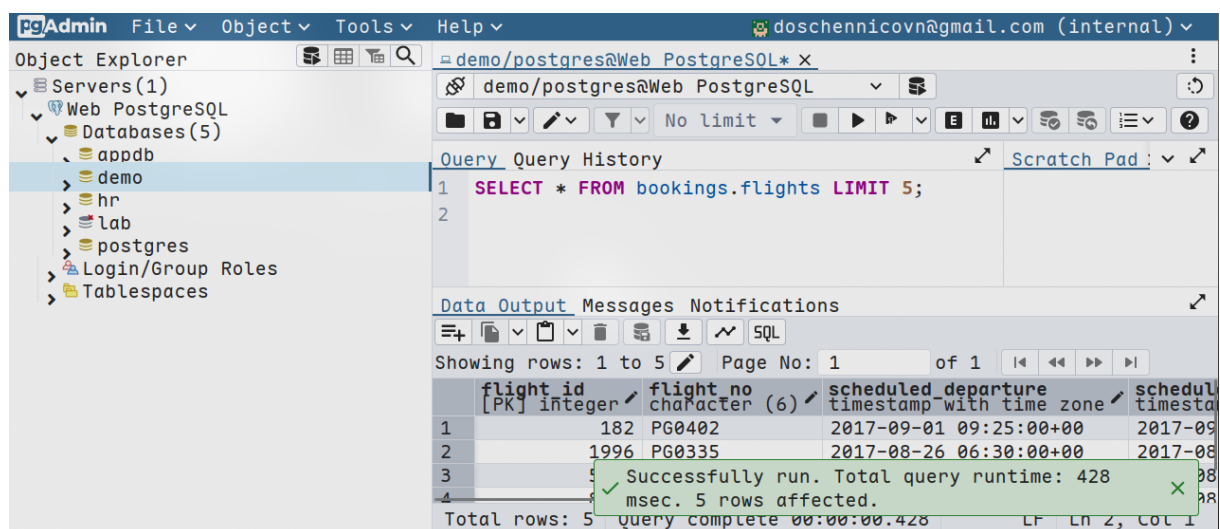
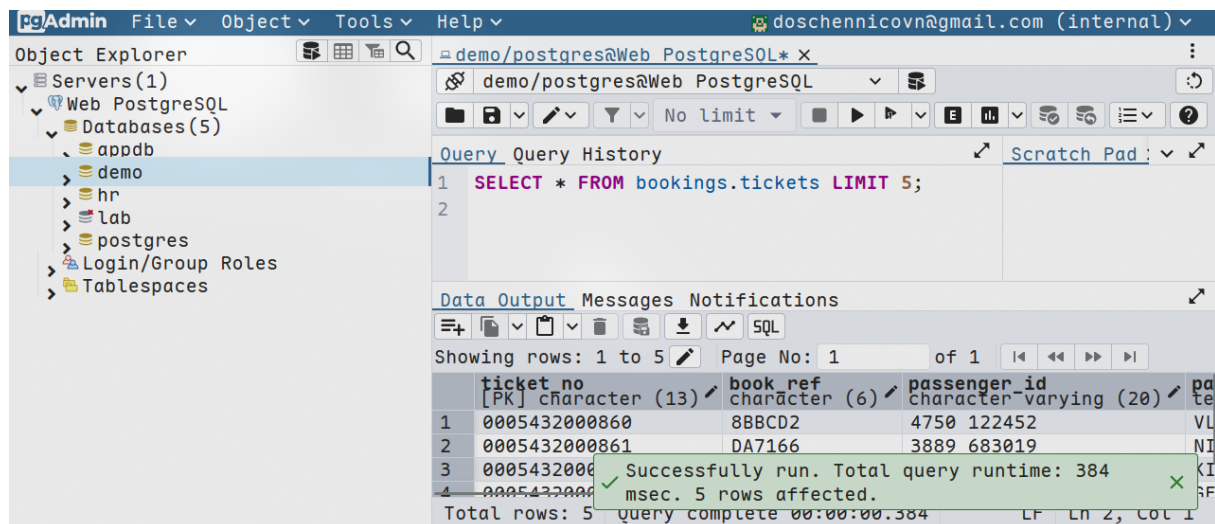
Я открыл в pgAdmin дерево объектов и изучил структуру. В схеме booking находятся основные таблицы:

Таблица	Описание
bookings	содержит данные о бронированиях
tickets	хранит информацию о билетах и пассажирах
flights	сведения о рейсах
airports	справочник аэропортов
aircrafts	справочник моделей самолетов
seats	информация о местах в самолётах и классах обслуживания
ticket_flights	связь билетов и рейсов
boarding_passes	посадочные талоны с указанием мест в самолёте

Для проверки содержимого таблиц я выполнил несколько тестовых запросов, например:

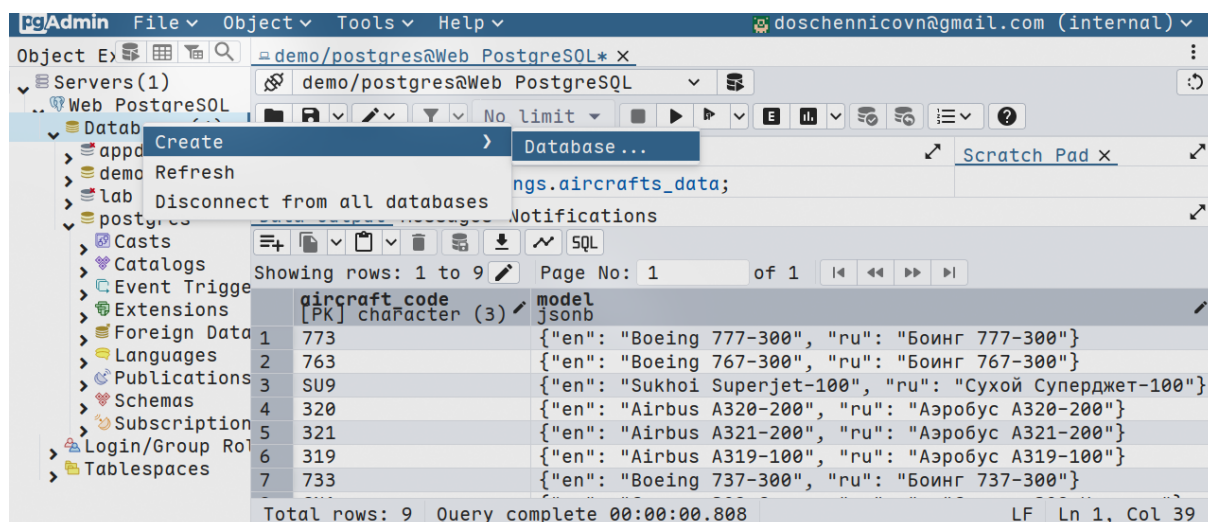
```
SELECT * FROM bookings.bookings LIMIT 5;  
SELECT * FROM bookings.tickets LIMIT 5;  
SELECT * FROM bookings.flights LIMIT 5;
```



