**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»**

**Обучающийся** Проскуряков Роман Владимирович

**Факультет** прикладной информатики

**Группа** K3239

**Направление подготовки** 09.03.03 Прикладная информатика

**Образовательная программа** Мобильные и сетевые технологии 2023

**Преподаватель** Говорова Марина Михайловна

Санкт-Петербург

2024/2025

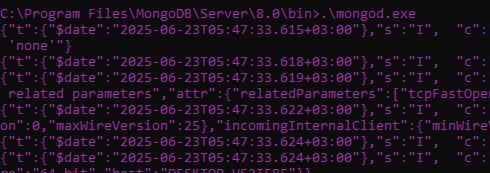
**Цель работы:** овладеть практическими навыками работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.

**Практическое задание:**

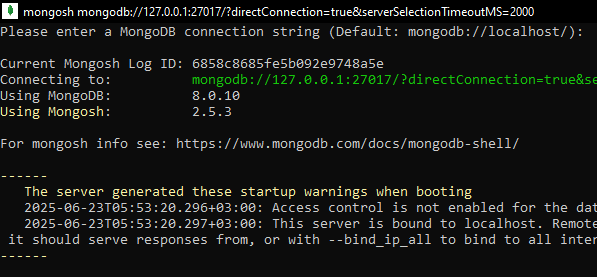
1. ВВЕДЕНИЕ В СУБД MONGODB. УСТАНОВКА MONGODB. НАЧАЛО РАБОТЫ С БД
2. CRUD-ОПЕРАЦИИ В СУБД MONGODB. ВСТАВКА ДАННЫХ. ВЫБОРКА ДАННЫХ
3. АПРОСЫ К БАЗЕ ДАННЫХ MONGODB.ВЫБОРКА ДАННЫХ. ВЛОЖЕННЫЕ ОБЪЕКТЫ. ИСПОЛЬЗОВАНИЕ КУРСОРОВ. АГРЕГИРОВАННЫЕ ЗАПРОСЫ. ИЗМЕНЕНИЕ ДАННЫХ
4. ССЫЛКИ И РАБОТА С ИНДЕКСАМИ В БАЗЕ ДАННЫХ MONGODB

**Выполнение:**

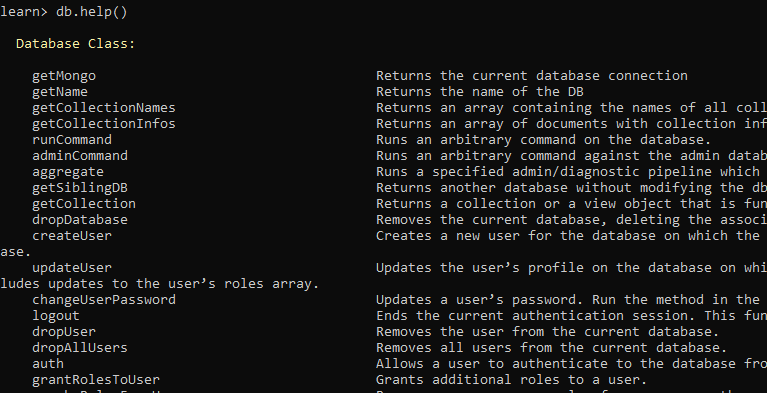
1. **ВВЕДЕНИЕ В СУБД MONGODB. УСТАНОВКА MONGODB. НАЧАЛО РАБОТЫ С БД**
   1. **Установите MongoDB для обеих типов систем (32/64 бита).**

****

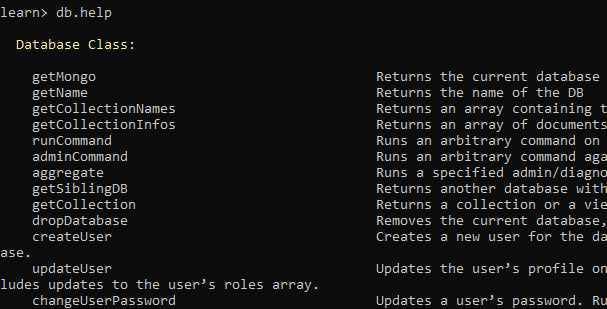
* 1. **Проверьте работоспособность системы запуском клиента mongo.**

