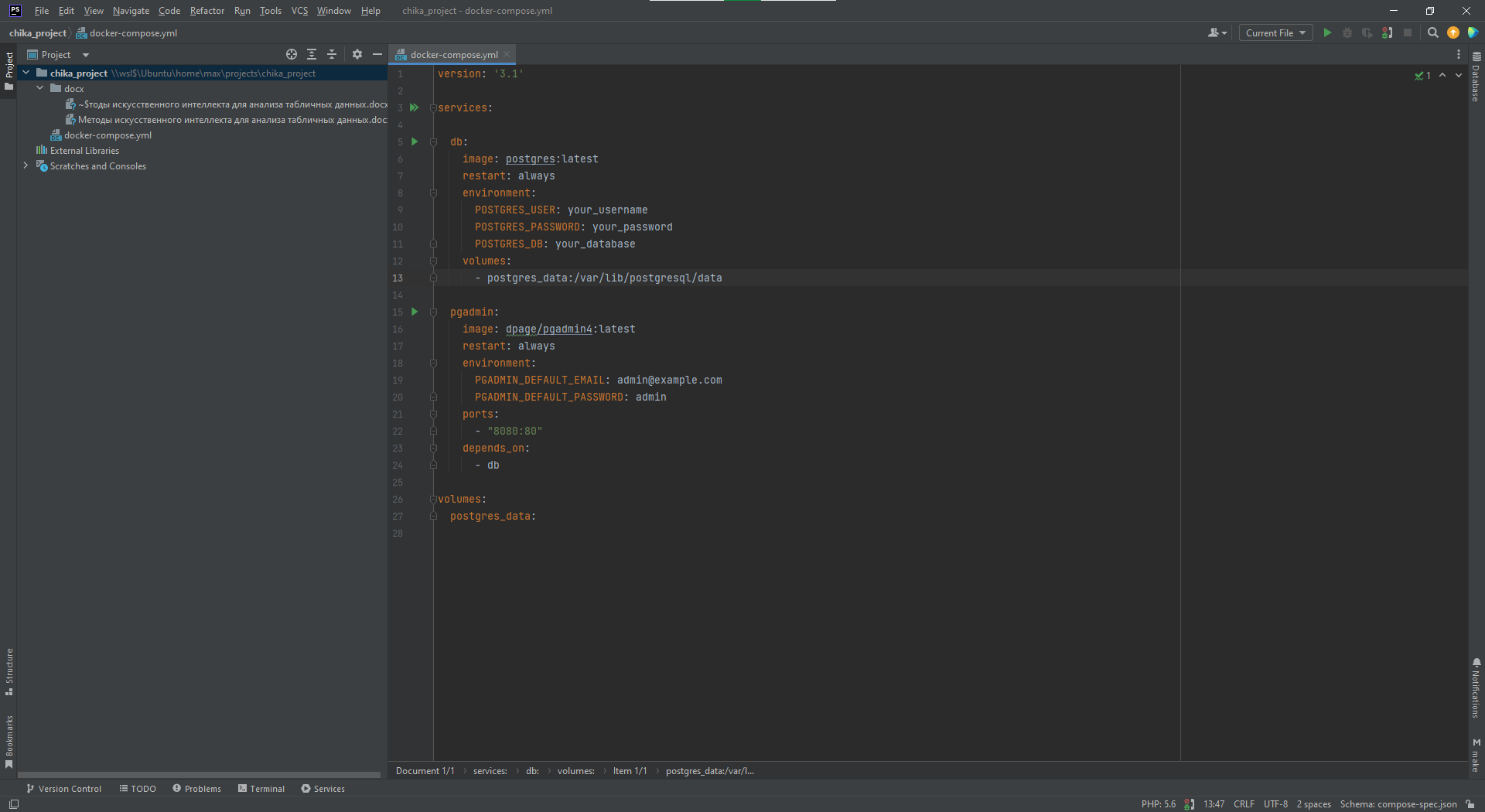
Методы искусственного интеллекта для анализа табличных данных

# Лабораторная работа №1. Подключение к базе данных и создание простейших запросов.

Для удобства я буду использвать PhpStorm и Docker (для его работы под windows уже предустановлена WSL)

Создадим новый проект   
и в нем файл docker-compose.yml с заготовкой под СУБД и pgadmin:

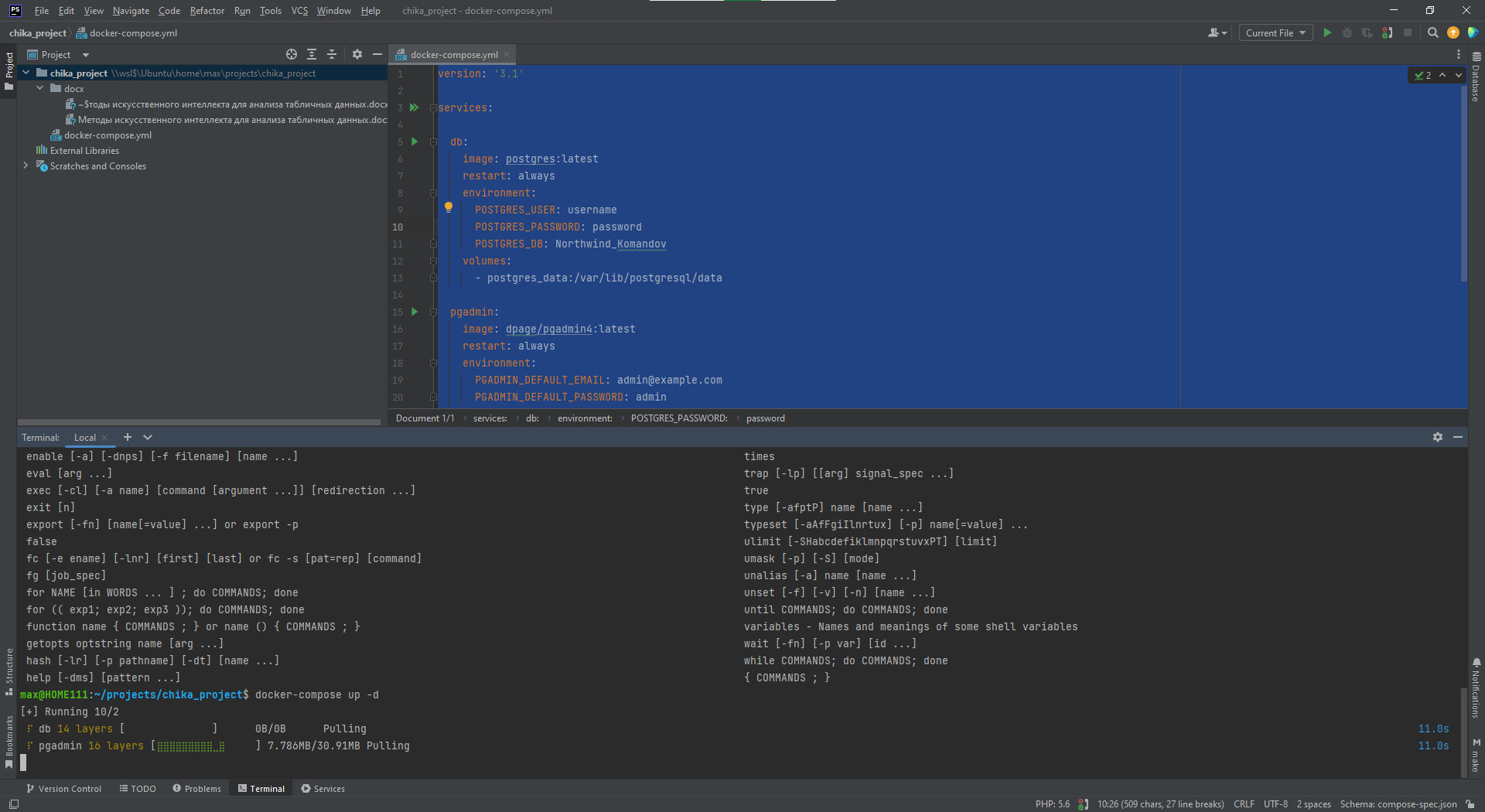


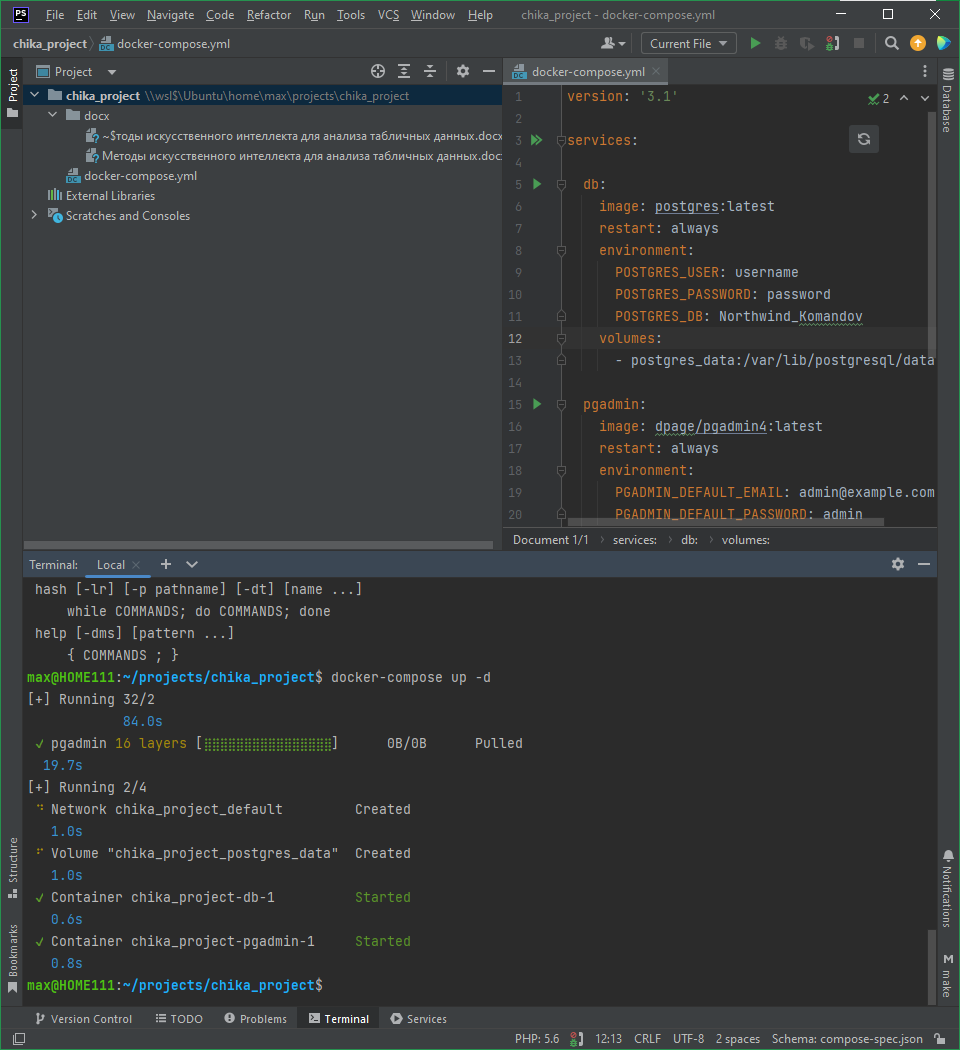
Изменим название БД в соответствии с требованиям:

version: '3.1'  
  
services:  
  
 db:  
 image: postgres:latest  
 restart: always  
 environment:  
 POSTGRES\_USER: username  
 POSTGRES\_PASSWORD: password  
 POSTGRES\_DB: Northwind\_Komandov  
 volumes:  
 - postgres\_data:/var/lib/postgresql/data  
  
 pgadmin:  
 image: dpage/pgadmin4:latest  
 restart: always  
 environment:  
 PGADMIN\_DEFAULT\_EMAIL: admin@example.com  
 PGADMIN\_DEFAULT\_PASSWORD: admin  
 ports:  
 - "8080:80"  
 depends\_on:  
 - db  
  
volumes:  
 postgres\_data:

