

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Введение в нереляционные базы данных»
Тема: Каталог котиков MongoDB

Студент гр. 8383	_____	Костарев К.В.
Студент гр. 8383	_____	Мололкин К.А.
Студент гр. 8383	_____	Федоров И.А.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург

2021

ЗАДАНИЕ

НА ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

Студенты

Костарев К.В.

Мололкин К.А.

Федоров И.А.

Группа 8383

Тема проекта: Каталог котиков MongoDB

Исходные данные:

Необходимо реализовать приложение – каталог пород котиков (frontend + backend + СУБД MongoDB)

Содержание пояснительной записки:

«Содержание», «Введение», «Качественные требования к решению»,
«Сценарии использования», «Модель данных», «Разработка приложения»,
«Заключение»

Предполагаемый объем пояснительной записки:

Не менее 10 страниц.

Дата выдачи задания:

Дата сдачи реферата:

Дата защиты реферата:

Студент гр. 8383

Костарев К.В.

Студент гр. 8383

Мололкин К.А.

Студент гр. 8383

Федоров И.А.

Преподаватель

Заславский М.М.

АННОТАЦИЯ

Было реализовано веб-приложение – каталог пород котиков с возможностью просмотра отдельной породы, таблицы всех пород, фильтрации пород по каждому из параметров (название породы, характеристики и др.), а также импорта данных из базы данных и экспорта в нее. В качестве СУБД в рамках изучения курса используется MongoDB. Была изучена функциональность и производительность СУБД.

SUMMARY

We developed a web application, which is a catalog of SEAL breeds with the ability to view an individual breed, a table of all breeds, filter breeds by each of the parameters (breed name, characteristics, etc.), as well as importing data from the database and exporting to it. MongoDB is used as a DBMS in the course study. The functionality and performance of the DBMS was studied.

СОДЕРЖАНИЕ

Введение	6
Качественные требования к решению	7
1. Сценарии использования	
1.1. Прототип пользовательского интерфейса	8
1.2. Описание сценариев	8
2. Модель данных	12
2.1. Схема базы данных	12
2.2. Оценка объема информации	13
2.3. Примеры запросов	14
2.4. Сравнение с AQL	15
3. Разработанное приложение	19
3.1. Краткое описание	19
3.2. Используемые технологии и ссылка на приложение	19
Заключение	20

ВВЕДЕНИЕ

Целью данной работы является разработка веб-приложения – каталога пород котиков, с использованием в качестве СУБД MongoDB.

Было реализовано полноценное приложение – сайт с каталогом пород. Для разработки фронтенда был использован фреймворк Angular, а для бэкенда использовался фреймворк Spring.

