**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: Сачук А. А.

Факультет: ИКТ

Группа: К3239 Преподаватель: Говорова М.М.



Санкт-Петербург 2023

**Оглавление**

Цель работы ................................................................................................................................................. 3

Выполнение задания......................................................................................................................................3

Вывод .......................................................................................................................................................... 28

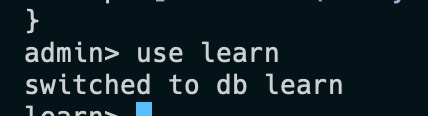
**Цель:** овладеть практическими навыками работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.

**Оборудование**: компьютерный класс.

**Программное обеспечение**: СУБД MongoDB 4+, 6.0.6 (текущая).

**Выполнение работы**

**2.1.1** 1) Создайте базу данных learn.



2) Заполните коллекцию единорогов unicorns:

3) Используя второй способ, вставьте в коллекцию единорогов документ:

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание**

4) Проверьте содержимое коллекции с помощью метода find:

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

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

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

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Самцы – db.unicorns.find({gender: 'm'}).sort({name: 1})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**2.2.2** Модифицируйте запрос для вывода списков самцов единорогов, исключив из результата информацию о предпотениях и поле

Самцы - db.unicorns.find({gender: 'm'}, {loves: 0, vampires: 0}).sort({name: 1})



Самки - db.unicorns.find({gender: 'f'}, {loves: 0, vampires: 0}).sort({name: 1}).limit(3)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**2.2.3** Вывести список единорогов в обратном порядке добавления.

db.unicorns.find().sort({$natural: -1})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**2.2.4** Вывести список единорогов с названием первого любимого предпочтения, исключив идентификатор

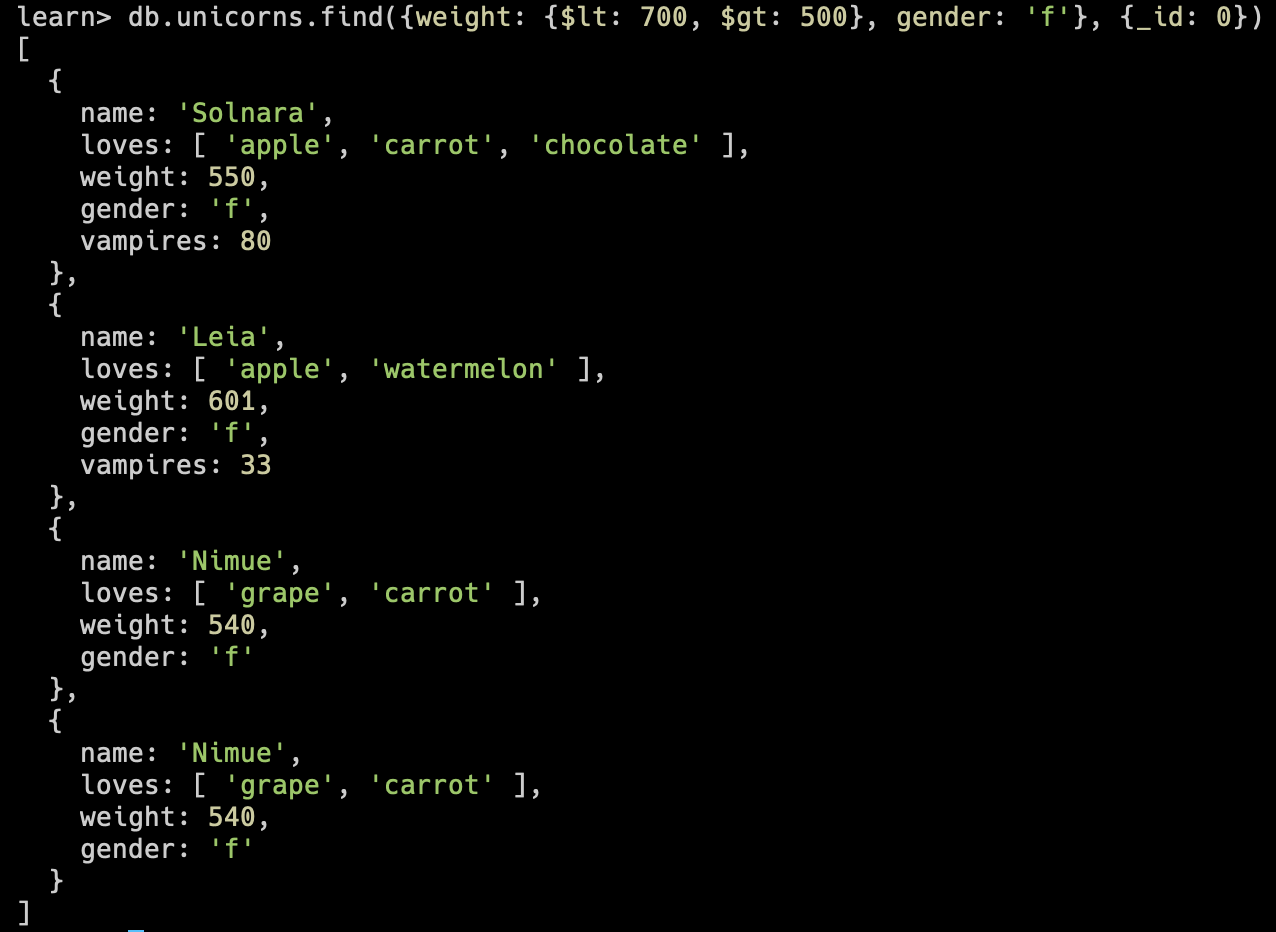
**db.unicorns.find({}, {\_id: 0, loves: {$slice: [0, 1]}})**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание**

