

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

```
df1 = pandas.read_csv('/content/drive/MyDrive/Datasets/cat_breeds_details.csv')
len(df1)

62

[13] df1[df1.duplicated()]

   name  length  origin  min_life_expectancy  max_life_expectancy  min_weight
55  American  12 to 15 inches  United States      15.0            20.0        7.0

[5] df1.drop_duplicates(inplace=True)

[6] len(df1)

61

1  name,length,origin,min_life_expectancy,max_life_expectancy,min_weight,max
2  Abyssinian,12 to 16 inches,Southeast Asia,9.0,15.0,6.0,10.0,3,3,2,5,5,3,5
3  Aegean,Medium,Greece,9.0,10.0,7.0,10.0,5,3,4,4,5,4,4,3
4  American Bobtail,Medium,United States and Canada,11.0,15.0,8.0,13.0,4,4,4
5  American Shorthair,12 to 15 inches,United States,15.0,20.0,7.0,12.0,3,3,4
6  American Wirehair,Medium to large,United States,14.0,18.0,8.0,15.0,4,3,5
7  Aphrodite Giant,Large,Cyprus,12.0,15.0,11.0,24.0,4,4,3,4,5,2,4,4
8  Arabian Mau,Medium,Saudi Arabia,12.0,14.0,8.0,16.0,4,2,4,4,4,5,4,3
9  Asian,Medium,Great Britain,15.0,15.0,10.0,10.0,5,1,4,4,5,3,4,4
10  Sokoke,Medium,Kenya,9.0,10.0,6.0,10.0,4,1,3,4,3,4,4,4
11  Somali,11 to 14 inches,United States,11.0,16.0,6.0,10.0,5,4,2,5,5,3,5,5
12  Sphynx,13 to 15 inches,Canada,8.0,14.0,6.0,12.0,5,1,2,4,5,2,4,5
13  American Shorthair,12 to 15 inches,United States,15.0,20.0,7.0,12.0,3,3,4
14  Tonkinese,12 to 15 inches,United States and Thailand,10.0,16.0,6.0,12.0,5
15  Toyger,Medium,United States,10.0,15.0,7.0,15.0,5,1,3,4,4,4,4,4
16  Turkish Angora,Medium,"Ankara, Turkey",12.0,18.0,5.0,9.0,5,2,3,5,3,4,4,3
```

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.

					ic	origin	length_from	length_to	
					se	Ethiopia	12	16	
24	Donskoy	Medium		12.0	15.0	se	Greece	14	15
25	Japanese Bobtail	Medium		9.0	15.0	se	United States	15	16
26	Javanese	Small to medium		10.0	15.0	se	United States	12	15
27	Khao Manee	10 to 12 inches		10.0	12.0	se	United States	18	23

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:

```
from bs4 import BeautifulSoup
import requests
import lxml
import pandas as pd
```

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



Selkirk Rex [Show Details](#) [Compare](#)

Origin: [United States](#)

Crossbreed

Lap cat: **Yes** ✓

Size: **Medium**

Hypoallergenic: **No** ✗

Lifespan: 14-15 years

[«](#)
[1](#)
[2](#)
[3](#)
[4](#)
[»](#)

```

▶ <div data-key="848">...</div>
▶ <div data-key="790">...</div>
▶ <div data-key="838">...</div>
▼ <ul class="pagination">
  ▶ <li class="prev disabled">...</li>
  ▼ <li class="active">
    ...
    <a href="/en/cat-breeds?page=1" data-page="0">
      ...
    </a>
  </li>
  ▶ <li>...</li>
  ▶ <li>...</li>
  ▶ <li>...</li>
  ▶ <li class="next">...</li>
</ul>
</div>
</div>

```

row

div.col-md-10.col-xs-12

div.breed-list

div#w0.list-view

ul.pagination

Styles

Computed

Layout

Event Listeners

DOM Breakpoints

Properties

Filter

hov .cls

element.style {

}

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:

```
headers = {
    'User-Agent': 'Mozilla/5.0',
    'Accept': 'text/html,application/xhtml+xml',
    'Connection': 'keep-alive'
}

url = "https://dogell.com/en/cat-breeds?page=1"
response = requests.get(url=url, headers=headers)
response.encoding = 'utf-8'
soup = BeautifulSoup(response.text, 'lxml')
```

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

```
pag_tag = soup.find('ul', {'class': 'pagination'})
link_list = pag_tag.find_all('a')
link_list = [elem['href'] for elem in link_list[:-1]]
```

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:

The screenshot displays the 'Cat Breeds A-Z List' on the Dogell website. It shows a list of breeds with filters for 'Showing 1-20 of 79 items' and 'Sort by Popularity Rank'. Two breeds are highlighted: 'Exotic Shorthair' and 'Ragdoll'. The 'Exotic Shorthair' entry shows details like 'Origin: United States', 'Crossbreed', 'Lap cat: Yes', 'Size: Medium', 'Hypoallergenic: No', and 'Lifespan: 12-15 years'. To the right, a BeautifulSoup HTML tree view shows the structure of the page, with the 'Exotic Shorthair' entry highlighted in yellow. The tree view shows the following structure:

```

<div class="summary">
  <div data-key="812">
    <div class="categorie-wrap">
      <div class="row">
        <div class="col-sm-3 col-xs-12">
          <div class="categories-info">
            <div class="row breed-property">
              <div class="row breed-property">
                <div class="col-md-4 breed-property-item">
                  <a href="/en/cat-breeds/size/medium-3" class="breed-link">
                    <strong>Medium</strong>
                  </a>
                </div>
                <div class="col-md-4 breed-property-item">
                  <div class="col-md-4 breed-property-item">
                    <strong>Lifespan: 12-15 years</strong>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

The bottom part of the screenshot shows a detailed view of the 'Exotic Shorthair' entry in the HTML tree, highlighting the 'categories-img' block and the 'breed-link' element.

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