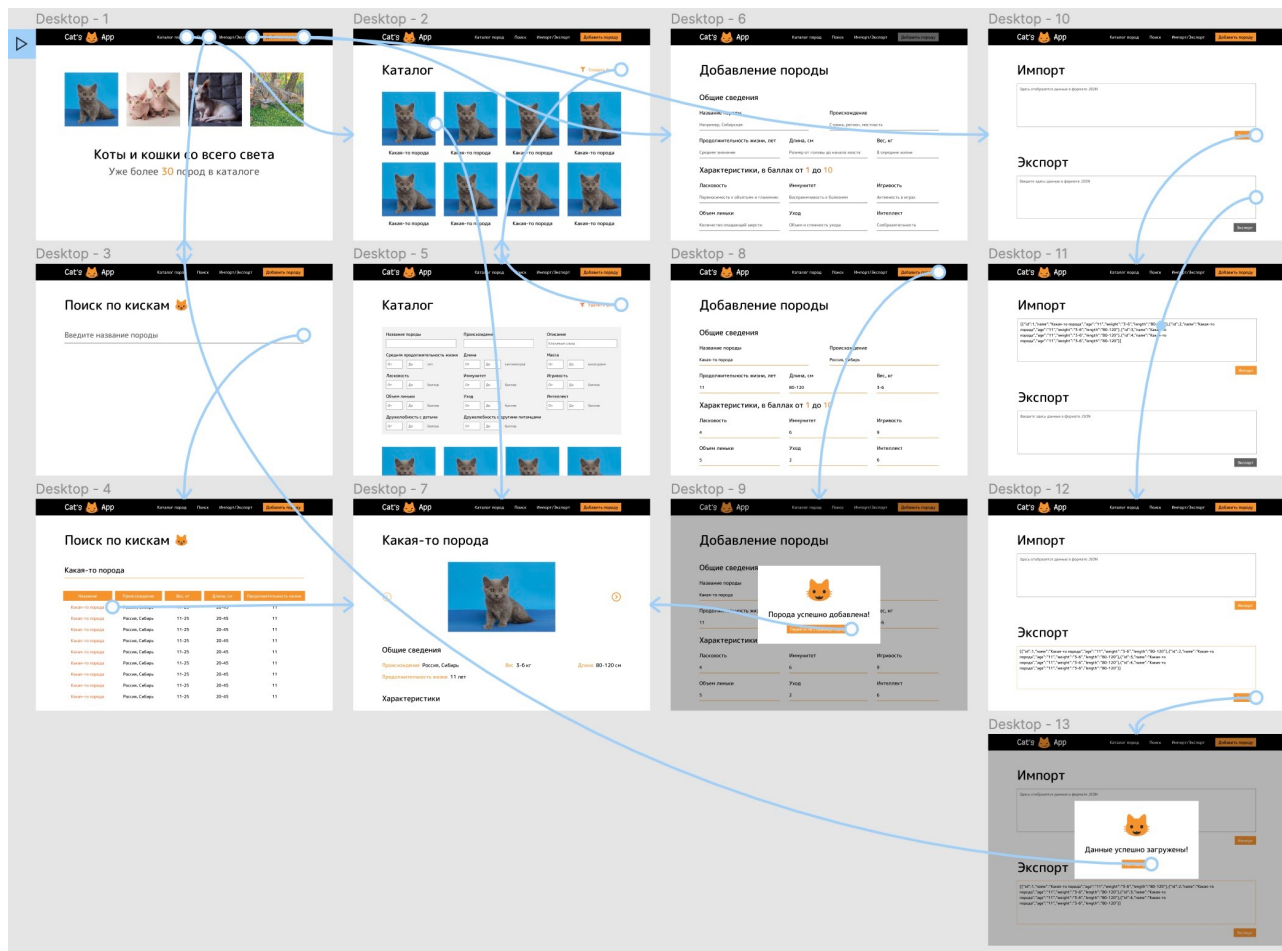
КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

Требуется разработать веб-приложение с использованием СУБД MongoDB и следующим функционалом: просмотр каталога пород котиков, просмотра отдельной породы, фильтрации пород по каждому параметру, импорта и экспорта данных с интерфейсом, описанным в прототипе.

1. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

1.1. Прототип пользовательского интерфейса

Прототип интерфейса приложения был реализован в Figma.



1.2. Описание сценариев

Сценарий – Добавление новой породы:

Действующее лицо: Пользователь

Основной сценарий:

1. Пользователь нажимает кнопку "Добавить породу" в верхней области страницы.
2. Пользователь попадает на страницу добавления породы, и вводит данные для добавления:
 - Средняя продолжительность жизни
 - Длину

- Массу
- Ласковость
- Иммуитет
- Игривость
- Объем линьки
- Уход
- Интеллект
- Дружелюбность с детьми
- Дружелюбность с другими питомцами
- Фотографию породы
- Краткое описание

3. Пользователь нажимает "Submit".

Альтернативный сценарий:

1. Пользователь закрывает страницу без нажатия на кнопку "Submit" (порода просто не добавляется).
2. Пользователь ввел не все обязательные поля.
3. Пользователь ввел неверные данные в некоторые поля.

Результат: Добавлена новая порода в таблицу.

Сценарий – Поиск породы

Действующее лицо: Пользователь

Предусловие: Пользователь находится на любой странице приложения.

Основной сценарий:

1. Пользователь нажимает кнопку "Поиск" в верхней области страницы.
2. Пользователь попадает на страницу поиска породы и вводит в поле название интересующей породы.
3. Пользователь нажимает "Enter".

Альтернативный сценарий:

1. Пользователь не вводит ничего в поле.

2. Пользователь нажимает на иконку конкретной породы и переходит на ее страницу.

Результат: Список совпадений по запросу.

Сценарий – Просмотр каталога пород

Действующее лицо: Пользователь

Предусловие: Пользователь находится на любой странице приложения.

Основной сценарий:

1. Пользователь нажимает кнопку "Каталог".
2. Пользователь попадает на страницу с каталогом пород.
3. Пользователь просматривает породы на странице и переходит на следующую часть с помощью кнопки "далее".