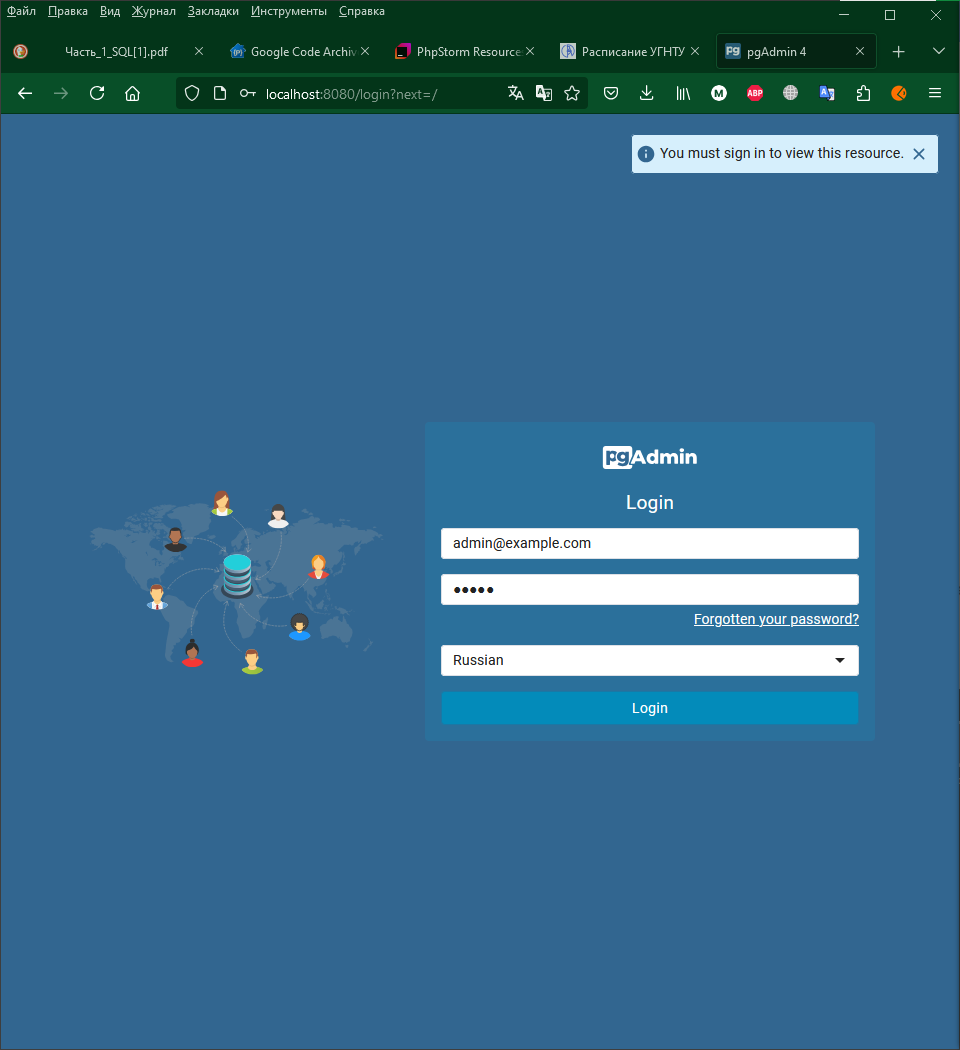
выполним команду docker-compose up -d после чего ожидаем

загрузки и поднятия контейнеров:

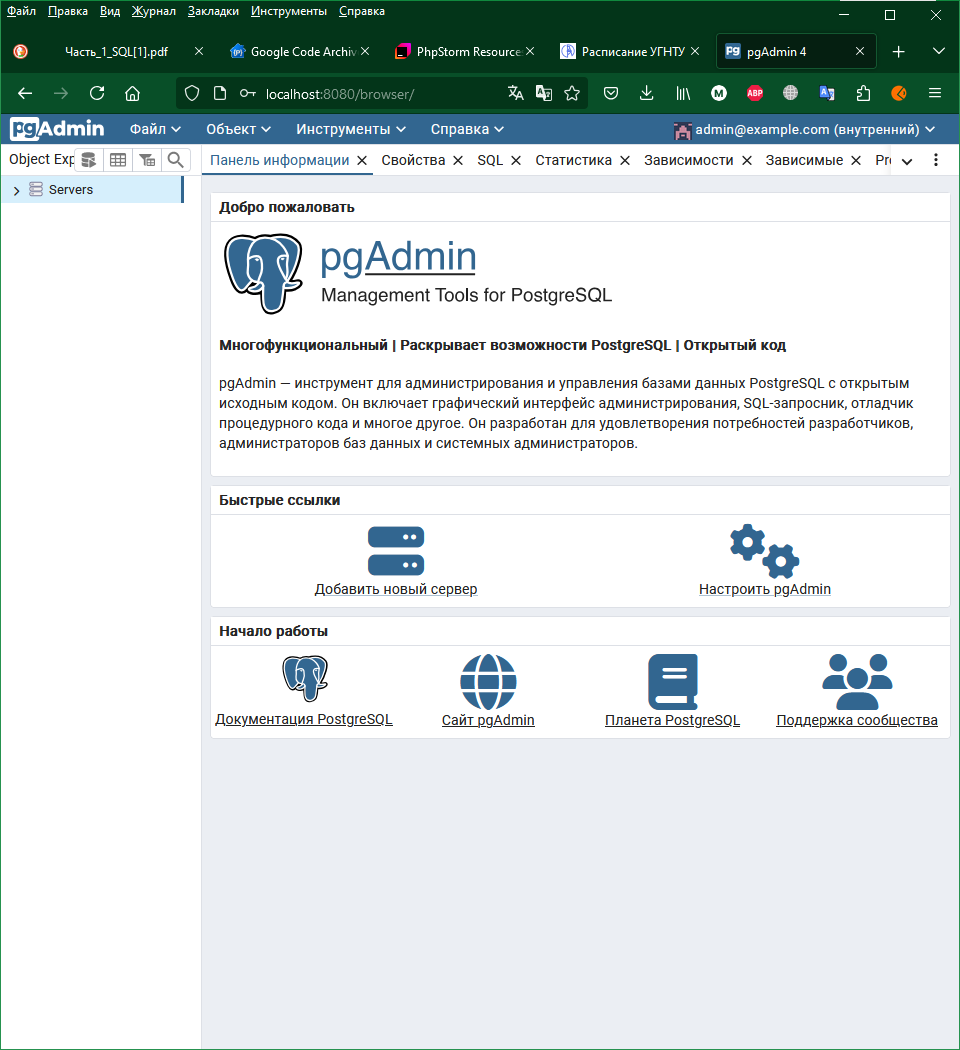


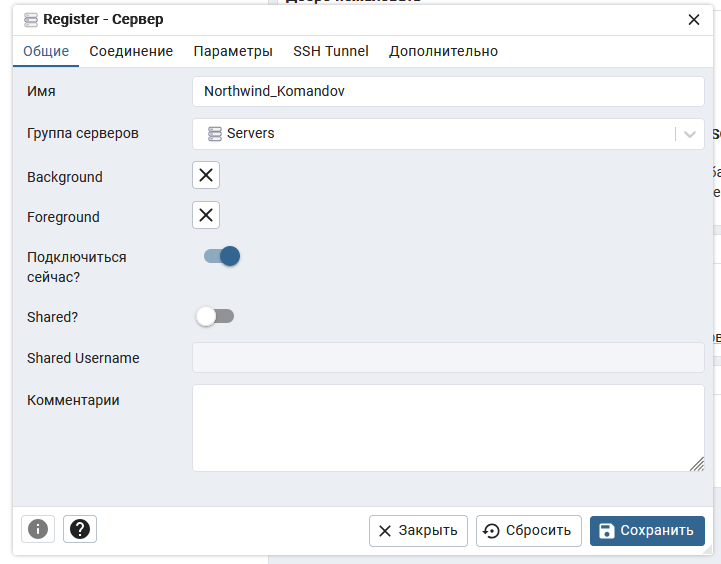
Контейнеры загружены и подняты:  


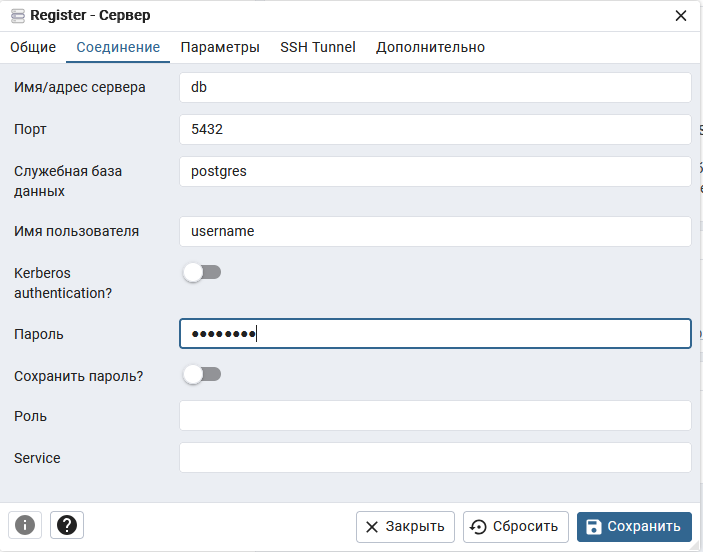
Переходим на созданный сервер админа по адресу <http://localhost:8080> и вводим указные ранее данные в файле docker-compose.yml:



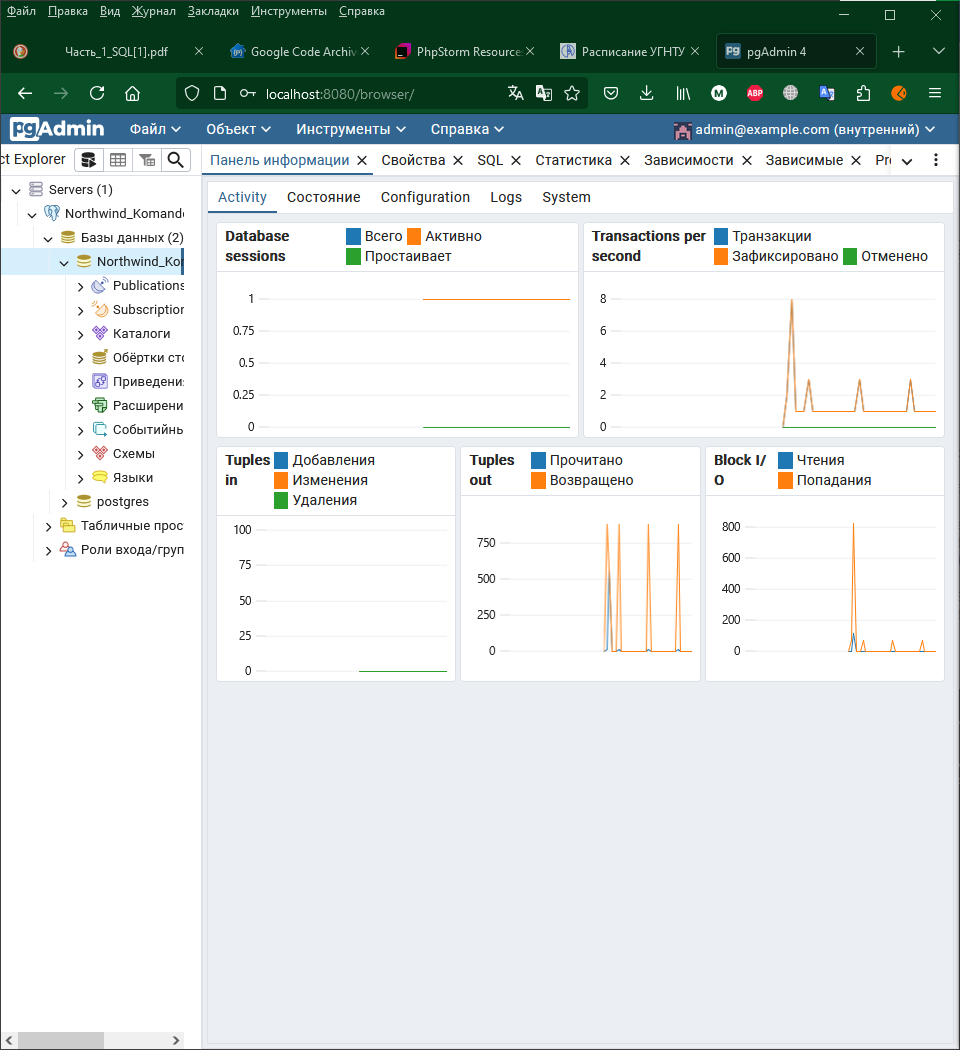
Мы успешно вошли

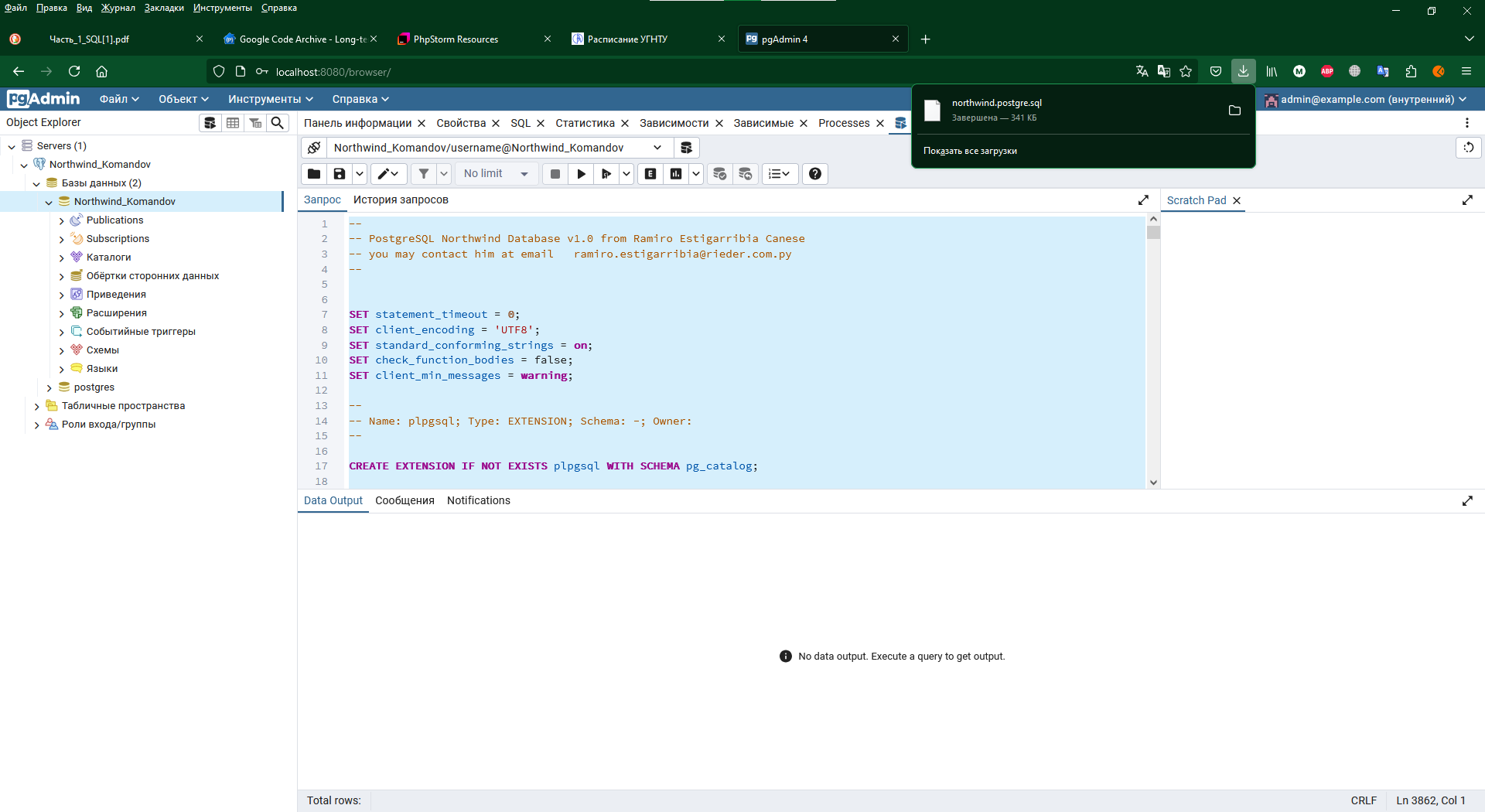


Подключаемся к БД:  