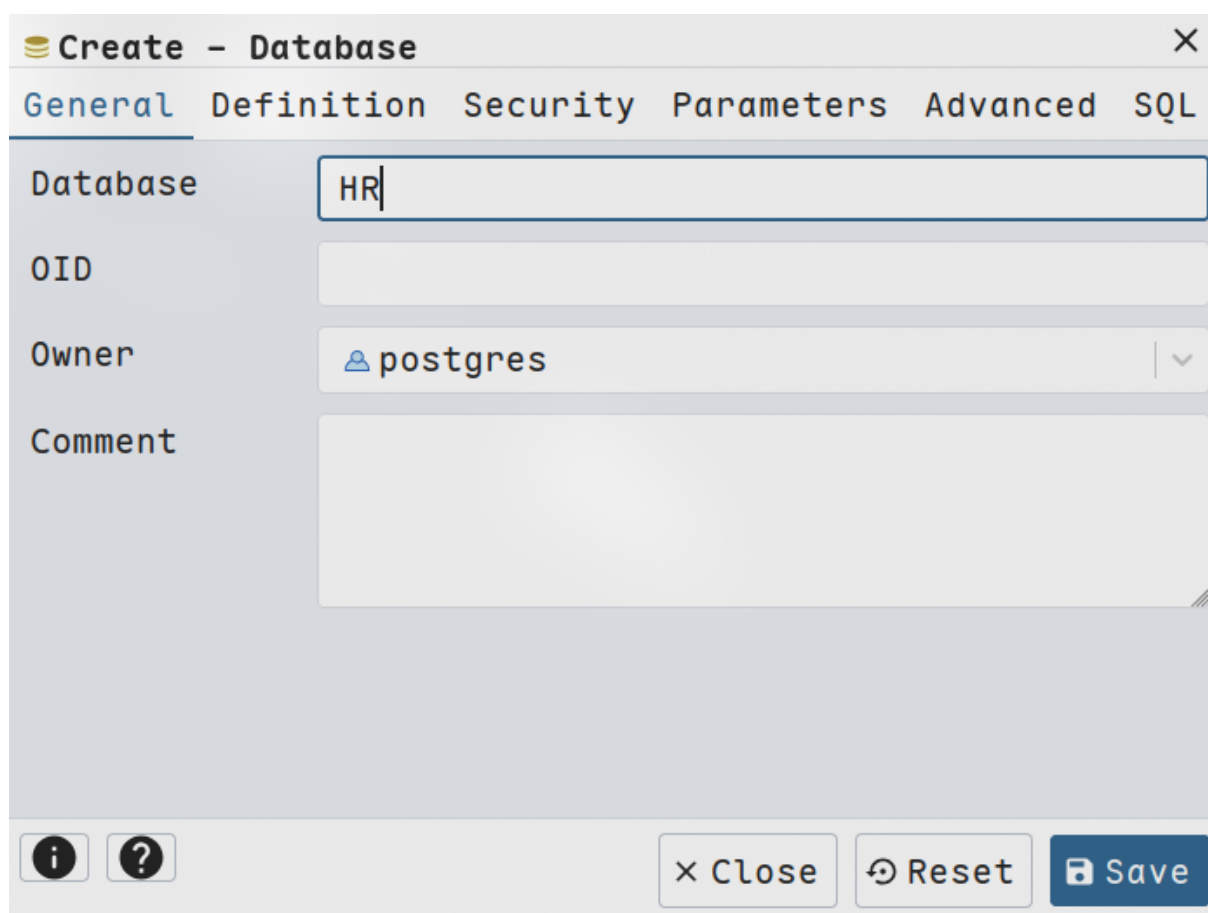


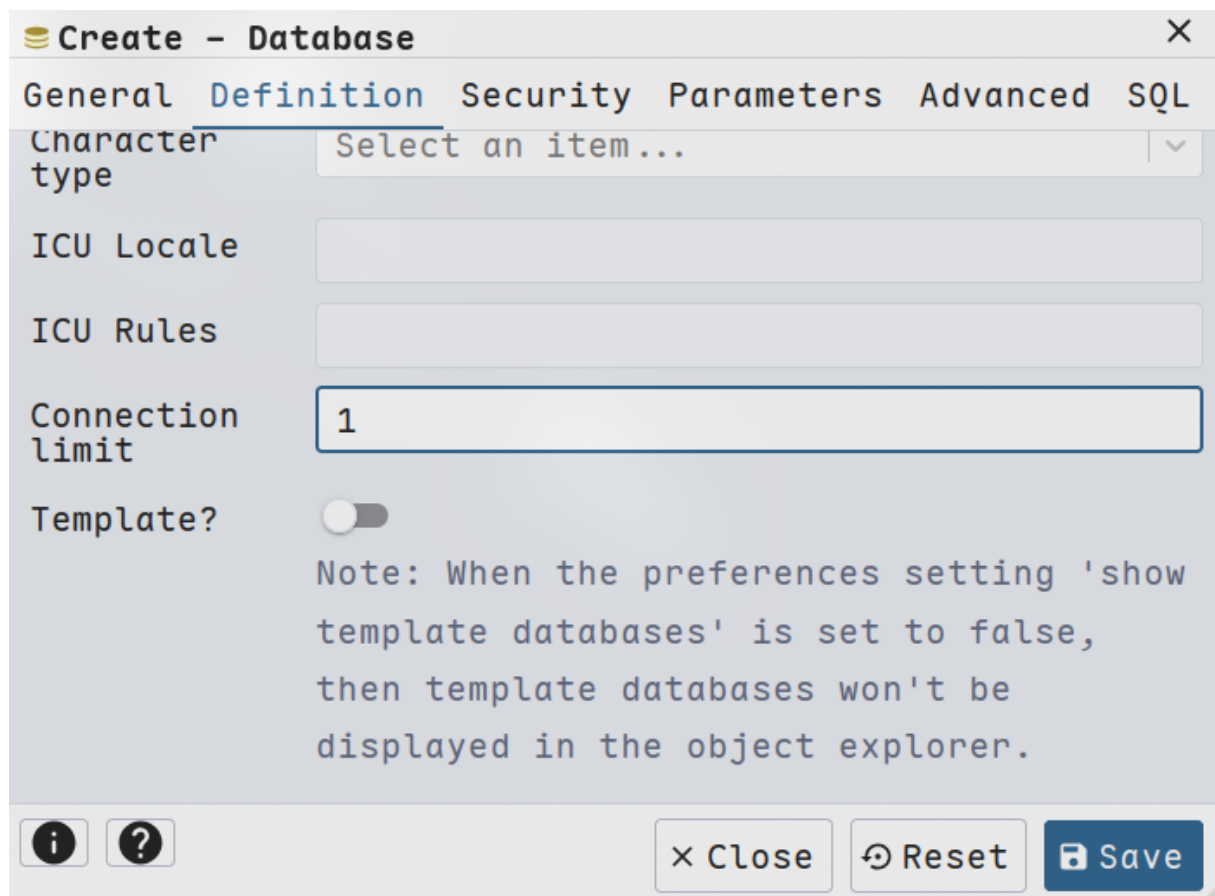
Задание 4. Создание учебной базы данных HR средствами pgAdmin.