Альтернативный сценарий:

1. Пользователь на странице каталога нажимает на "Добавить фильтр" и вводит данные в соответствующие поля.
2. Пользователь нажимает на иконку конкретной породы и переходит на ее страницу.

Результат: Список пород, хранящихся в каталоге.

Сценарий – Просмотр конкретной породы.

Предусловие: Пользователь находится на странице каталога всех пород и нажимает на иконку породы; Пользователь находится на странице Поиска породы и нажимает на иконку породы; Пользователь находится на странице Добавления новой породы и после успешной операции нажимает кнопку "Перейти на страницу породы".

Основной сценарий:

1. Пользователь просматривает характеристики породы.

Результат: страница конкретной породы.

Сценарий – Импорт данных

Предусловие: Пользователь находится на любой странице.

Основной сценарий:

1. Пользователь нажимает кнопку "Импорт/Экспорт".
2. Пользователь попадает на страницу Импорта/Экспорта.
3. Пользователь вводит в поле БД в JSON-формате.
4. Пользователь нажимает кнопку "Импорт".

Альтернативный сценарий:

1. Пользователь использует API для импорта БД.
2. Пользователь выходит без нажатия "Импорт" (ничего не добавляется).

Результат: Импорт БД пользователя.

Сценарий – Экспорт

Предусловие: Пользователь находится на любой странице.

Основной сценарий:

1. Пользователь нажимает кнопку "Импорт/Экспорт".
2. Пользователь попадает на страницу Импорта/Экспорта.
3. Пользователь нажимает кнопку "Экспорт".

Результат: Пользователь получает БД в формате JSON.

2. МОДЕЛЬ ДАННЫХ

2.1. Схема базы данных

Каждая порода является отдельным документом. Идентификатор документа - идентификатор породы `_id`.

Документ имеет следующие свойства:

- `name` - название породы
- `origin` - место происхождения
- `averageLifespan` – средняя продолжительность жизни
- `description` – описание породы
- `images` – массив ссылок на изображения породы
- `length` - объект со свойствами `from`, `to` (длина от и до)
- `weight` - объект со свойствами `from`, `to` (вес от и до)
- `characteristics` - объект со свойствами `gentleness`, `immunity`, `playfulness`, `molt`, `care`, `intelligence`, `childFriendliness`, `petFriendliness` – характеристики породы в баллах от 1 до 10.

Модель документа:

```
{
  "_id": "<ObjectId>",
  "name": "Сибирская",
  "origin": "Россия, Сибирь",
  "averageLifespan": 5,
  "length": {
    "from": 20,
    "to": 30
  },
  "weight": {
    "from": 3,
    "to": 15
  },
  "characteristics": {
    "gentleness": 5,
    "immunity": 5,
    "playfulness": 6,
    "molt": 3,
    "care": 8,
    "intelligence": 1,
    "childFriendliness": 10,
    "petFriendliness": 7
  },
}
```

```

    "description": "Очень крутая порода",
    "images": [
      "https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.proplan.ru%2Fcat%2Farticle%2Fsibirskaya-koshka%2F&psig=A0vVaw2LsjAvG9dsdK9mtZ31eQjk&ust=1635078597624000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCPjUu8zE4PMCFQAAAAAdAAAAABAD",
      "https://upload.wikimedia.org/wikipedia/commons/thumb/c/c8/%D0%92%D0%B5%D1%82%D0%B5%D1%80%D0%BE%D0%BA_%D0%A1%D0%B8%D0%B1%D0%B5%D1%80%D0%B8%D1%8F.jpg/1200px-%D0%92%D0%B5%D1%82%D0%B5%D1%80%D0%BE%D0%BA_%D0%A1%D0%B8%D0%B1%D0%B5%D1%80%D0%B8%D1%8F.jpg"
    ]
  }

```

2.2. Оценка объема информации

MongoDB, чистый объем:

- name - 30b
- origin - 60b
- lifespan - 4b
- length - 8b (from - 4b, to - 4b)
- weight - 8b (from - 4b, to - 4b)
- characteristics - 32b (8 * 4b)
- description - 1000b
- images - 765 (3 * 255b) (в среднем хранится 3 картинки)

Чистый объем равен $N * (30b + 60b + \dots + 765b)$, то есть $1907N$ байт, где N - количество пород.