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

**db.unicorns.find({weight: {$lt: 700, $gt: 500}, gender: 'f'}, {\_id: 0})**

****

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

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

**Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание**

**2.3.3** Найти всех единорогов, не имеющих ключ vampires

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

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**2.3.4** Вывести список упорядоченный список имен самцов единорогов с информацией об их первом предпочтении

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

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**3.1.1**

1. Создайте коллекцию towns, включающую следующие документы: db.towns.insert(…)
2. Сформировать запрос, который возвращает список городов с независимыми мэрами (party="I"). Вывести только название города и информацию о мэре.

db.towns.find({'mayor.party': 'I'}, {'name': 1, 'mayor.name': 1})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Сформировать запрос, который возвращает список беспартийных мэров (party отсутствует). Вывести только название города и информацию о мэре.

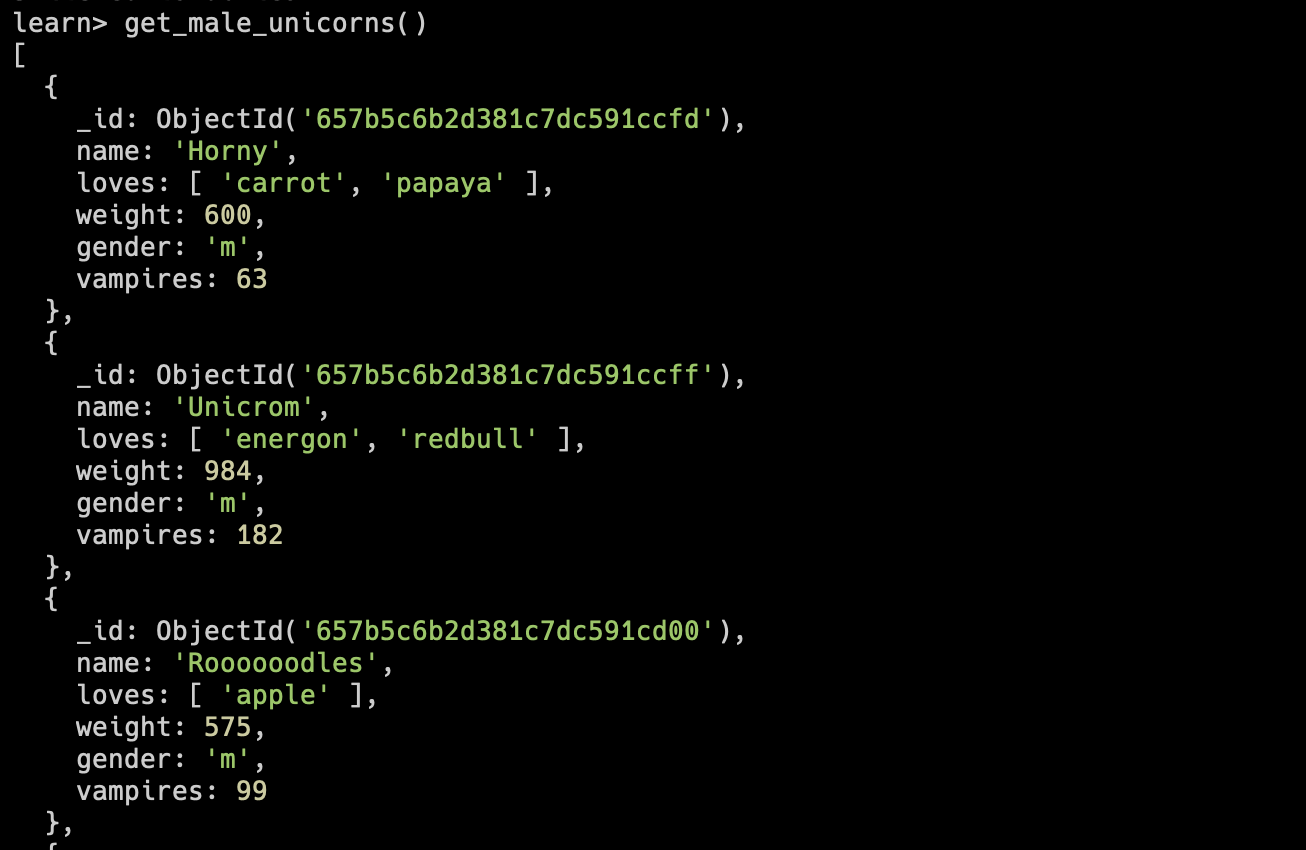
db.towns.find({'mayor.party': {$exists: false}}, {'name': 1, mayor: 1})

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**3.1.2.** 1)Сформировать функцию для вывода списка самцов единорогов.

function get\_male\_unicorns() { return db.unicorns.find({gender: "m"}) }



1. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.

