Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

Специальность 09.02.07 Информационные системы и программирование

Квалификация: Программист

ПРАКТИЧЕСКИЕ РАБОТЫ

По дисциплине «Технология разработки и защиты баз данных»

Листов: 48

|  |  |
| --- | --- |
| Выполнил: студент  Студент группы П50-4-21  Игошев Р.В. | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_\_Ю. В. Севастьянов  «\_\_\_\_» \_\_\_\_\_\_\_\_ 2023 года |

Москва 2023

**ОГЛАВЛЕНИЕ**

[ПРАКТИЧЕСКАЯ РАБОТА №1 3](#_Toc148611191)

[ПРАКТИЧЕСКАЯ РАБОТА №2 17](#_Toc148611192)

[ПРАКТИЧЕСКАЯ РАБОТА №3 34](#_Toc148611193)

[ПРАКТИЧЕСКАЯ РАБОТА №4 43](#_Toc148611194)

# ПРАКТИЧЕСКАЯ РАБОТА №1

Тема: «Знакомство с PostgreSQL»

Цель работы: научиться работать с PostgreSql, используя её консольную и графическую оболочку, создать 5 таблиц в качестве закрепления полученных знаний.

Начнём с скачивания PostgreSQL, скачать его можно на этом сайте:

<https://www.postgresql.org/download/>



Рисунок 1 – Страница загрузки PostgreSql

Вначале установки придется выбрать компоненты для скачивания. Выбираем все. После создаем пароль для БД и указываем порт. Пароль опциональный, порт лучше оставить по умолчанию. После установки запускаем psql.



Рисунок 2 – Авторизация psql и смена кодировки

Здесь мы вошли в БД и поменяли кодировку на кириллицу. Попробуем узнать информацию о нынешних командах.

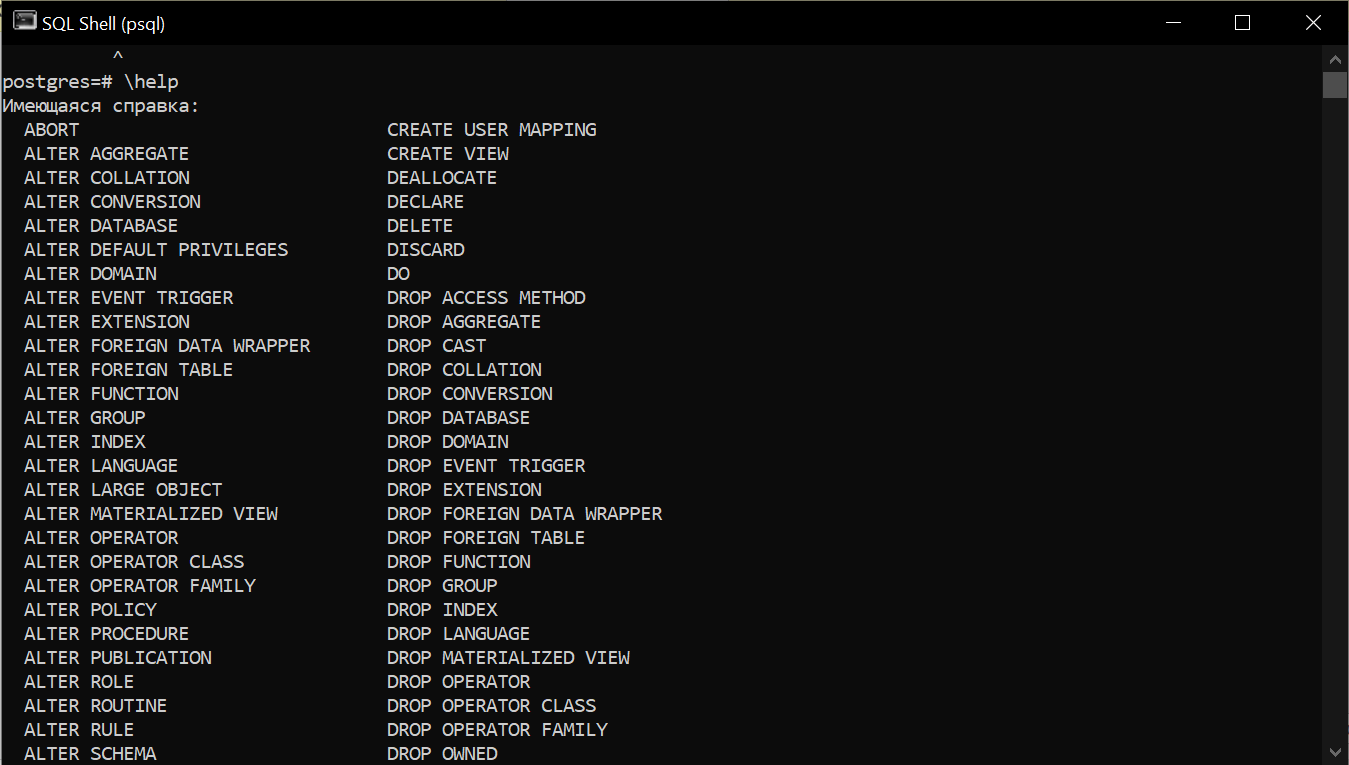


Рисунок 3 – Справка по командам

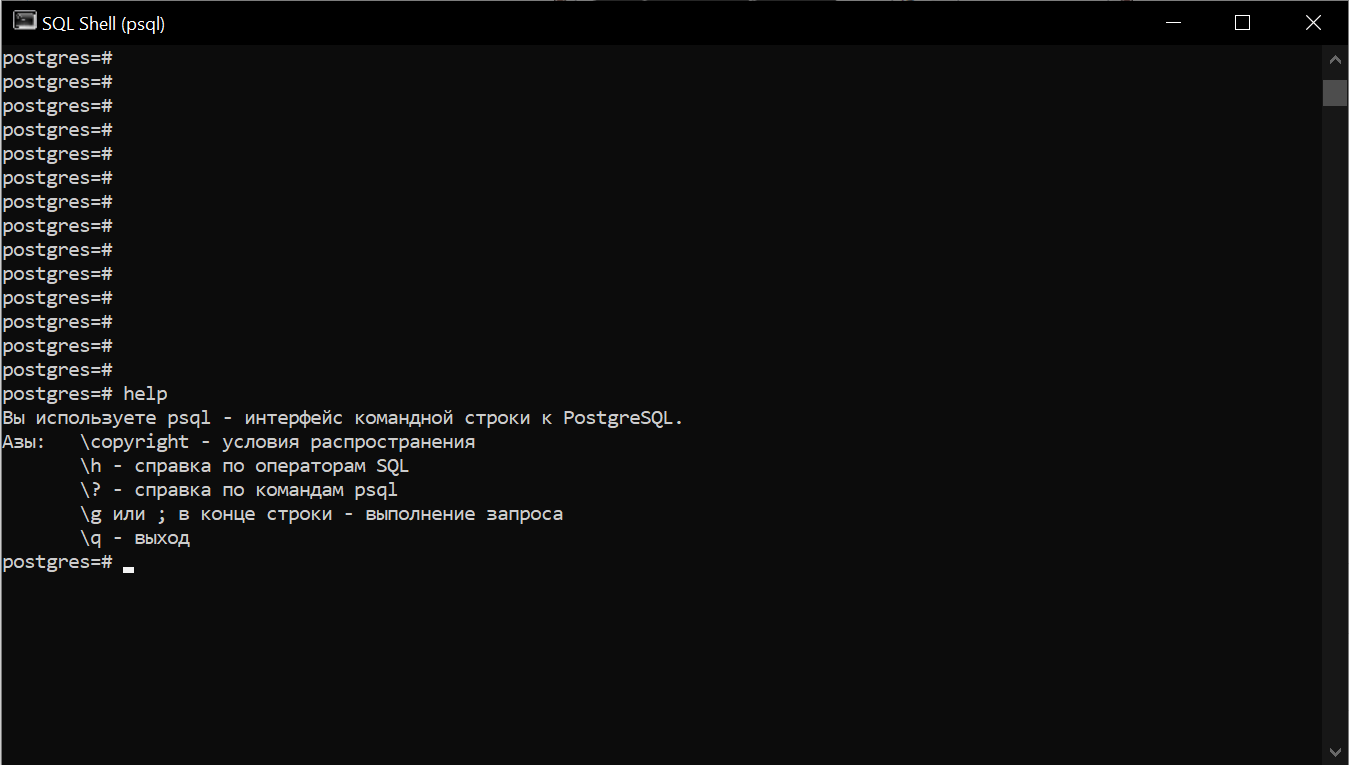


Рисунок 4 – Справка по справкам

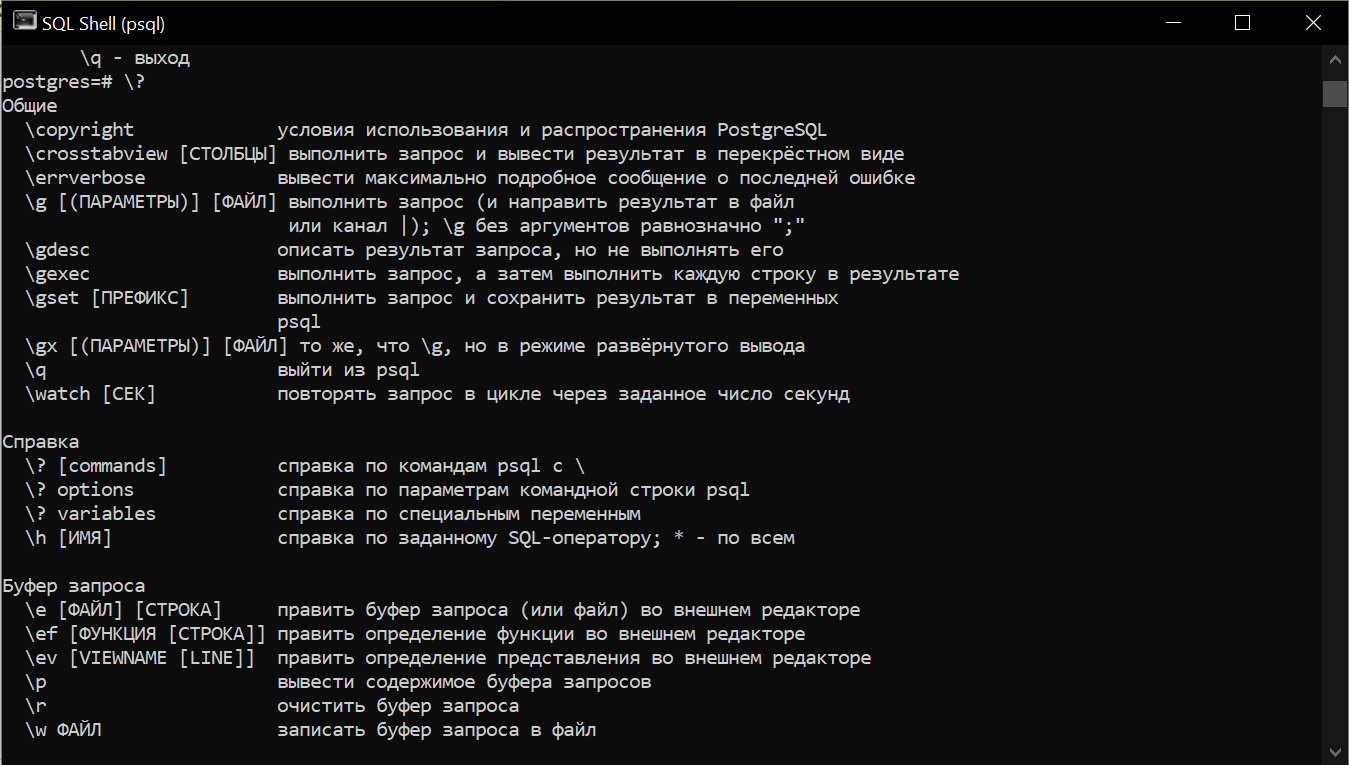


Рисунок 5 – Справка по командам

Попробуем повзаимодействовать с консолью:

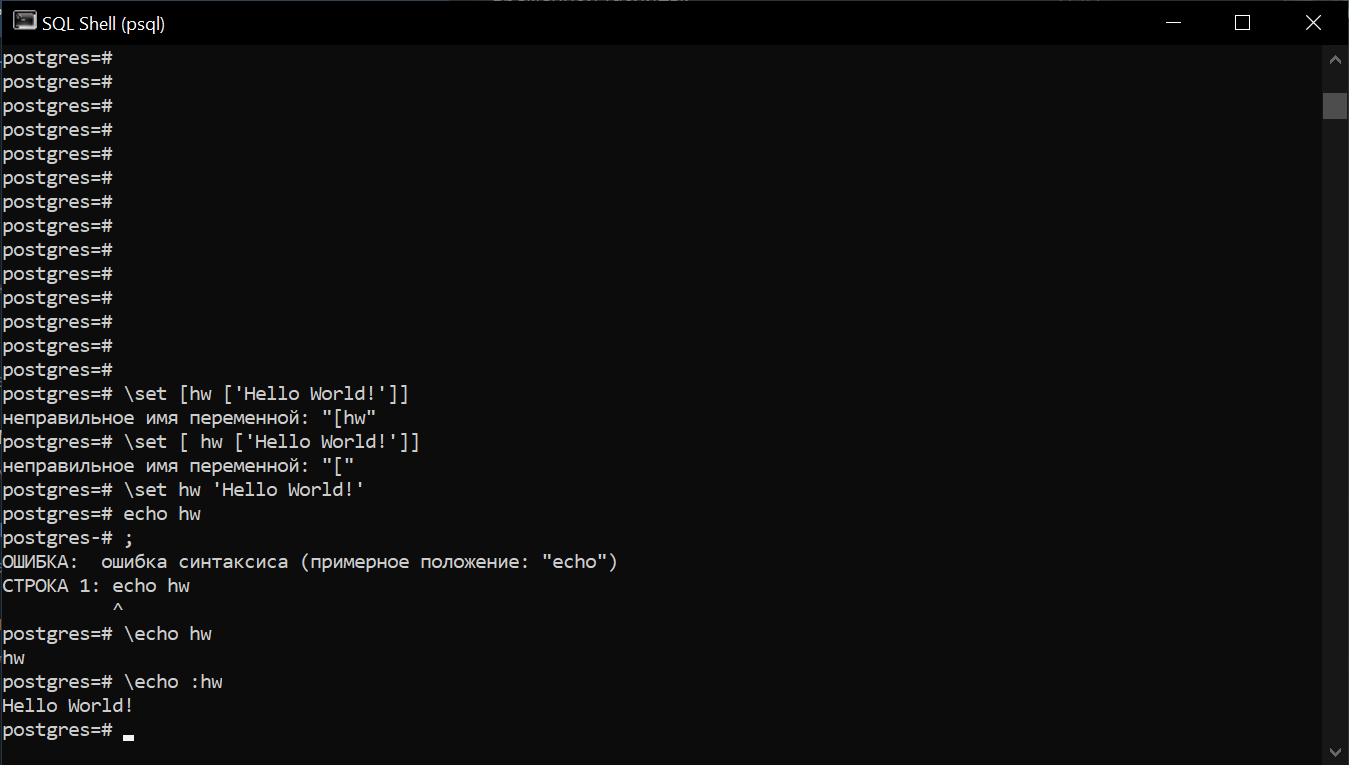


Рисунок 6 – Вывод данных через команды в консоль

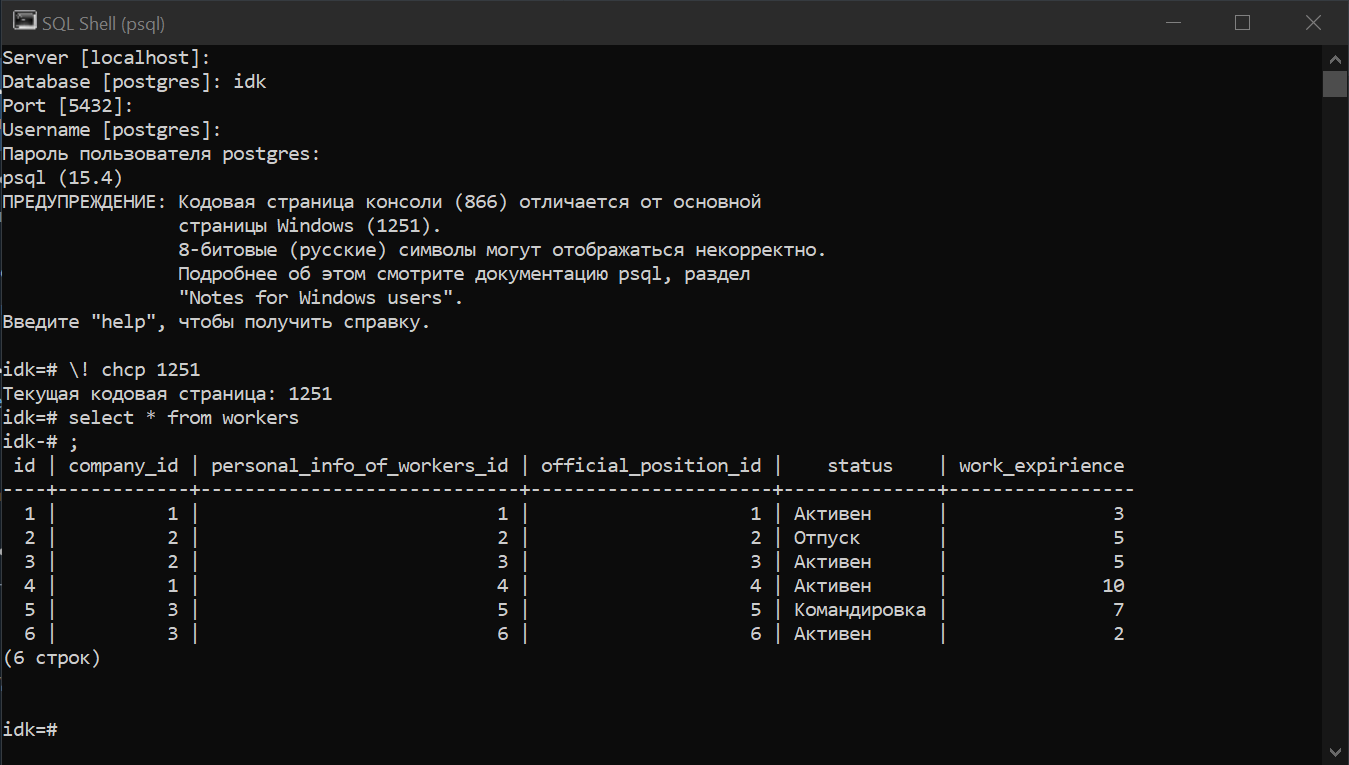


Рисунок 7 – Подключение к другой БД и вывод оттуда данных



Рисунок 8 – Вывод данных с условием из таблицы

Теперь переходим в pgAdmin4. pgAdmin 4 - это бесплатная и открытая платформа администрирования баз данных PostgreSQL. Она предоставляет широкий набор инструментов и функций для управления и администрирования баз данных PostgreSQL. MS SQL и PostgreSQL — это две разные системы управления базами данных (СУБД) с отличиями в функциональности, синтаксисе SQL и поддержке операционных систем.

