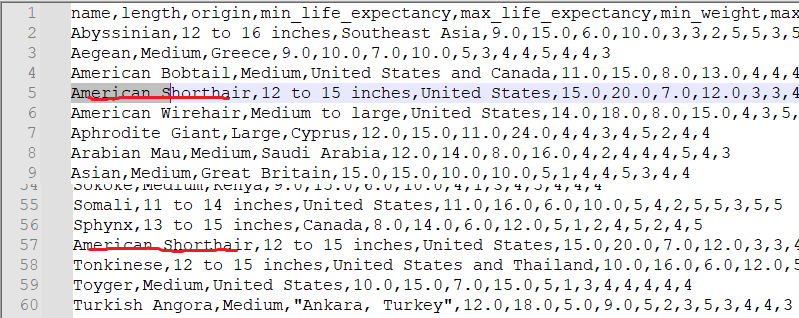
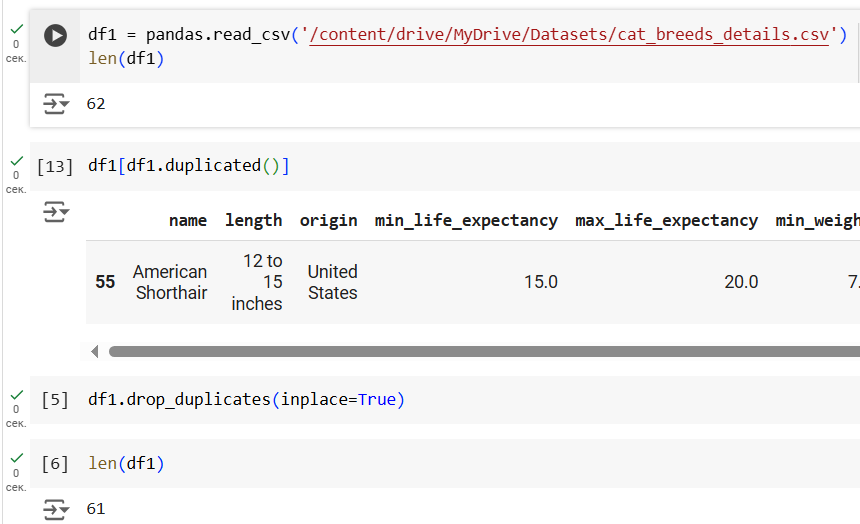
В отчете коды представлены в виде скриншотов (для подсветки синтаксиса), с запросами можно ознакомиться в файлах .sql папки задания.

Скриншоты «возможного» интерфейса сайта сделаны в figma и нужны лишь для наглядности изложения.

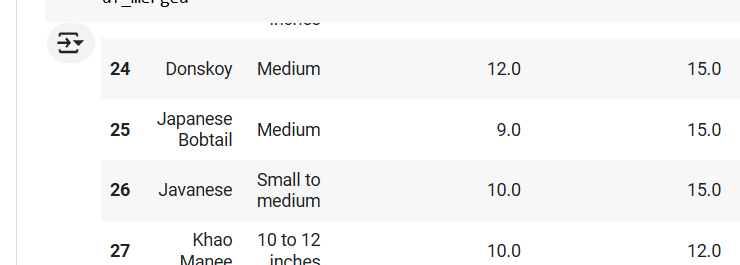
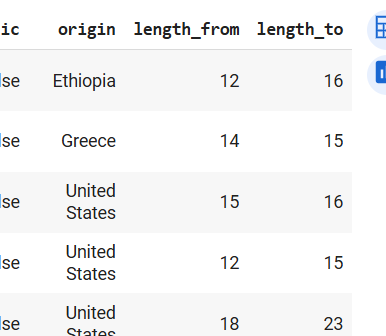
Для выполнения работы был выбран набор данных уникальных пород котов и их характеристик из сервиса Kaggle <https://www.kaggle.com/datasets/warcoder/cat-breeds-details>. Как будет описано ниже, набор будет дополнен несколькими свойствами из других данных.

Для удобства работы с данными их необходимо подготовить.

Имеются дубликаты, поэтому их нужно удалить

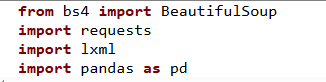


Также можно увидеть, что длины котов представлены в виде текста и в разных форматах. Была проведена «нормализация», с разбиением на два столбца length\_from и length\_to.