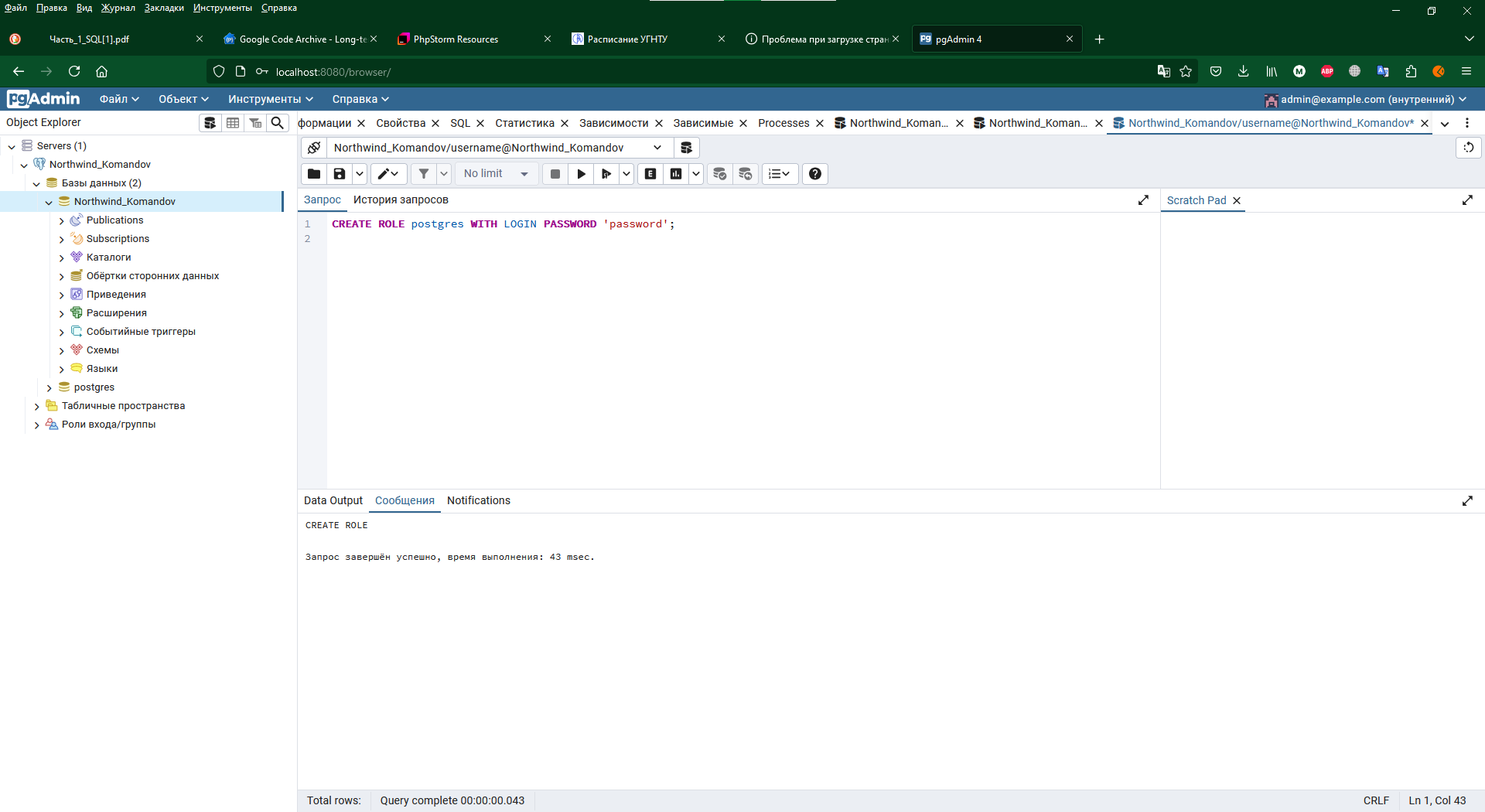


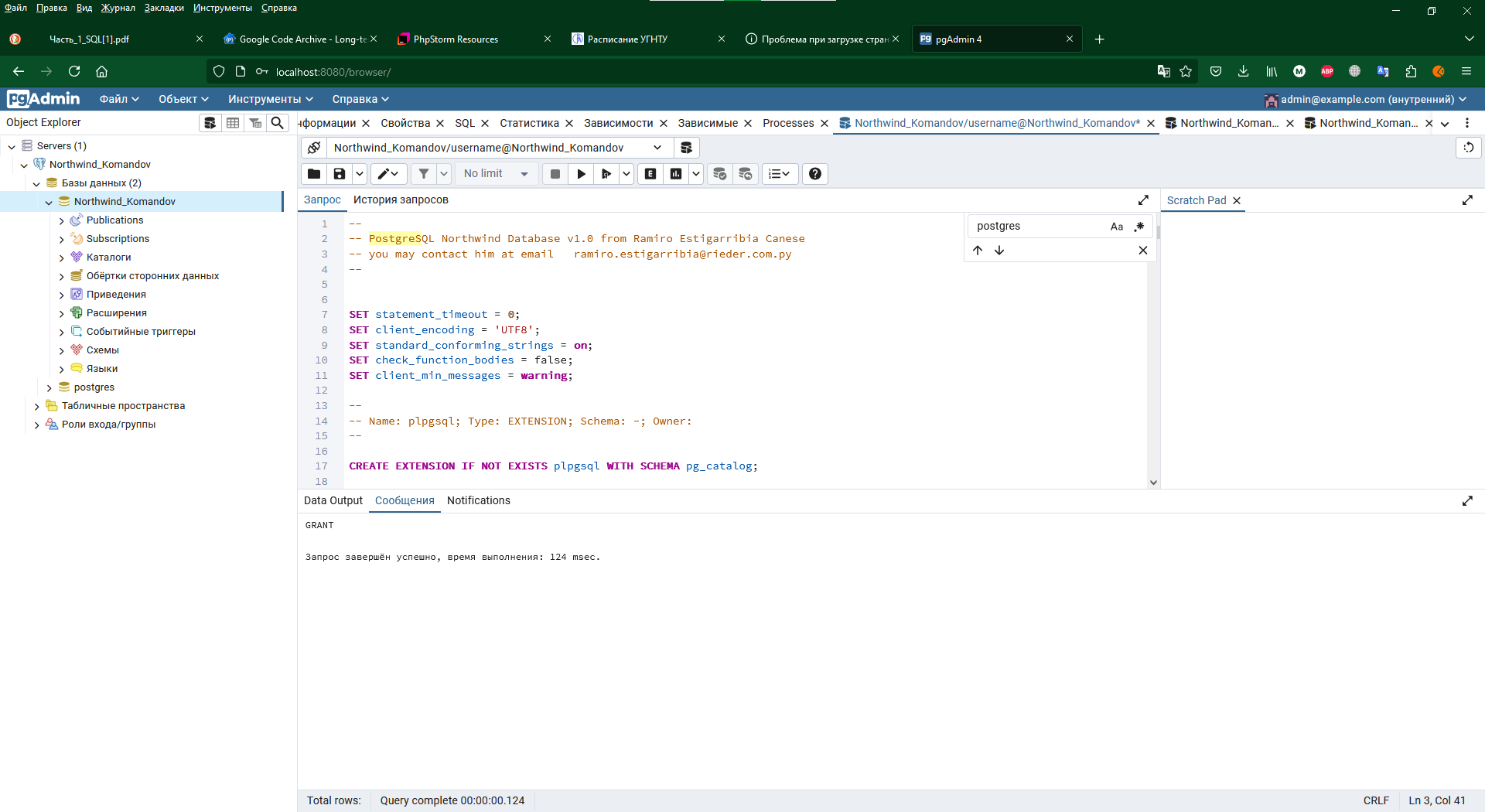
Нажимаем кнопку сохранить



Перетаскиваем ранее скачанный файл на окно запроса 

После выполнения получаем ошибку   
ERROR: role "postgres" does not exist

Создадим эту роль:  


После чего запрос будет успешно выполнен:  


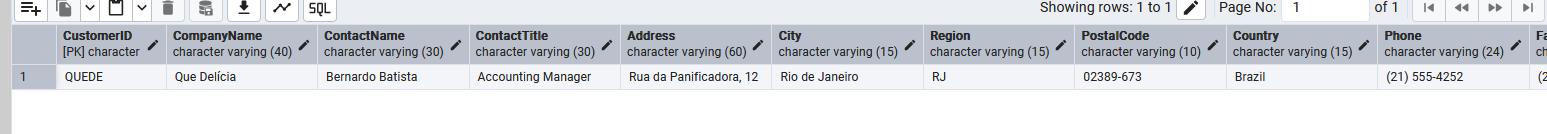
## Выполнение запросов

Select:

-- 1. Запрос, с помощью которого можно получить пользователей из таблицы Customers, имена которых начинаются на «B».

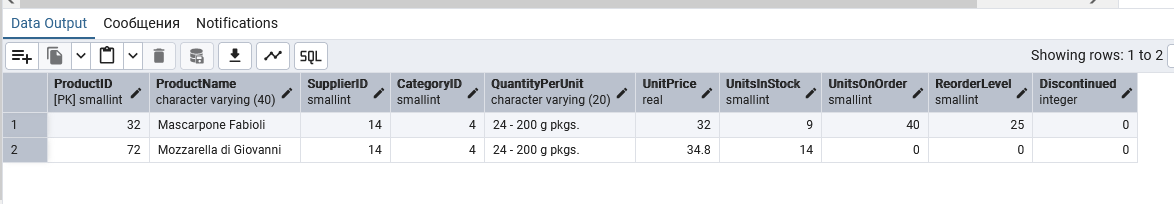
SELECT \* FROM customers

WHERE "ContactName" LIKE 'B%';



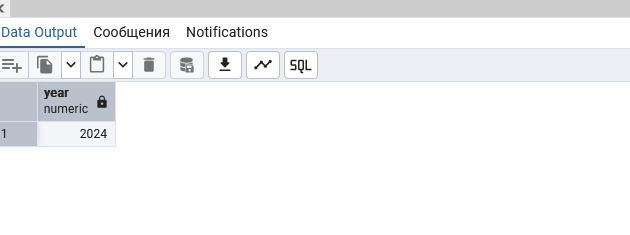
-- 2. Запрос, с помощью которого можно получить товары из таблицы Products, название которых начинается на «M» и заканчивается на «i»:

SELECT \* FROM products WHERE "ProductName" LIKE 'M%i';



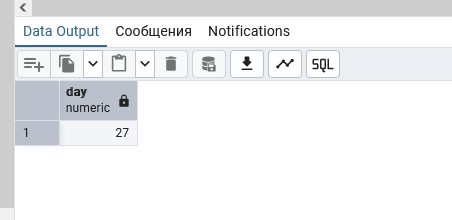
-- 4 Запрос, получающий год из даты 27.06.2024:

SELECT EXTRACT(YEAR FROM DATE '2024-06-27') AS year;



-- 5 Запрос, получающий число из даты 27.06.2024:

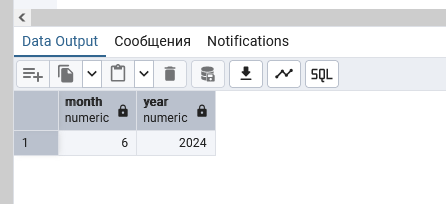
SELECT EXTRACT(DAY FROM DATE '2024-06-27') AS day;

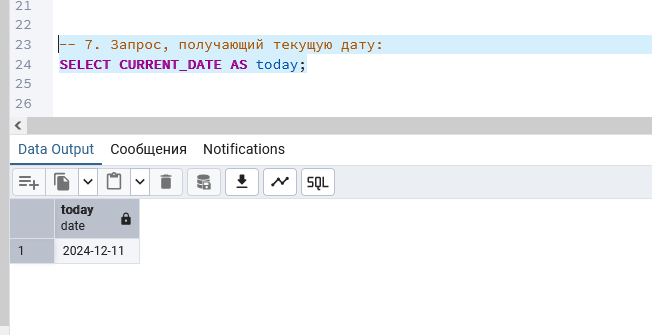


-- 6. Запрос, получающий месяц и год из даты 27.06.2024:

SELECT EXTRACT(MONTH FROM DATE '2024-06-27') AS month,

EXTRACT(YEAR FROM DATE '2024-06-27') AS year;



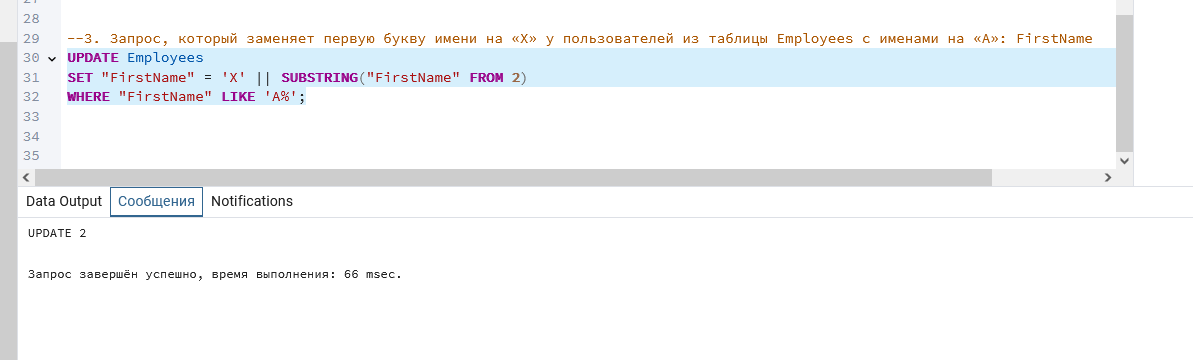


Update:

UPDATE Employees

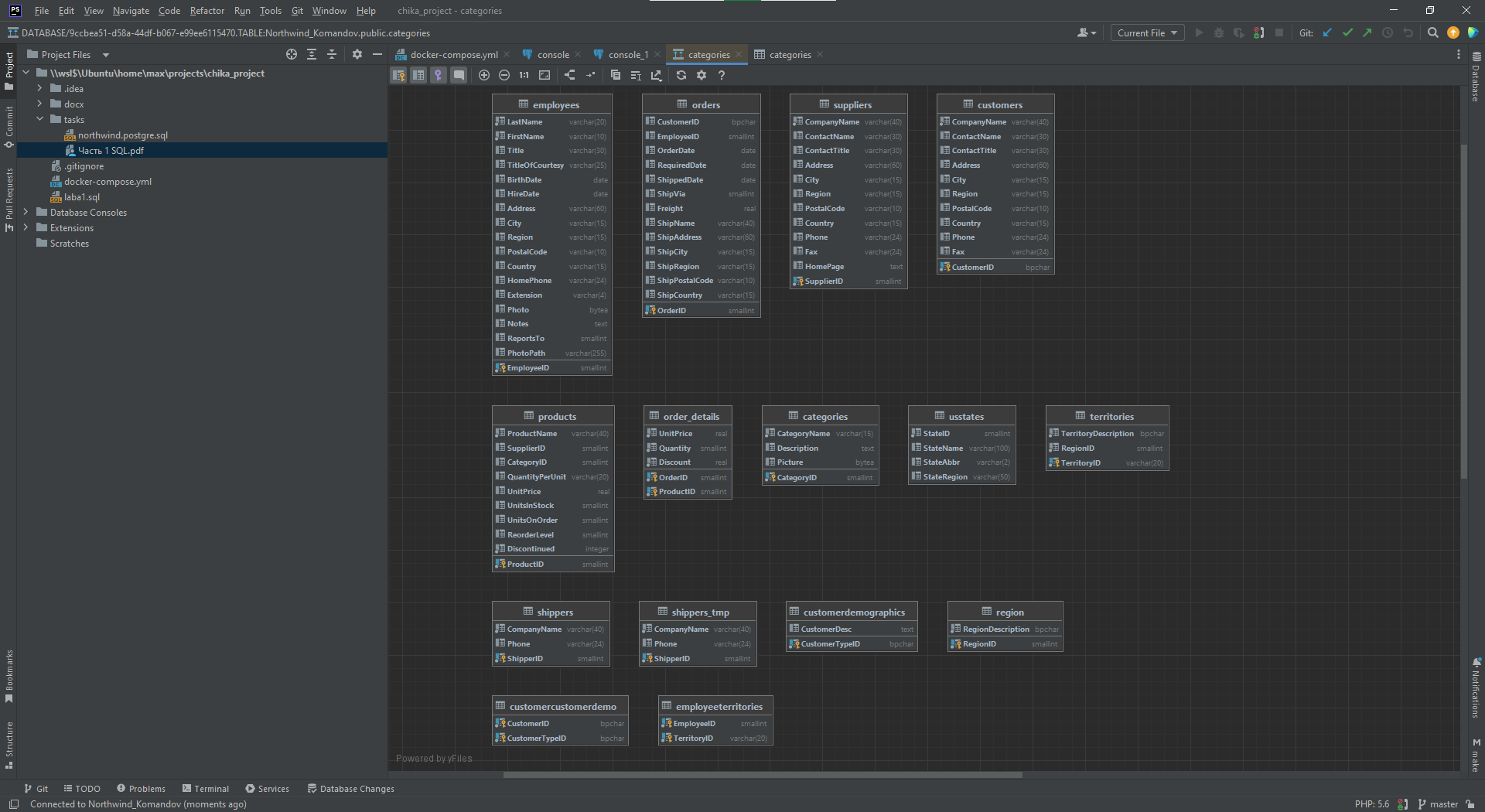
SET "FirstName" = 'X' || SUBSTRING("FirstName" FROM 2)

WHERE "FirstName" LIKE 'A%';



# Лабораторная работа №2. DDL и DML.

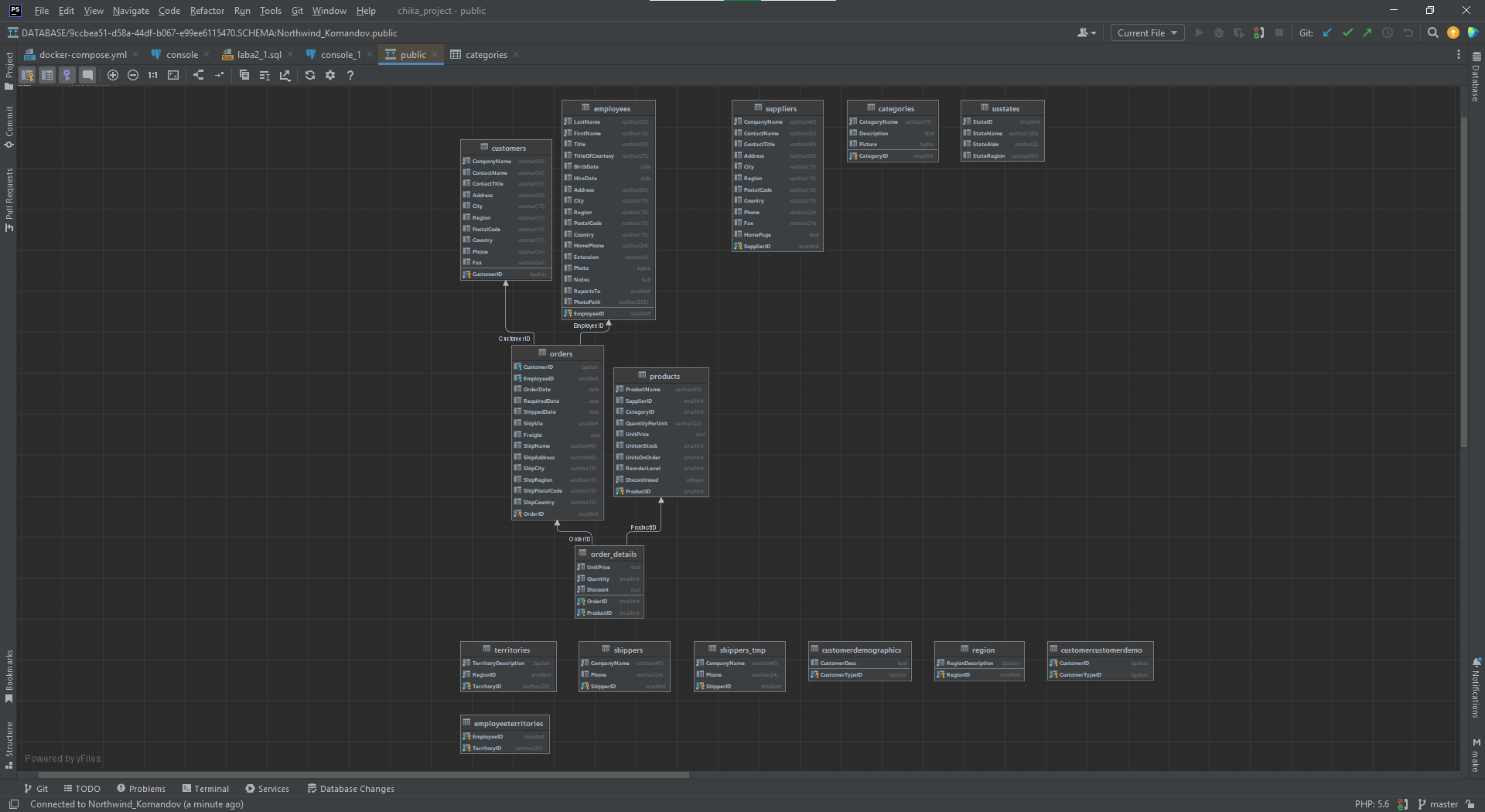
1. Необходимо, используя операторы DDL и DML, добавить ограничения внешних ключей для таблиц из учебной базы данных Northwind (для этого необходимо открыть схему БД (Схема-public-ERD For Schema), найти одноименные столбцы и связать их).

Взглянем на полную схему используя phpStorm и увидим полное отсувие связей между таблицами:  


Создадим их с помощью запроса :

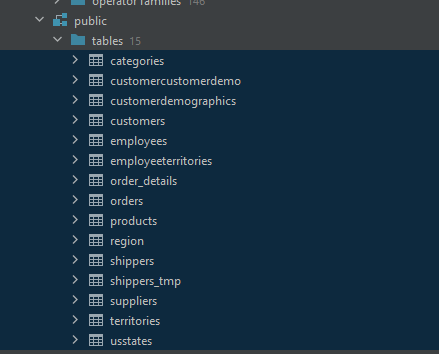
ALTER TABLE public.orders  
 ADD CONSTRAINT fk\_orders\_customers  
 FOREIGN KEY ("CustomerID") REFERENCES public.customers("CustomerID");  
  
ALTER TABLE public.orders  
 ADD CONSTRAINT fk\_orders\_employees  
 FOREIGN KEY ("EmployeeID") REFERENCES public.employees("EmployeeID");  
  
ALTER TABLE public.order\_details  
 ADD CONSTRAINT fk\_order\_details\_orders  
 FOREIGN KEY ("OrderID") REFERENCES public.orders("OrderID");  
  
ALTER TABLE public.order\_details  
 ADD CONSTRAINT fk\_order\_details\_products  
 FOREIGN KEY ("ProductID") REFERENCES public.products("ProductID");