****

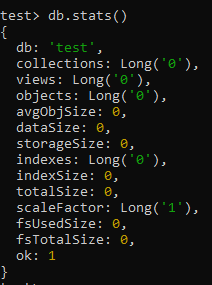
* 1. **Выполните методы:**
     1. **db.help()**

****

* + 1. **db.help**

****

* + 1. **db.stats()**

****

* 1. **Создайте БД learn.**

use learn

****

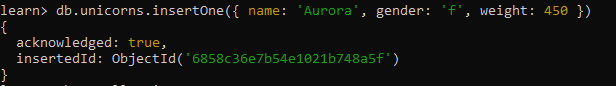
* 1. **Получите список доступных БД.**

show dbs



* 1. **Создайте коллекцию unicorns, вставив в нее документ {name: 'Aurora', gender: 'f', weight: 450}.**

db.unicorns.insertOne({ name: 'Aurora', gender: 'f', weight: 450 })



* 1. **Просмотрите список текущих коллекций.**

show collections



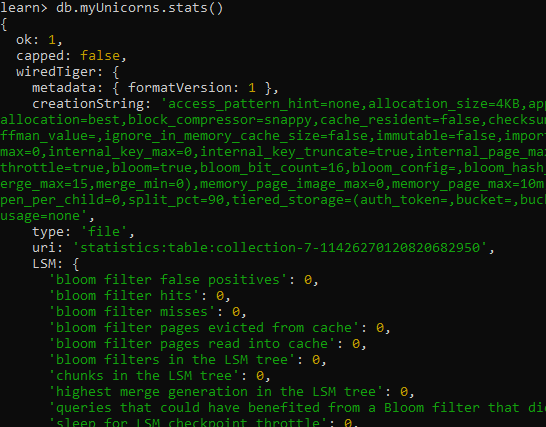
* 1. **Переименуйте коллекцию unicorns.**

db.unicorns.renameCollection('myUnicorns')



* 1. **Просмотрите статистику коллекции.**

db.myUnicorns.stats()



* 1. **Удалите коллекцию.**

db.myUnicorns.drop()

****

* 1. **Удалите БД learn.**

db.dropDatabase()

****

1. **2 CRUD-ОПЕРАЦИИ В СУБД MONGODB. ВСТАВКА ДАННЫХ. ВЫБОРКА ДАННЫХ**
   1. **ВСТАВКА ДОКУМЕНТОВ В КОЛЛЕКЦИЮ**
      1. **Создайте базу данных learn и заполните коллекцию unicorns указанными документами.**

use learn

db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63})

... (остальные 10 insert)

var doc = {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}

db.unicorns.insert(doc)

** **

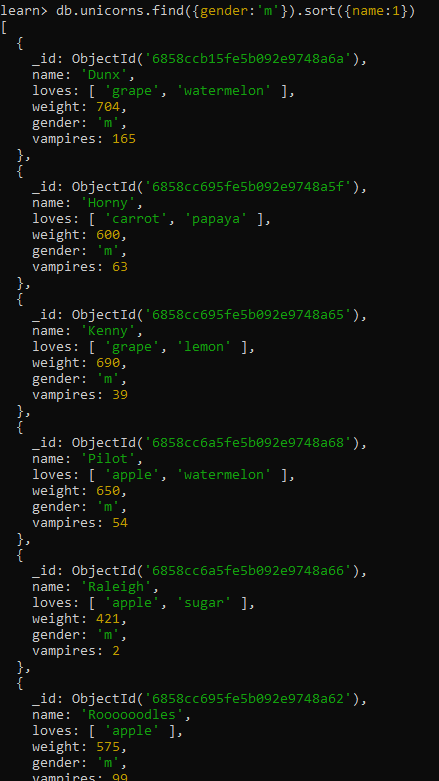
* 1. **ВЫБОРКА ДАННЫХ ИЗ БД**
     1. **Сформируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени. Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций findOne и limit.**