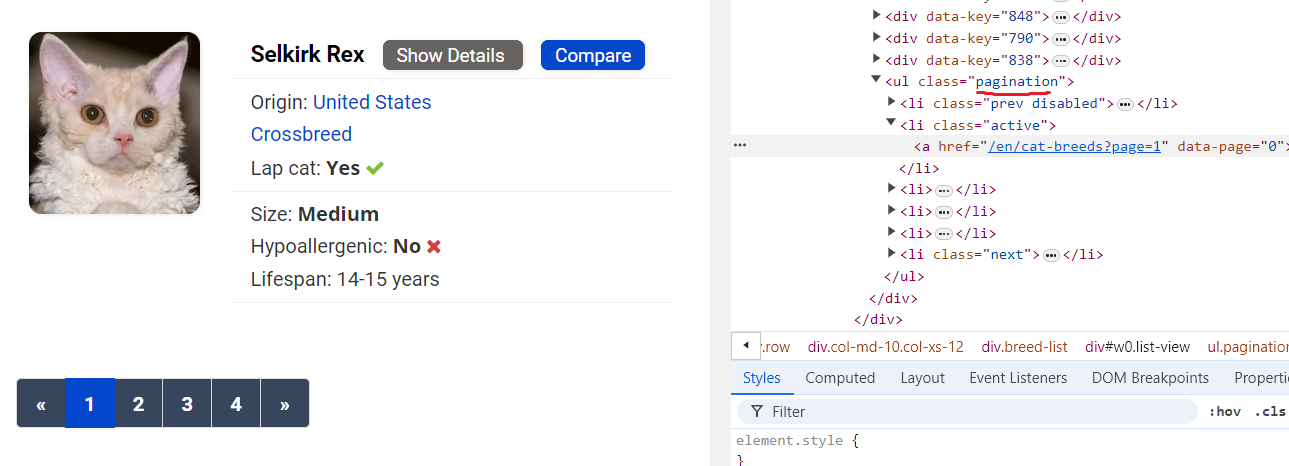
 

Имеющийся набор данных дополним с помощью анализа сайта пород кошек <https://dogell.com/en/cat-breeds>, внешний вид которого показан на рисунках ниже.

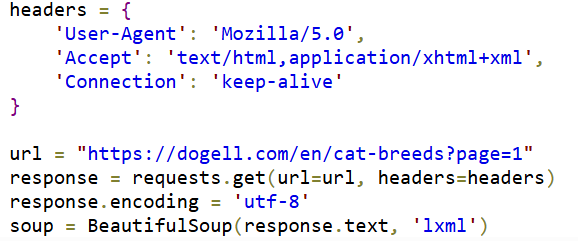
Для получения данных использовались библиотеки для веб-парсинга, такие как Beautifulsoup4 и Requests:



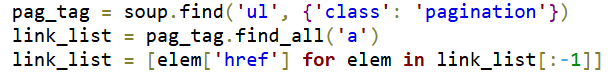
Страница имеет пагинацию, поэтому сначала необходимо определить все ссылки для анализа. Элемент пагинации на сайте имеет следующий вид:



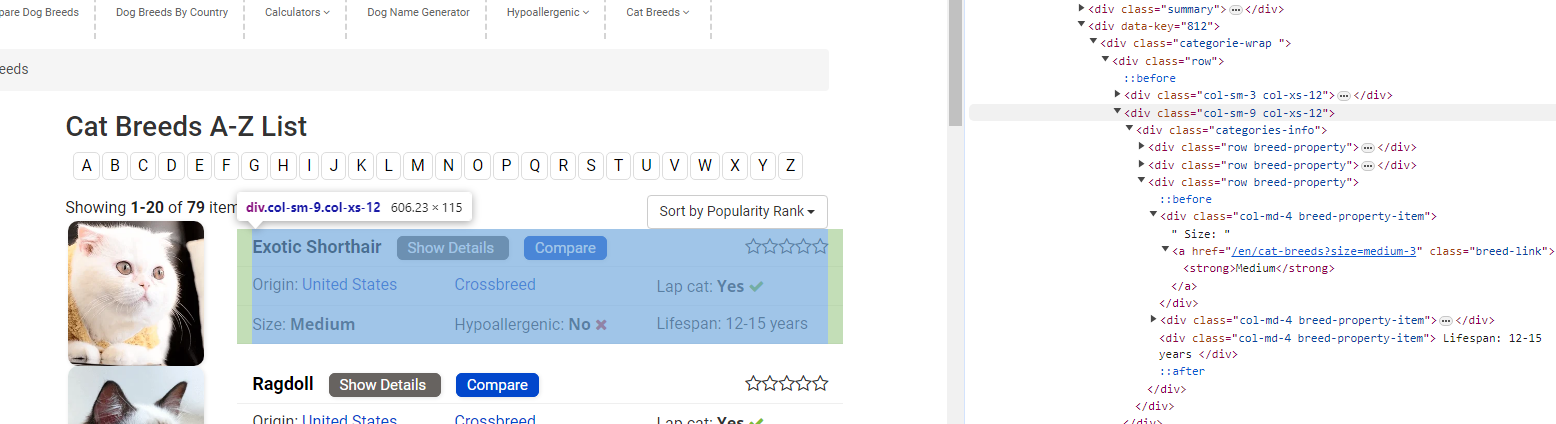
Определим страницы парсинга данных с помощью следующего отрывка кода. Сначала получим страницу, настроив заголовок запроса (чтобы сервер воспринимал как пользователя, а не автоматизированный скрипт) и преобразовав кодировку:

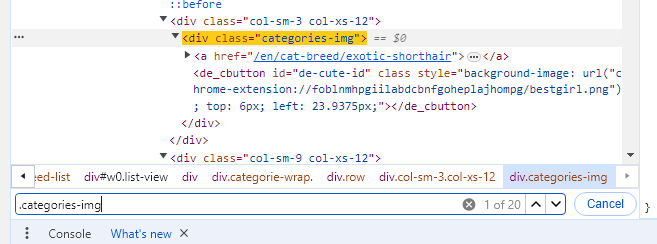


После чего можно найти набор страниц. Последний элемент исключен, т.к. относится к «стрелке» и указывает на следующую страницу:

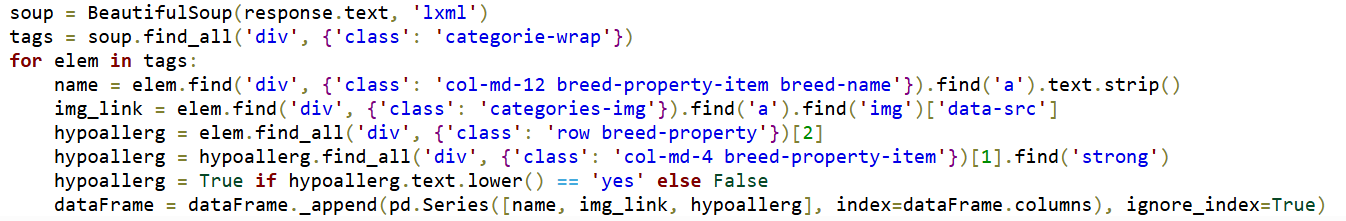


Получив все ссылки, можно проанализировать каждую страницу для определения дополнительных параметров – ссылки на изображение породы, регион и новое свойство «гипоаллергенности». Расположение ссылки и структура блоков с информацией о породе показаны на рисунке ниже:





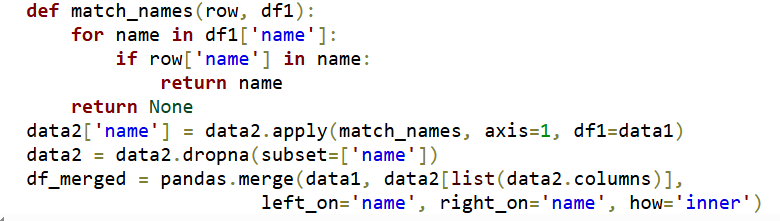
С помощью следующего скрипта можно получить необходимые свойства, которые будут сохранены в dataframe (для сохранения в csv).



В результате был получен следующий набор данных:



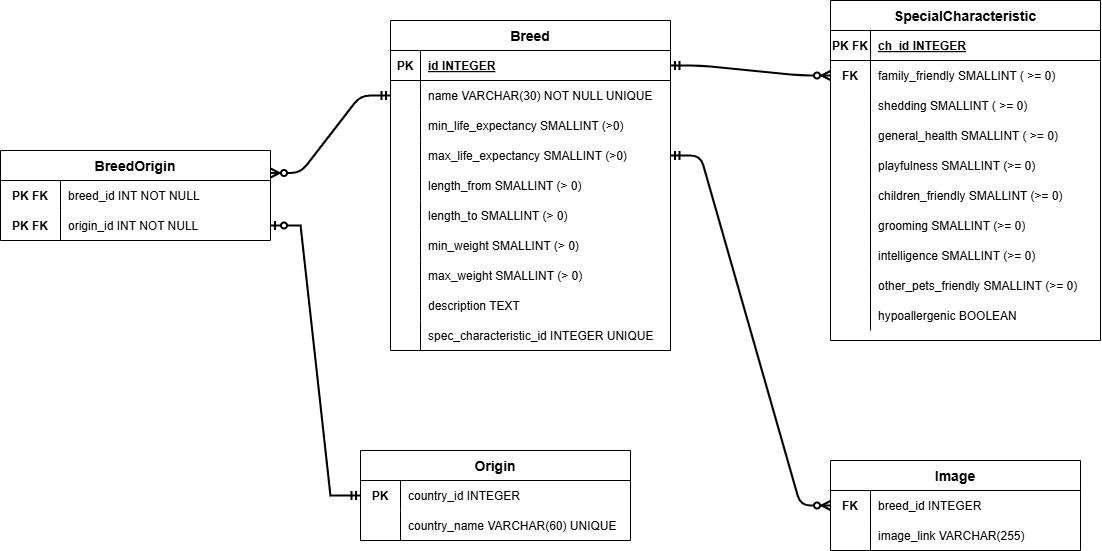
После этого было проведено «объединение» данных. Для этого названия порог приводится к одному виду и затем в результирующий dataframe сохраняется пересечение данных:



Результирующий набор данных представлен в файле mainData2DB.csv.

Далее, для более удобного импорта данных в таблицы базы данных сразу из csv-файлов, единый датафрейм pandas был разбит на несколько наборов, соответствующих таблицам БД.