Я нажал ПКМ на узле "Databases" и выбрал меню "Create" затем "Database...".

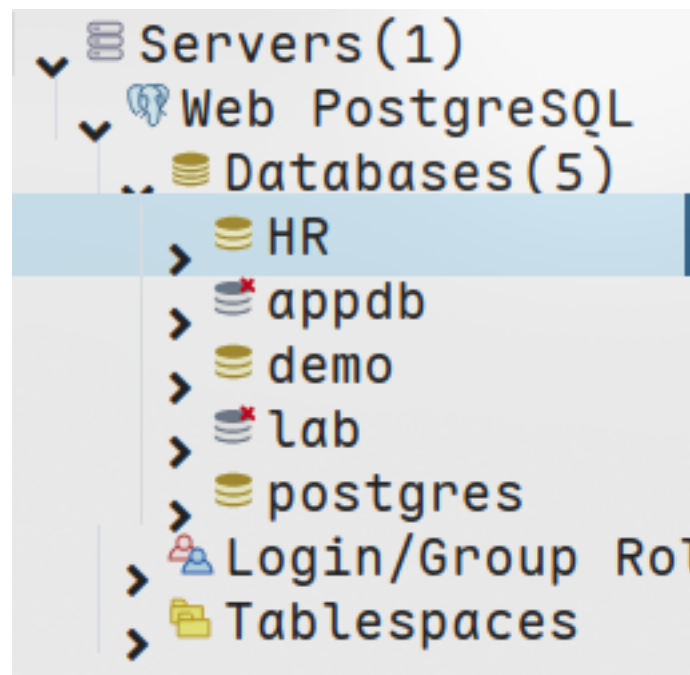


Затем в открывшемся окне я выставил следующие настройки.