MongoDB, фактический объем:

- _id - 15b
- name - 34b
- origin - 66b
- lifespan - 12b
- length - 20b (from - 8b, to - 6b)
- weight - 20b (from - 8b, to - 6b)

- characteristics - 160b (gentleness - 14b, immunity - 12b, playfulness - 15b, molt - 8b, care - 8b, intelligence - 15b, childFriendliness - 21b, petFriendliness - 19b)
- description - 1011b
- images - 770b

Фактический объем равен $N * (15b + 34b + \dots + 770b)$, то есть 2108N байт, где N - количество пород.

Избыточность модели данных равна $2108N / 1907N$.

Скорость запроса на поиск породы по id равен $O(B)$, где B – размер таблицы пород.

2.3. Примеры запросов

1. Добавление породы

```
breed = {
  "name": "Сибирская",
  "origin": "Россия, Сибирь",
  "averageLifespan": 5,
  "length": {
    "from": 20,
    "to": 30
  },
  "weight": {
    "from": 3,
    "to": 15
  },
  "characteristics": {
    "gentleness": 5,
    "immunity": 5,
    "playfulness": 6,
    "molt": 3,
    "care": 8,
    "intelligence": 1,
    "childFriendliness": 10,
    "petFriendliness": 7
  },
  "description": "Очень крутая порода",
  "image":
    "https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.proplan.ru%2Fcat%2Farticle%2Fsibirskaya-koshka%2F&psig=A0vVaw2LsjAvG9dsdK9mtZ31eQjk&ust=1635078597624000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCPjUu8zE4PMCFQAAAAAdAAAAABAD"
}

db.breeds.insertOne(breed)
```

2. Получение всех пород

```
db.breeds.find()
```

3. Поиск по id

```
db.breeds.find({_id: ObjectId("6176bc9dcf931294d4ee22af")})
```

4. Поиск по фильтру

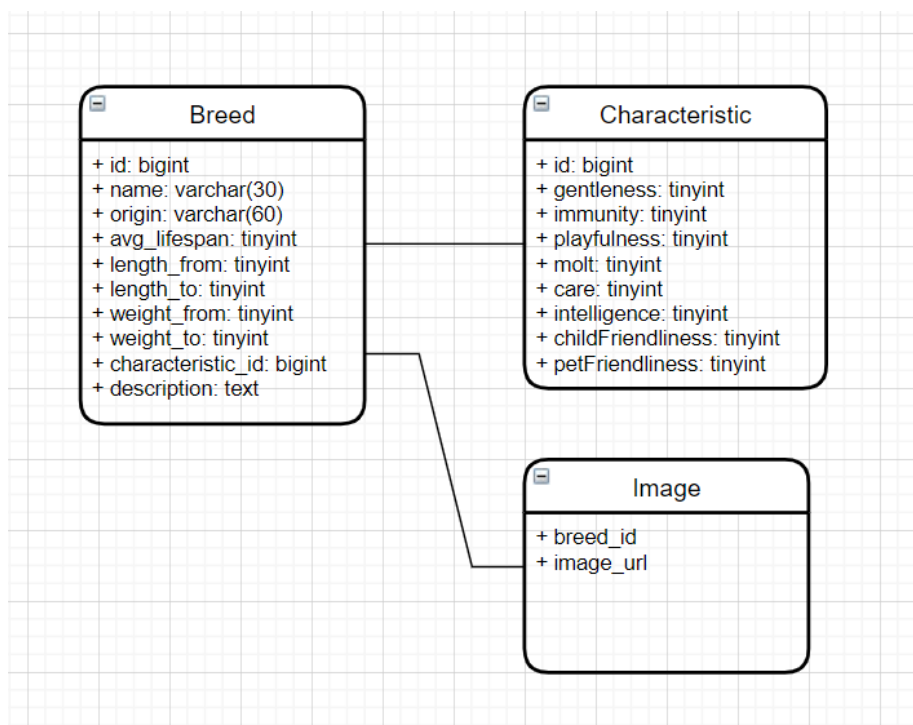
```
db.breeds.find({
  "averageLifespan": {"$gte": 3, "$lte": 7},
  "length.from": 10,
  "length.to": 20,
  "weight.from": 10,
  "weight.to": 20,
  "characteristics.gentleness": {"$gte": 3, "$lte": 7},
  // ...
})
```

