Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет
по лабораторной работе №6 «Работа с БД в СУБД MongoDB »

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Оспельников А. В.
Факультет: ИКТ
Группа: К3240
Преподаватель: Говорова М.М.

Санкт-Петербург 2023

Оглавление

Цель работы

Овладеть практическими навыками работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.

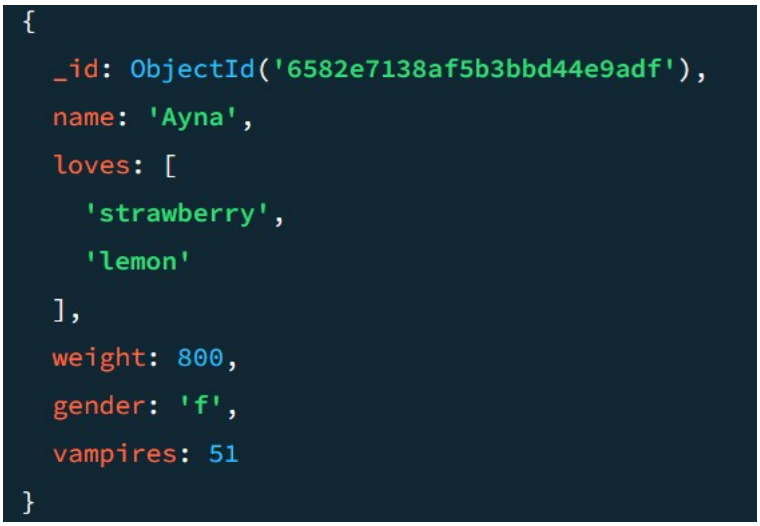
Практическое задание

1. Создать запросы и представления на выборку данных к базе данных MongoDB.
2. Составить запросы на модификацию данных (insert, update, remove).
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. Создать простой и составной индексы для запроса и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду explain.

Выполнение

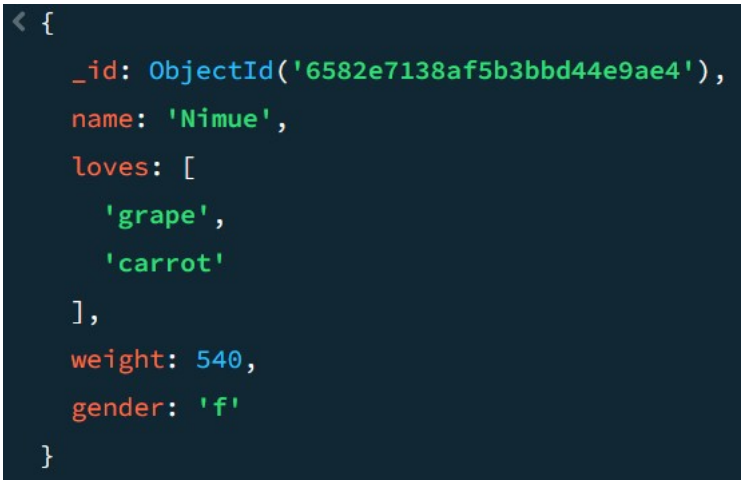
1) Сформируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.

db.unicorns.find({gender: 'f'}).limit(3).sort({name: 1})
db.unicorns.find({gender: 'm'}).sort({name: 1})

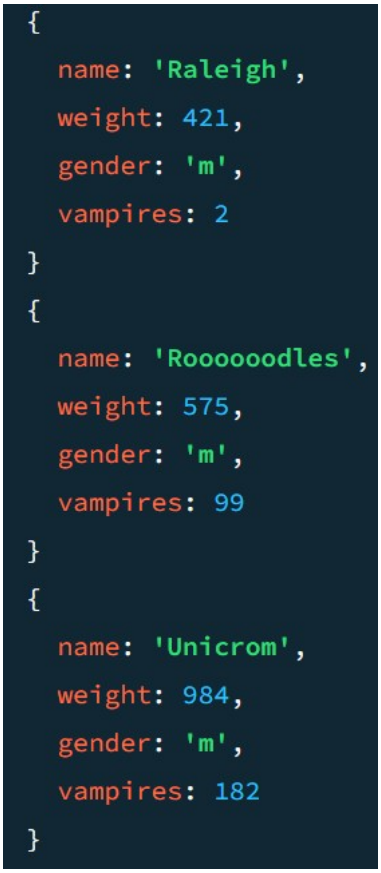


2) Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций findOne и limit.