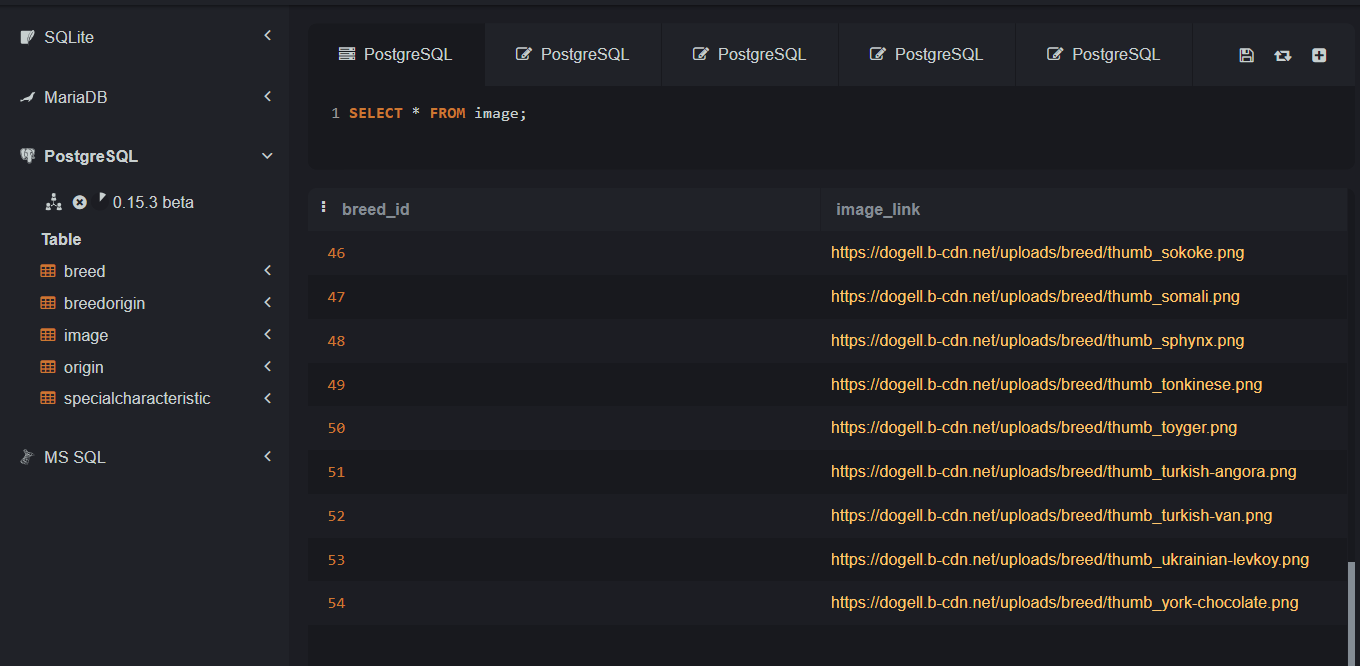




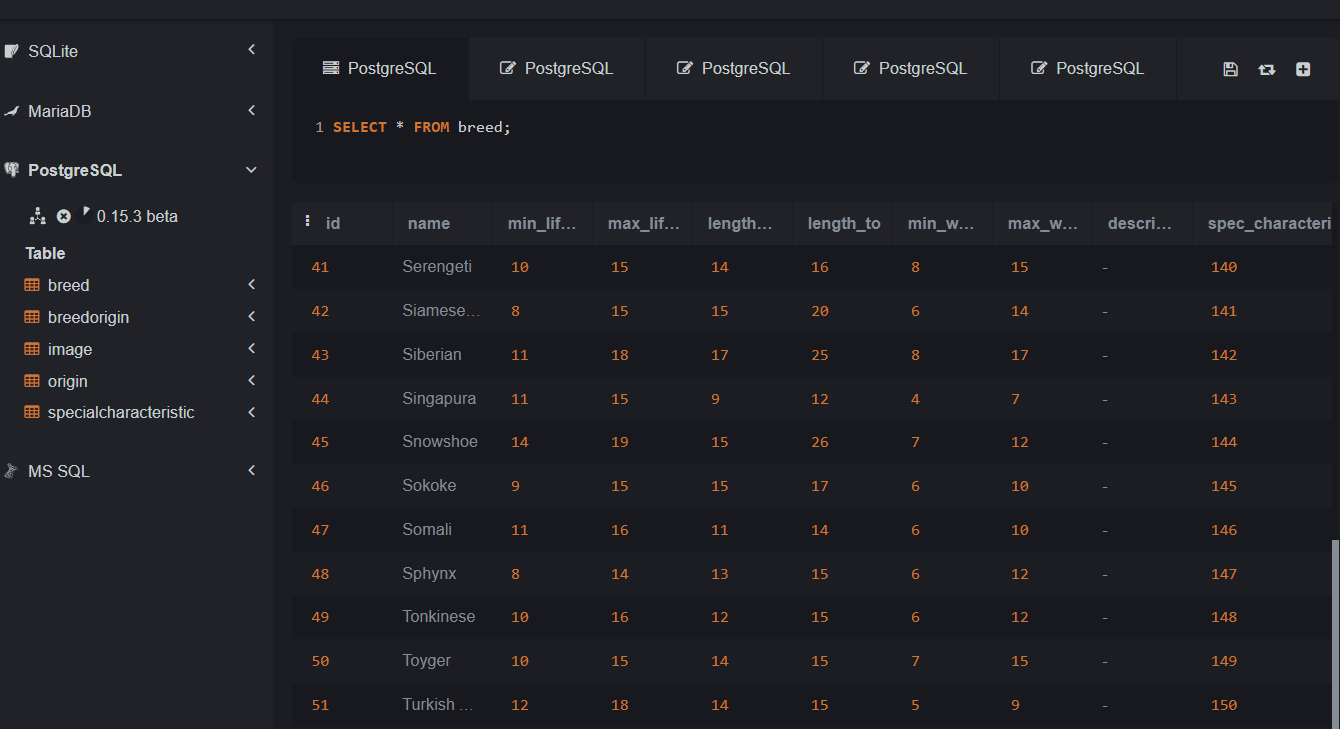
Запросы

**1. Простые запросы выборки данных таблиц.**

Самый простой запрос – выборка данных из таблицы с помощью select. Результат для таблицы Image может иметь следующий вид:

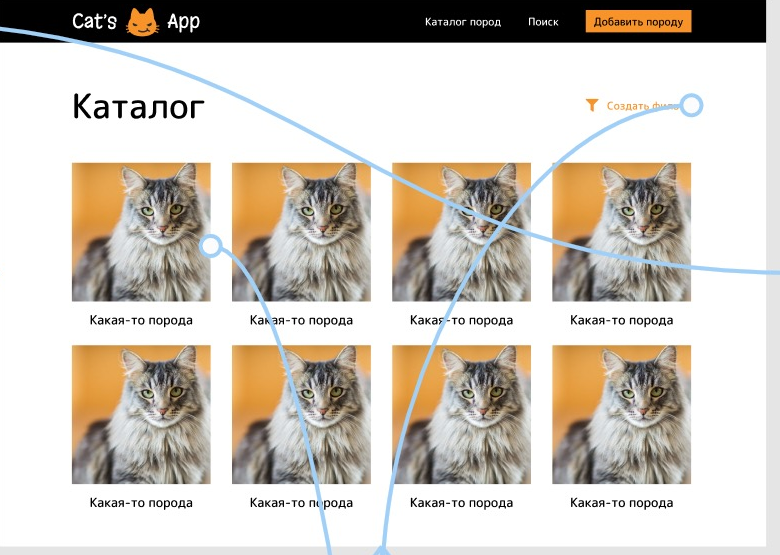


Для таблицы Breed результат будет иметь следующий вид:

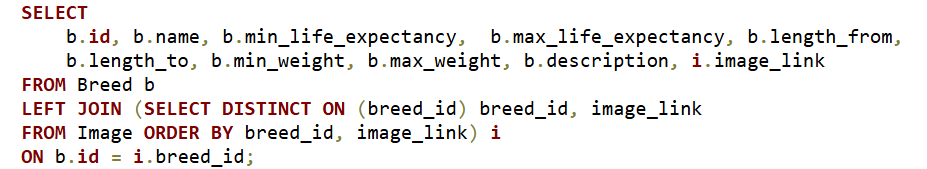


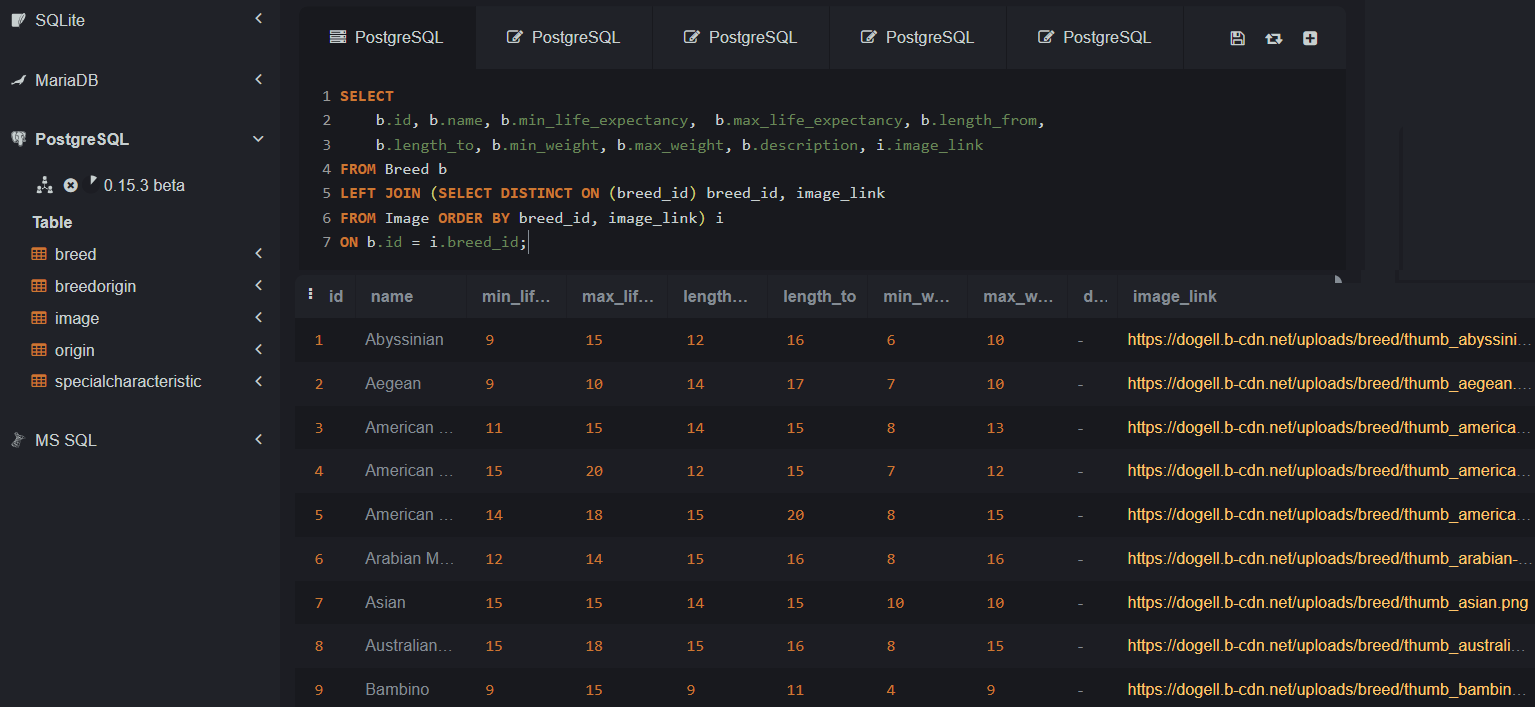
**2. Запросы-выборки и добавление данных**

Более реалистичным для использования являются следующие запросы. Пусть у нас имеется сервис, который предоставляет информацию о породах котов. Допустим в нем имеется веб-страница «каталог», которая представляет из себя список или «сетку» из пород. В такое сетке информация будет приводиться кратко, вероятно, с картинками:

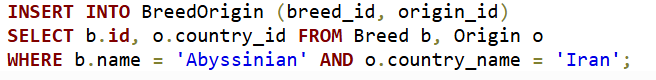


Тогда, запрос для такого сценария будет выглядеть как показано ниже. Т.к. возможно соответствие нескольких картинок для породы в таблице Image, то в запросе выбирается только одна ссылка на изображение.