5. Поиск попадания в промежуток

```
w = 18
db.breeds.find({'$and': [{'weight.from': {'$lte': w}}, {'weight.to': {'$gte': w}}]})
```

2.4. Сравнение с SQL

Схема БД:



Список сущностей:

1. Breed - сущность породы. Включает в себя следующие атрибуты:
 - a. id: bigint - уникальный идентификатор породы
 - b. name: varchar(30) - название породы
 - c. origin: varchar(60) - место происхождения
 - d. avg_lifespan: tinyint - средняя продолжительность жизни
 - e. length_from: tinyint - минимальная длина особей
 - f. length_to: tinyint - максимальная длина особей
 - g. weight_from: tinyint - минимальный вес особей
 - h. weight_to: tinyint - максимальный вес особей
 - i. characteristic_id: bigint - уникальный идентификатор характеристик породы
 - j. description: text - описание породы
2. Characteristic - сущность характеристик породы. Включает в себя следующие атрибуты:
 - a. id: bigint - уникальный идентификатор характеристик
 - b. gentleness: tinyint - ласковость
 - c. immunity: tinyint - иммунитет
 - d. playfulness: tinyint - игривость
 - e. molt: tinyint - объем линьки
 - f. care: tinyint - уход
 - g. intelligence: tinyint - интеллект
 - h. childFriendliness: tinyint - дружелюбность с детьми
 - i. petFriendliness: tinyint - дружелюбность с другими питомцами
3. Image - сущность картинок. Включает в себя:
 - a. breed_id - идентификатор сущность породы, к которой принадлежит данная картинка
 - b. image_url - ссылка на картинку

Чистый объем:

Breed:

- name - 30b
- origin - 60b
- avg_lifespan - 1b
- length_from - 1b
- length_to - 1b
- weight_from - 1b
- weight_to - 1b
- description - 1000b

Characteristic:

- gentleness - 1b
- immunity - 1b
- playfulness - 1b
- molt - 1b
- care - 1b
- intelligence - 1b
- childFriendliness - 1b
- petFriendliness - 1b

Image:

- image_url - 255b

Чистый объем равен $N * (\text{size}(\text{Breed}) + \text{size}(\text{Characteristic}) + 3 * \text{size}(\text{image}))$, то есть 1867N байт, где N - количество пород.

Фактический объем:

Breed:

- id - 8b
- name - 30b
- origin - 60b
- avg_lifespan - 1b

- length_from - 1b
- length_to - 1b
- weight_from - 1b
- weight_to - 1b
- characteristic_id - 8b
- description - 1000b

Characteristic:

- id - 8b
- gentleness - 1b
- immunity - 1b
- playfulness - 1b
- molt - 1b
- care - 1b
- intelligence - 1b
- childFriendliness - 1b
- petFriendliness - 1b

Image:

- breed_id - 8b
- image_url - 255b

Фактический объем равен $N * (\text{size}(\text{Breed}) + \text{size}(\text{Characteristic}) + 3 * \text{size}(\text{image}))$, то есть $1915N$ байт, где N - количество пород.

Избыточность модели данных равна $1915N / 1867N$.

Мы можем сделать вывод, что SQL требует меньше памяти: $1915N$ против $2108N$ у MongoDB. Написание SQL запросов занимает больше времени. Время доступа к данным является преимуществом MongoDB, потому что в SQL данные хранятся в 2 таблицах, а в MongoDB 1 коллекция (вложенные документы, данные не в нормальной форме).

3. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

3.1. Краткое описание

Backend представляет собой приложение на языке Java, реализованное с помощью фреймворка Spring. Спецификация Rest API:

- `get /api/v1/breed` - возвращает все породы хранящиеся в бд, подходит и для импорта из дб
- `get /api/v1/breed?<"parameter"="value">` - поиск с фильтрацией
- `post /api/v1/breeds` - отправить тело (новая бд) в виде массива json объектов

Frontend – это одностраничное Angular приложение с использованием Rest API backend.

3.2. Используемые технологии и ссылка на приложение

Используемые технологии:

- Frontend: HTML, CSS, TypeScript, Angular 12
- Backend: Java, Spring
- СУБД: MongoDB

Репозиторий в Github: <https://github.com/moevm/nosql2h21-cats-mongodb>

ЗАКЛЮЧЕНИЕ

Было реализовано веб-приложение – каталог пород котиков, изучена функциональность и производительность СУБД MongoDB и было проведено сравнение с SQL.