самцы

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

самки, первые три

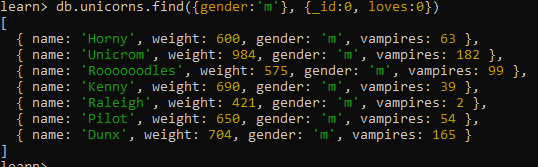
db.unicorns.find({gender:'f'}).sort({name:1}).limit(3)

****

****

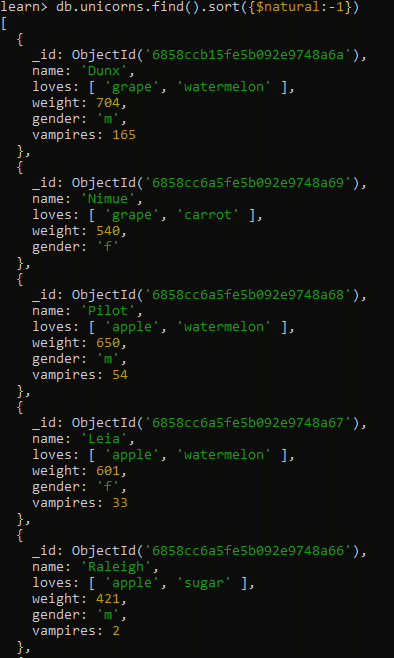
* + 1. **Модифицируйте запрос для вывода списков самцов единорогов, исключив из результата информацию о предпочтениях и поле.**

db.unicorns.find({gender:'m'}, {\_id:0, loves:0})

****

* + 1. **Вывести список единорогов в обратном порядке добавления.**

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



* + 1. **Вывести список единорогов с названием первого любимого предпочтения, исключив идентификатор.**

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

****

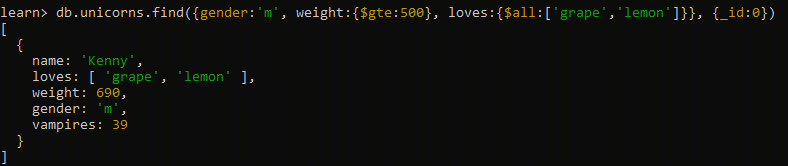
* 1. **ЛОГИЧЕСКИЕ ОПЕРАТОРЫ**
     1. **Вывести список самок единорогов весом от полутонны до 700 кг (без \_id)**

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



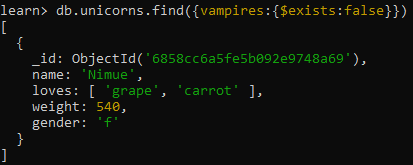
* + 1. **Вывести список самцов единорогов весом ≥ 500 кг, предпочитающих grape и lemon (без \_id).**

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



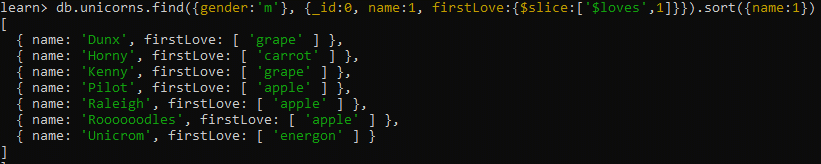
* + 1. **Найти всех единорогов, не имеющих ключ vampires.**

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

****

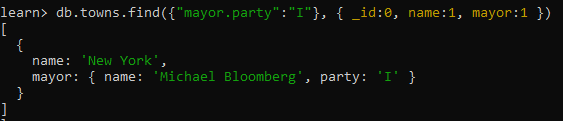
* + 1. **Вывести список упорядоченный список имен самцов единорогов с информацией об их первом предпочтении.**