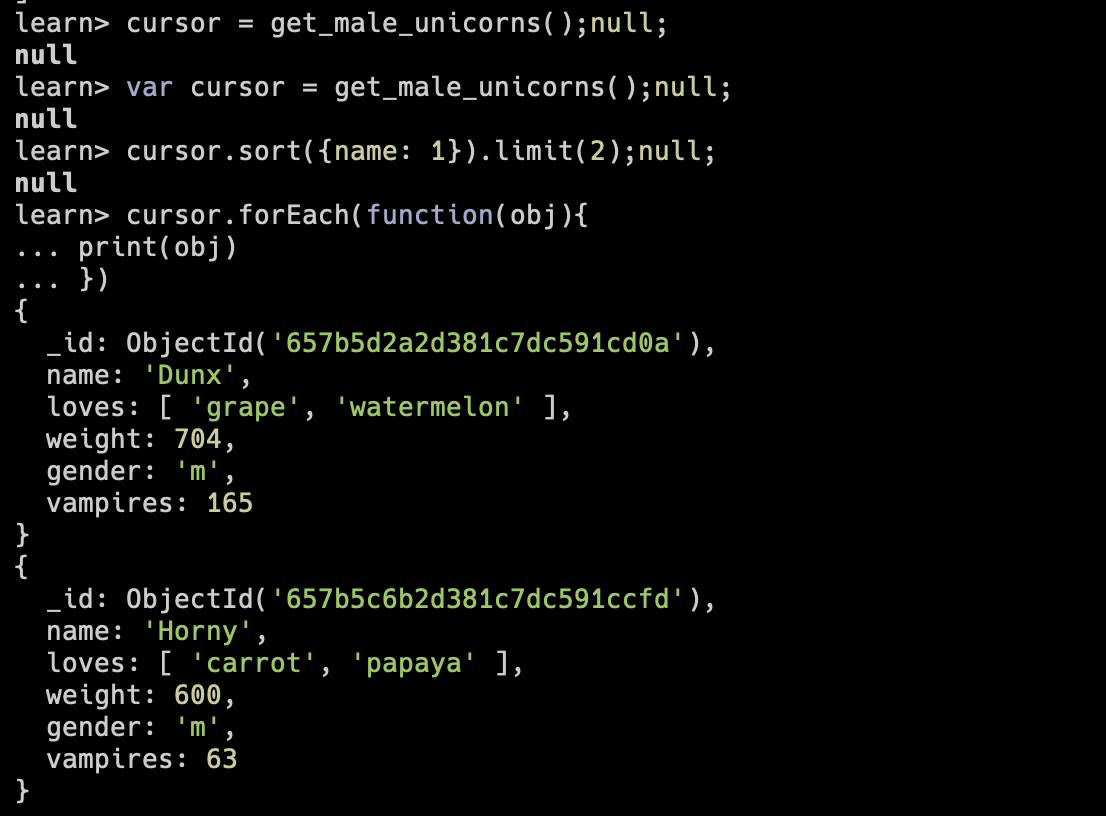
* var cursor = get\_male\_unicorns();null;
* cursor.sort({name: 1}).limit(2);null;

1. Вывести результат, используя forEach.

* cursor.forEach(function(obj){

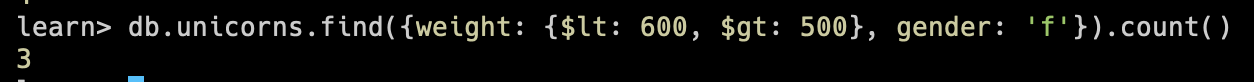
print(obj)

})



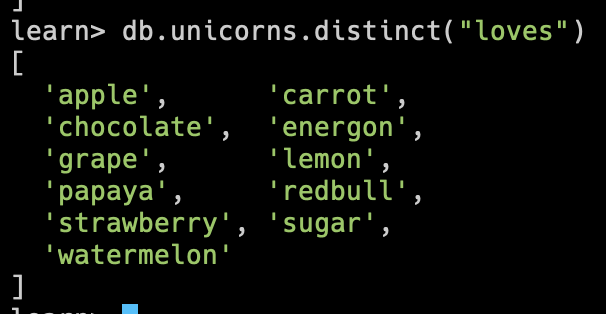
**3.2.1** Вывести количество самок единорогов весом от полутонны до 600 кг.

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

****

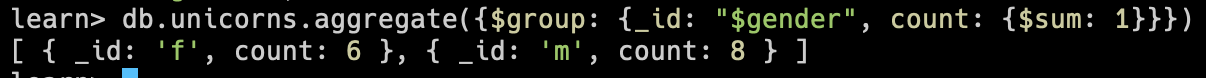
**3.2.2** Вывести список предпочтений.

db.unicorns.distinct("loves")