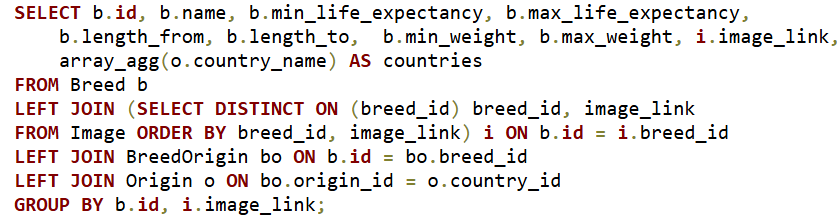


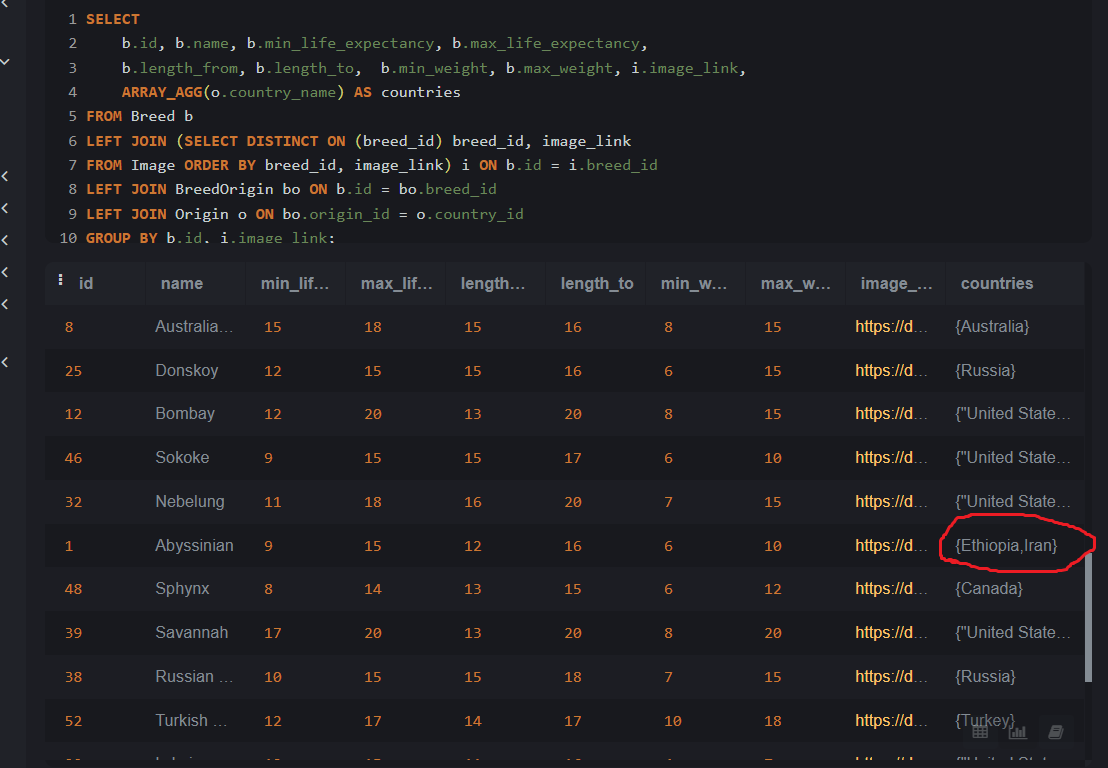
Т.к. в наборе каждой породе пока соответствует только 1 страна, однако может быть так, что необходимо внести изменения (структура БД позволяет соответствие одной породе несколько стран происхождения). Тогда необходимо выполнить запрос на добавление. Пусть известны название породы и название страны, тогда вставка будет иметь вид:



Запросы на добавление данных в таблицы также представлены в приложении.