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



1. **ЗАПРОСЫ К БАЗЕ ДАННЫХ MONGODB. ВЫБОРКА ДАННЫХ. ВЛОЖЕННЫЕ ОБЪЕКТЫ. ИСПОЛЬЗОВАНИЕ КУРСОРОВ. АГРЕГИРОВАННЫЕ ЗАПРОСЫ. ИЗМЕНЕНИЕ ДАННЫХ**
   1. **ЗАПРОС К ВЛОЖЕННЫМ ОБЪЕКТАМ**
      1. **Создайте коллекцию towns, включающую следующие документы:**
         1. **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"}, { \_id:0, name:1, mayor:1 })

****

* + - 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}}, { \_id:0, name:1, mayor:1 })

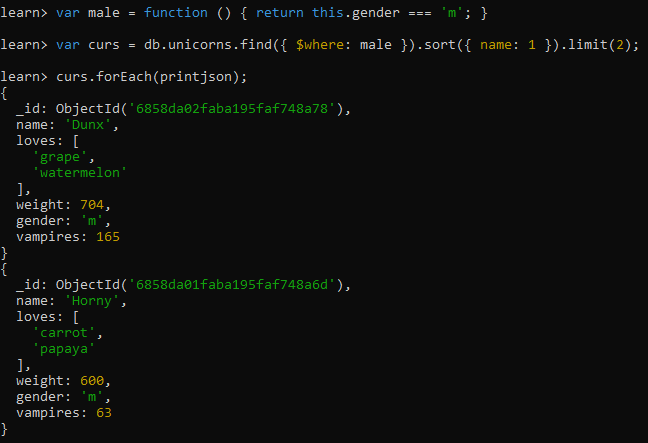


* + 1. **Сформировать функцию для вывода списка самцов единорогов. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке. Вывести результат, используя forEach.**

var male = function () { return this.gender === 'm'; }

var curs = db.unicorns.find({ $where: male }).sort({ name: 1 }).limit(2);

curs.forEach(printjson);

****

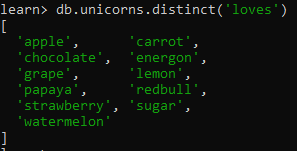
* 1. **АГРЕГИРОВАННЫЕ ЗАПРОСЫ**
     1. **Вывести количество самок единорогов весом от полутонны до 600 кг.**

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

****

* + 1. **Вывести список предпочтений (loves).**

db.unicorns.distinct('loves')

****

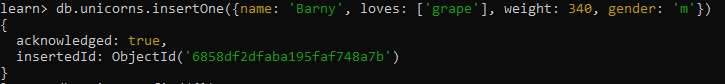
* + 1. **Посчитать количество единорогов обоих полов.**

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

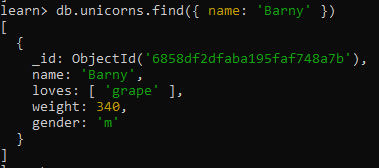
****

* 1. **РЕДАКТИРОВАНИЕ ДАННЫХ**
     1. **Сохранить документ Barny в коллекции unicorns и проверить содержимое.**

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

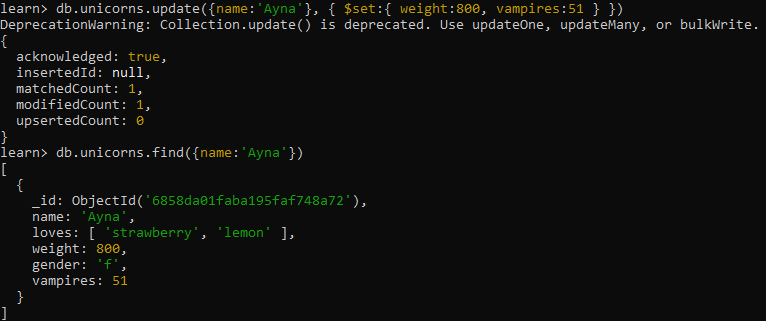
****

db.unicorns.find({ name: 'Barny' })