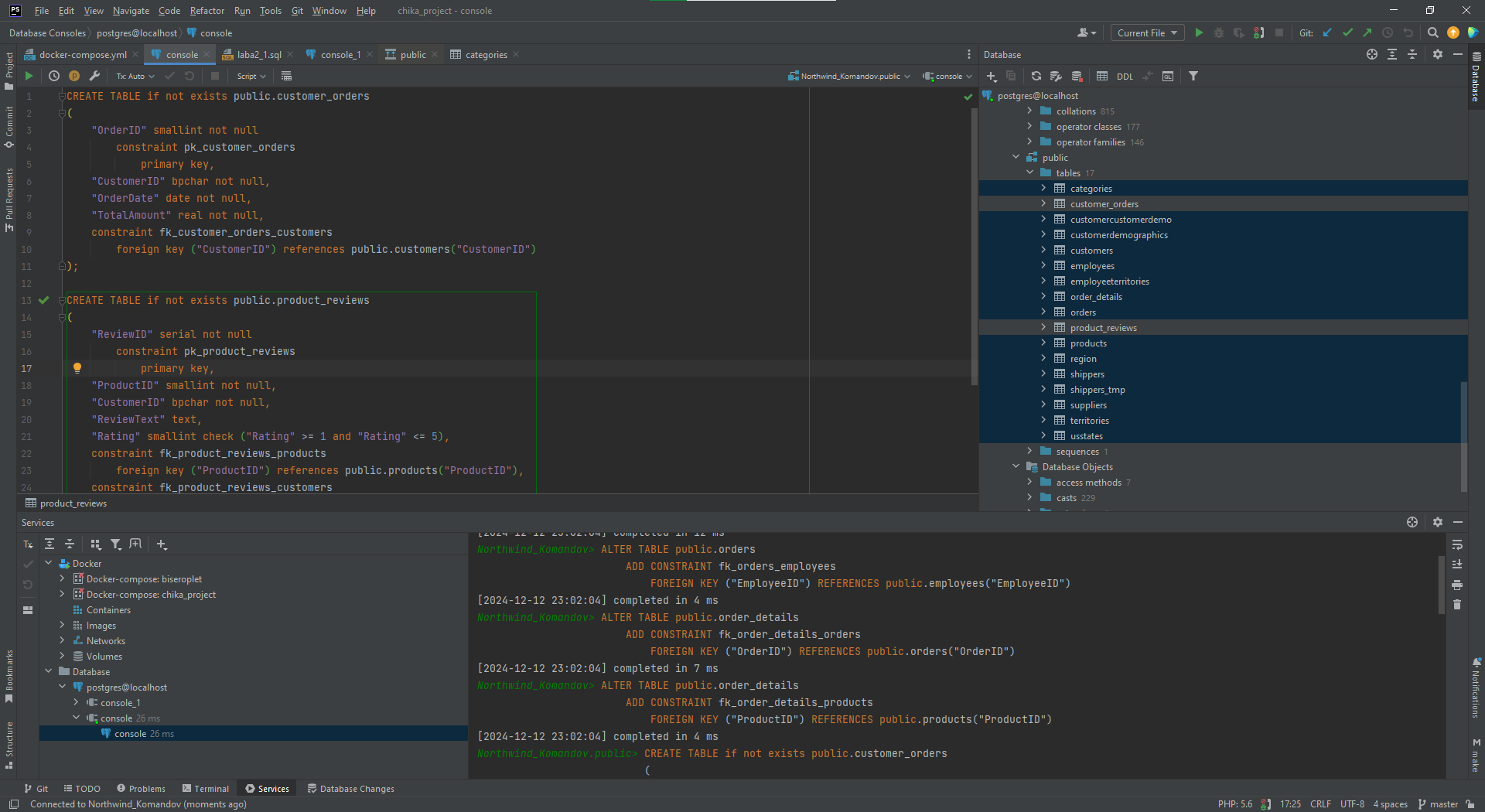


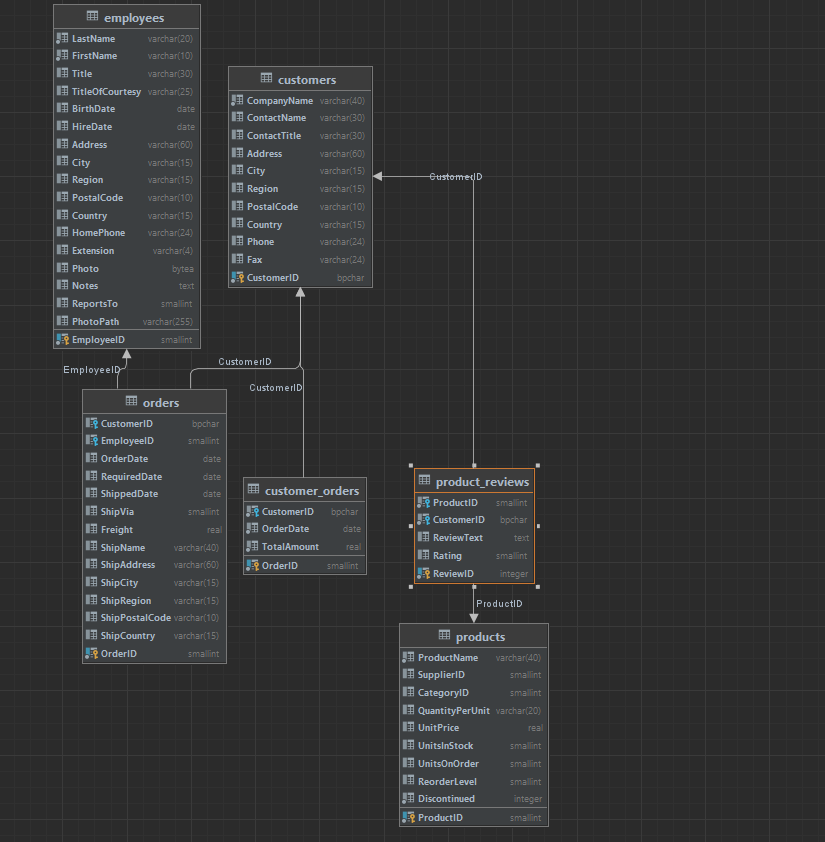


1. Создать таблицы (2-3 штуки), подходящие по контексту к данной базе данных. Добавить к созданным таблицам необходимые ограничения первичных и внешних ключей.

Просмотрим список существующих таблиц:

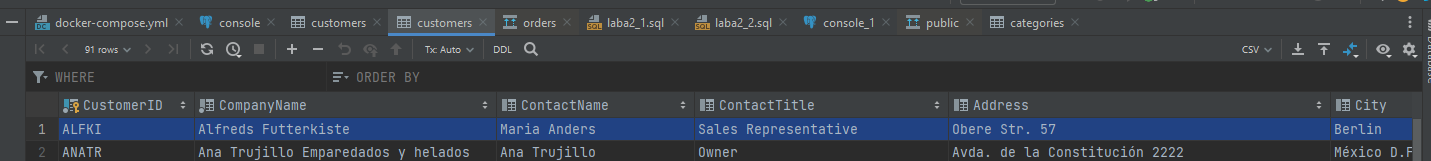


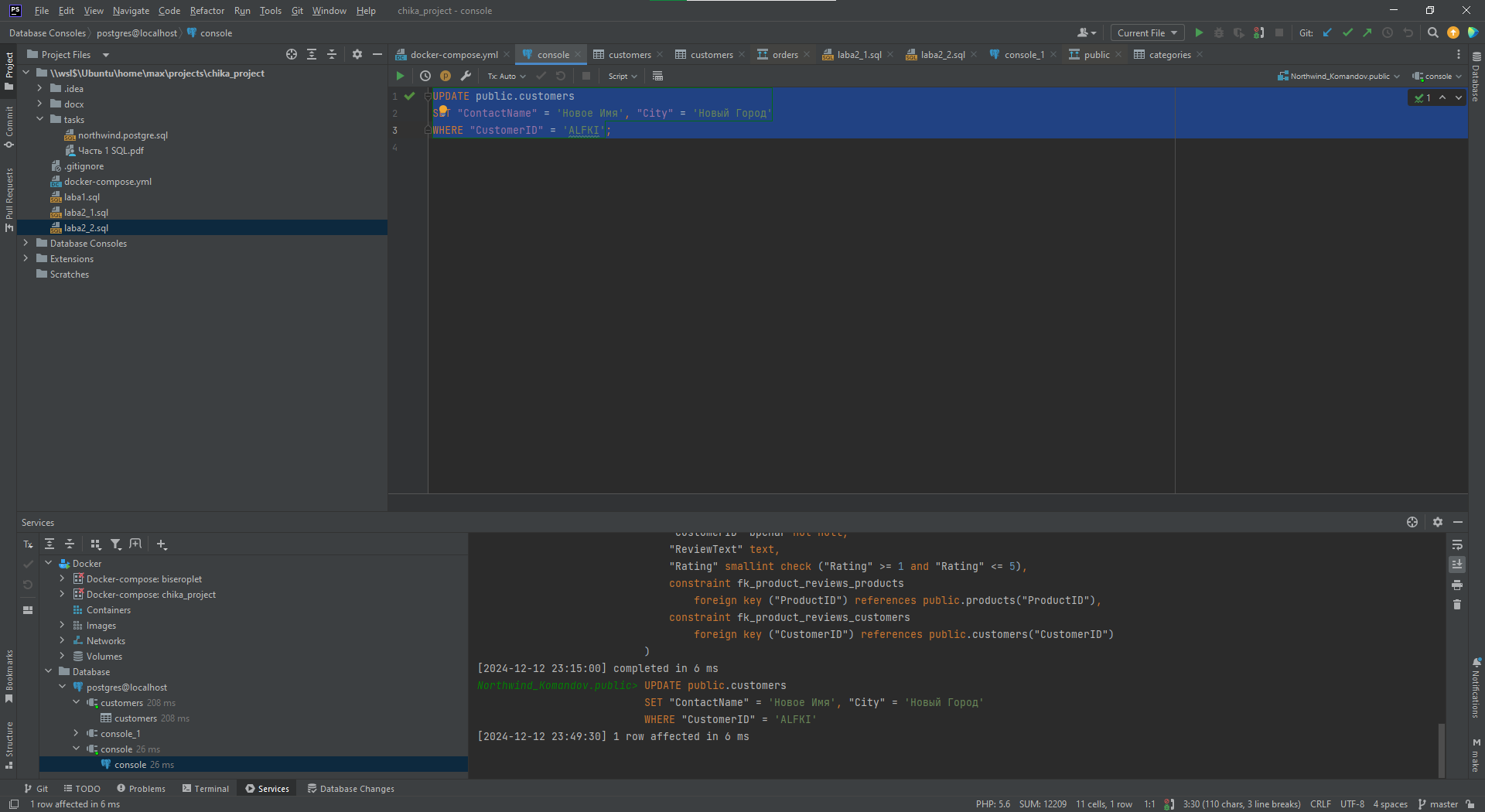
Добавим с помощью запроса новые вместе со связями:  
  


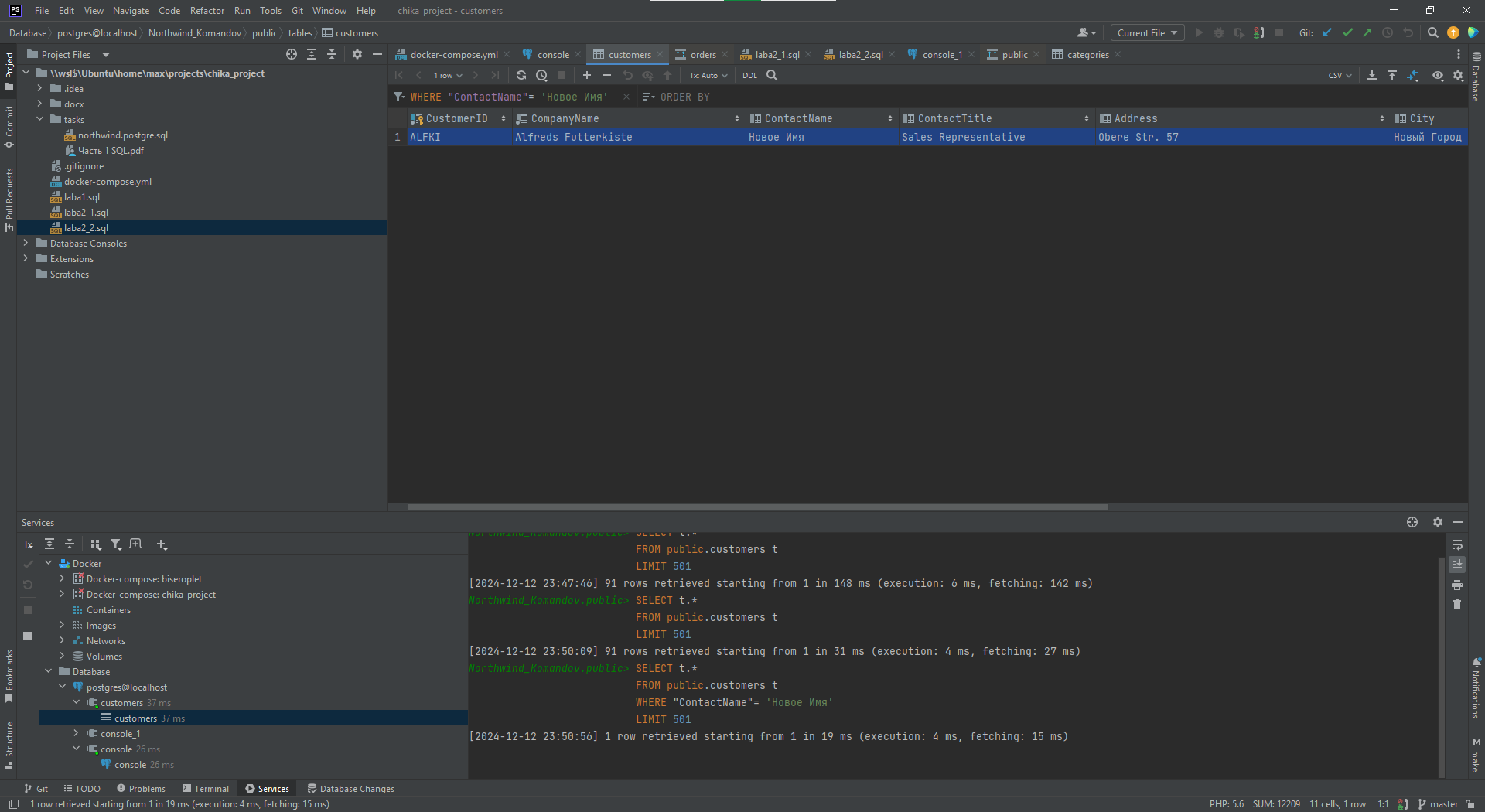


1. Изменить данные в таблице customers (одну строчку) на свои собственные (с использованием команды UPDATE).