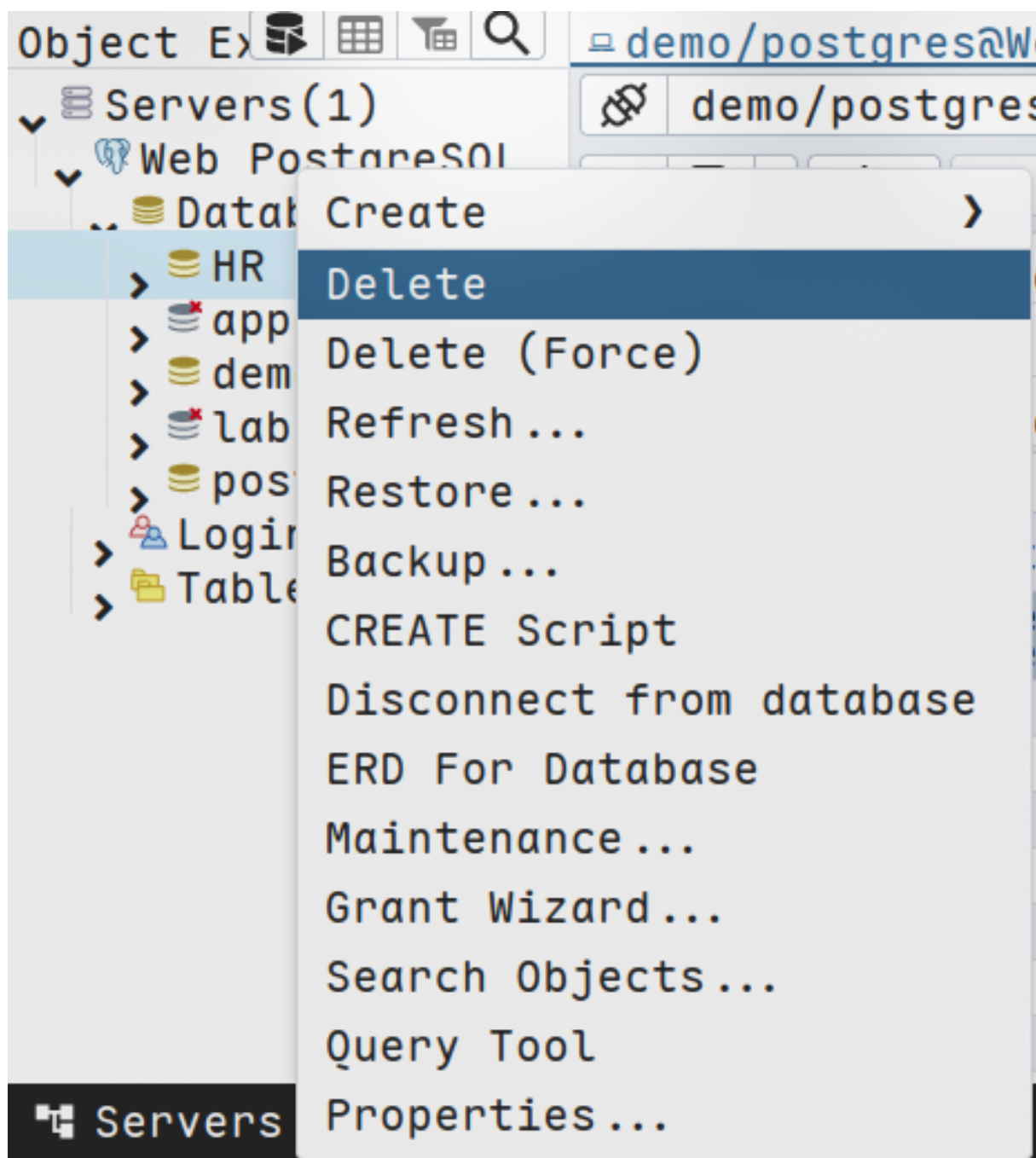




Затем я сохранил настройки. В “Object Explorer” появилась созданная база.