****

* + 1. **Изменить Ayna: вес = 800, vampires = 51 и проверить коллекцию.**

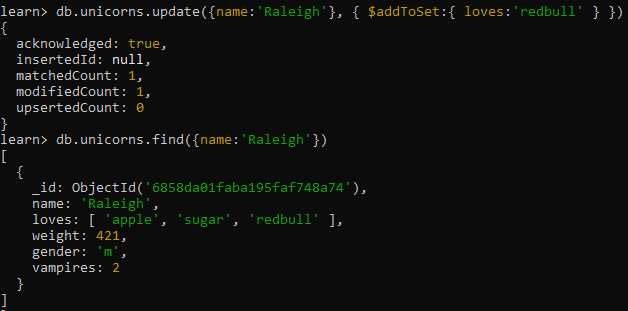
db.unicorns.update({name:'Ayna'}, { $set:{ weight:800, vampires:51 } })

db.unicorns.find({name:'Ayna'})****

* + 1. **Изменить Raleigh: добавить предпочтение redbull и проверить коллекцию.**

db.unicorns.update({name:'Raleigh'}, { $addToSet:{ loves:'redbull' } })

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

****

* + 1. **Всем самцам увеличить vampires на 5 и проверить коллекцию.**

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

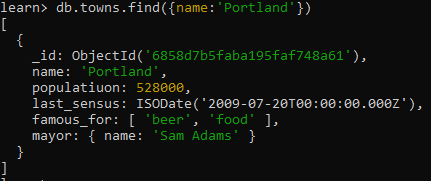
db.unicorns.find({gender:'m'}, {name:1, vampires:1})

****

* + 1. **Обновить город Portland: мэр беспартийный и проверить коллекцию towns.**

db.towns.update({name:'Portland'}, { $unset:{ 'mayor.party':1 } })

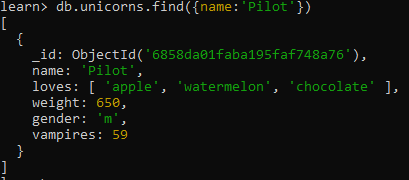
db.towns.find({name:'Portland'})

****

* + 1. **Изменить Pilot: теперь любит chocolate и проверить коллекцию.**

db.unicorns.update({name:'Pilot'}, { $addToSet:{ loves:'chocolate' } })

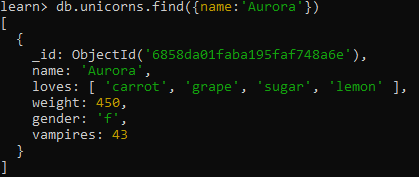
db.unicorns.find({name:'Pilot'})

****

* + 1. **Изменить Aurora: теперь любит sugar и lemon и проверить коллекцию.**

db.unicorns.update({name:'Aurora'}, { $addToSet:{ loves:{ $each:['sugar','lemon'] } } })

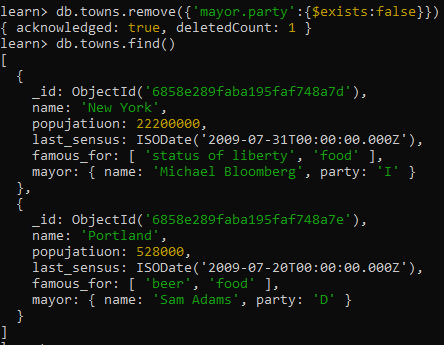
db.unicorns.find({name:'Aurora'})

****

* 1. **УДАЛЕНИЕ ДАННЫХ ИЗ КОЛЛЕКЦИИ**
     1. **Создайте коллекцию towns, включающую следующие документы:**
        1. **Удалите документы с беспартийными мэрами. Проверьте содержание коллекции.**

db.towns.remove({'mayor.party':{$exists:false}})