Исходные данные:

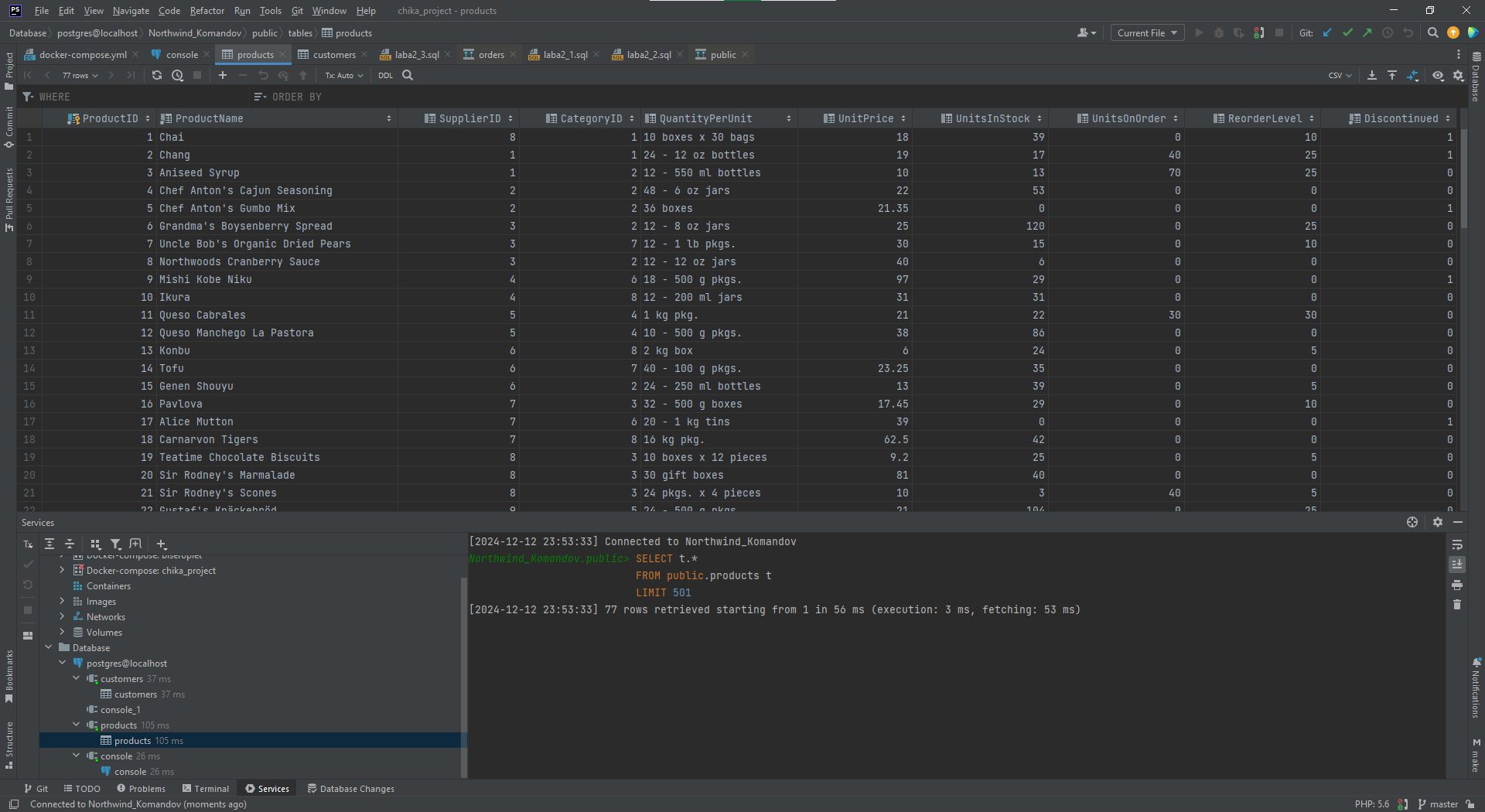


Напишем скрипт для замены и выполним его:  


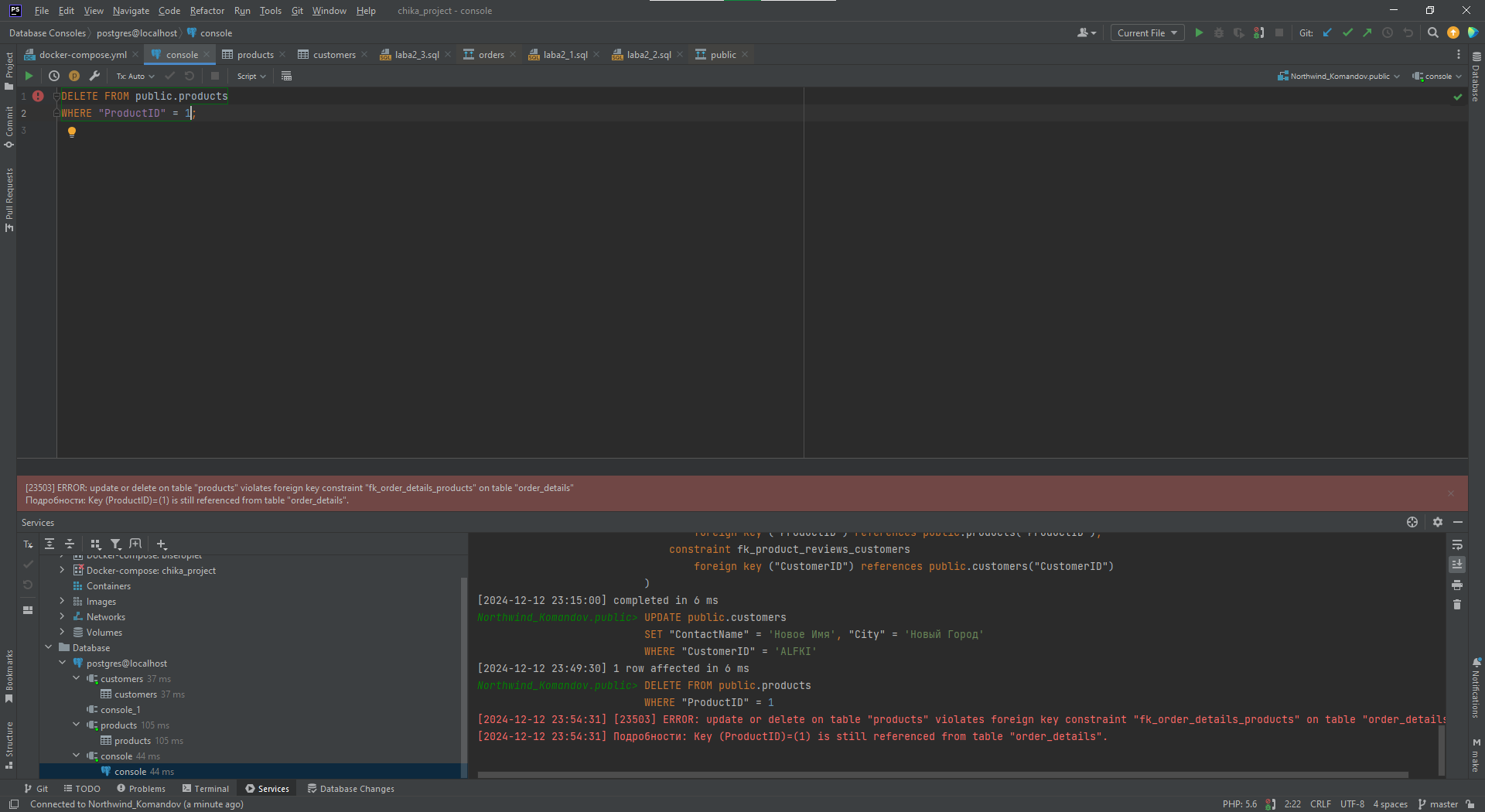
Изменённые данные:  


4. Проверить, что будет при попытке удаления одной записи из таблицы products (с использованием команды DELETE).