После этого я удалил базу через ПКМ.

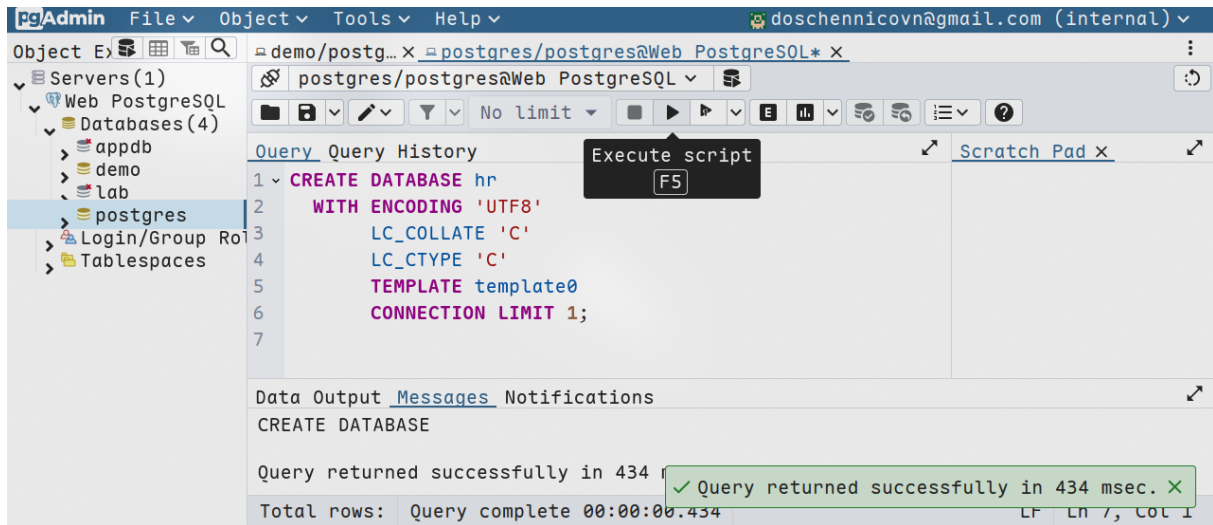


Задание 5. Создание учебной базы данных HR средствами PL/pgSQL

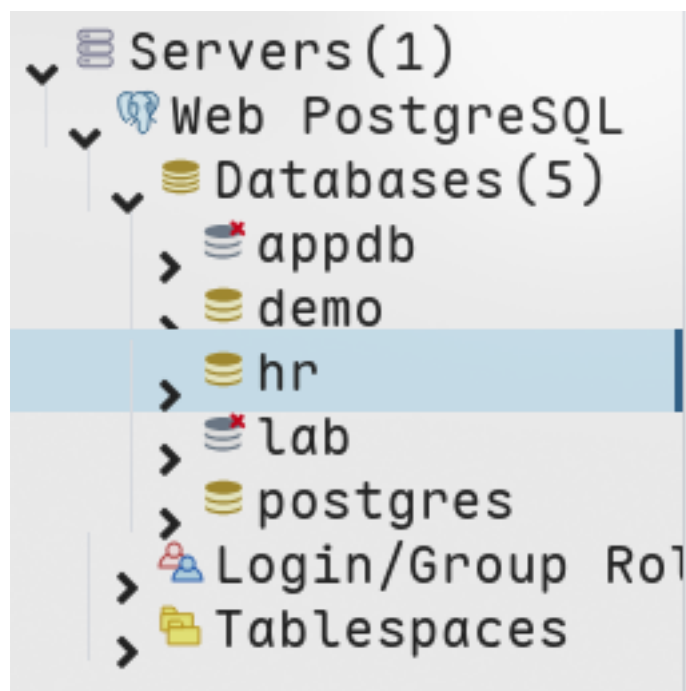
Я открыл “Query Tool” базы данных postgres. И прописал следующий скрипт.

```
CREATE DATABASE hr
WITH ENCODING 'UTF8'
LC_COLLATE 'C'
LC_CTYPE 'C'
```

```
TEMPLATE template0  
CONNECTION LIMIT 1;
```