****

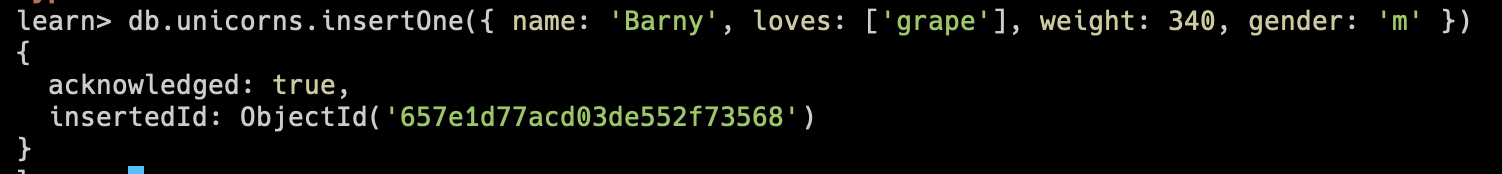
**3.2.3** Посчитать количество особей единорогов обоих полов.

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

****

**3.3.1** 1)Выполнить команду

db.unicorns.insertOne({ name: 'Barny', loves: ['grape'], weight: 340, gender: 'm' })

****

1. Проверить содержимое коллекции unicorns

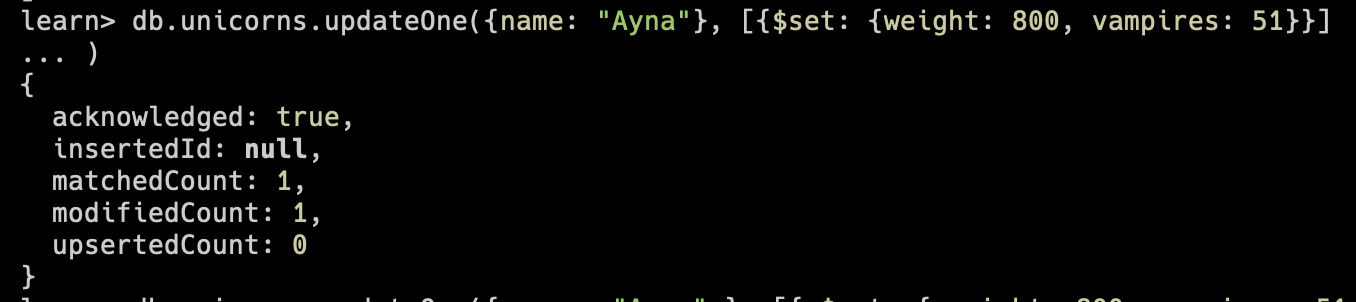
db.unicorns.find(ObjectId('657e1d77acd03de552f73568'))

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**3.3.2** Для самки единорога Ayna внести изменения в БД: теперь ее вес 800, она убила 51 вапмира.

db.unicorns.updateOne({ name: "Ayna" }, [{ $set: { weight: 800, vampires: 51}}]



1. Проверить содержимое коллекции unicorns.

db.unicorns.find({ name: "Ayna" })

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**3.3.3**

1. Для самца единорога Raleigh внести изменения в БД: теперь он любит рэдбул.

db.unicorns.updateOne({name: "Raleigh"}, [{$set: {loves: ["redbull"]}}]

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Проверить содержимое коллекции unicorns.

db.unicorns.find({ name: "Raleigh" })

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**3.3.4**

1. Всем самцам единорогов увеличить количество убитых вапмиров на 5.

db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Проверить содержимое коллекции unicorns.

db.unicorns.find({gender: 'm'})

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание** **Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**3.3.5**

1. Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.

db.towns.updateOne({name: "Portland"}, [{$unset: "mayor.party"}])

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Проверить содержимое коллекции towns

db.towns.find({name: "Portland"})

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**3.3.6**

1. Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.

db.unicorns.updateOne({name: "Pilot"}, {$push: {loves: "chocolate"}})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Проверить содержимое коллекции unicorns.

db.unicorns.find({name: "Pilot"})

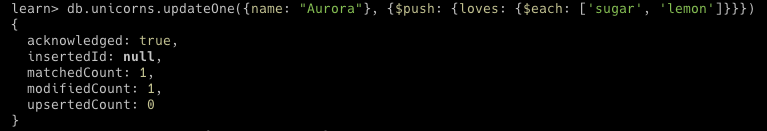
**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**3.3.7**

1. Изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и лимоны.

db.unicorns.updateOne({name: "Aurora"}, {$push: {loves: {$each: ['sugar', 'lemon']}}})



1. Проверить содержимое коллекции unicorns.

db.unicorns.find({name: "Aurora"})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**3.4.1** Создайте коллекцию towns, включающую следующие документы:



1. Удалите документы с беспартийными мэрами.

db.towns.deleteMany({$not: {$exists: "mayor.party"}})



1. Проверьте содержание коллекции.

db.towns.find()

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Очистите коллекцию.

db.towns.drop()



1. Просмотрите список доступных коллекций.

show collections

****

**4.1.1** 1)Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.

db.locations.insertMany([

{"\_id": "MGL", "name": "Mystic Glade", "description": "A hidden glade surrounded by ancient forests, where magic permeates the air."},

{"\_id": "CRY", "name": "Crystal Haven", "description": "A serene valley adorned with crystalline formations, reflecting the colors of the unicorns' enchantments."},

{"\_id": "STL", "name": "Starlight Meadow", "description": "A meadow bathed in perpetual starlight, where unicorns graze under the celestial glow."},

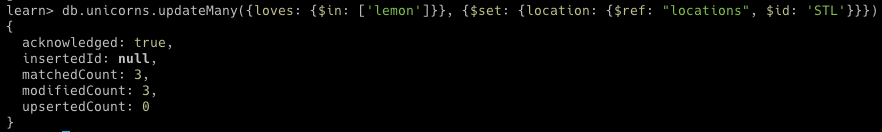
])

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Включите для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания.

db.unicorns.updateMany({loves: {$in: ['lemon']}}, {$set: {location: {$ref: "locations", $id: 'STL'}}})



1. Проверьте содержание коллекции едиорогов.

db.unicorns.find({loves: {$in: ['lemon']}})

**Изображение выглядит как текст, снимок экрана

Автоматически созданное описание**

**4.2.1** Проверьте, можно ли задать для коллекции unicorns индекс для ключа name с флагом unique.

db.numbers.find({value: {$in: [9999, 9998, 9997]}})

****

В коллекции было 2 единорога с именем ‘Nimue’

db.unicorns.deleteOne({name: "Nimue"})

Попробуем еще раз навесить индекс

****

**4.3.1**

1. Получите информацию обо всех индексах коллекции unicorns.

db.unicorns.getIndexes()

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Удалите все индексы, кроме индекса для идентификатора.

db.unicorns.dropIndex("name\_1")



1. Попытайтесь удалить индекс для идентификатора.

db.unicorns.dropIndex("\_id\_")



**4.4.1**

1. Создайте объемную коллекцию numbers, задействовав курсор:

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

1. Выберите последних четыре документа.

db.numbers.find({value: {$in: [9999, 9998, 9997]}})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)

db.numbers.explain("executionStats").find({value: {$in: [9999, 9998, 9997]}})

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

1. Создайте индекс для ключа value.

db.numbers.ensureIndex({value: 1})



1. Получите информацию обо всех индексах коллекции numbres.

db.numbers.getIndexes()

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Выполните запрос 2.

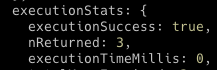
db.numbers.find({value: {$in: [9999, 9998, 9997]}})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?

db.numbers.explain("executionStats").find({value: {$in: [9999, 9998, 9997]}})



1. Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?

Супер ускорение для запросов с фильтрацией по конкретным значениям. Если бы мы выбирали 4 последних документа с помощью сортировки, то индексы бы никак не ускорили выполнение запроса.

**Вывод**

В ходе лабораторной работы были освоены практические навыки по созданию, функций, документов и коллекций в СУБД MongoDB. Были созданы функции на выборку данных, а также были созданы запросы для удаления, замены и вставки данных разными способами. Также был создан курсор и различные запросы на выборку данных.