```
soup = BeautifulSoup(response.text, 'lxml')
tags = soup.find_all('div', {'class': 'categorie-wrap'})
for elem in tags:
    name = elem.find('div', {'class': 'col-md-12 breed-property-item breed-name'}).find('a').text.strip()
    img_link = elem.find('div', {'class': 'categories-img'}).find('a').find('img')['data-src']
    hypoallerg = elem.find_all('div', {'class': 'row breed-property'})[2]
    hypoallerg = hypoallerg.find_all('div', {'class': 'col-md-4 breed-property-item'})[1].find('strong')
    hypoallerg = True if hypoallerg.text.lower() == 'yes' else False
    dataFrame = dataFrame._append(pd.Series([name, img_link, hypoallerg], index=dataFrame.columns), ignore_index=True)
```

В результате был получен следующий набор данных:






name	img_link	hypoallergenic	origin
Exotic Shorthair	https://dogell.b-cdn.net/uploads/breed/thumb_exotic-shorthair.png	False	United States
Ragdoll	https://dogell.b-cdn.net/uploads/breed/thumb_ragdoll.png	False	United States
British Shorthair	https://dogell.b-cdn.net/uploads/breed/thumb_british-shorthair.png	False	United Kingdom
Maine Coon	https://dogell.b-cdn.net/uploads/breed/thumb_maine-coon.png	False	United States
Persian	https://dogell.b-cdn.net/uploads/breed/thumb_persian.png	False	Iran
Sphynx	https://dogell.b-cdn.net/uploads/breed/thumb_sphynx.png	True	Canada
Devon Rex	https://dogell.b-cdn.net/uploads/breed/thumb_devon-rex.png	True	United Kingdom
Scottish Fold	https://dogell.b-cdn.net/uploads/breed/thumb_scottish-fold.png	False	United Kingdom
American Shorthair	https://dogell.b-cdn.net/uploads/breed/thumb_american-shorthair.png	False	United States
Abyssinian	https://dogell.b-cdn.net/uploads/breed/thumb_abyssinian.png	False	Ethiopia
Russian Blue	https://dogell.b-cdn.net/uploads/breed/thumb_russian-blue.png	True	Russia
Bengal	https://dogell.b-cdn.net/uploads/breed/thumb_bengal.png	True	United States
Cornish Rex	https://dogell.b-cdn.net/uploads/breed/thumb_cornish-rex.png	True	United Kingdom
Norwegian Forest	https://dogell.b-cdn.net/uploads/breed/thumb_norwegian-forest.png	False	Norway
Siamese	https://dogell.b-cdn.net/uploads/breed/thumb_siamese.png	True	Thailand
Oriental	https://dogell.b-cdn.net/uploads/breed/thumb_oriental.png	True	United States
Ocicat	https://dogell.b-cdn.net/uploads/breed/thumb_ocicat.png	True	United States
Tonkinese	https://dogell.b-cdn.net/uploads/breed/thumb_tonkinese.png	False	Canada
Birman	https://dogell.b-cdn.net/uploads/breed/thumb_birman.png	False	France
Selkirk Rex	https://dogell.b-cdn.net/uploads/breed/thumb_selkirk-rex.png	False	United States
Burmese	https://dogell.b-cdn.net/uploads/breed/thumb_burmese.png	True	Thailand
Siberian	https://dogell.b-cdn.net/uploads/breed/thumb_siberian.png	True	Russia
Donskoy	https://dogell.b-cdn.net/uploads/breed/thumb_donskoy.png	True	Russia
Thai	https://dogell.b-cdn.net/uploads/breed/thumb_thai.png	True	Thailand
Dwelf	https://dogell.b-cdn.net/uploads/breed/thumb_dwelf.png	False	United States
Napoleon	https://dogell.b-cdn.net/uploads/breed/thumb_napoleon.png	True	United States
Sokoke	https://dogell.b-cdn.net/uploads/breed/thumb_sokoke.png	False	United States
Serengeti	https://dogell.b-cdn.net/uploads/breed/thumb_serengeti.png	False	United States
Peterbald	https://dogell.b-cdn.net/uploads/breed/thumb_peterbald.png	False	Russia

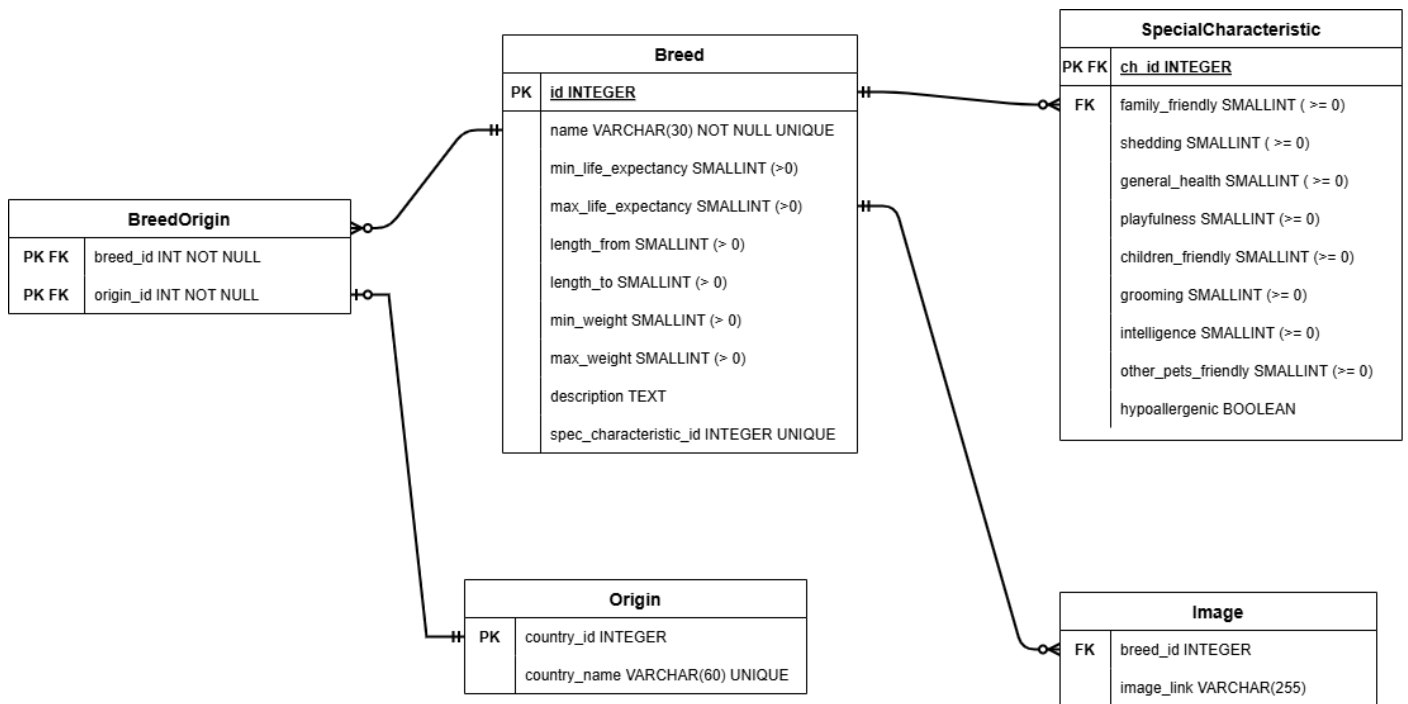
После этого было проведено «объединение» данных. Для этого названия порог приводится к одному виду и затем в результирующий dataframe сохраняется пересечение данных:

```
def match_names(row, df1):
    for name in df1['name']:
        if row['name'] in name:
            return name
    return None
data2['name'] = data2.apply(match_names, axis=1, df1=data1)
data2 = data2.dropna(subset=['name'])
df_merged = pandas.merge(data1, data2[list(data2.columns)],
                          left_on='name', right_on='name', how='inner')
```

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.

-  breedOriginTable.csv
-  breedTable.csv
-  imageTable.csv
-  originTable.csv
-  specialTable.csv



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:

SQLite	PostgreSQL	PostgreSQL	PostgreSQL	PostgreSQL	PostgreSQL
MariaDB					
PostgreSQL					
0.15.3 beta					
Table					
breed					
breedorigin					
image					
origin					
specialcharacteristic					
MS SQL					

1 SELECT * FROM image;	
breed_id	image_link
46	https://dogell.b-cdn.net/uploads/breed/thumb_sokoke.png
47	https://dogell.b-cdn.net/uploads/breed/thumb_somali.png
48	https://dogell.b-cdn.net/uploads/breed/thumb_sphynx.png
49	https://dogell.b-cdn.net/uploads/breed/thumb_tonkinese.png
50	https://dogell.b-cdn.net/uploads/breed/thumb_toyger.png
51	https://dogell.b-cdn.net/uploads/breed/thumb_turkish-angora.png
52	https://dogell.b-cdn.net/uploads/breed/thumb_turkish-van.png