db.unicorns.findOne({gender: 'f', loves: 'carrot'})



3) *Модифицируйте запрос для вывода списков самцов единорогов, исключив из результата информацию о предпотениях и поле.*
db.unicorns.find({gender: 'm'}, {loves: 0, _id: 0}).sort({name: 1})



4) *Вывести список единорогов в обратном порядке добавления.*
db.unicorns.find().sort({ $natural: -1 })

5) *Вывести список единорогов с названием первого любимого предпочтения, исключив идентификатор.*

db.unicorns.find({}, {loves: {$slice: 1}, _id: 0})

```
{
  name: 'Raleigh',
  loves: [
    'apple'
  ]
}
{
  name: 'Roooooodles',
  loves: [
    'apple'
  ]
}
{
  name: 'Unicrom',
  loves: [
    'energon'
  ]
}
```

6) *Вывести список самок единорогов весом от полутонны до 700 кг, исключив вывод идентификатора*

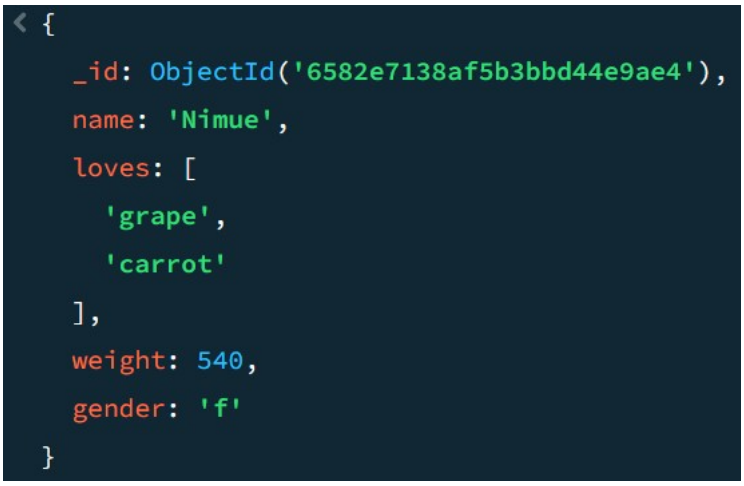
db.unicorns.find({gender: 'f', weight: {$lt : 700}}, {_id: 0})

***7**) Вывести список самцов единорогов весом от полутонны и предпочитающих grape и lemon, исключив вывод идентификатора.*

db.unicorns.find({weight: {$gt : 500}, loves: {$all : ["grape", "lemon"]}}, {_id: 0})

8) *Найти всех единорогов, не имеющих ключ vampires.*

db.unicorns.find({vampires: {$exists: false}})



9) *Вывести список упорядоченный список имен самцов единорогов с информацией об их первом предпочтении.*

db.unicorns.find({gender: 'm'}, {name: 1, _id: 0, loves: {$slice: 1}}).sort({name: 1})

10) *Сформировать функцию для вывода списка самцов единорогов. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке. Вывести результат, используя forEach.*

males = function() { return db.unicorns.find({gender: "m"}) }
var cursor = males().limit(2).sort({name: 1}); null;

cursor.forEach(function(obj){ print(obj.name); })

```
Dunx
Horny
```

11) *Вывести количество самок единорогов весом от полутонны до 600 кг.*

db.unicorns.find({gender: 'f', weight: {$gt : 500, $lt : 600}}).count()

```
2
```