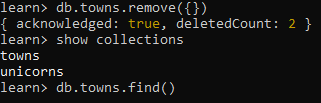
db.towns.find()

****

* + - 1. **Очистите коллекцию. Просмотрите список доступных коллекций.**

db.towns.remove({})

show collections

****

1. **ССЫЛКИ И РАБОТА С ИНДЕКСАМИ В БАЗЕ ДАННЫХ MONGODB**
   1. **ССЫЛКИ В БД**
      1. **Создать коллекцию зон обитания, добавить ссылки у нескольких единорогов, проверить unicorns**

db.habitats.insert({\_id:'forest', full:'Enchanted Forest', descr:'Dense magical woods'})

db.habitats.insert({\_id:'moon', full:'Moon (satellite)', descr:'A cold, airless wasteland'})

db.unicorns.update({name:'Horny'}, { $set:{ habitat:{ $ref:'habitats', $id:'forest' } } })

db.unicorns.update({name:'Dunx'}, { $set:{ habitat:{ $ref:'habitats', $id:'moon' } } })

db.unicorns.find({habitat:{$exists:true}})

****

* 1. **НАСТРОЙКА ИНДЕКСОВ**
     1. **Проверить возможность установки unique‑индекса на name в коллекции unicorns.**

db.unicorns.createIndex({name:1}, {unique:true})

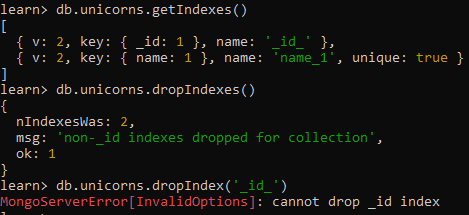
****

* 1. **УПРАВЛЕНИЕ ИНДЕКСАМИ**
     1. **Получить все индексы коллекции unicorns, удалить все кроме \_id\_, попытаться удалить \_id\_**

db.unicorns.getIndexes()

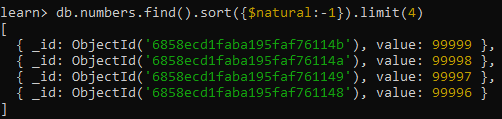
db.unicorns.dropIndexes()

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

****

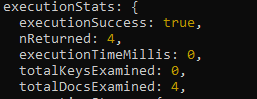
* 1. **ПЛАН ЗАПРОСА**
     1. **Создайте объемную коллекцию numbers, задействовав курсор:**
        1. **Выберите последних четыре документа.**

db.numbers.find().sort({$natural:-1}).limit(4)

****

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

db.numbers.explain('executionStats').find().sort({$natural:-1}).limit(4)

****

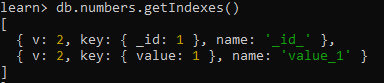
* + - 1. **Создайте индекс для ключа value.**

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

****

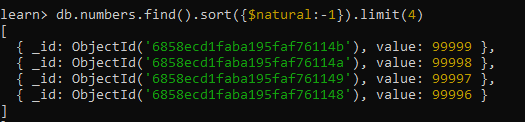
* + - 1. **Получите информацию о всех индексах коллекции numbres.**

db.numbers.getIndexes()

****

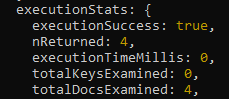
* + - 1. **Выполните запрос 2.**

db.numbers.find().sort({$natural:-1}).limit(4)

****

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

db.numbers.explain('executionStats').find().sort({$natural:-1}).limit(4)

****

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

Время запросов одинаково и равно 0. Скорее всего данных слишком мало. Но у друга изменения были, так что я всё осознал.

**Выводы**

MongoDB очень похоже по функционалу и логике на sql. Многие функции называются так же. Он похожи как два языка программирования с разными стилями. В ходе выпо1лнения лабораторной работы были получены базовые навыки управления базами данных MongoDB, что будет полезно для дальнейшей работы с базами данных и разработки эффективных баз данных.