53	https://dogell.b-cdn.net/uploads/breed/thumb_ukrainian-levkoy.png
54	https://dogell.b-cdn.net/uploads/breed/thumb_york-chocolate.png

Для таблицы `Breed` результат будет иметь следующий вид:

SQLite

MariaDB

PostgreSQL

0.15.3 beta

Table

breed

breedorigin

image

origin

specialcharacteristic

MS SQL

PostgreSQL

PostgreSQL

PostgreSQL

PostgreSQL

PostgreSQL

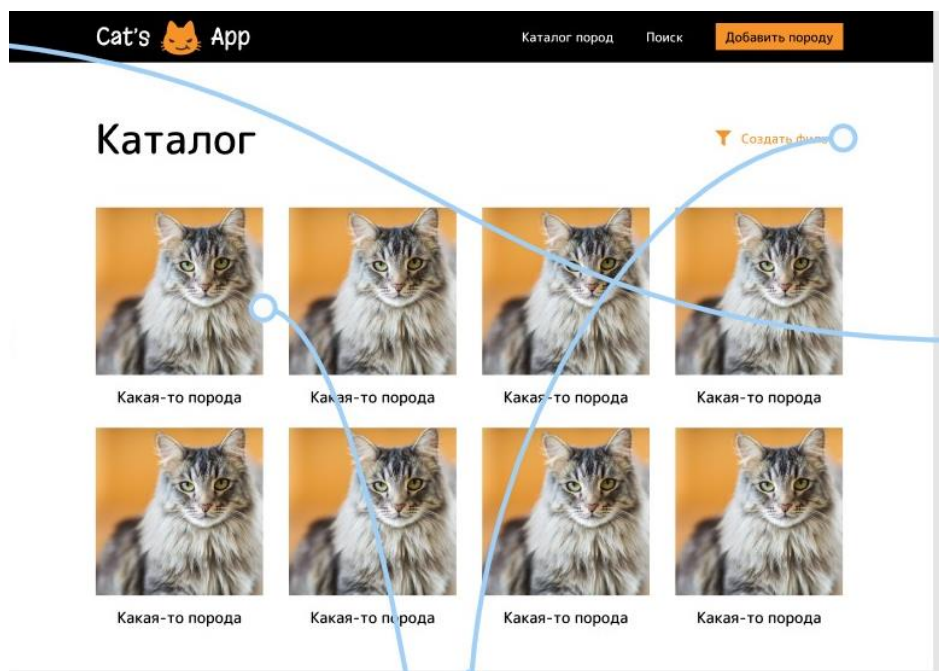
1

SELECT * FROM breed;

id	name	min_lif...	max_lif...	length...	length_to	min_w...	max_w...	descri...	spec_characteri
41	Serengeti	10	15	14	16	8	15	-	140
42	Siamese...	8	15	15	20	6	14	-	141
43	Siberian	11	18	17	25	8	17	-	142
44	Singapura	11	15	9	12	4	7	-	143
45	Snowshoe	14	19	15	26	7	12	-	144
46	Sokoke	9	15	15	17	6	10	-	145
47	Somali	11	16	11	14	6	10	-	146
48	Sphynx	8	14	13	15	6	12	-	147
49	Tonkinese	10	16	12	15	6	12	-	148
50	Toyger	10	15	14	15	7	15	-	149
51	Turkish ...	12	18	14	15	5	9	-	150

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 такой сетке информация будет приводиться кратко, вероятно, с картинками:

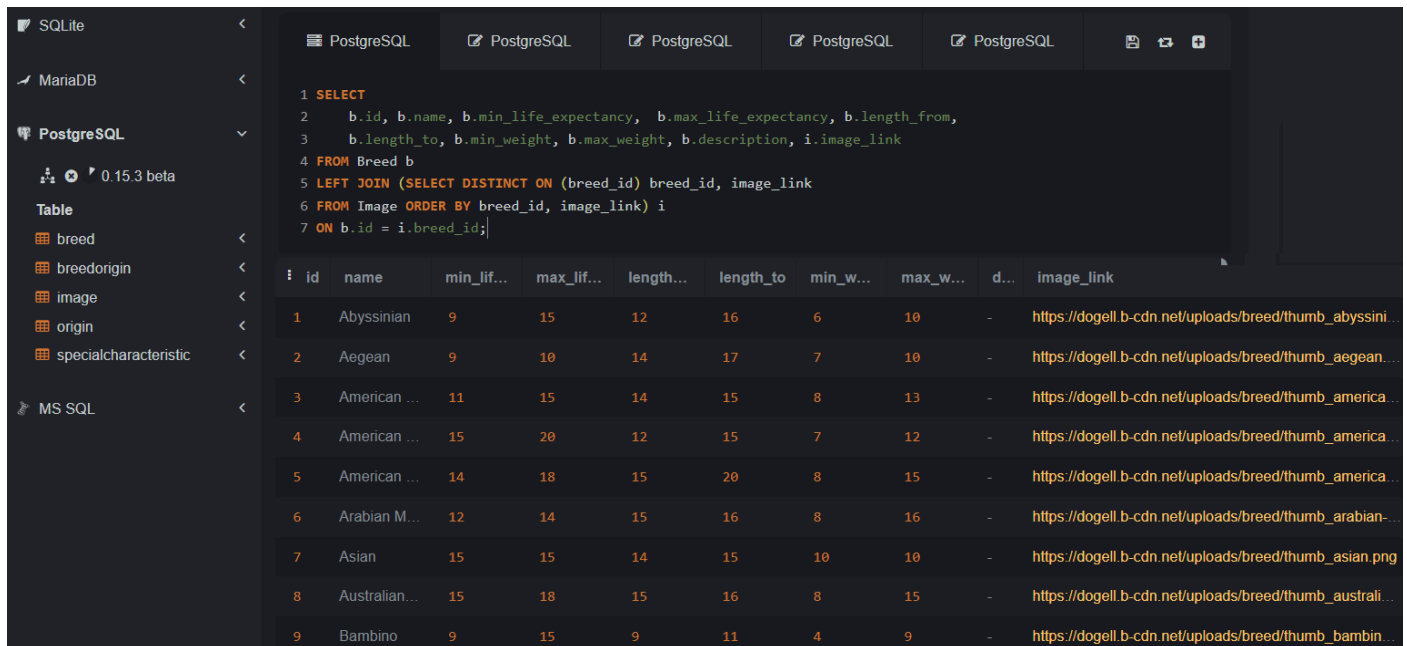


Тогда, запрос для такого сценария будет выглядеть как показано ниже. Т.к. возможно соответствие нескольких картинок для породы в таблице Image, то в запросе выбирается только одна ссылка на изображение.