1. Владение и лицензирование:

- MS SQL: Разработана и владеется компанией Microsoft. Существуют различные версии MS SQL Server, включая платные и бесплатные.

- PostgreSQL: Разработана и поддерживается открытым сообществом PostgreSQL Global Development Group. Она распространяется бесплатно и имеет открытый исходный код.

2. Поддерживаемые операционные системы:

- MS SQL: Поддерживает только Windows и Linux.

- PostgreSQL: Поддерживает большинство операционных систем, включая Windows, Linux, macOS и другие UNIX-подобные системы.

3. Синтаксис SQL:

- MS SQL: Использует T-SQL (Transact-SQL), который является расширением SQL с добавлением некоторых дополнительных функций и операторов.

- PostgreSQL: Поддерживает стандартный SQL с дополнительными возможностями, такими как хранимые процедуры, триггеры и пользовательские типы данных.

4. Расширяемость и гибкость:

- MS SQL: Обладает широким набором инструментов и функций, но ограничена в плане расширяемости, так как большинство расширений являются платными.

- PostgreSQL: Имеет открытый исходный код, что делает его более гибким и расширяемым. Существует большое количество расширений, доступных бесплатно.

5. Репликация и высокая доступность:

- MS SQL: Предлагает встроенные функции репликации и высокой доступности, такие как Always On Availability Groups.

- PostgreSQL: Также имеет возможности репликации и высокой доступности, такие как Streaming Replication и Logical Replication.

6. Производительность:

- MS SQL: Часто считается быстрее в некоторых сценариях, особенно при использовании оптимизированных индексов и хранимых процедур.

- PostgreSQL: Также обладает хорошей производительностью, особенно при использовании корректно настроенных индексов и запросов.

Начнём писать код. Код запроса можно начать писать, открыв в контекстном меню БД положение Query Tool.

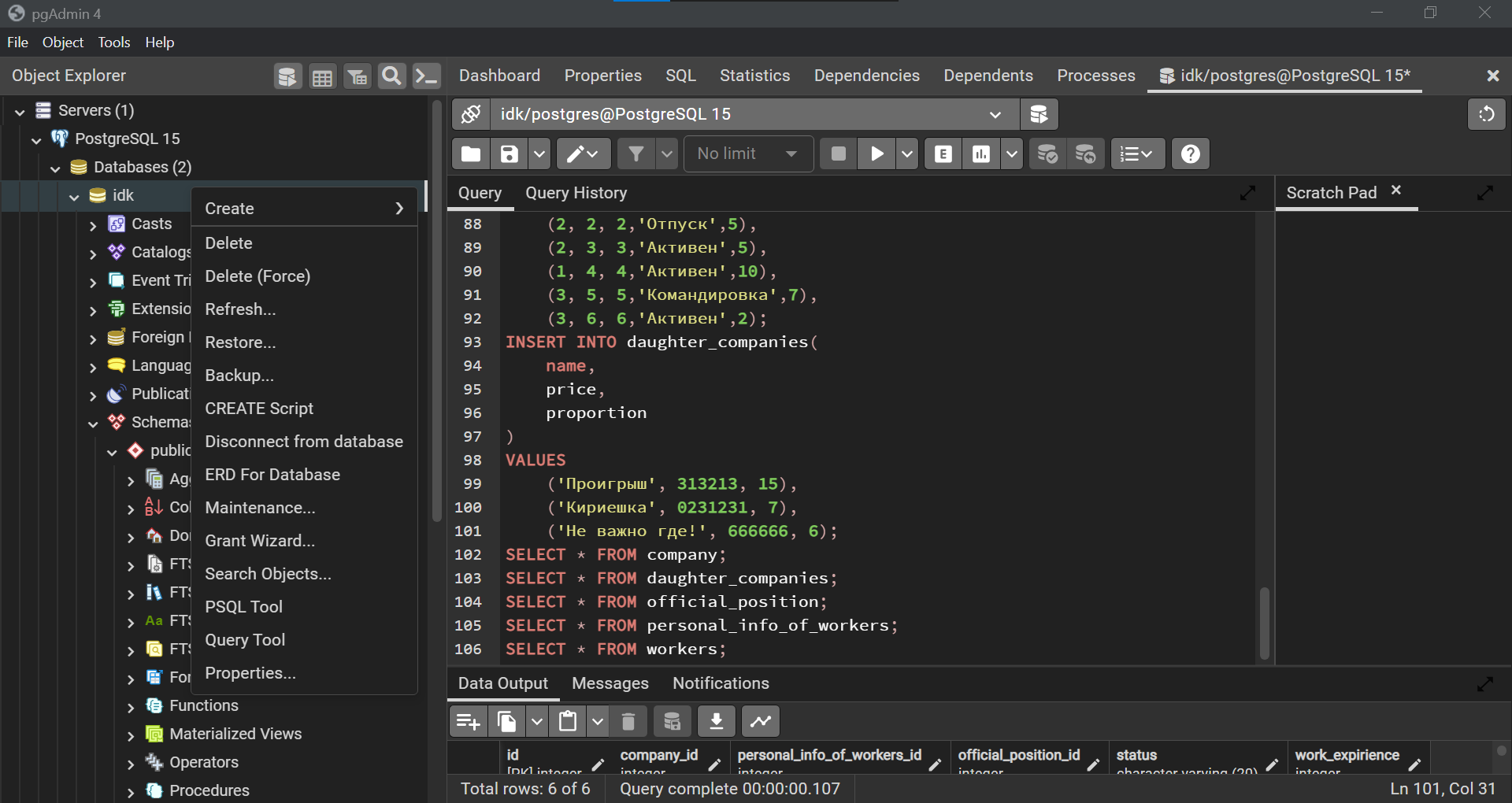


Рисунок 9 – pgAdmin4

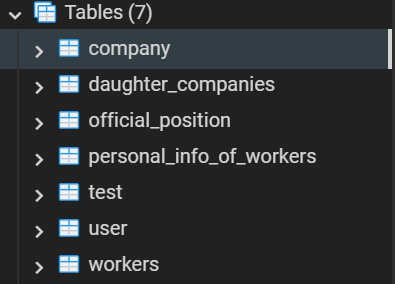


Рисунок 10 – Список таблиц в БД

Результаты работы вывода данных из БД:

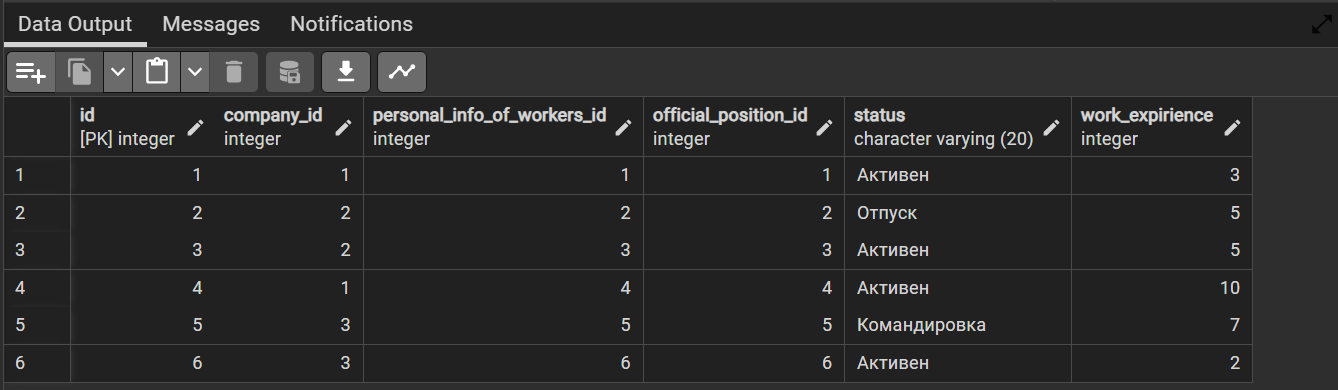


Рисунок 11 – Вывод данных таблицы company

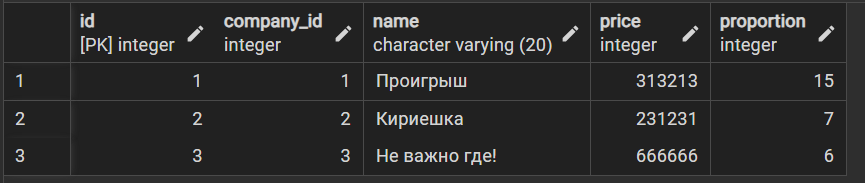


Рисунок 12 – Вывод данных таблицы daughter\_companies

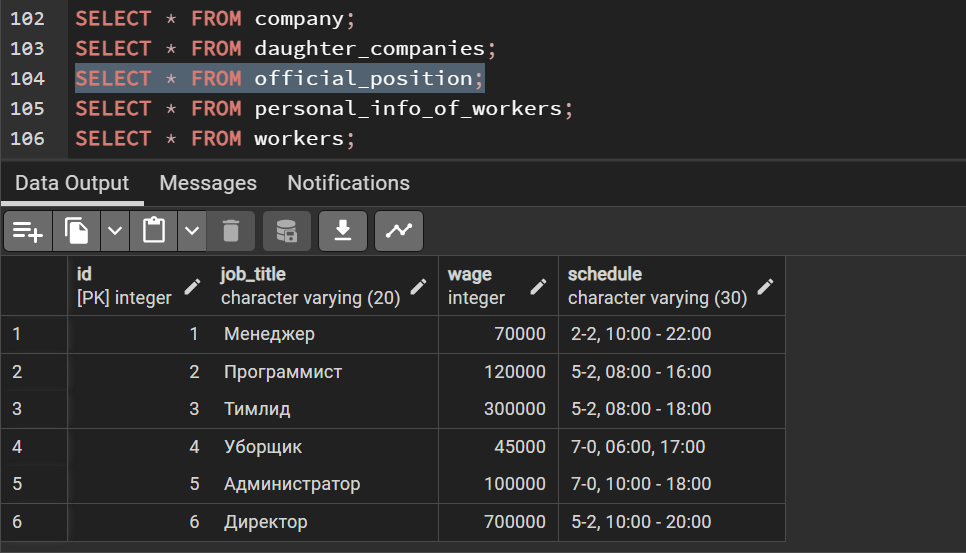


Рисунок 13 – Вывод данных таблицы official\_position

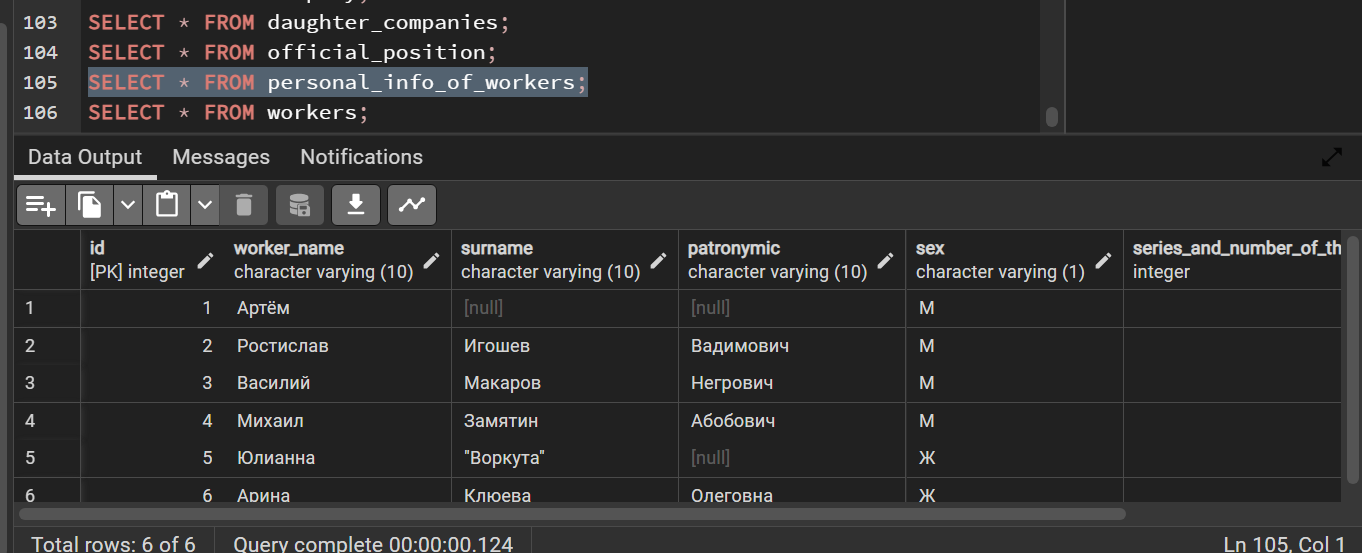


Рисунок 14 – Вывод данных таблицы personal\_info\_of\_workers



Рисунок 15 - Вывод данных таблицы workers

Некоторые части кода запроса:

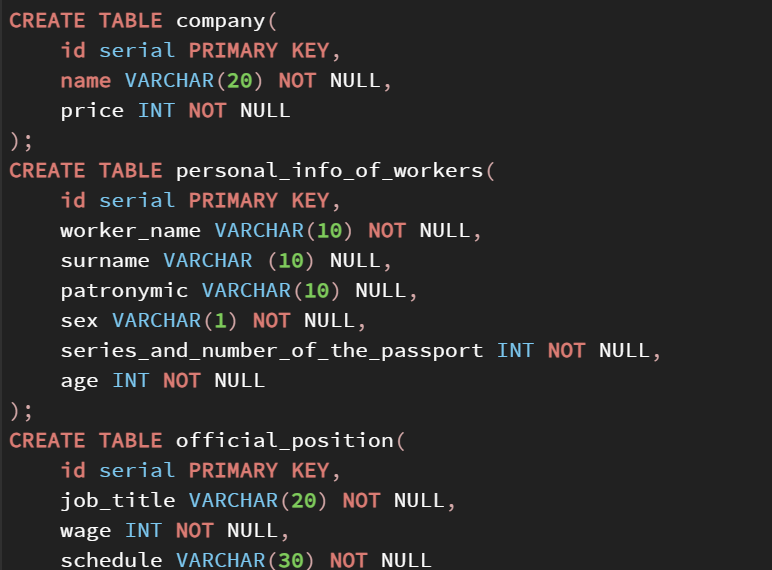


Рисунок 16 – Код создания таблиц

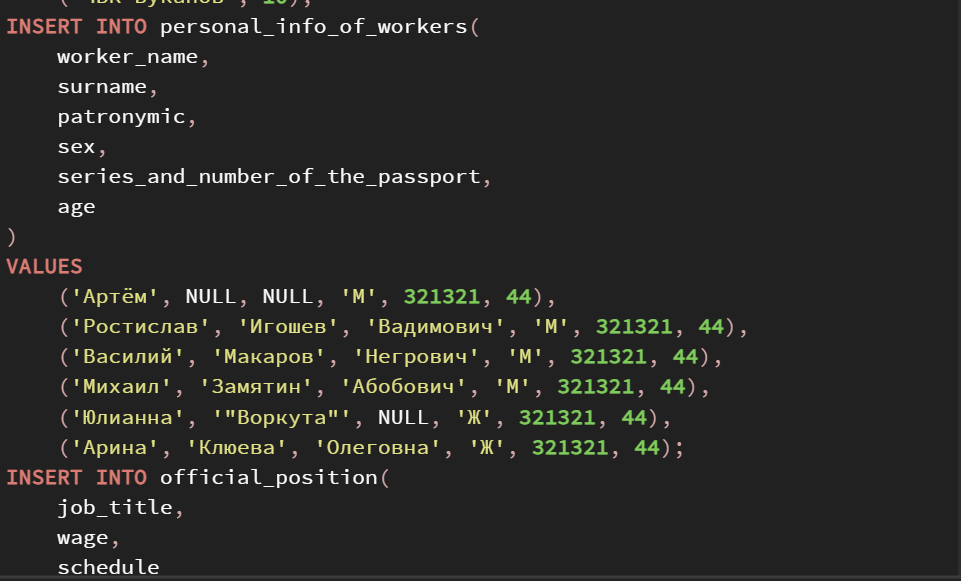


Рисунок 17 – Вставка данных в таблицу

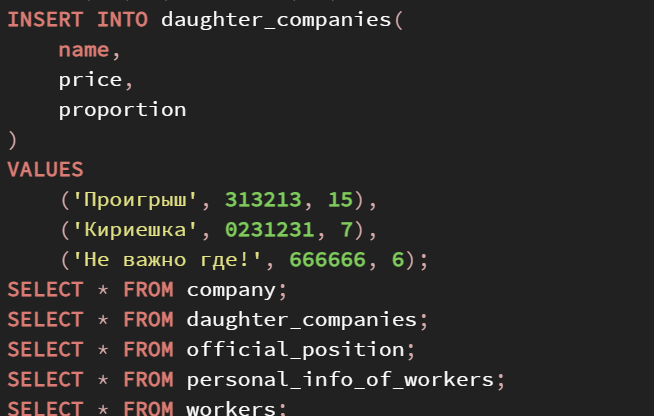


Рисунок 18 – Вставка данных в таблицу и вывод данных

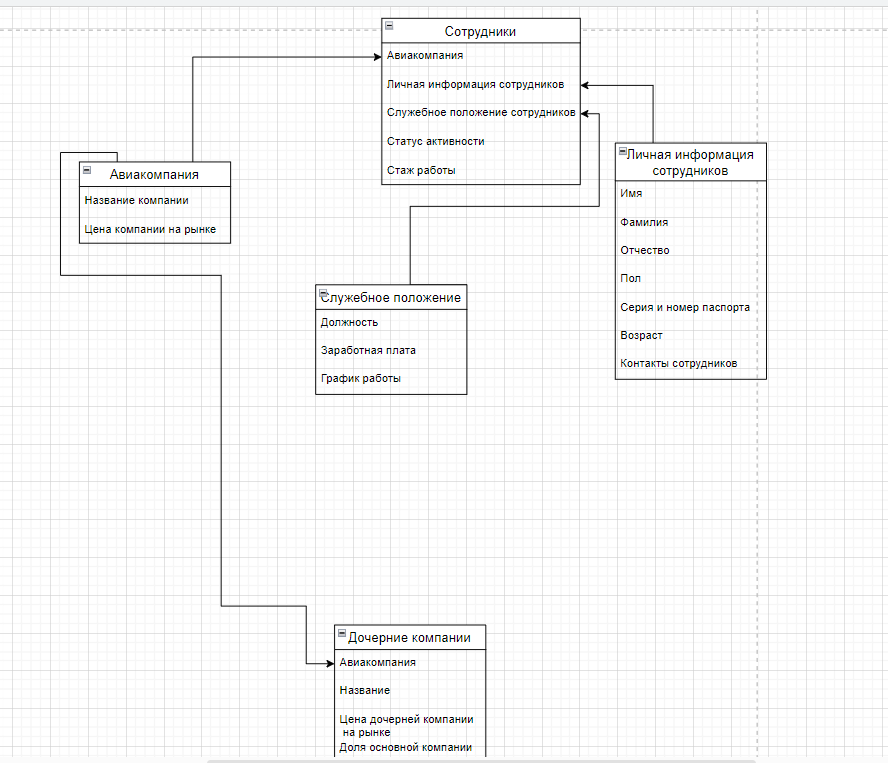


Рисунок 19 – Логическая модель данных

Полный код запроса:

CREATE TABLE company(

id serial PRIMARY KEY,

name VARCHAR(20) NOT NULL,

price INT NOT NULL

);

CREATE TABLE personal\_info\_of\_workers(

id serial PRIMARY KEY,

worker\_name VARCHAR(10) NOT NULL,

surname VARCHAR (10) NULL,

patronymic VARCHAR(10) NULL,

sex VARCHAR(1) NOT NULL,

series\_and\_number\_of\_the\_passport INT NOT NULL,

age INT NOT NULL

);

CREATE TABLE official\_position(

id serial PRIMARY KEY,

job\_title VARCHAR(20) NOT NULL,

wage INT NOT NULL,

schedule VARCHAR(30) NOT NULL

);

CREATE TABLE workers(

id serial PRIMARY KEY,

company\_id INT,

personal\_info\_of\_workers\_id INT,

official\_position\_id INT,

status varchar(20),

work\_expirience INT,

FOREIGN KEY (company\_id)

REFERENCES company(id),

FOREIGN KEY (personal\_info\_of\_workers\_id)

REFERENCES personal\_info\_of\_workers(id),

FOREIGN KEY (official\_position\_id)

REFERENCES official\_position(id)

);

CREATE TABLE daughter\_companies(

id serial PRIMARY KEY,

company\_id INT,

name VARCHAR(20) NOT NULL,

price INT NOT NULL,

proportion INT NOT NULL,

FOREIGN KEY (company\_id)

REFERENCES company(id)

);

INSERT INTO company(

name,

price

)

VALUES

('Аэрофлот', 1400000),

('ООО МПТ', 9999999),

('ЧВК Буканов', 10);

INSERT INTO personal\_info\_of\_workers(

worker\_name,

surname,

patronymic,

sex,

series\_and\_number\_of\_the\_passport,

age

)

VALUES

('Артём', NULL, NULL, 'M', 321321, 44),

('Ростислав', 'Игошев', 'Вадимович', 'M', 321321, 44),

('Василий', 'Макаров', 'Негрович', 'M', 321321, 44),

('Михаил', 'Замятин', 'Абобович', 'M', 321321, 44),

('Юлианна', '"Воркута"', NULL, 'Ж', 321321, 44),

('Арина', 'Клюева', 'Олеговна', 'Ж', 321321, 44);

INSERT INTO official\_position(

job\_title,

wage,

schedule

)

VALUES

('Менеджер', 70000, '2-2, 10:00 - 22:00'),

('Программист', 120000, '5-2, 08:00 - 16:00'),

('Тимлид', 300000, '5-2, 08:00 - 18:00'),

('Уборщик', 45000, '7-0, 06:00, 17:00'),

('Администратор', 100000, '7-0, 10:00 - 18:00'),

('Директор', 700000, '5-2, 10:00 - 20:00');

INSERT INTO workers(

company\_id,

personal\_info\_of\_workers\_id,

official\_position\_id,

status,

work\_expirience

)

VALUES

(1, 1, 1,'Активен',3),

(2, 2, 2,'Отпуск',5),

(2, 3, 3,'Активен',5),

(1, 4, 4,'Активен',10),

(3, 5, 5,'Командировка',7),

(3, 6, 6,'Активен',2);

INSERT INTO daughter\_companies(

name,

price,

proportion

)

VALUES

('Проигрыш', 313213, 15),

('Кириешка', 0231231, 7),

('Не важно где!', 666666, 6);

SELECT \* FROM company;

SELECT \* FROM daughter\_companies;

SELECT \* FROM official\_position;

SELECT \* FROM personal\_info\_of\_workers;

SELECT \* FROM workers;

Вывод: научились работать с PostgreSql, используя её консольную и графическую оболочку, создали 5 таблиц в качестве закрепления полученных знаний.

# ПРАКТИЧЕСКАЯ РАБОТА №2

Тема: «DCL»

Цель работы: Создать роли и пользователей, выдать права и проверить работоспособность.

Для начала следует создать самих пользователей с паролями и присвоить им нужные разрешения.

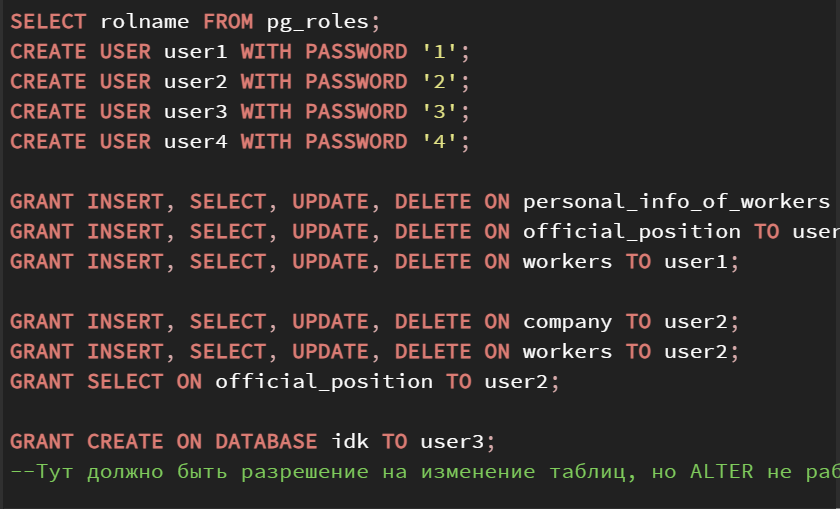
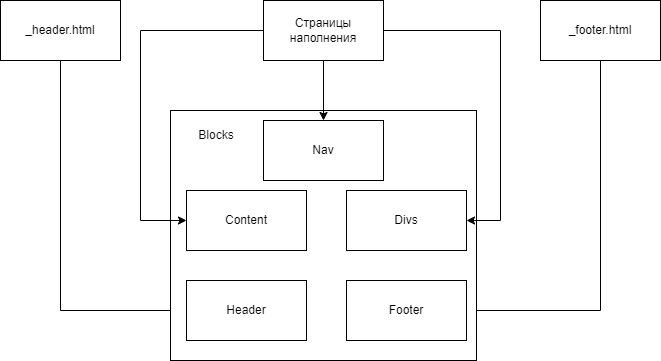


Рисунок 20 – Код запроса



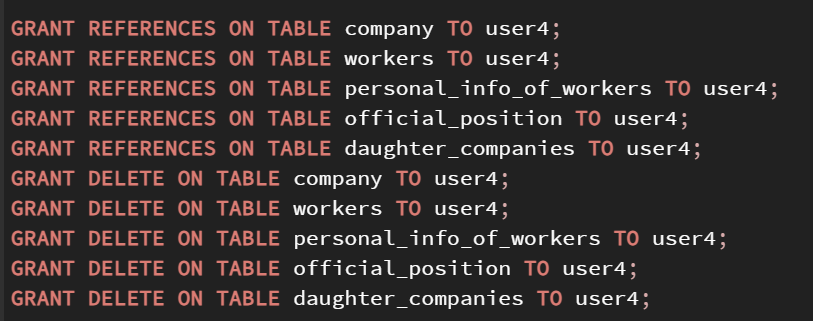


Рисунок 21 – Код запроса присвоения прав к четвертому пользователю



Рисунок 22 – Смена владельца таблицы

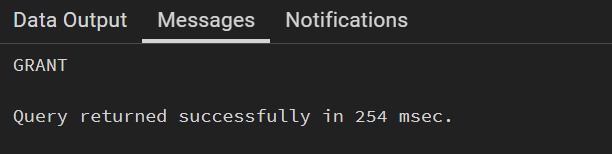


Рисунок 23 – Выполнение запроса

Код выполнился. Попробуем создать таблицу:

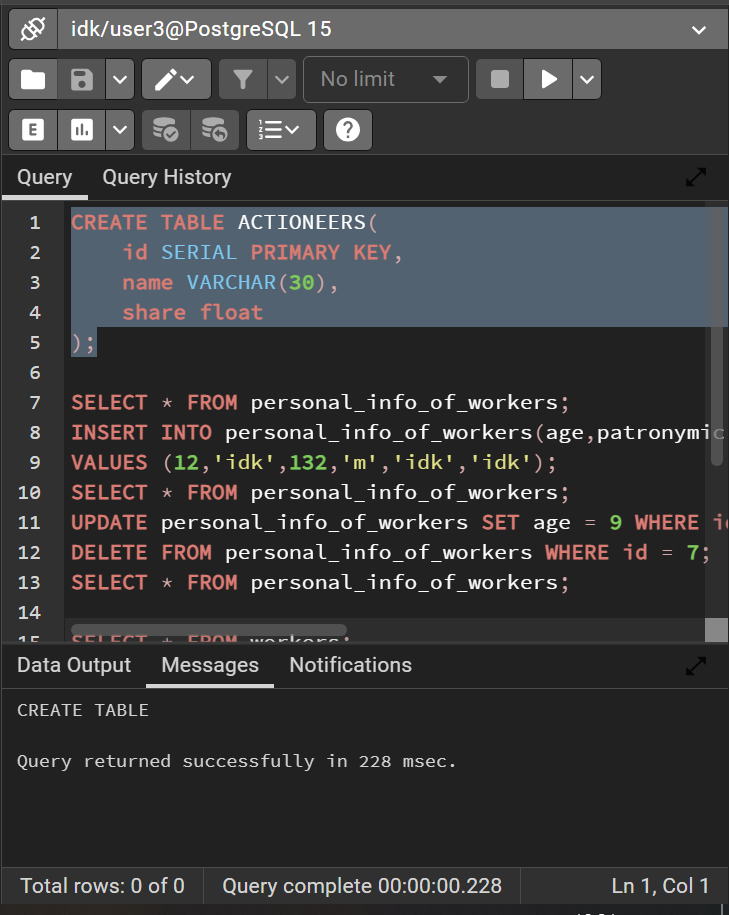


Рисунок 24 – Попытка создать таблицу

Как видно, таблица успешно создалась.

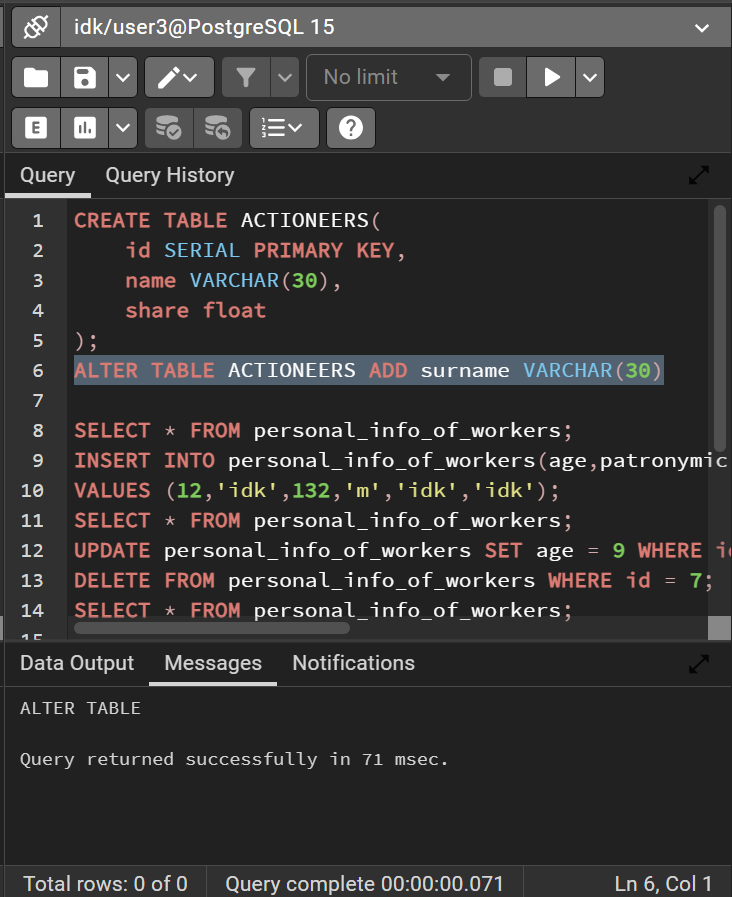


Рисунок 25 – Изменение созданной таблицы

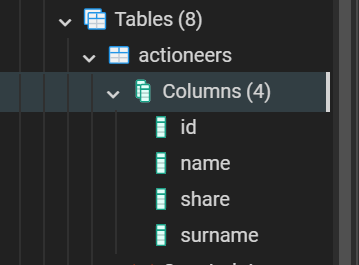


Рисунок 26 – Созданная таблица

На очереди – создание ролей.

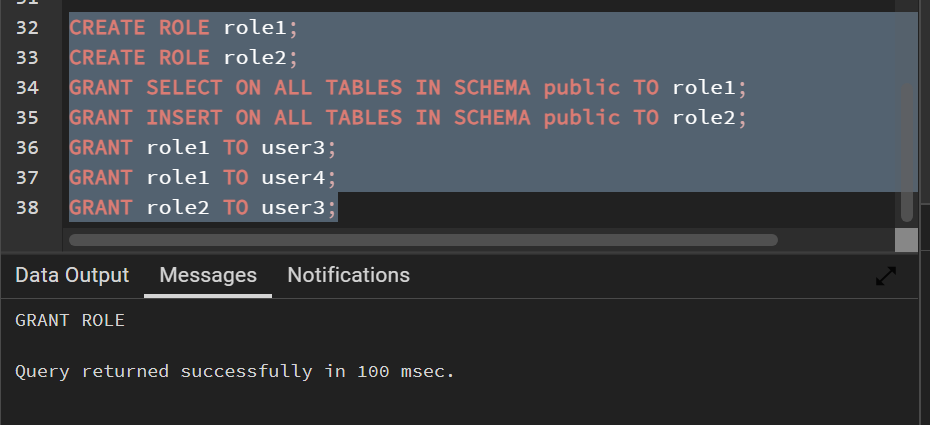


Рисунок 27 – Создание ролей и добавление в них участников

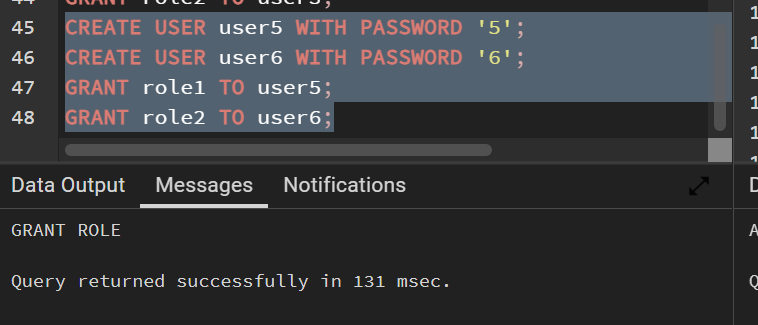


Рисунок 28 – Создание новых пользователей и добавление их в роли

Теперь необходимо проверить созданные выше разрешения и роли. Ниже будут предоставлены результаты работы:

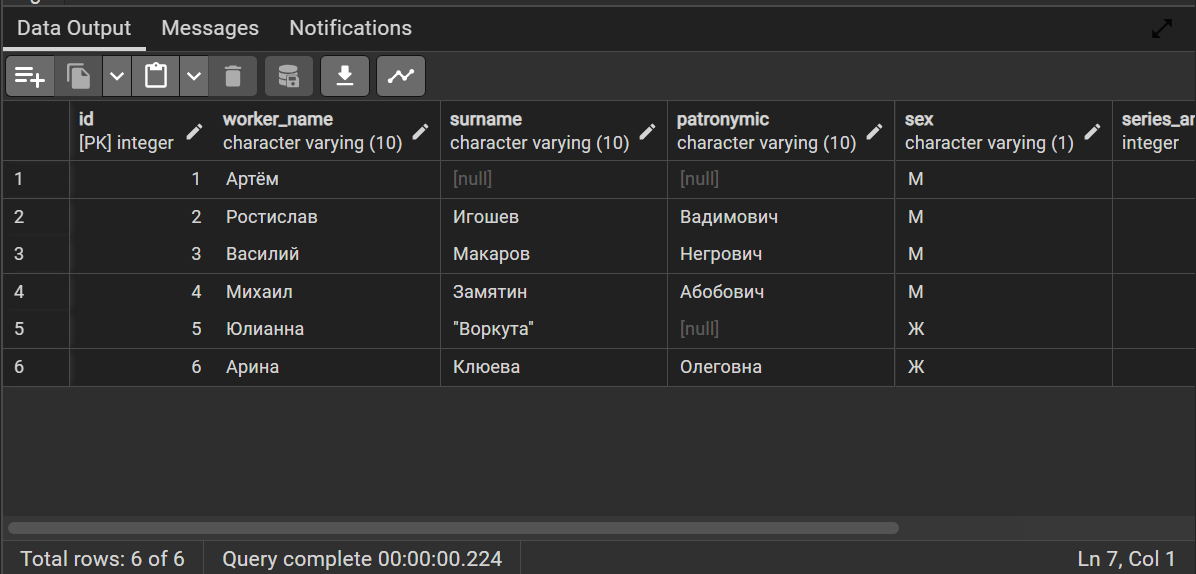


Рисунок 29 – Таблица до изменений



Рисунок 30 – Таблица после добавления информации

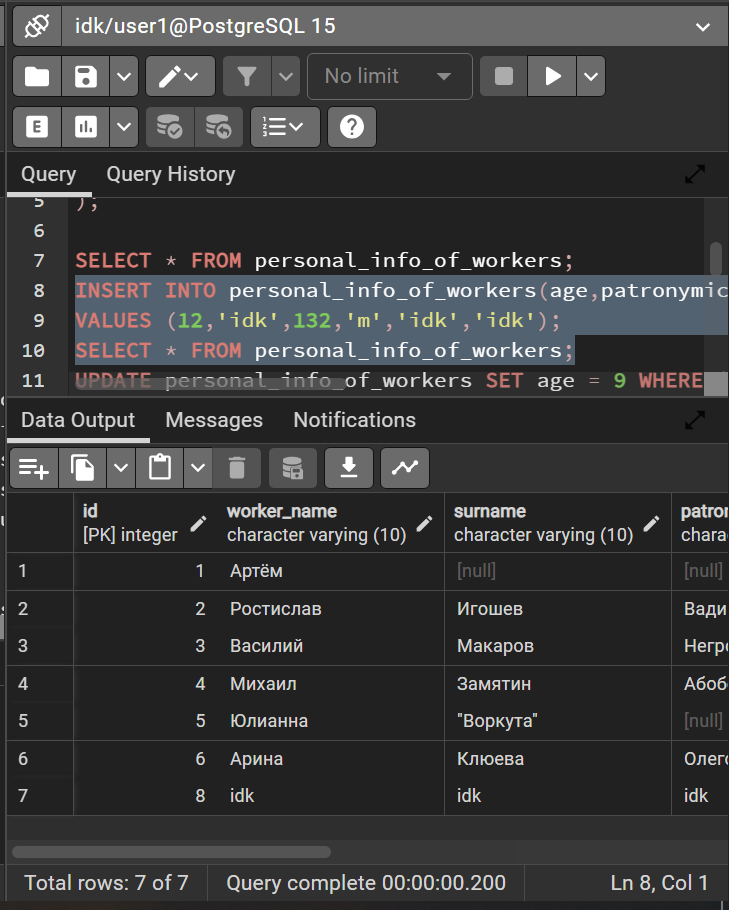


Рисунок 31 – Добавление информации, но видно подключение

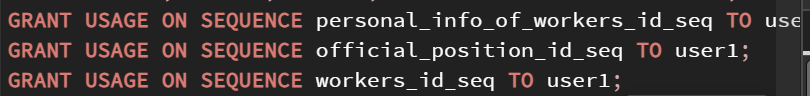


Рисунок 32 – Необходимые разрешения

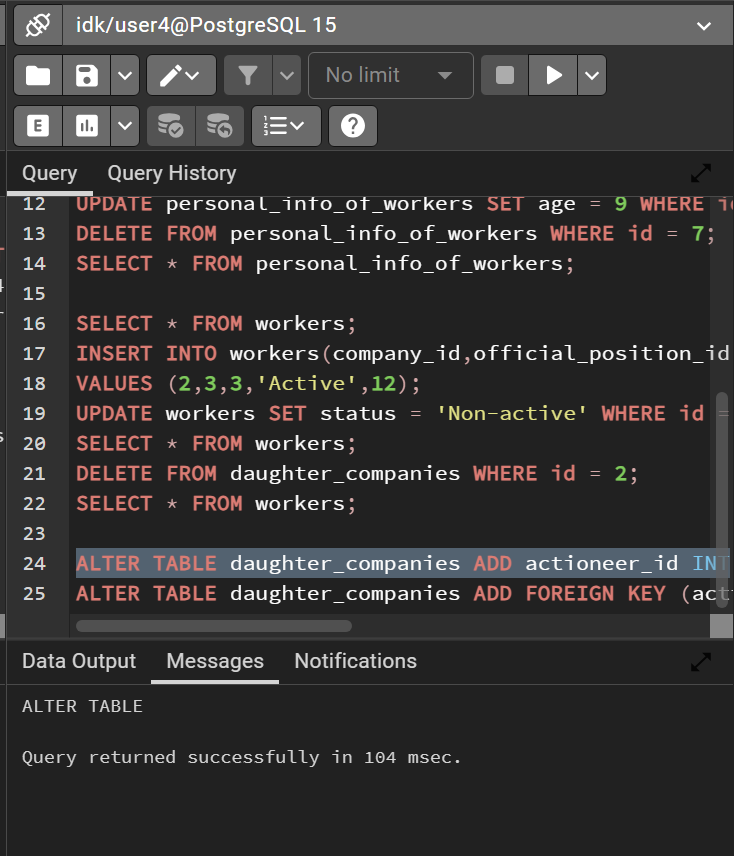


Рисунок 33 – Создание столбца

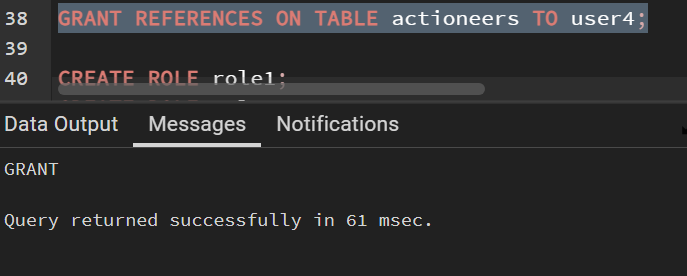


Рисунок 34 – Добавление разрешения на новую таблицу

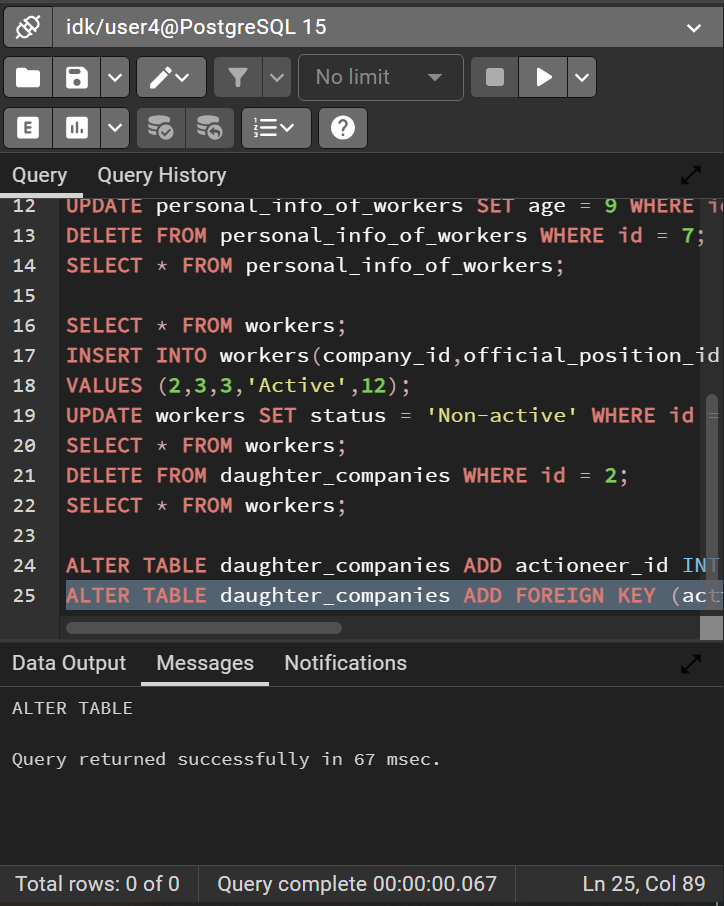


Рисунок 35 – Добавление внешнего ключа

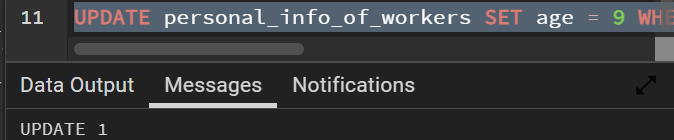


Рисунок 36 – Изменение таблицы

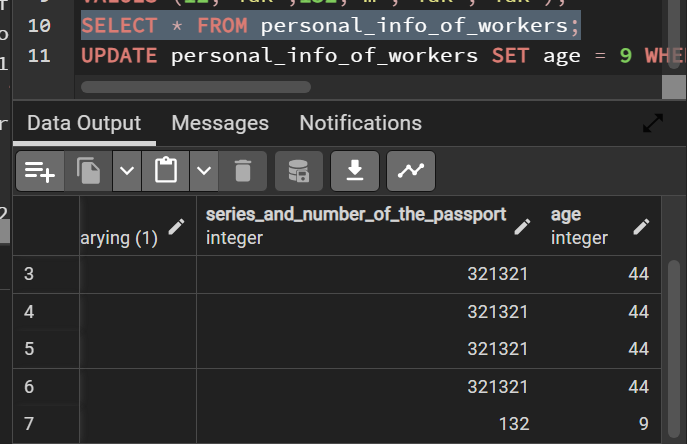


Рисунок 37 – Вывод данных таблицы

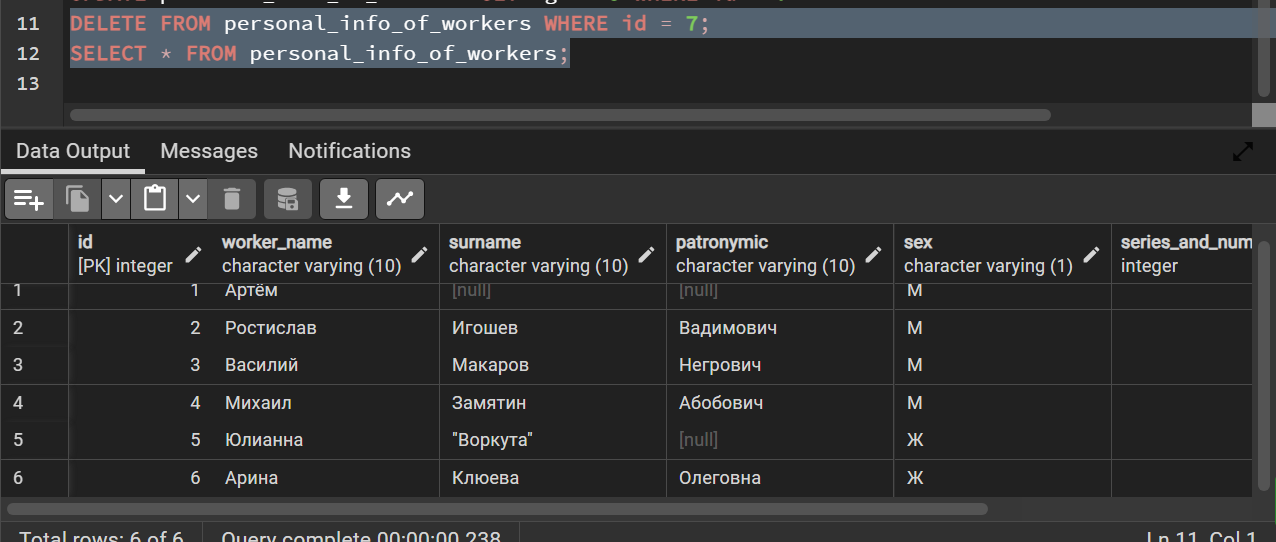


Рисунок 38 – Удаление данных из таблицы

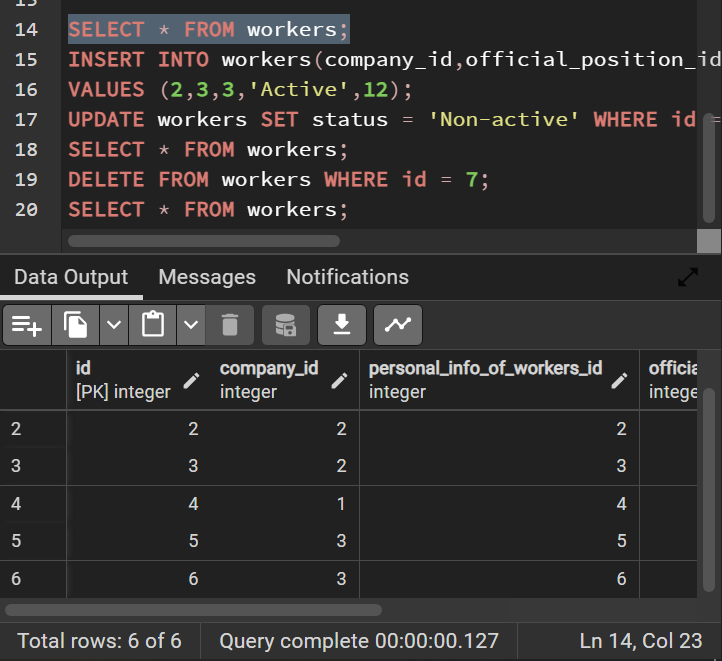


Рисунок 39 – Таблица до изменений

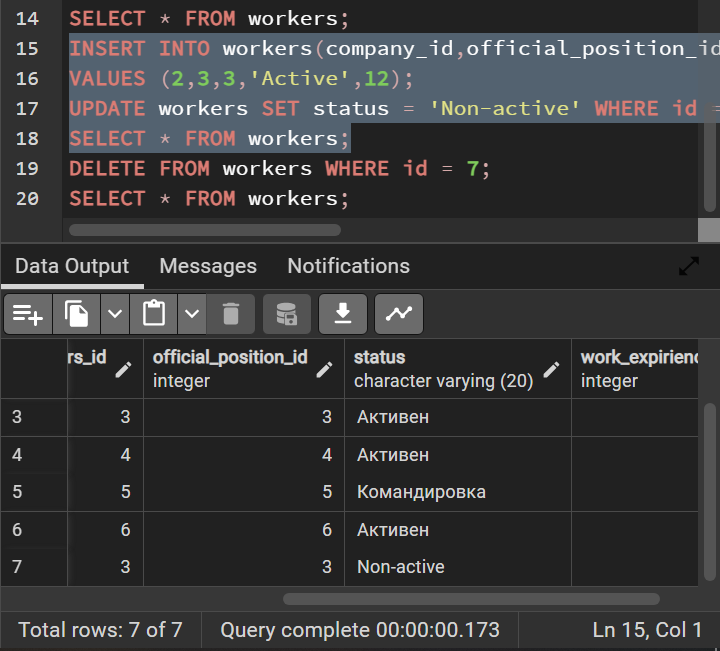


Рисунок 40 – Добавление данных в таблицу и изменение их

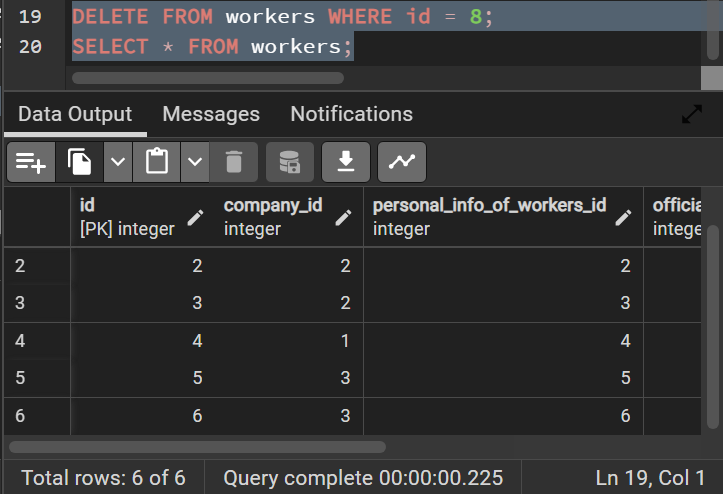


Рисунок 41 – Удаление данных из таблицы

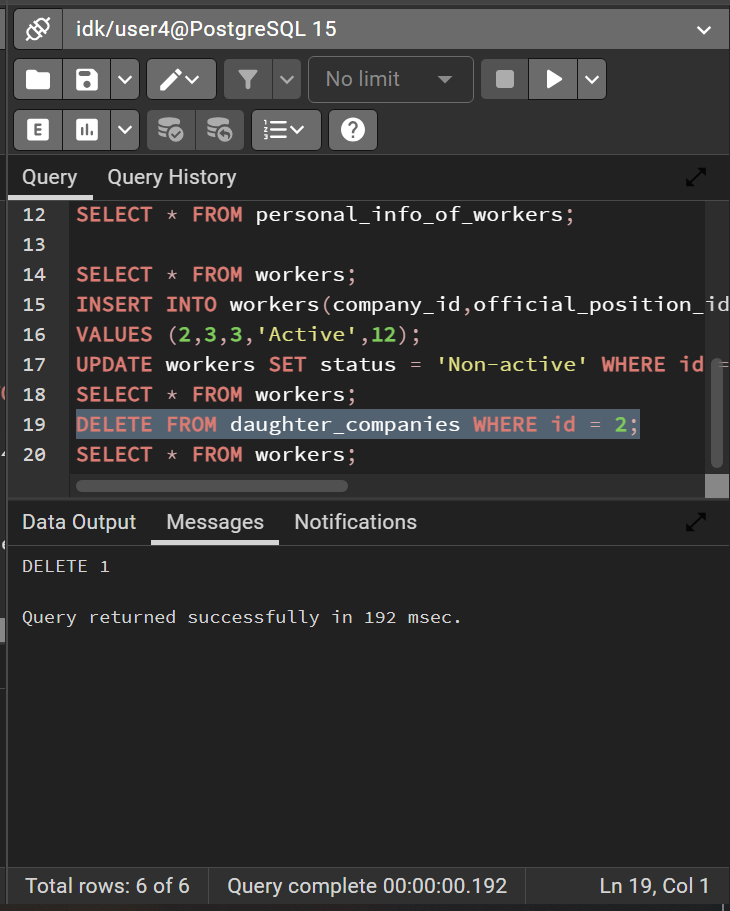


Рисунок 42 – Удаление данных из таблицы

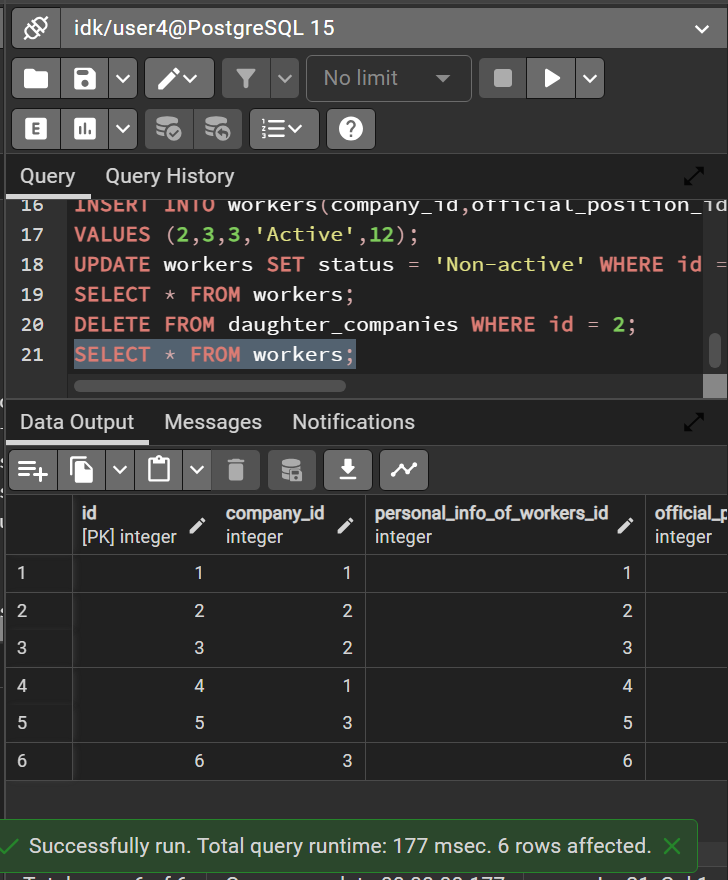


Рисунок 43 – Вывод данных из таблицы (Доступно благодаря роли)

Вывод: Создали роли и пользователей, выдали права и проверили их работоспособность.

# ПРАКТИЧЕСКАЯ РАБОТА №3

Тема: «Маршрутизация»