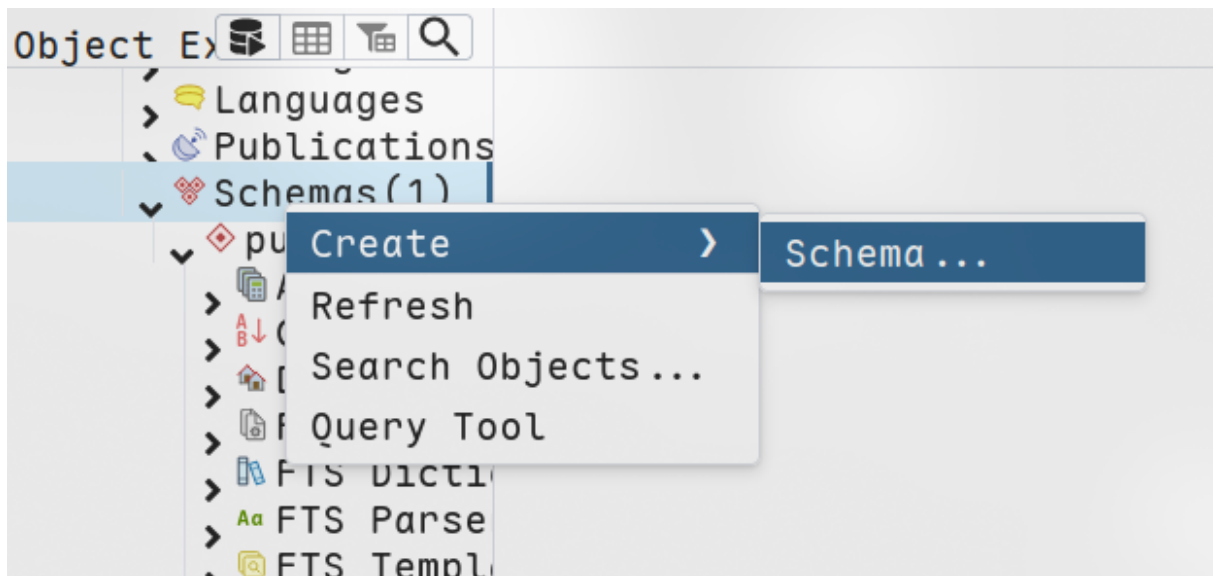


После этого в списке баз появилась созданная.

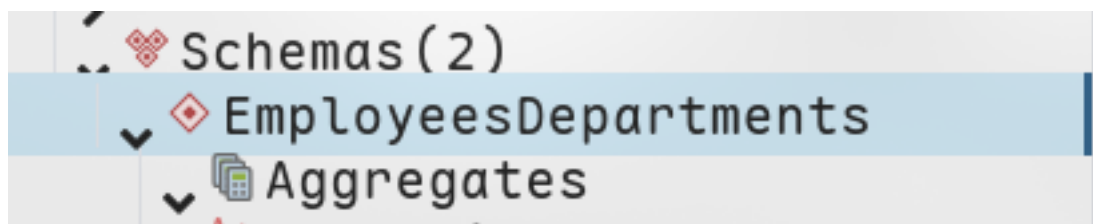


Задание 6. Создание схем.

Я развернул вкладку для создания схем.