```

SELECT
    b.id, b.name, b.min_life_expectancy, b.max_life_expectancy, b.length_from,
    b.length_to, b.min_weight, b.max_weight, b.description, i.image_link
FROM Breed b
LEFT JOIN (SELECT DISTINCT ON (breed_id) breed_id, image_link
FROM Image ORDER BY breed_id, image_link) i
ON b.id = i.breed_id;

```



The screenshot shows a database client interface with a sidebar on the left containing database connections: SQLite, MariaDB, PostgreSQL (selected), and MS SQL. The main area displays a PostgreSQL query and its results. The query is the same as the one in the previous block. The results are shown in a table with 10 columns: id, name, min_lif..., max_lif..., length..., length_to, min_w..., max_w..., d..., and image_link. The table contains 9 rows of data, each representing a dog breed and its associated image link.

id	name	min_lif...	max_lif...	length...	length_to	min_w...	max_w...	d...	image_link
1	Abyssinian	9	15	12	16	6	10	-	https://dogell.b-cdn.net/uploads/breed/thumb_abyssini...
2	Aegean	9	10	14	17	7	10	-	https://dogell.b-cdn.net/uploads/breed/thumb_aegean...
3	American ...	11	15	14	15	8	13	-	https://dogell.b-cdn.net/uploads/breed/thumb_america...
4	American ...	15	20	12	15	7	12	-	https://dogell.b-cdn.net/uploads/breed/thumb_america...
5	American ...	14	18	15	20	8	15	-	https://dogell.b-cdn.net/uploads/breed/thumb_america...
6	Arabian M...	12	14	15	16	8	16	-	https://dogell.b-cdn.net/uploads/breed/thumb_arabian...
7	Asian	15	15	14	15	10	10	-	https://dogell.b-cdn.net/uploads/breed/thumb_asian.png
8	Australian...	15	18	15	16	8	15	-	https://dogell.b-cdn.net/uploads/breed/thumb_australi...
9	Bambino	9	15	9	11	4	9	-	https://dogell.b-cdn.net/uploads/breed/thumb_bambin...

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:

```

INSERT INTO BreedOrigin (breed_id, origin_id)
SELECT b.id, o.country_id FROM Breed b, Origin o
WHERE b.name = 'Abyssinian' AND o.country_name = 'Iran';

```

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.

```

SELECT b.id, b.name, b.min_life_expectancy, b.max_life_expectancy,
    b.length_from, b.length_to, b.min_weight, b.max_weight, i.image_link,
    array_agg(o.country_name) AS countries
FROM Breed b
LEFT JOIN (SELECT DISTINCT ON (breed_id) breed_id, image_link
FROM Image ORDER BY breed_id, image_link) i ON b.id = i.breed_id
LEFT JOIN BreedOrigin bo ON b.id = bo.breed_id
LEFT JOIN Origin o ON bo.origin_id = o.country_id
GROUP BY b.id, i.image_link;

```

```


1 SELECT
2   b.id, b.name, b.min_life_expectancy, b.max_life_expectancy,
3   b.length_from, b.length_to, b.min_weight, b.max_weight, i.image_link,
4   ARRAY_AGG(o.country_name) AS countries
5 FROM Breed b
6 LEFT JOIN (SELECT DISTINCT ON (breed_id) breed_id, image_link
7 FROM Image ORDER BY breed_id, image_link) i ON b.id = i.breed_id
8 LEFT JOIN BreedOrigin bo ON b.id = bo.breed_id
9 LEFT JOIN Origin o ON bo.origin_id = o.country_id
10 GROUP BY b.id, i.image link:

```


id	name	min_lif...	max_lif...	length...	length_to	min_w...	max_w...	image_...	countries
8	Australia...	15	18	15	16	8	15	https://d...	{Australia}
25	Donskoy	12	15	15	16	6	15	https://d...	{Russia}
12	Bombay	12	20	13	20	8	15	https://d...	{"United State...
46	Sokoke	9	15	15	17	6	10	https://d...	{"United State...
32	Nebelung	11	18	16	20	7	15	https://d...	{"United State...
1	Abyssinian	9	15	12	16	6	10	https://d...	{Ethiopia,Iran}
48	Sphynx	8	14	13	15	6	12	https://d...	{Canada}
39	Savannah	17	20	13	20	8	20	https://d...	{"United State...
38	Russian ...	10	15	15	18	7	15	https://d...	{Russia}
52	Turkish ...	12	17	14	17	10	18	https://d...	{Turkey}

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.

Cat's  App Каталог пород Поиск

Какая-то порода



Общие сведения

Название породы

Например, Сибирская

Происхождение

Страна, регион, местность

Продолжительность жизни, лет

Длина, см

Вес, кг

Среднее значение

Размер от головы до начала хвоста

В середине жизни

Характеристики, в баллах от 1 до 10

Ласковость

Иммунитет

Игривость

Переносимость к объятиям и глажению

Восприимчивость к болезням

Активность в играх

Объем линьки

Уход

Интеллект

Количество опадающей шерсти

Объем и сложность ухода

Сообразительность

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:

```
SELECT
    b.name, s.family_friendly, s.shedding, s.general_health,
    s.playfulness, s.children_friendly, s.grooming,
    s.intelligence, s.other_pets_friendly, s.hypoallergenic
FROM Breed b
LEFT JOIN SpecialCharacteristic s
ON b.spec_characteristic_id = s.ch_id
WHERE b.id = 1;
```

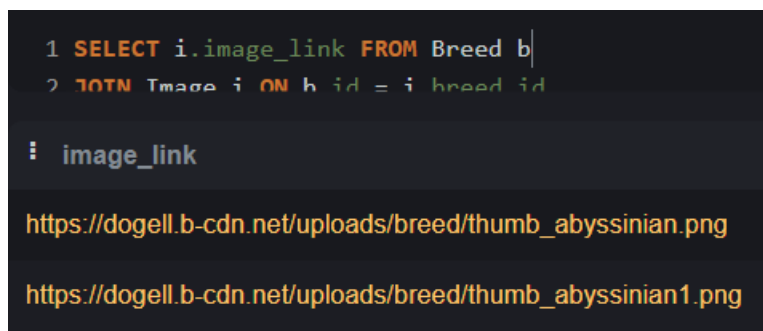


The screenshot shows a PostgreSQL query editor with a dark theme. The query is the same as the one above. Below the query, the results are displayed in a table with 10 columns: name, family_..., shedding, genera..., playfulness, childre..., groom..., intellig..., other_..., and hypoallerge... The first row of data shows 'Abyssinian' with values 3, 3, 2, 5, 5, 3, 5, 5, and f.

name	family_...	shedding	genera...	playful...	childre...	groomi...	intellig...	other_...	hypoallerge...
Abyssinian	3	3	2	5	5	3	5	5	f

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.

```
SELECT i.image_link FROM Breed b
JOIN Image i ON b.id = i.breed_id
WHERE b.id = 1;
```



The screenshot shows a PostgreSQL query editor with a dark theme. The query is the same as the one above. Below the query, the results are displayed in a table with 1 column: image_link. The first row of data shows an array of two URLs: https://dogell.b-cdn.net/uploads/breed/thumb_abyssinian.png and https://dogell.b-cdn.net/uploads/breed/thumb_abyssinian1.png.

image_link
https://dogell.b-cdn.net/uploads/breed/thumb_abyssinian.png https://dogell.b-cdn.net/uploads/breed/thumb_abyssinian1.png

Или

```
SELECT ARRAY_AGG(i.image_link) AS image_links
FROM Breed b
JOIN Image i ON b.id = i.breed_id
WHERE b.id = 1;
```

```
1 SELECT ARRAY_AGG(i.image_link) AS image_links
2 FROM Breed b
```

```
image_links
```

```
{https://dogell.b-cdn.net/uploads/breed/thumb_abyssinian.png,https://dogell.b-cdn.net/uploads/breed/thumb_abyssinian1.png}
```

3. Фильтрованный поиск

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

```
SELECT id, name FROM Breed
WHERE length_to > 18 AND max_weight <= 15;
```

```
1 SELECT ID, NAME FROM Breed
2 WHERE length_to > 18 AND max_weight <= 15;
3
```

id	name
5	American Wirehair
12	Bombay
18	California Spangled
32	Nebelung
42	Siamese Cat
45	Snowshoe

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).

```
SELECT id, name, avg_life_expectancy
FROM (SELECT id, name,
      (min_life_expectancy + max_life_expectancy) / 2 AS avg_life_expectancy
      FROM Breed) AS bs
WHERE avg_life_expectancy > 12 AND avg_life_expectancy < 14;
```

```
1 SELECT ID, NAME, avg_life_expectancy
2 FROM (SELECT ID, NAME,
```

id	name	avg_life_expectancy
22	Cornish Rex	13
23	Cyprus	13
25	Donskoy	13
29	Lykoi	13
35	Peterbald	13

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

```
SELECT b.id, b.name FROM Breed b
JOIN SpecialCharacteristic sc
ON b.spec_characteristic_id = sc.ch_id
WHERE sc.hypoallergenic = TRUE;
```

PostgreSQL	PostgreSQL	PostgreSQL	PostgreSQL	PostgreSQL	PostgreSQL	PostgreSQL
1	2	3	4	5	6	7
SELECT b.id, b.name FROM Breed b	JOIN SpecialCharacteristic sc					
id	name					
10	Bengal Cats					
16	Burmese					
18	California Spangled					
21	Colorpoint Shorthair					
22	Cornish Rex					
24	Devon Rex					
25	Donskoy					
27	Javanese					
33	Oriental Bicolor					

Более сложный фильтрационный запрос может иметь следующий вид:

```
SELECT b.id, b.name, s.children_friendly, s.intelligence, s.hypoallergenic
FROM Breed b
JOIN SpecialCharacteristic s ON b.spec_characteristic_id = s.ch_id
WHERE s.children_friendly >= 4 AND
s.intelligence > 4 AND s.hypoallergenic = TRUE;
```

```
1 SELECT b.id, b.name, s.children_friendly, s.intelligence, s.hypoallergenic
2 FROM Breed b
```

id	name	children_friendly	intelligence	hypoallergenic
10	Bengal Cats	5	5	t
16	Burmese	5	5	t
42	Siamese Cat	4	5	t
53	Ukrainian Levkoy	5	5	t

Заключение

В результате выполнения работы были освоены и применены инструменты 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.

```
INSERT INTO Origin ("country_name", "country_id") VALUES
('Ethiopia', '200'),
('Greece', '201'),
('United States', '202'),
('Iran', '203'),
('United Kingdom', '204'),
('Australia', '205'),
('France', '206'),
('Brazil', '207'),
('Thailand', '208'),
('Egypt', '209'),
('Cyprus', '210'),
('Russia', '211'),
('Japan', '212'),
('Singapore', '213'),
('Somalia', '214'),
('Canada', '215'),
('Turkey', '216'),
('Ukraine', '217');
```

```
INSERT INTO specialcharacteristic ("family_friendly", "shedding", "general_health", "playfulness", "children_friendly", "grooming", "intelligence", "ot
('3', '3', '2', '5', '5', '3', '5', '5', '0', '100'),
('5', '3', '4', '4', '5', '4', '4', '3', '0', '101'),
('4', '4', '4', '4', '4', '3', '4', '4', '0', '102'),
('3', '3', '4', '2', '4', '4', '4', '3', '0', '103'),
('4', '3', '5', '3', '4', '5', '4', '3', '0', '104'),
('4', '2', '4', '4', '4', '5', '4', '3', '0', '105'),
...
('4', '2', '4', '4', '4', '5', '4', '3', '1', '117'),
('4', '3', '4', '4', '4', '2', '4', '3', '0', '118'),
('4', '1', '4', '4', '1', '5', '5', '4', '0', '119'),
('5', '2', '2', '5', '4', '4', '4', '4', '1', '120'),
('5', '2', '3', '5', '5', '5', '4', '5', '1', '121');
```

```
INSERT INTO breed ("name","min_life_expectancy","max_life_expectancy","length_from","length_to"
('Abyssinian','9','15','12','16','6','10','1','100','-'),
('Aegean','9','10','14','17','7','10','2','101','-'),
('American Bobtail','11','15','14','15','8','13','3','102','-'),
('American Shorthair','15','20','12','15','7','12','4','103','-'),
('American Wirehair','14','18','15','20','8','15','5','104','-'),
('Arabian Mau','12','14','15','16','8','16','6','105','-'),
('Asian','15','15','14','15','10','10','7','106','-'),
('Australian Mist','15','18','15','16','8','15','8','107','-'),
('Bambino','9','15','9','11','4','9','9','108','-'),
('Bengal Cats','10','16','17','22','8','17','10','109','-'),
```

```
INSERT INTO breedorigin("breed_id","country_id") VALUES
```

```
('1','200'),
('2','201'),
('3','202'),
('4','202'),
('5','202'),
('6','203'),
...
('9','202'),
('10','202'),
('11','206');
```

```
INSERT INTO image ("image_link", "breed_id") VALUES
```

```
('https://dogell.b-cdn.net/uploads/breed/thumb_abyssinian.png','1'),
('https://dogell.b-cdn.net/uploads/breed/thumb_aegean.png','2'),
('https://dogell.b-cdn.net/uploads/breed/thumb_american-bobtail.png','3'),
('https://dogell.b-cdn.net/uploads/breed/thumb_american-shorthair.png','4'),
('https://dogell.b-cdn.net/uploads/breed/thumb_american-wirehair.png','5'),
('https://dogell.b-cdn.net/uploads/breed/thumb_arabian-mau.png','6'),
('https://dogell.b-cdn.net/uploads/breed/thumb_asian.png','7'),
('https://dogell.b-cdn.net/uploads/breed/thumb_australian-mist.png','8'),
('https://dogell.b-cdn.net/uploads/breed/thumb_bambino.png','9'),
('https://dogell.b-cdn.net/uploads/breed/thumb_bengal.png','10'),
('https://dogell.b-cdn.net/uploads/breed/thumb_birman.png','11'),
```