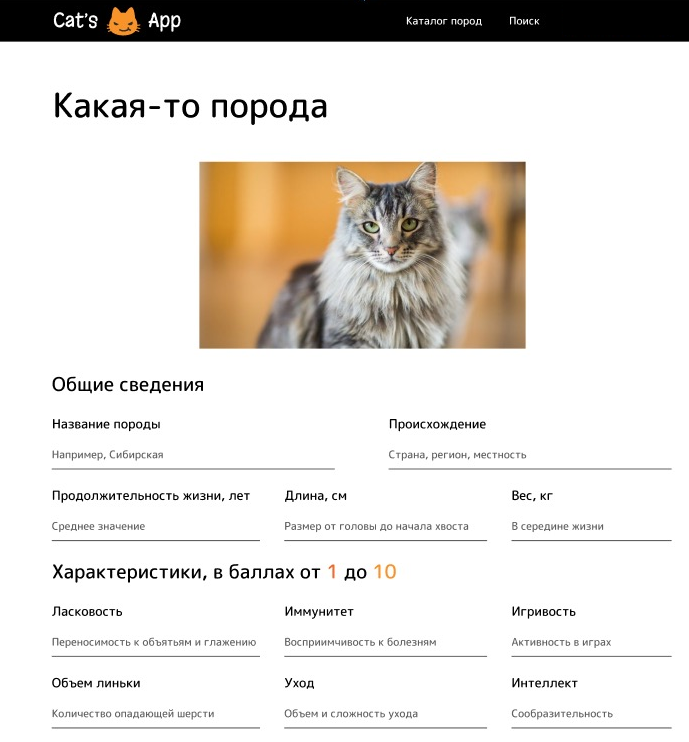
Предыдущий объединяющий запрос может быть расширен, если для «карточки» краткой информации также требуются страны происхождения/обитания. Тогда, если помимо базовой информации из Breed и ссылки на картинку требуются страны, то запрос примет вид, представленный ниже. Postgresql имеет тип данных «массив», поэтому список стран объединяется с помощью array\_agg.



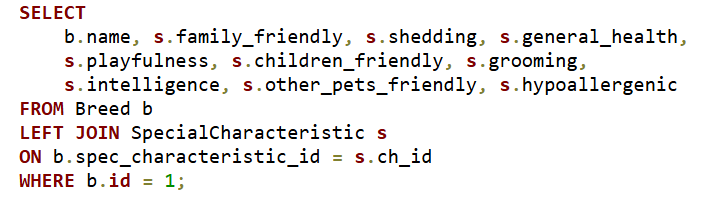


Дополнительно для корректности может использоваться функция COALESCE с некоторой заранее заданной строкой, например ‘No data’, если нет информации о странах для породы.

В данном случае мы рассматривали сетку пород с краткой информацией. Следующим сценарием может быть переход по ссылке на одну из пород (кнопка или нажатие на картинку). После перехода будет открыта страница конкретной породы уже с полной информацией.

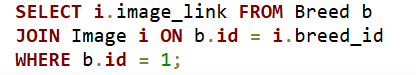


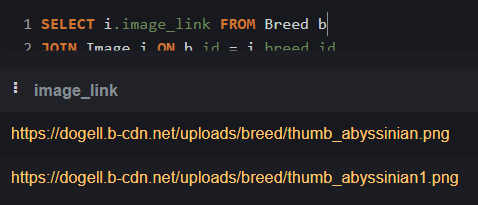
Часть данных у нас уже есть (базовая информация, одна картинка и список стран). Тогда, с учетом знания id породы для получения дополнительных данных должен быть выполнен следующий запрос:



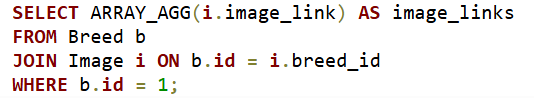


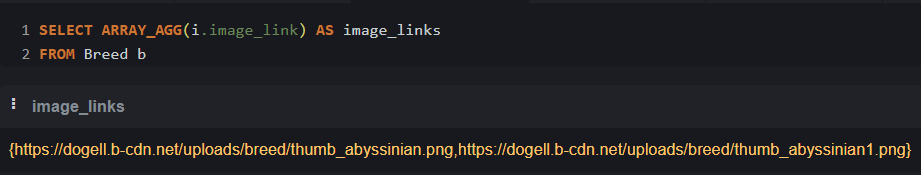
Оказавшись на странице определенной породы, мы имеем картинку породы и всю информацию. Возможен сценарий, который подразумевает наличие кнопки «галерея» породы, для перехода на страницу, со всеми фотографиями данной породы. Как уже упоминалось, БД допускает соответствие породе нескольких фотографий. Тогда запрос для этого может быть представлен в двух вариантах – получение столбца результата, или массива-результата.