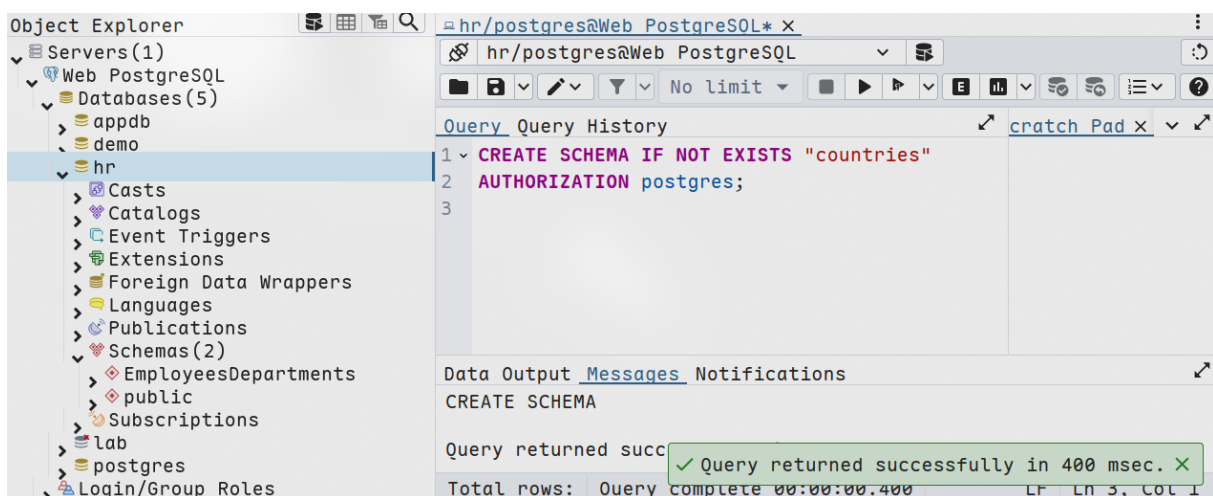


И создал схему "EmployeesDepartments". В "Object Explorer" видно что схема создана.

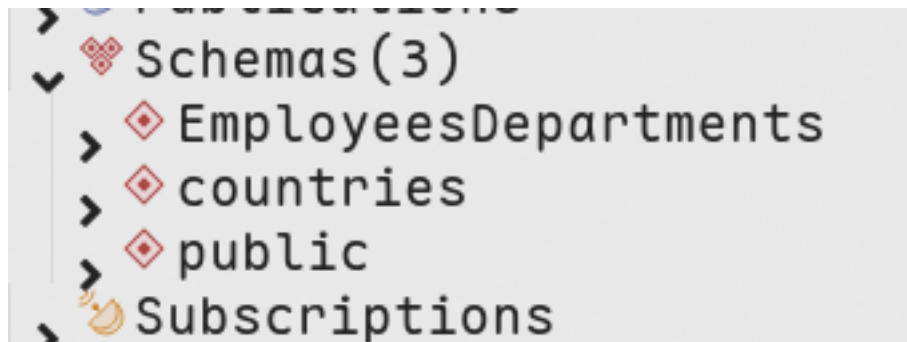


Чтобы создать вторую схему, я открыл "Query Tool" к базе hr. И прописал следующие запросы:

```
CREATE SCHEMA IF NOT EXISTS "countries"
AUTHORIZATION postgres;
```



В "Object Explorer" видна добавленная схема:



Выводы.

В ходе выполнения работы я освоил базовые приёмы установки и настройки СУБД PostgreSQL, а также работу с клиентским приложением pgAdmin. Было развернуто необходимое программное обеспечение в контейнерах Docker.

В процессе работы удалось:

- установить PostgreSQL и pgAdmin
- загрузить и подключить учебную базу данных и выполнить тестовые запросы к ней
- ознакомиться со структурой Demo и основными сущностями
- создать учебную базу данных двумя способами — через графический интерфейс pgAdmin и с использованием SQL-оператора `CREATE DATABASE`
- создать в базе две схемы: одну средствами pgAdmin, вторую — с помощью SQL-запроса.

При выполнении работы возникли следующие трудности:

- при создании базы данных через SQL возникала ошибка с параметрами локали. Она была устранена с помощью указания `TEMPLATE template0` и нейтральной локали `C`
- при подключении pgAdmin к серверу пришлось настраивать отдельную Docker-сеть и регистрировать контейнеры, чтобы обеспечить доступность сервиса по адресу `db.fymio.us`.