12) *Вывести список предпочтений.*

db.unicorns.distinct("loves")

```
[
  'apple',      'carrot',
  'chocolate',  'energon',
  'grape',      'lemon',
  'papaya',     'redbull',
  'strawberry', 'sugar',
  'watermelon'
]
```

13) *Посчитать количество особей единорогов обоих полов*.

db.unicorns.aggregate({"$group": {_id: "$gender", count: {$sum: 1}}})

```
{
  _id: 'm',
  count: 7
}
{
  _id: 'f',
  count: 5
}
```

14) *Для самки единорога* `Ayna` *внести изменения в БД: теперь ее вес 800, она убила 51 вапмира*

db.unicorns.updateOne({name: "Ayna"}, {$set: {name: "Ayna", weight: 800, vampires: 51}}, {upsert: true})

```
{
  _id: ObjectId('6582e7138af5b3bbd44e9adf'),
  name: 'Ayna',
  loves: [
    'strawberry',
    'lemon'
  ],
  weight: 800,
  gender: 'f',
  vampires: 51
}
```


15) *Для самца единорога* `Raleigh` *внести изменения в БД: теперь он любит рэдбул.*
db.unicorns.updateOne({name: "Raleigh"}, {$set: {name: "Raleigh", loves: ['Redbull']}}, {upsert: true})


```
{
  _id: ObjectId('6582e7138af5b3bbd44e9ae1'),
  name: 'Raleigh',
  loves: [
    'Redbull'
  ],
  weight: 421,
  gender: 'm',
  vampires: 7
}
```


16) *Всем самцам единорогов увеличить количество убитых вапмиров на 5.*
db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}})


```
{
  _id: ObjectId('6582e7138af5b3bbd44e9ae1'),
  name: 'Raleigh',
  loves: [
    'Redbull'
  ],
  weight: 421,
  gender: 'm',
  vampires: 2
}
```

```
{
  _id: ObjectId('6582e7138af5b3bbd44e9ae1'),
  name: 'Raleigh',
  loves: [
    'Redbull'
  ],
  weight: 421,
  gender: 'm',
  vampires: 7
}
```

17) *Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.*
db.towns.updateOne({name: "Portland"}, {$unset: {"mayor.party": 1}})

18) *Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.*
db.unicorns.updateOne({name: "Pilot"}, {$push: {loves: "chocolate"}})

19) *Изменить информацию о самке единорога* `Aurora`*: теперь она любит еще и сахар, и лимоны.*
db.unicorns.updateOne({name: "Aurora"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}})

20) *Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.*
db.unicorns.update({name: 'Horny'}, {$set: {area: {$ref: "areas", $id: "cone"}}})

```
_id: "cone"
name: "Cone forest"
description: "Big forest with most of trees as cones"
```

```
_id: "des"
name: "Desert"
description: "Sand area with lack of water"
```

```
_id: "plains"
name: "Plains"
description: "Green area with good living conditions"
```

21) 1. *Создайте объемную коллекцию* `numbers`*, задействовав курсор:*

```
for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
```

2. *Выберите последних четыре документа.*
3. *Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра* `executionTimeMillis`*)*
4. *Создайте индекс для ключа* `value`*.*
5. *Получите информацию о всех индексах коллекции* `numbres`*.*
6. *Выполните запрос 2.*
7. *Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?*
8. *Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?*

Выполнив задание, я заметил, что без индекса последние 4 элемента выдаются за 73 мс, а с идексом 59 мс. Таким образом, эффективность индексов примерно 20%

Вывод

В данной лабораторной работе выполнены различные запросы на создание функций и курсоров, а также на модификацию данных: вставка, изменение и удаление. Были выполнены запросы без индекса и созданы планы запросов через EXPLAIN, далее были созданы индексы для различных запросов, в случае использования индекса получилось выиграть 19.8%.