Цель работы: Ознакомиться с построение маршрутов в Django и реализовать маршрутизацию по web-приложению.

Для начала следует создать функции возврата страниц html, они же view-функции.

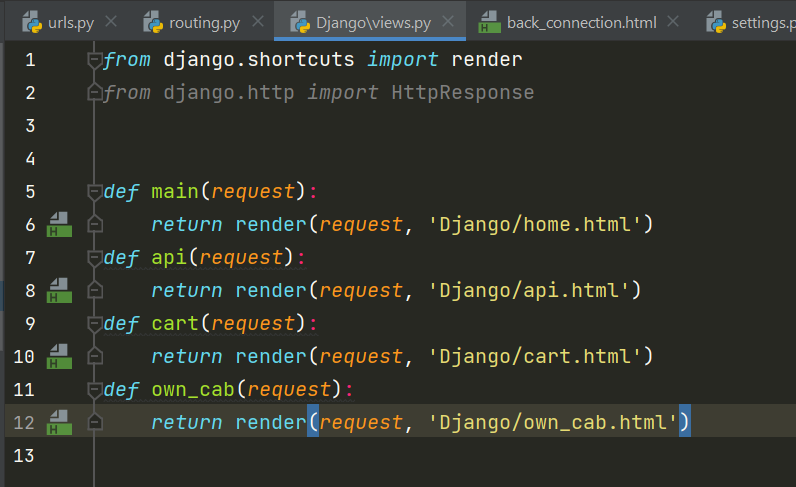


Рисунок 44 – Views проекта

Теперь, создаем url пути.

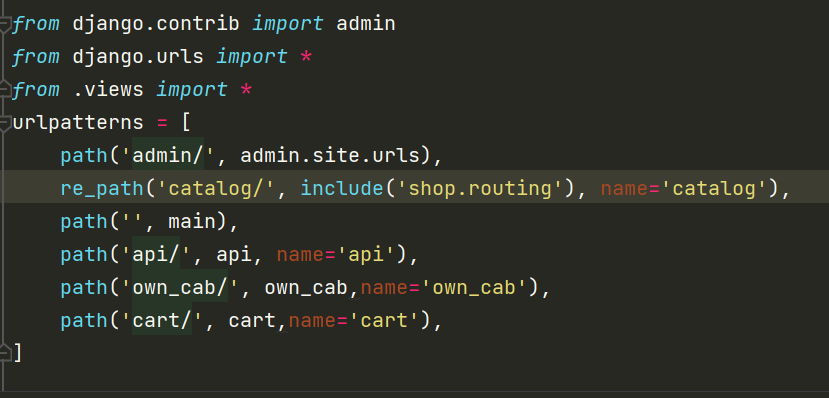


Рисунок 45 – Url проекта

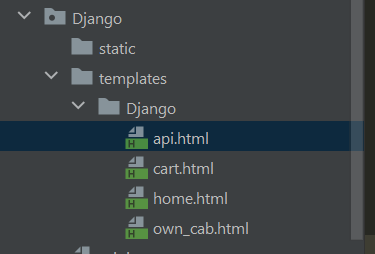


Рисунок 46 – Структура проекта

Теперь следует приступить к созданию html-файлов, которые и будут отображать информацию на сайте.

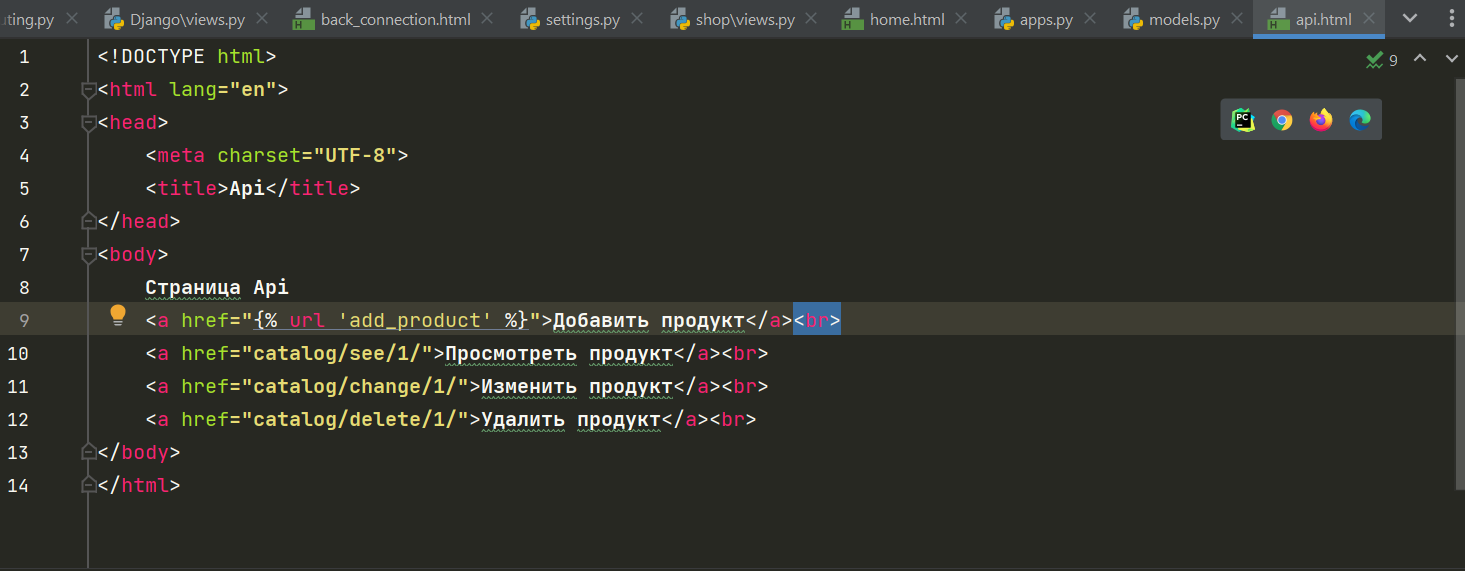


Рисунок 47 – Страница Апи

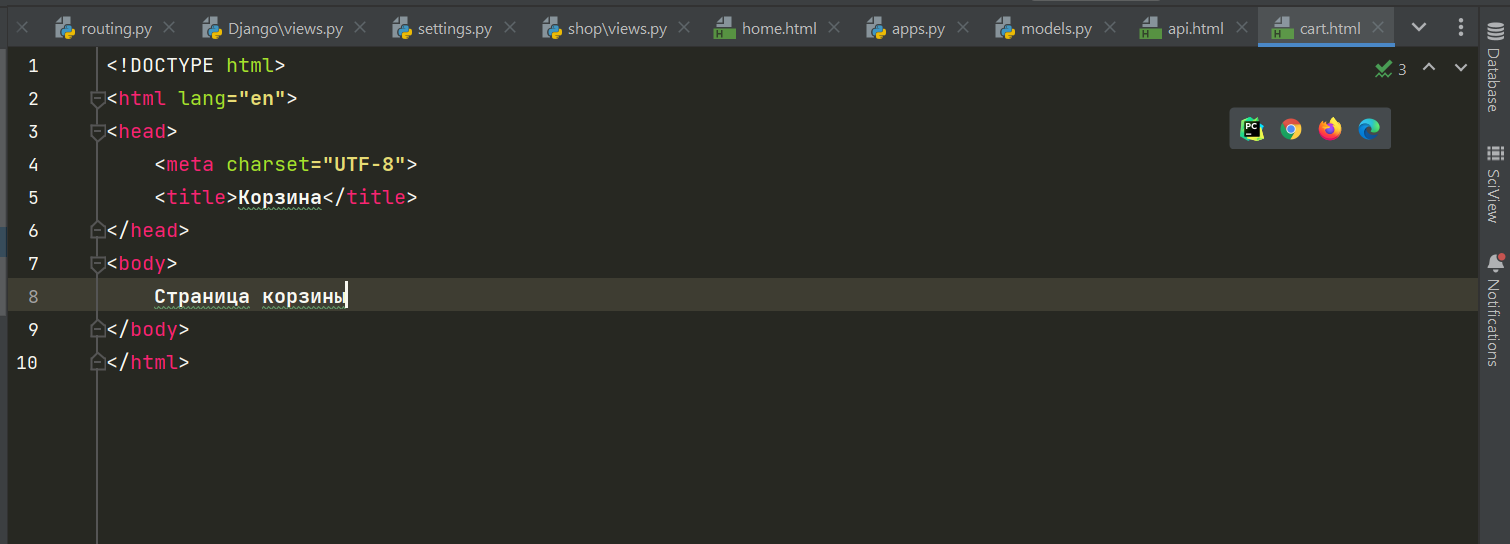


Рисунок 48 – Страница корзины

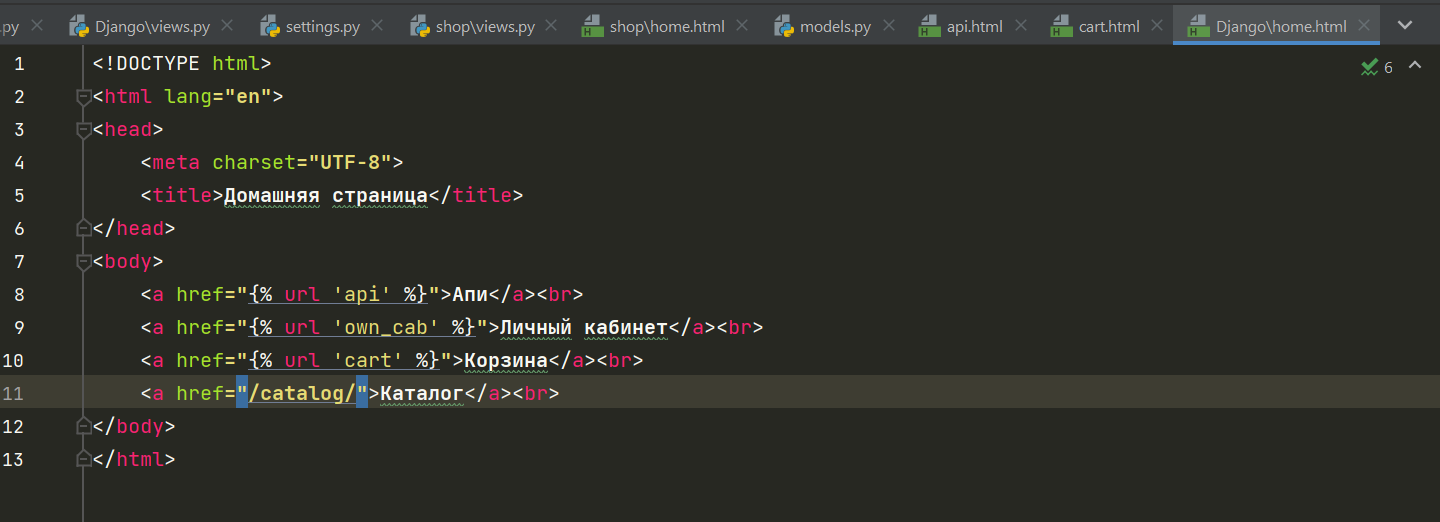


Рисунок 49 – Страница каталога

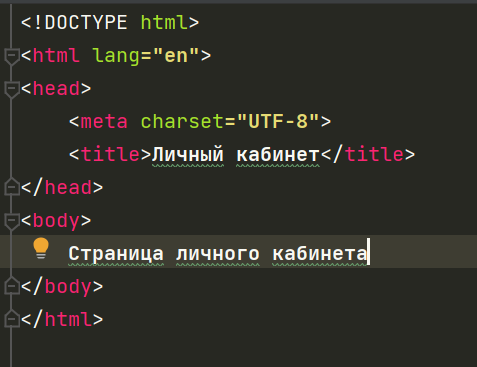


Рисунок 50 – Страница личного кабинета

После следует приступить к настройке приложения. В него нужно добавить пути адресации с переменными внутри, и реализовать функции для возврата верстки.

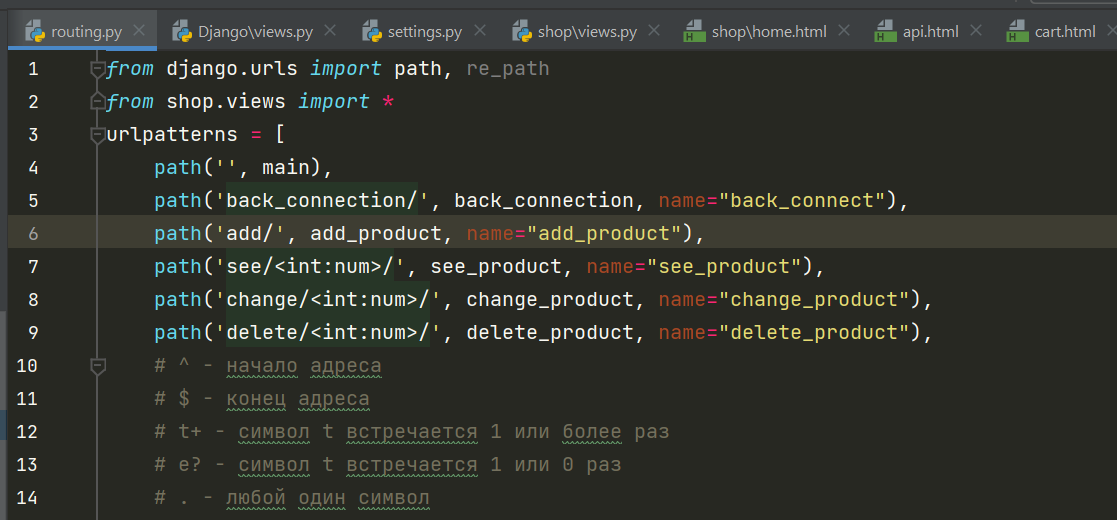


Рисунок 51 – Файл адресации

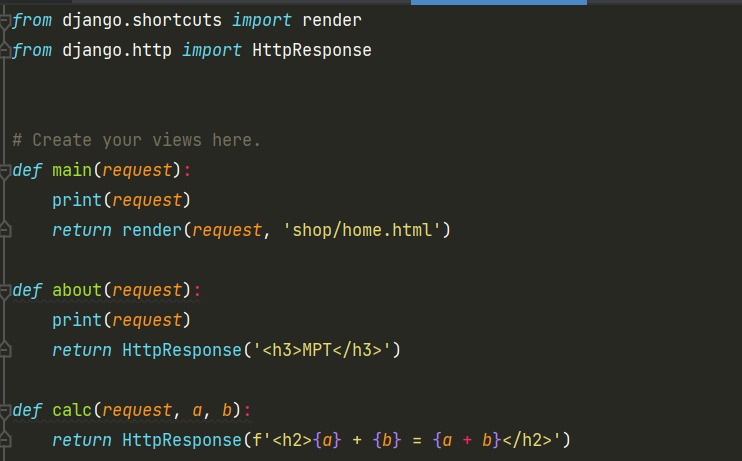


Рисунок 52 – Views функции приложения

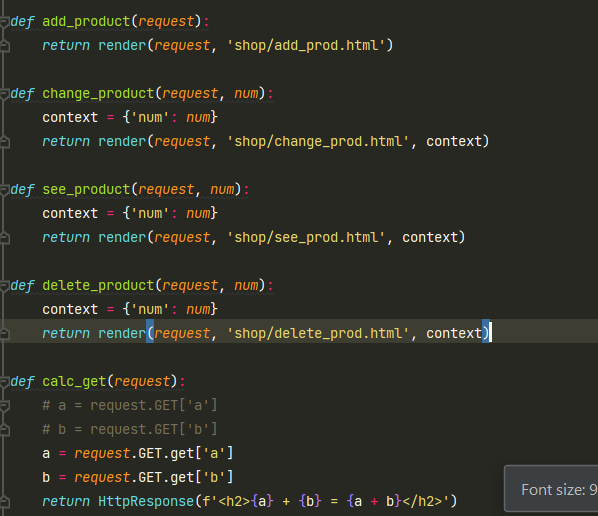


Рисунок 53 – Views функции приложения

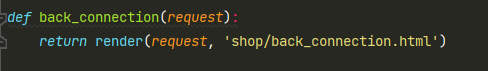


Рисунок 54 – View функция приложения

Структура html-файлов будет выглядеть таким образом:

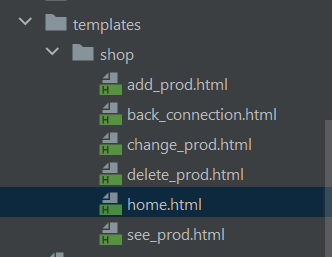


Рисунок 55 – Структура приложения

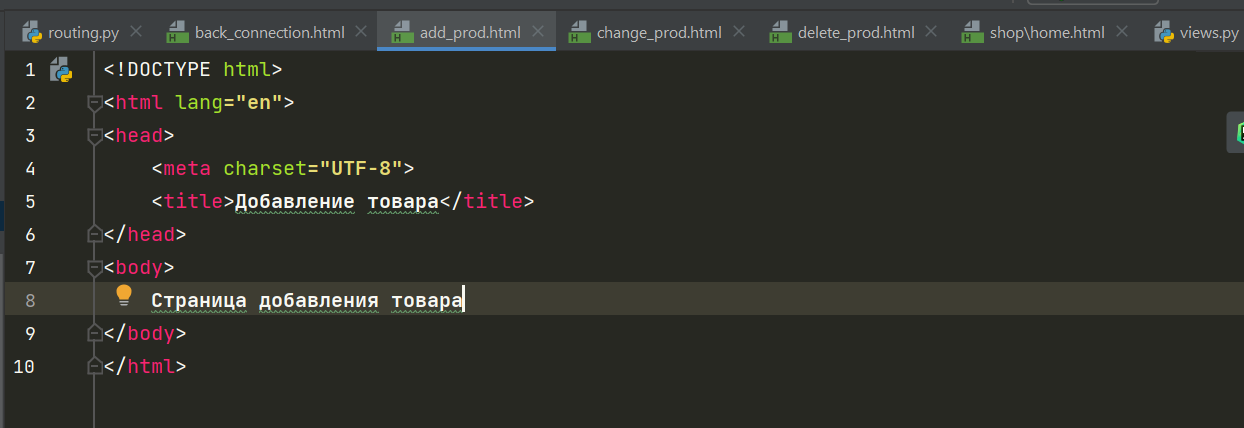


Рисунок 56 – Страница добавления товара

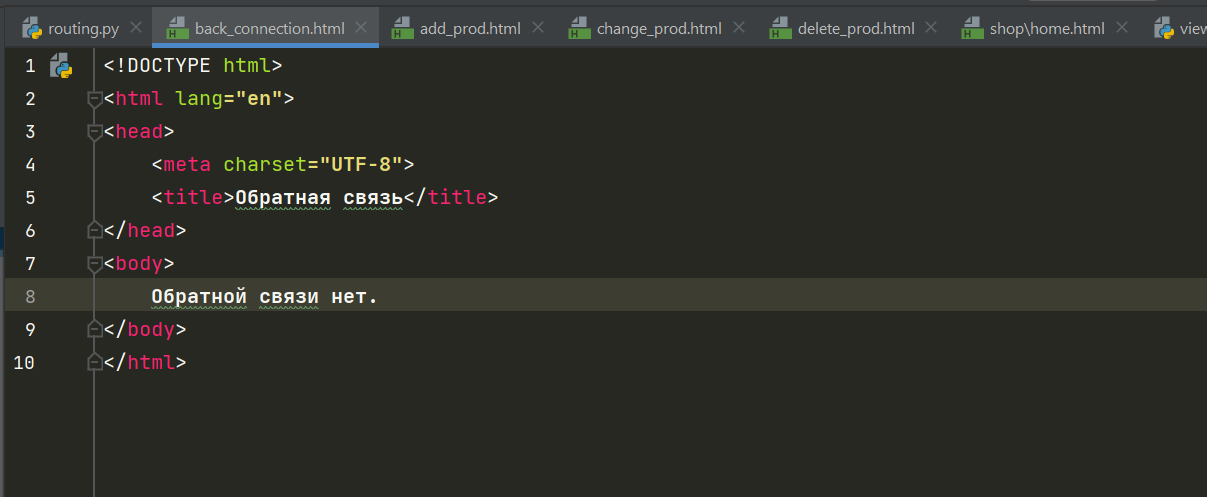


Рисунок 57 – Страница обратной связи



Рисунок 58 – Страница изменения товара

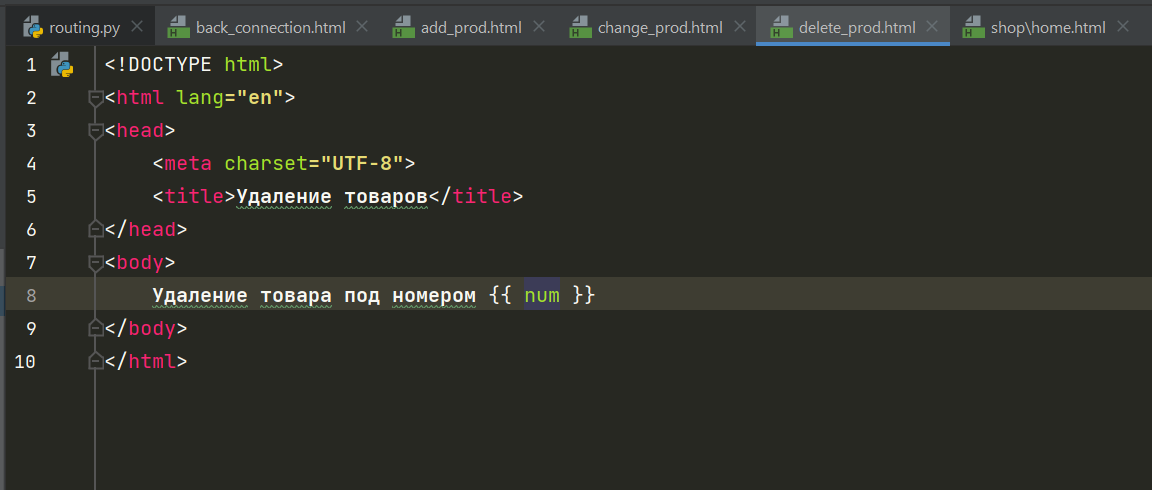


Рисунок 59 – Страница удаления товара



Рисунок 60 – Страница каталога

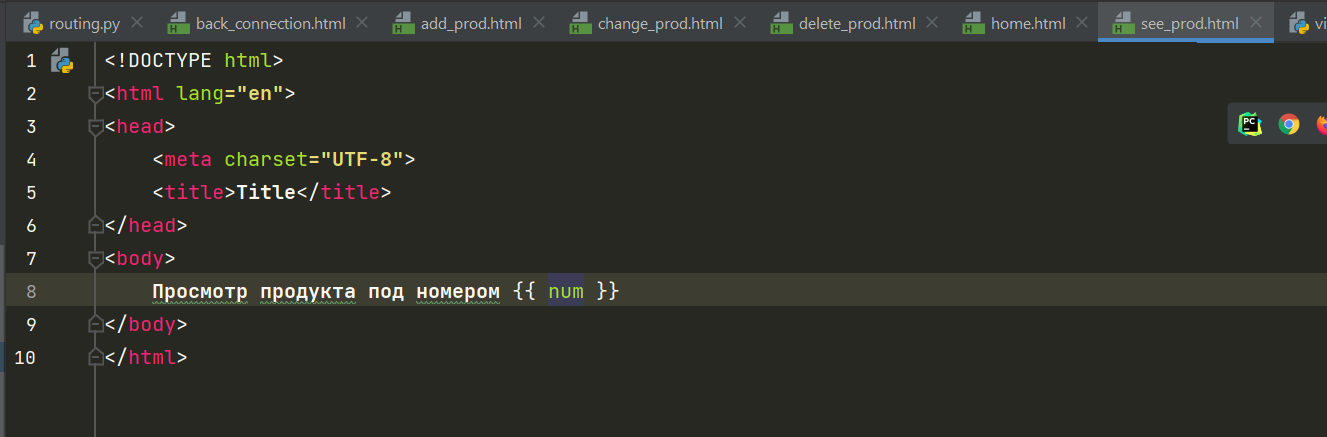


Рисунок 61 – Страница просмотра

Результаты работы:

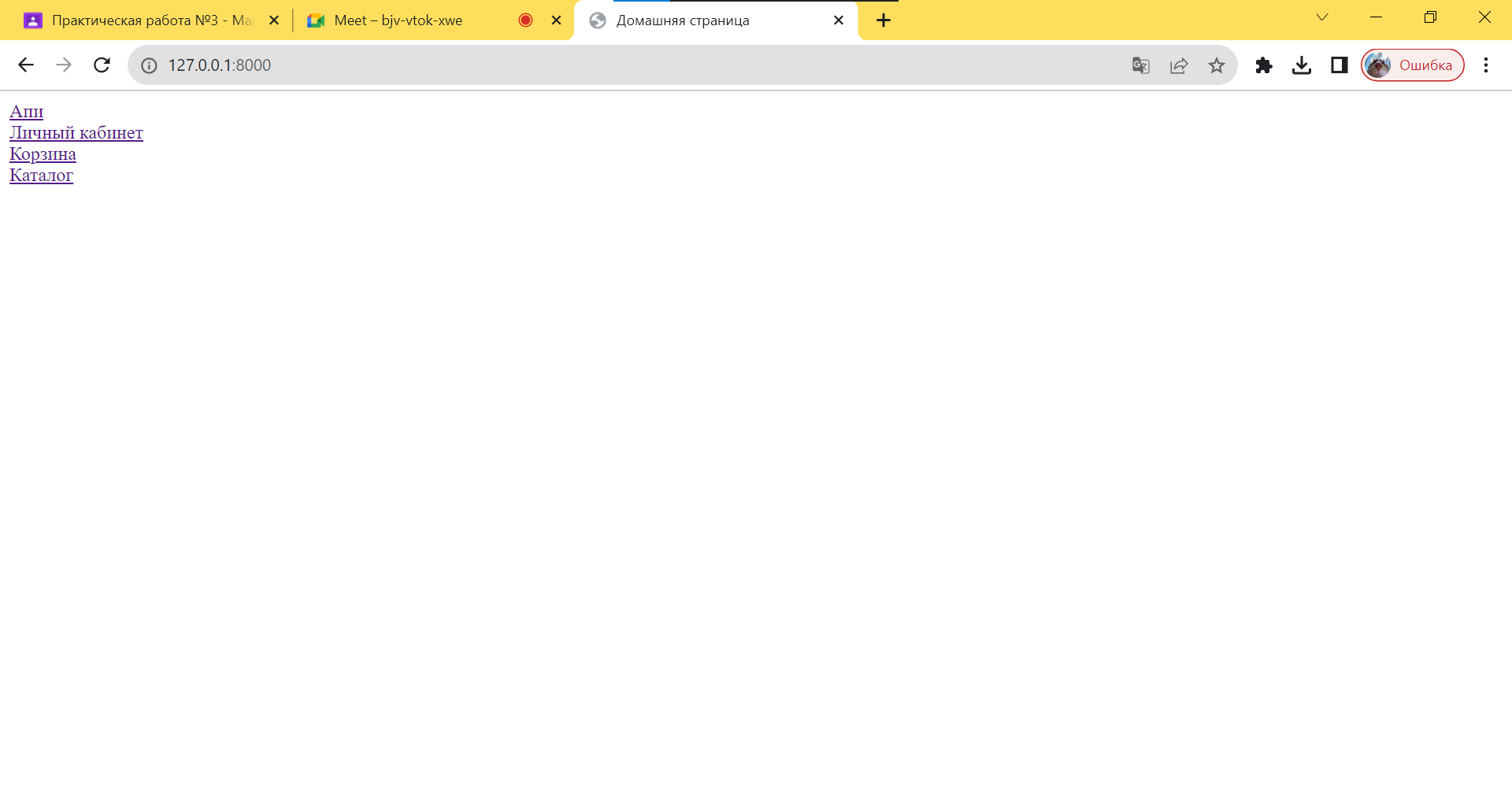


Рисунок 62 – Домашняя страница

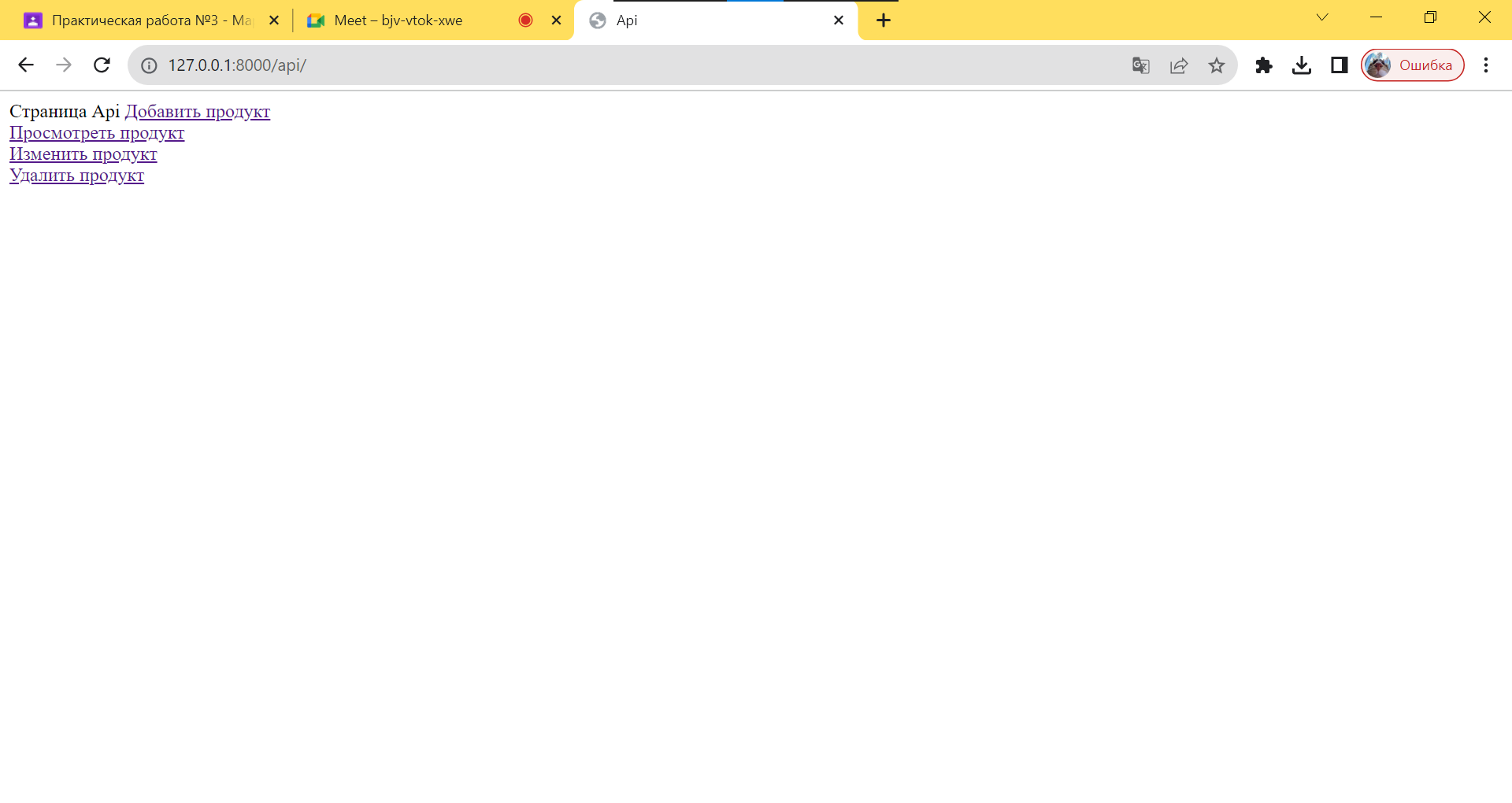


Рисунок 63 – Страница Апи

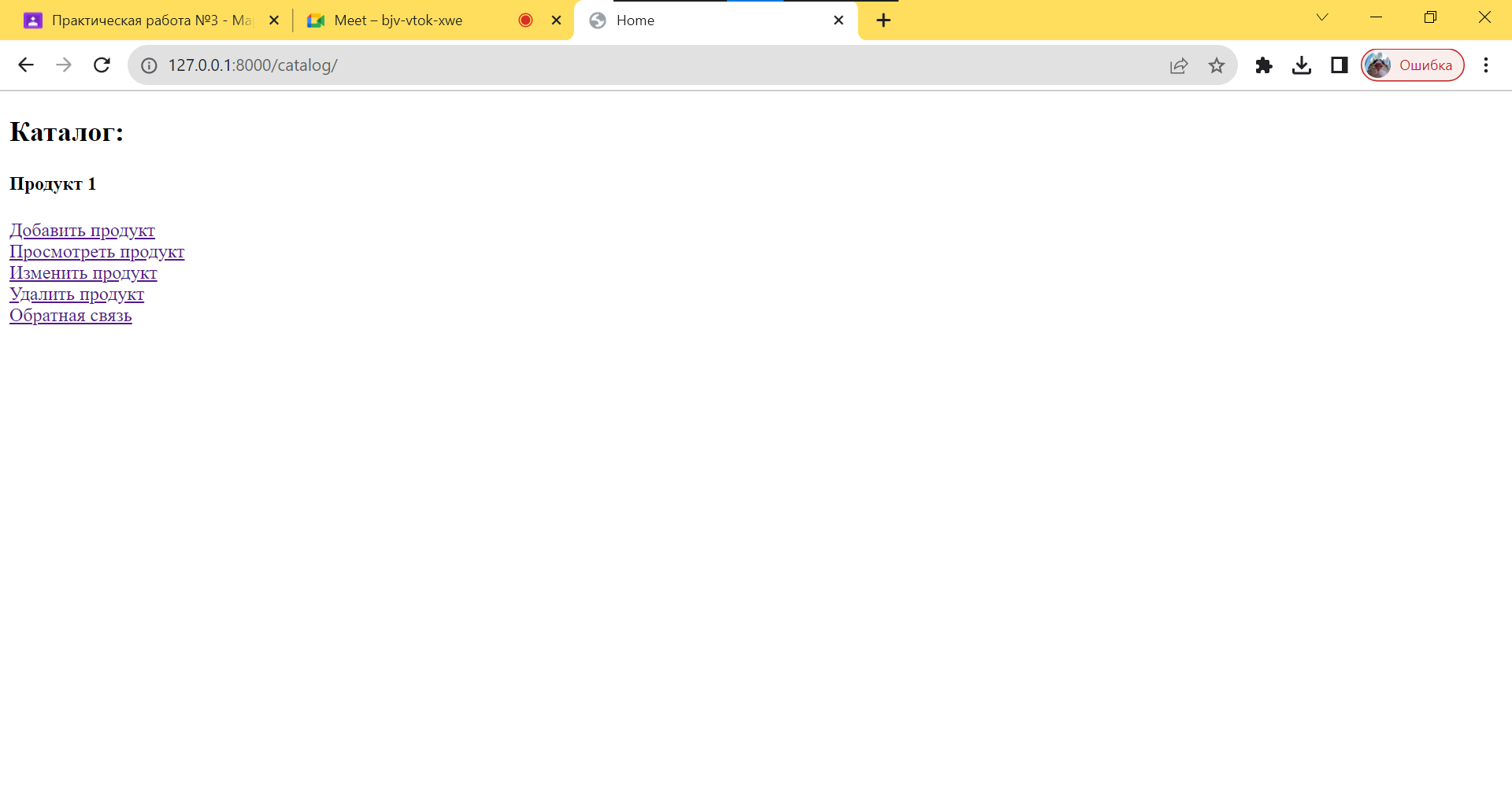


Рисунок 64 – Страница каталога

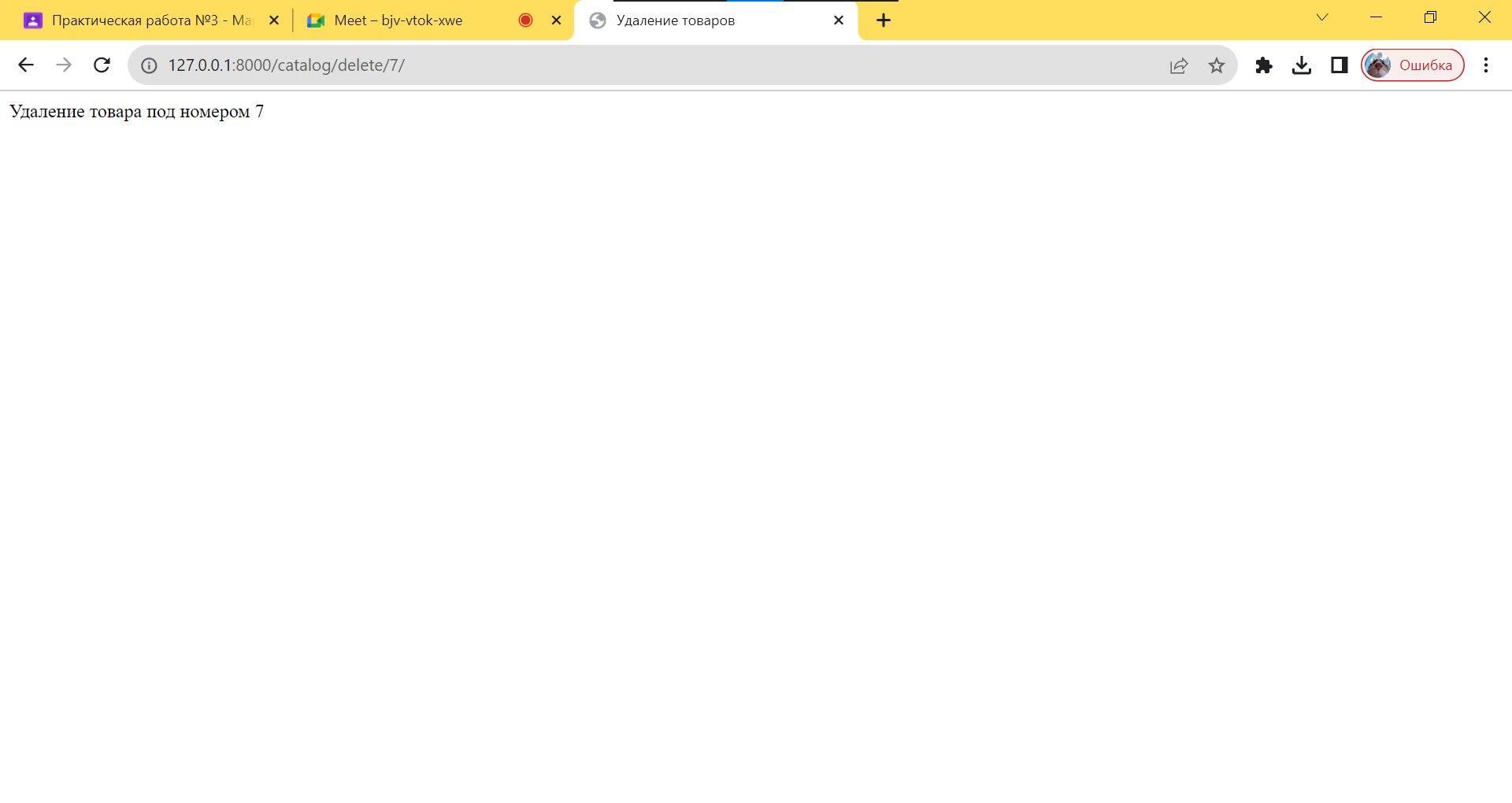


Рисунок 65 – Страница удаления товара под номером 7

Вывод: в ходе работы была изучена адресация и маршрутизация, взаимодействие проекта и приложений, передача данных между файлами и отображение полученных данных на странице верстки.

# ПРАКТИЧЕСКАЯ РАБОТА №4

Тема: «Шаблоны»

Цель работы: Ознакомиться с работой шаблонов и реализовать наследование шаблонов.

Начать следует с самого скелета. Расположение блоков в данном файле будет влиять на отображение всех прочих страниц, которые используют данный файл в качестве скелета, при чем вне зависимости от расположения блоков на самой странице.

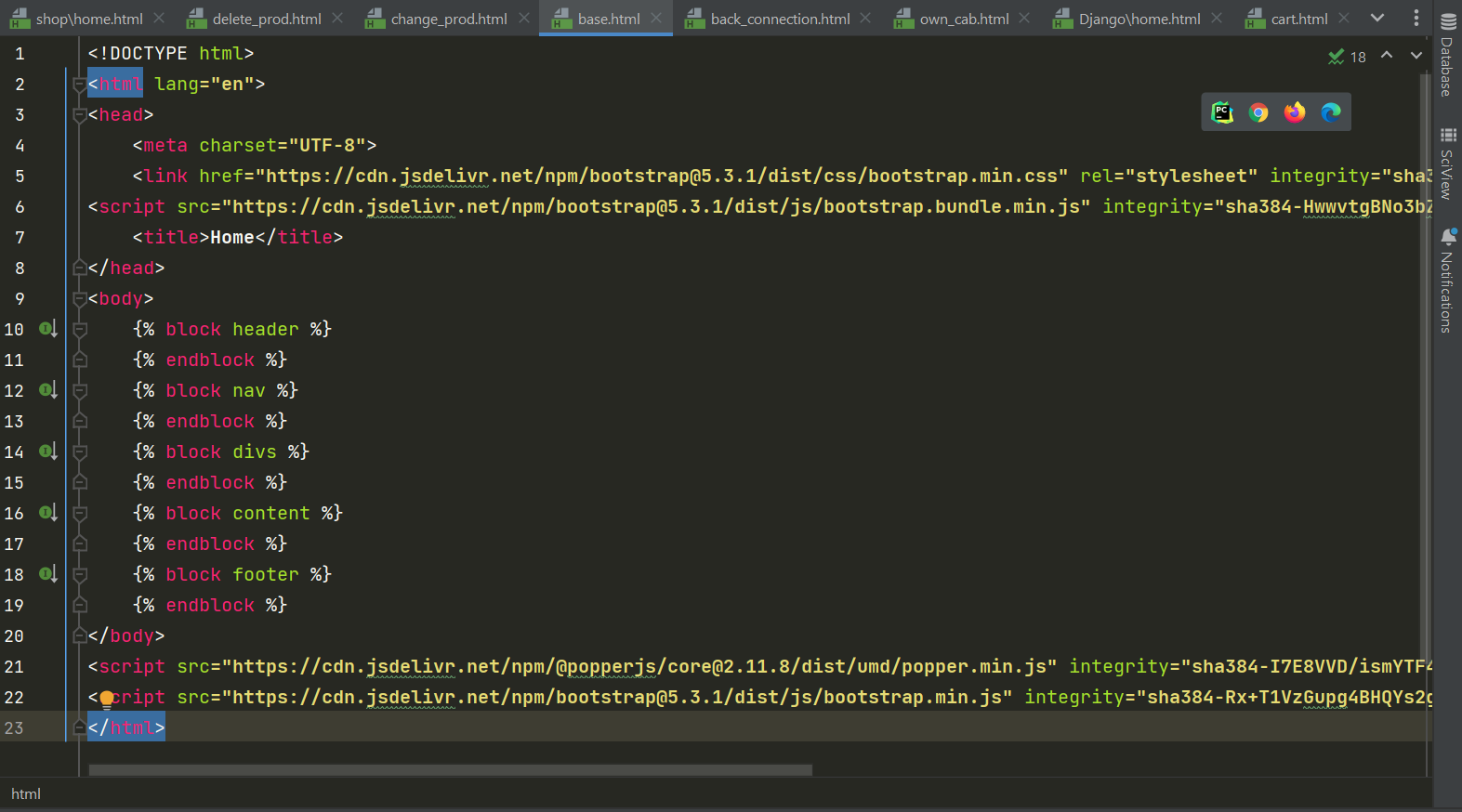


Рисунок 66 – Скелет блоков base.html

Также, для простоты использования и избегания самоповторения, можно создать файлы-шаблоны, которые будут использоваться в файлах страниц через include. В данном случае их будет два: header, «шапка» сайта, и footer, «подвал» сайта.

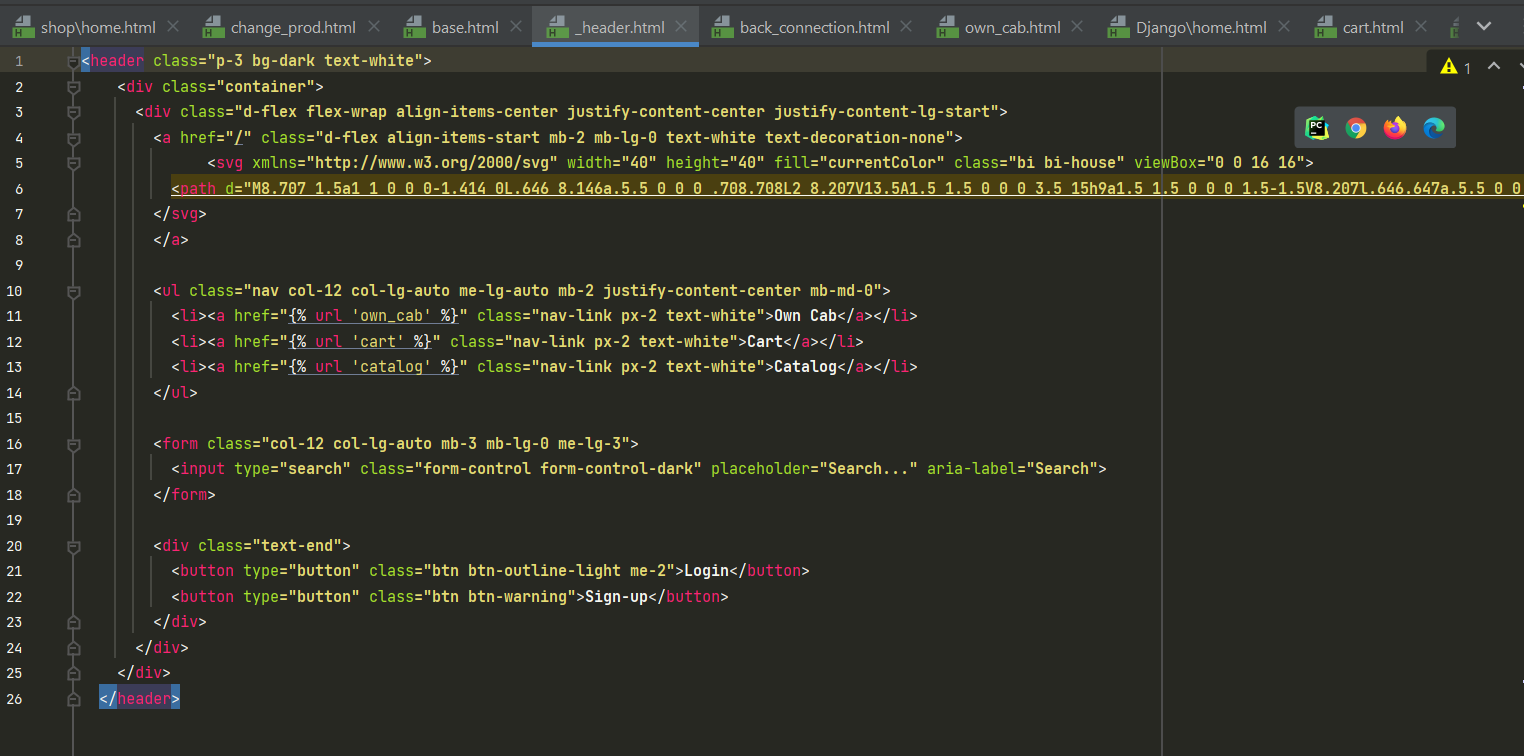


Рисунок 67 – Шаблон header.html

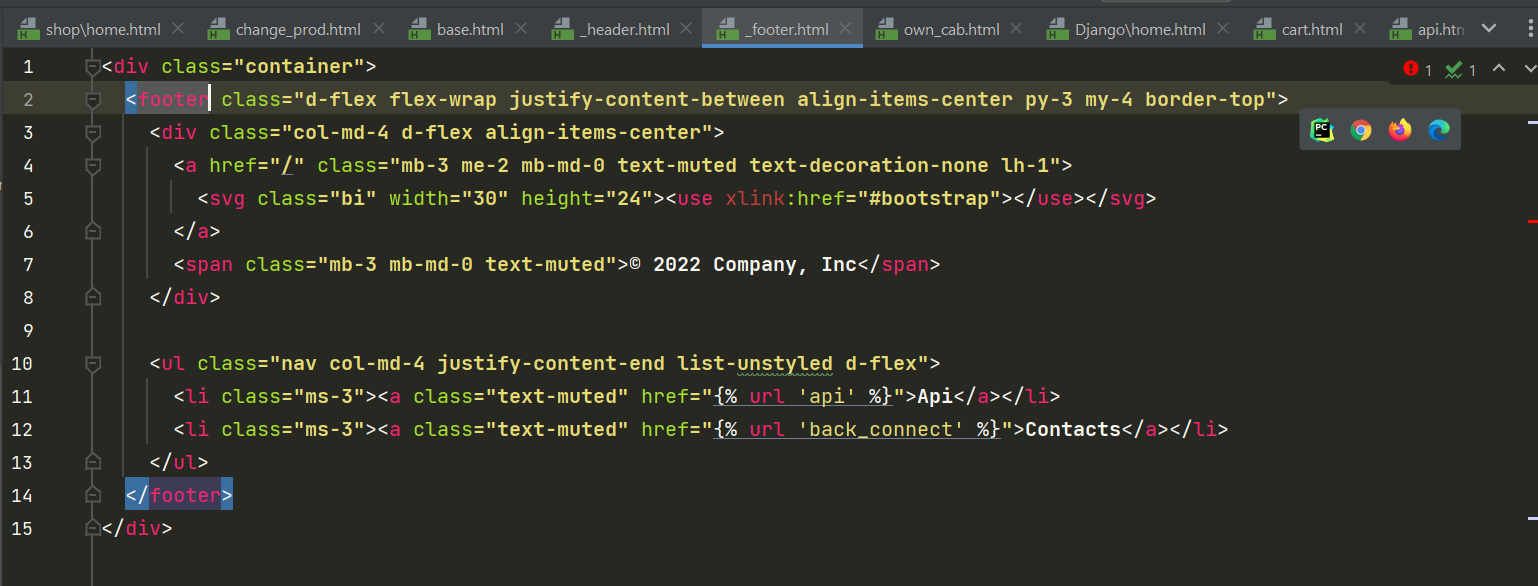


Рисунок 68 – Шаблон footer.html

Структура папки templates после добавления шаблонов и скелета выглядит следующим образом:

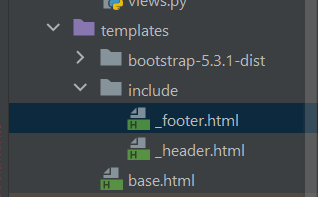


Рисунок 69 – Структура папки templates проекта

Для того, чтобы использовать созданный скелет, в первой строчке html файла необходимо прописать следующее:

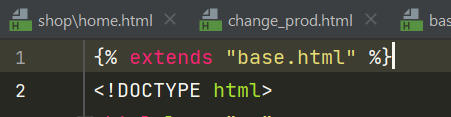


Рисунок 70 – Указание источника структуры

Примеры использования созданной структуры и её заполнения:

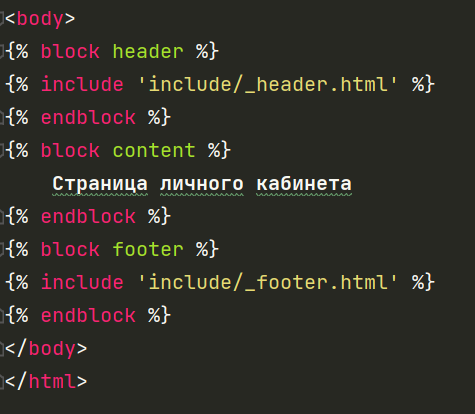


Рисунок 71 – Пример заполнения блоков на обычной странице

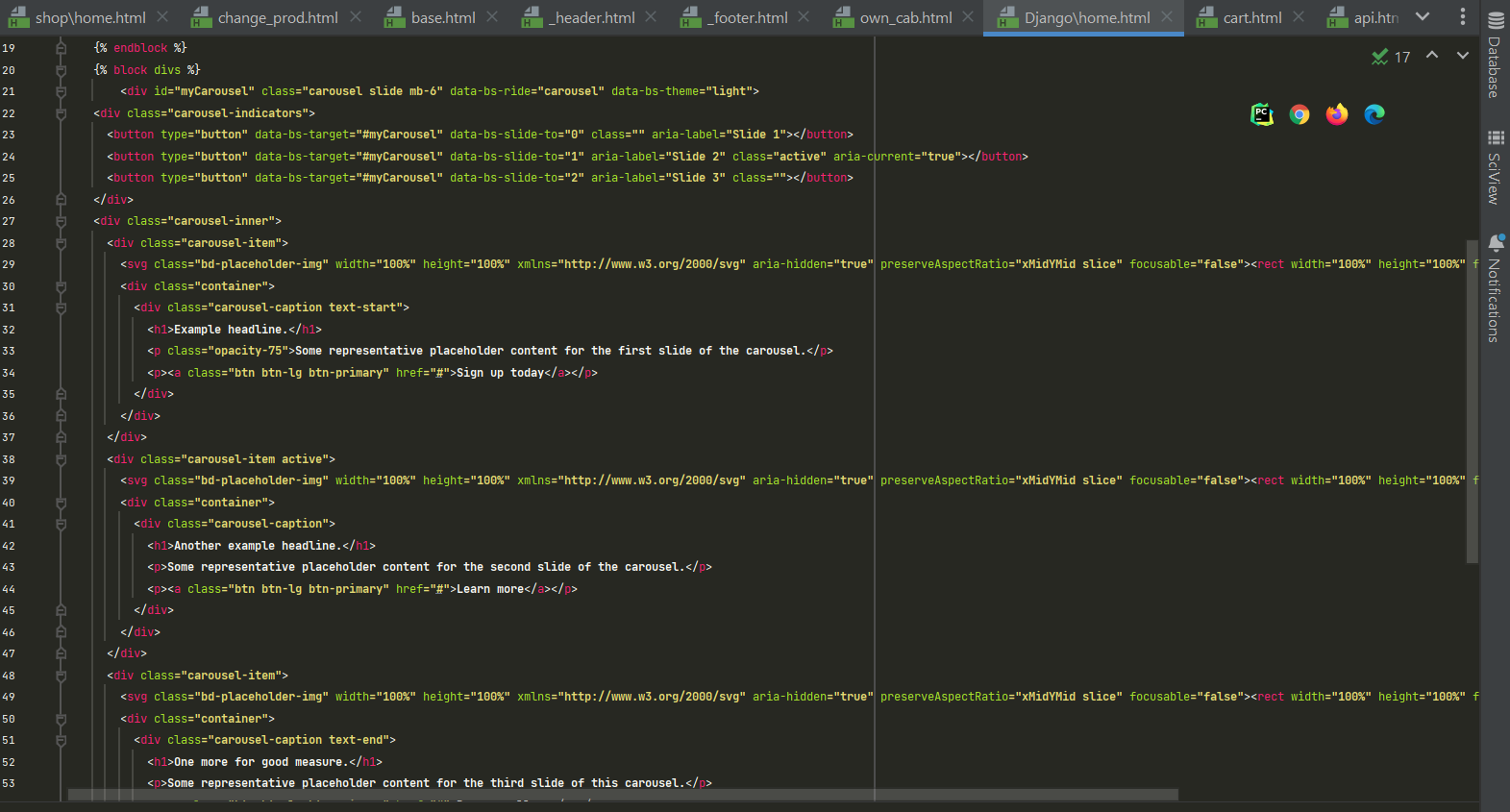


Рисунок 72 – Пример заполнения блока divs



Рисунок 73 – Пример заполнения блока nav

Результаты работы:

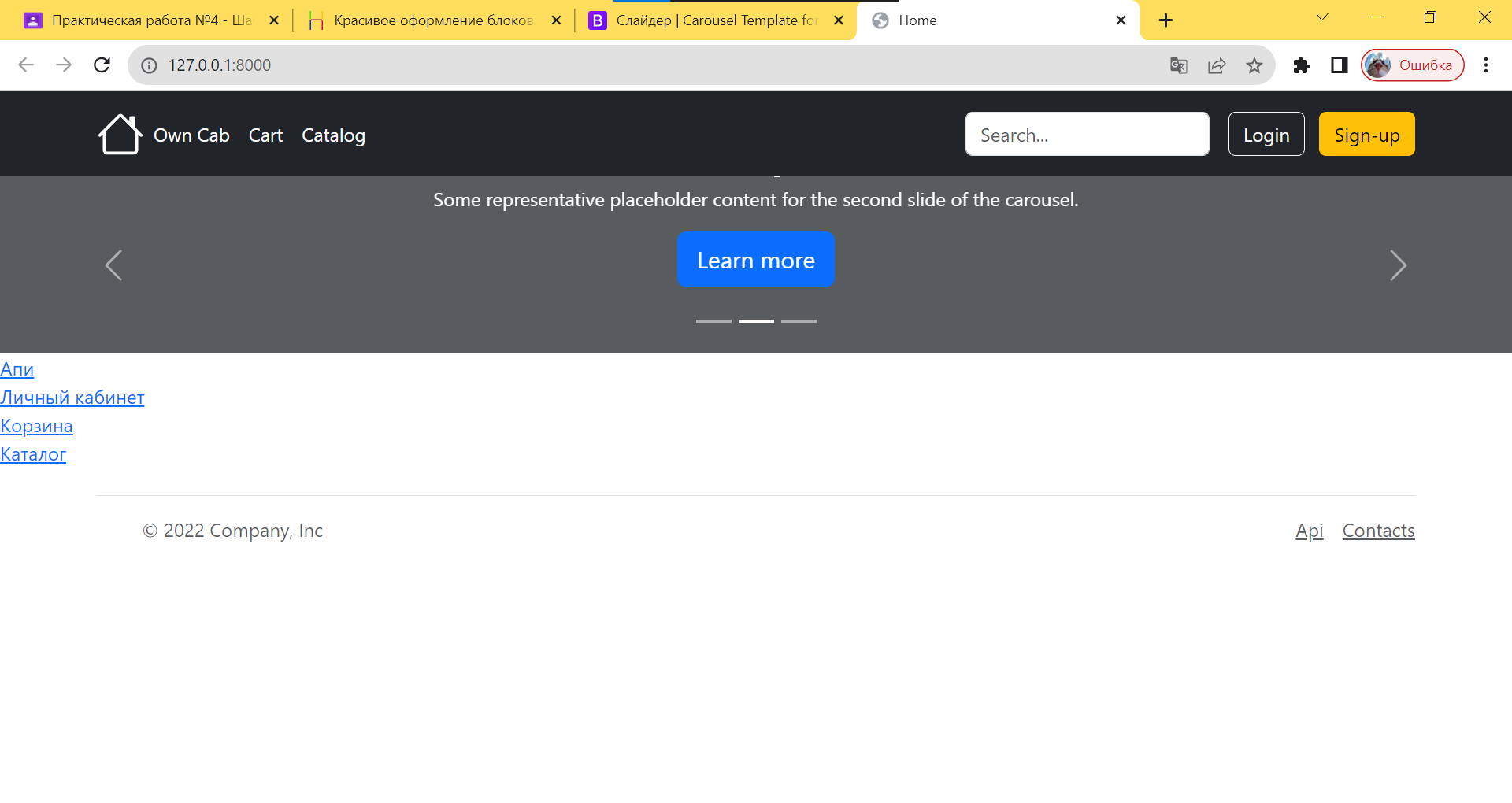


Рисунок 74 – Результат работы основной страницы с блоком divs

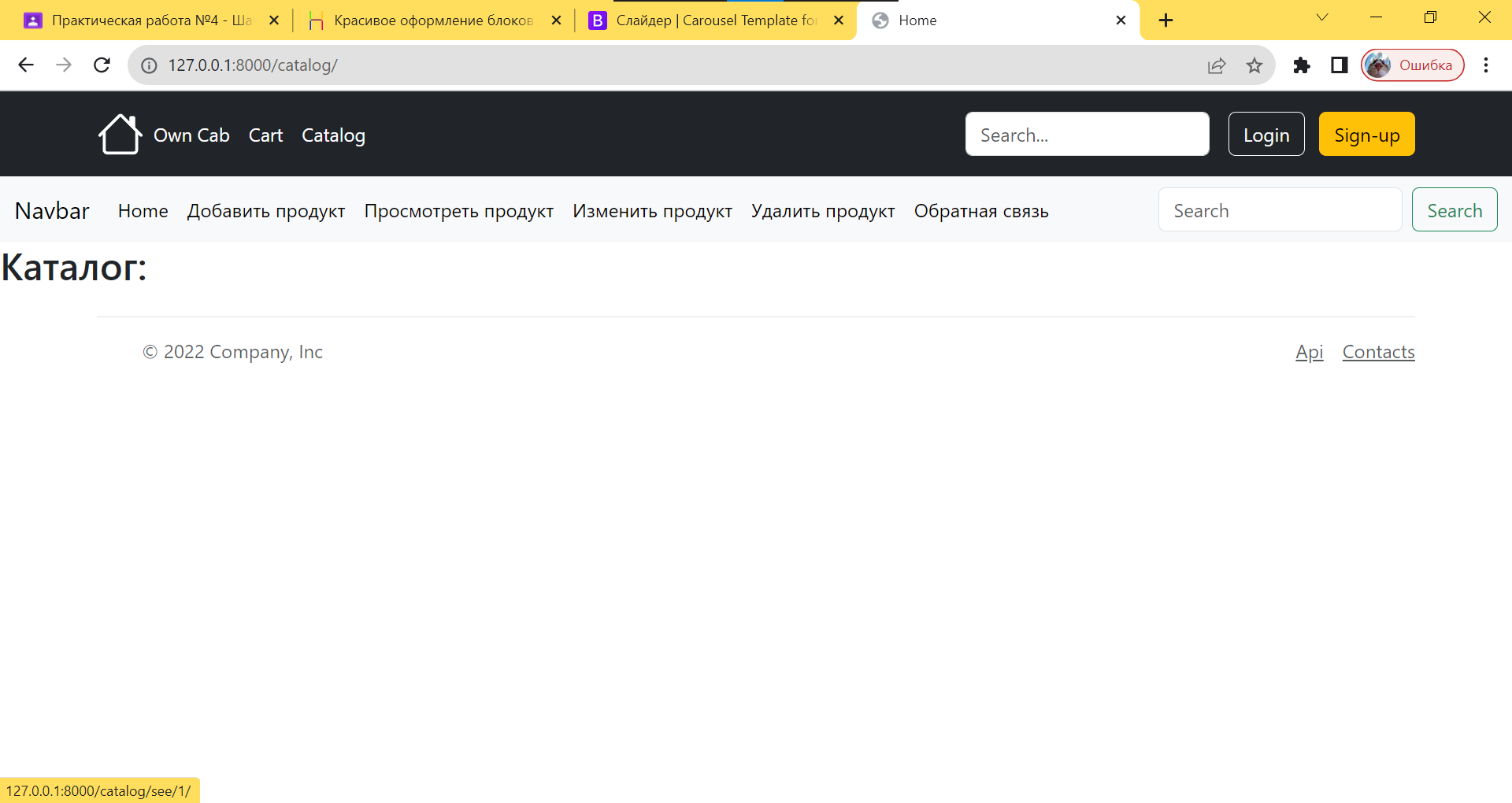


Рисунок 75 – Результат работы страницы каталога с блоком nav

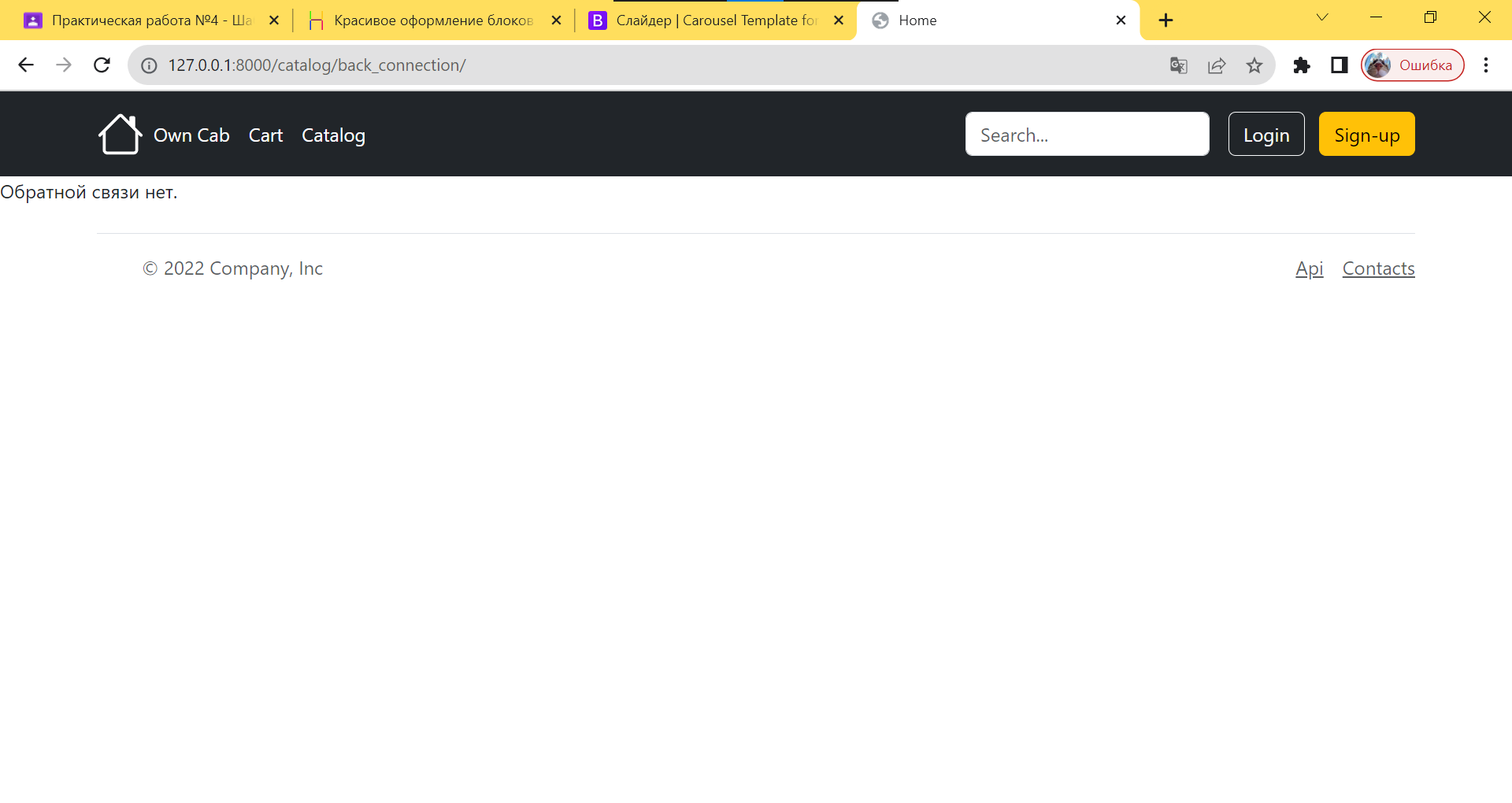


Рисунок 76 – Результат работы обычной страницы с header, content и footer

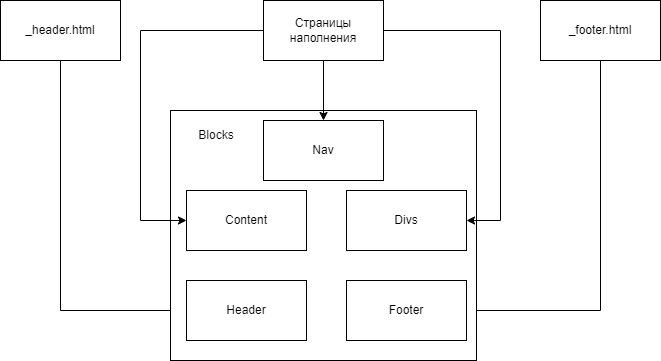


Рисунок 77 – Диаграмма структуры со связями

Вывод: научились работать с блоками и шаблонами в фреймворке Django, создали несколько блоков, два шаблона, заполнили блоки контентом и отразили их на странице.