Исходные данные таблицы products



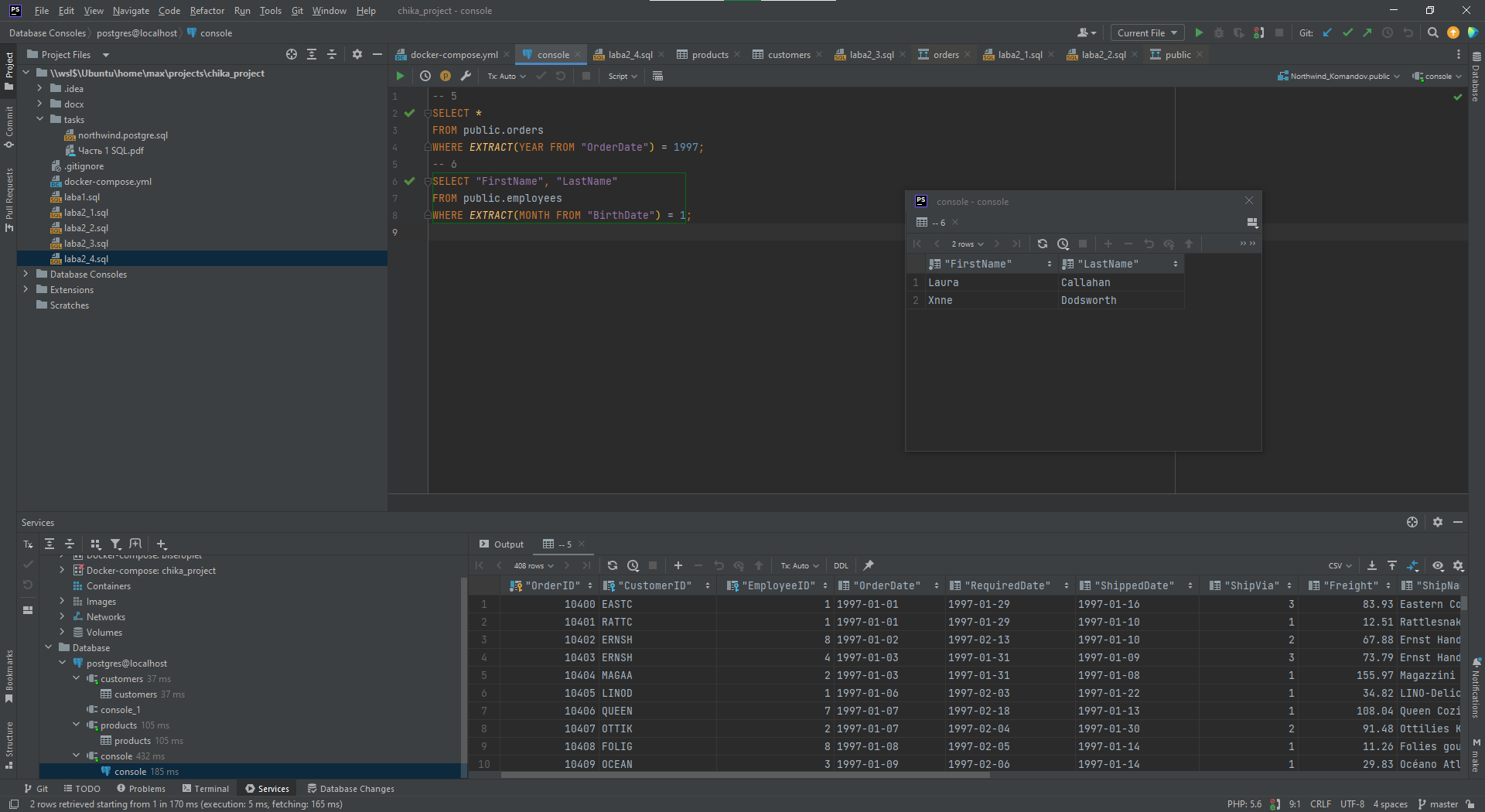
Получаем ошибку при попытке удаления:



5. Запрос, получающий заказы, сделанные в 1997-м году (по полю OrderDate в таблице Orders).

6. Вывести имена всех сотрудников, родившихся в январе.

Напишем запрос и выполним его:



# Лабораторная работа №3. Соединение данных.

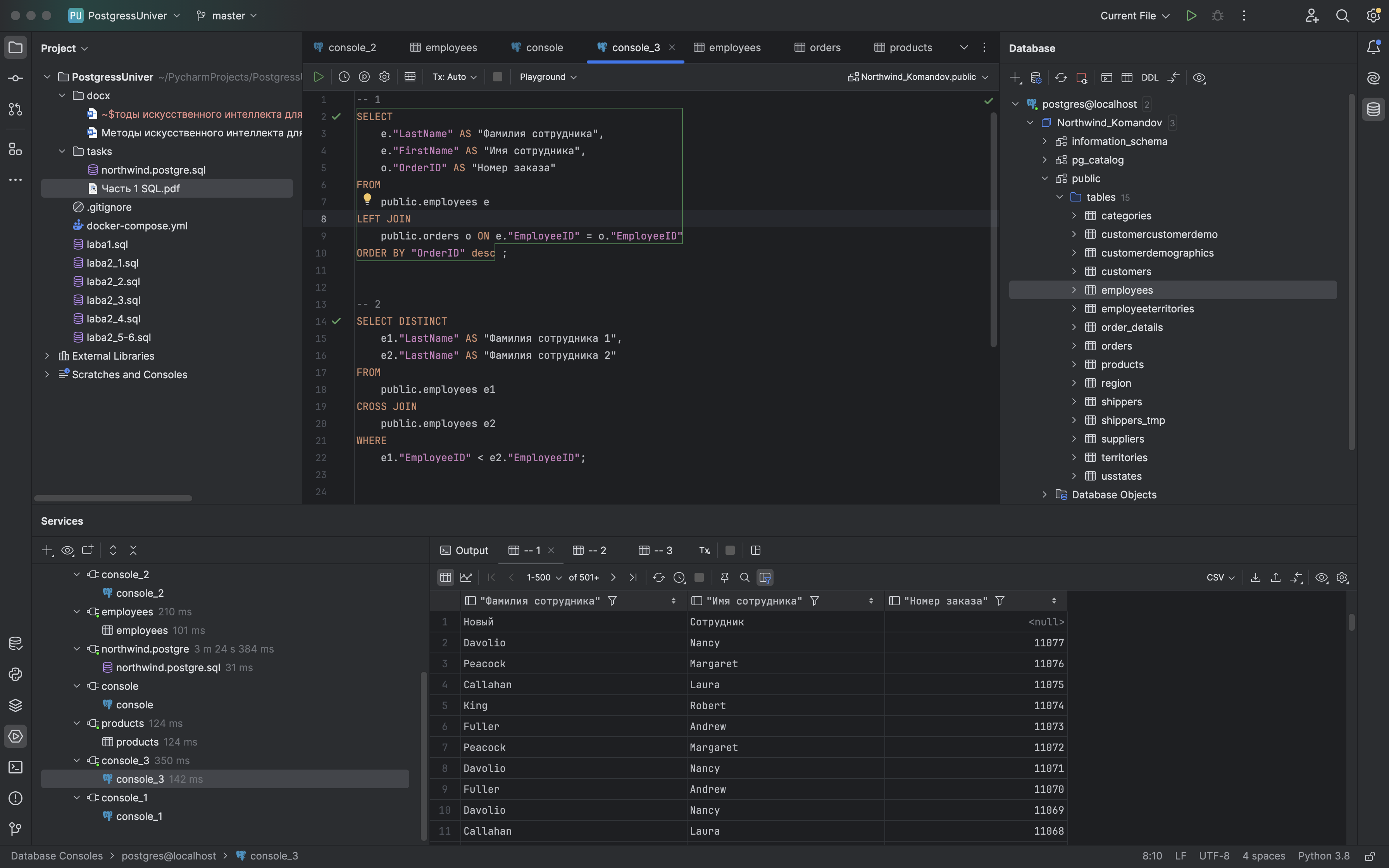
1. Требуется создать запрос, с помощью которого вывести три столбца:

- фамилия сотрудника;

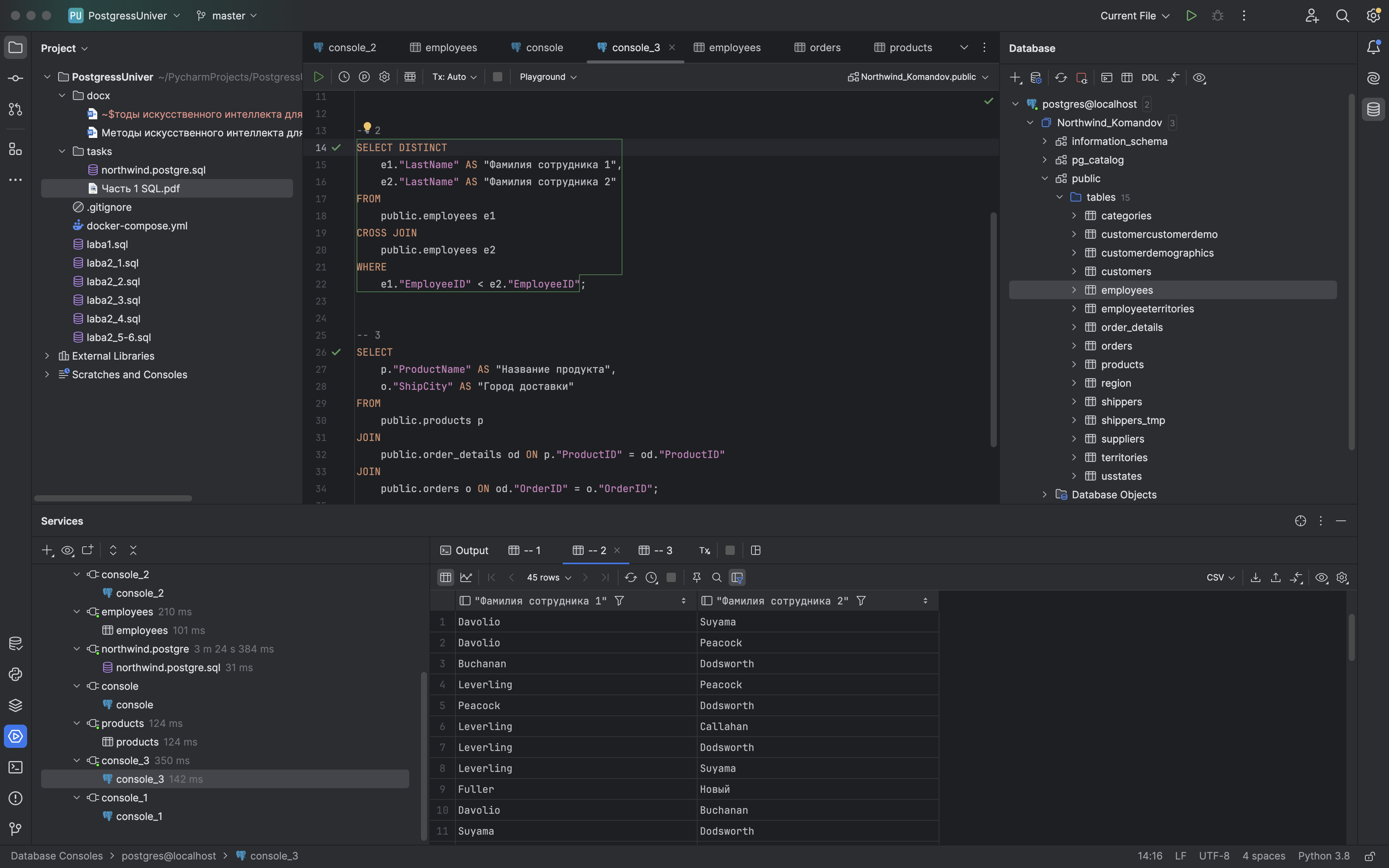
- имя сотрудника;

- номер заказа, за который он отвечает, также рассмотреть вариант вывода сотрудников, у которых пока нет заказов, за которые он отвечает (новенький сотрудник).

Необходимо использовать оператор JOIN.



2. Требуется написать запрос для получения всех возможных комбинаций фамилий сотрудников, исключив повторения. Необходимо использовать оператор JOIN.



3. Создать запрос, который выводит название продукта и города, куда данный продукт доставляли. Необходимо использовать оператор JOIN.