Или

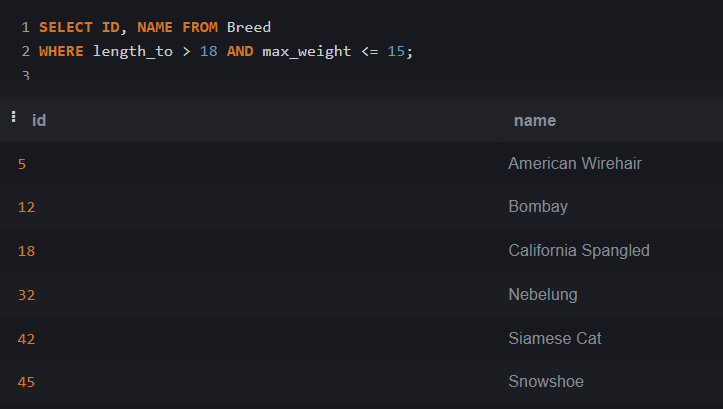




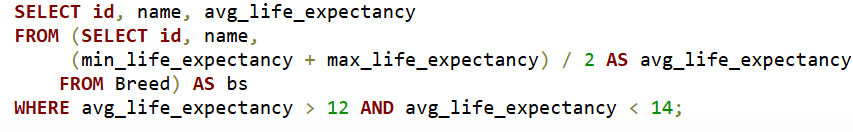
**3. Фильтрованный поиск**

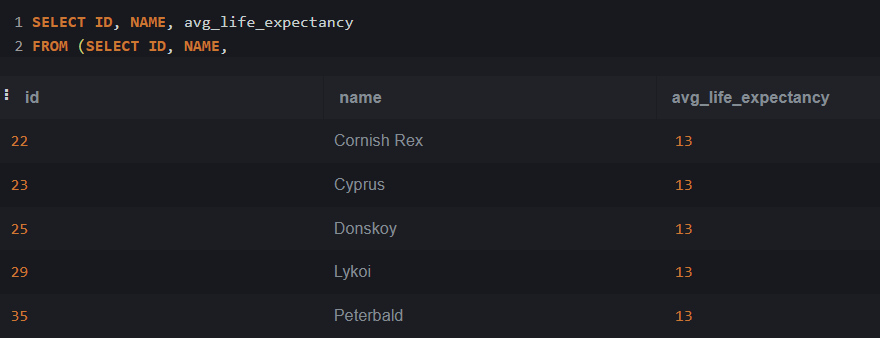
Допустим, мы хотим найти длинных котов, но не слишком толстых. Для этого нужен поиск с фильтрацией:



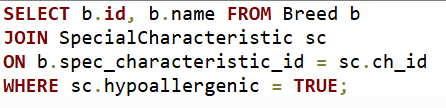


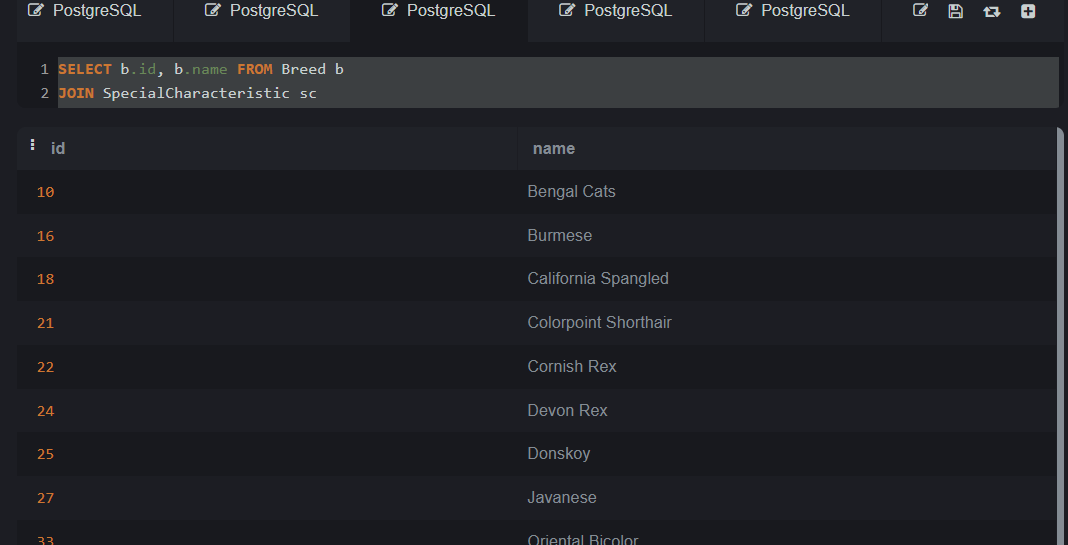
Допустим, что мы хотим найти породы, которые живут не слишком мало, но при этом мы не хотим «содержать» этих котиков на пенсии слишком долго. Тогда можно найти породы на основе средней продолжительности жизни (продолжительность в таблице задана от и до).



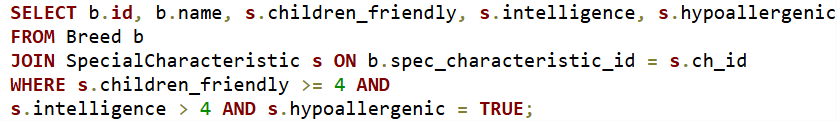


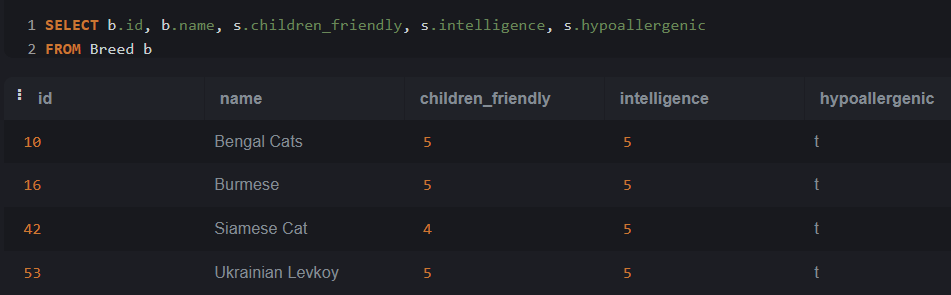
Как уже упоминалось, в сценарии использования возможно наличие страницы с «сеткой» пород. Допустим что необходимо найти все гипоаллегенные породы. Соответствующий запрос:





Более сложный фильтрационный запрос может иметь следующий вид:





Заключение

В результате выполнения работы были освоены и применены инструмента для обработки данных (pandas, numpy). Были использованы такие инструменты как BeautifulSoup4 и библиотека requests для парсинга информации, а также применены полученные знания по работе с базами данных и обработки с помощью языка SQL.

ПРИЛОЖЕНИЕ

Импорт сгенерированных csv файлов в базу данных.



