**Министерство науки и высшего образования Российской Федерации** федеральное государственное автономное образовательное учреждение высшего образования

# «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

**Отчет**

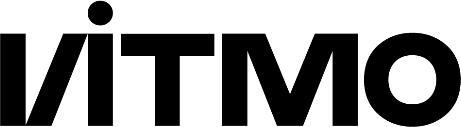
по лабораторной работе №6 «Работа с БД в СУБД MongoDB**»** по дисциплине **«Проектирование и реализация баз данных»**

Автор:Джаватов И.А.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

**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

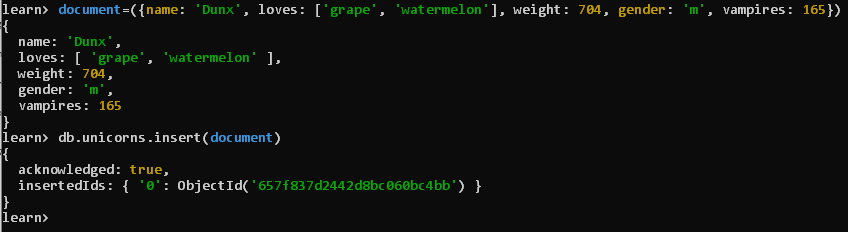
**Практическое задание 2.1.1:**

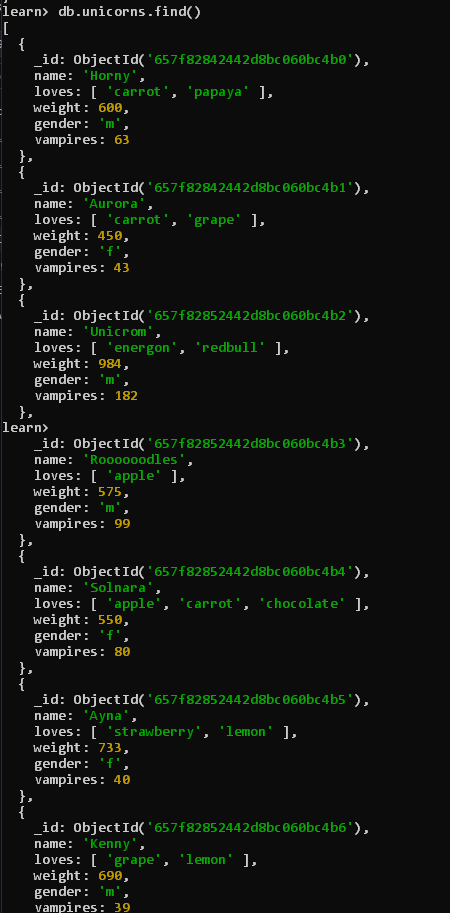
1. Создайте базу данных learn.
2. Заполните коллекцию единорогов unicorns:

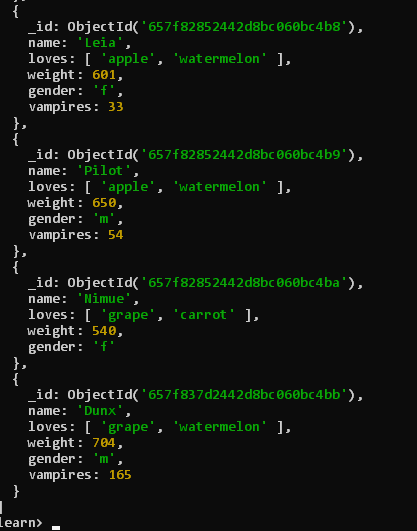




1. Используя второй способ, вставьте в коллекцию единорогов документ:



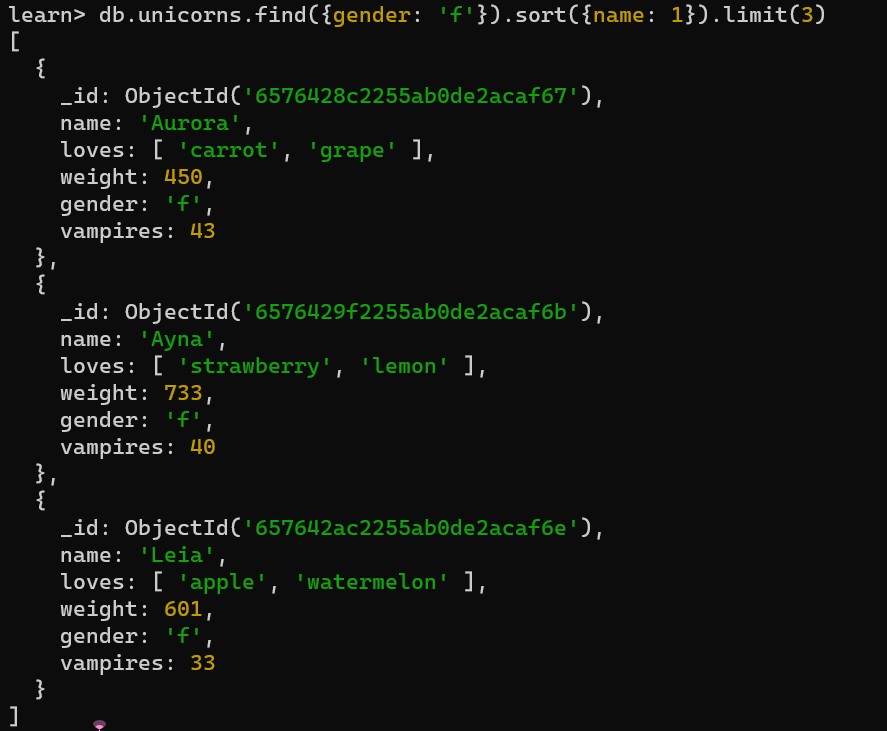
1. Проверьте содержимое коллекции с помощью метода find.



# Практическое задание 2.2.1:

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

Список самцов:

Список самок:

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

findOne:



Limit:



# Практическое задание 2.2.2:

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

# Практическое задание 2.2.3:

Вывести список единорогов в обратном порядке добавления.



11

# Практическое задание 2.1.4

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



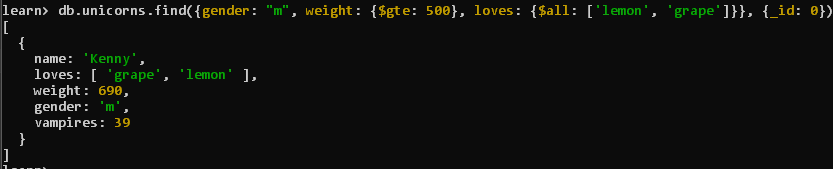
# Практическое задание 2.3.1

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

**

# Практическое задание 2.3.2

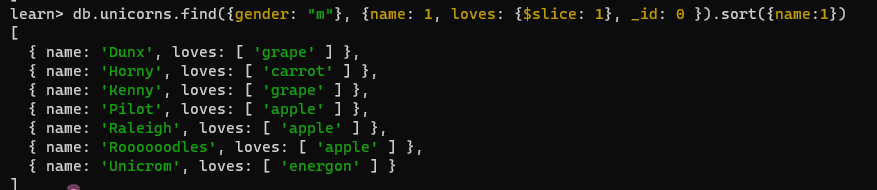
Вывести список самцов единорогов весом от полутонны и предпочитающих grape и lemon, исключив вывод идентификатора.



# Практическое задание 2.3.3



**Практическое задание 2.3.4**

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

# Практическое задание 3.1.1

1. Создайте коллекцию towns, включающую следующие документы

{name: "Punxsutawney ", populatiuon: 6200,

last\_sensus: ISODate("2008-01-31"), famous\_for: [""],

mayor: {

name: "Jim Wehrle"

}}

{name: "New York", populatiuon: 22200000,

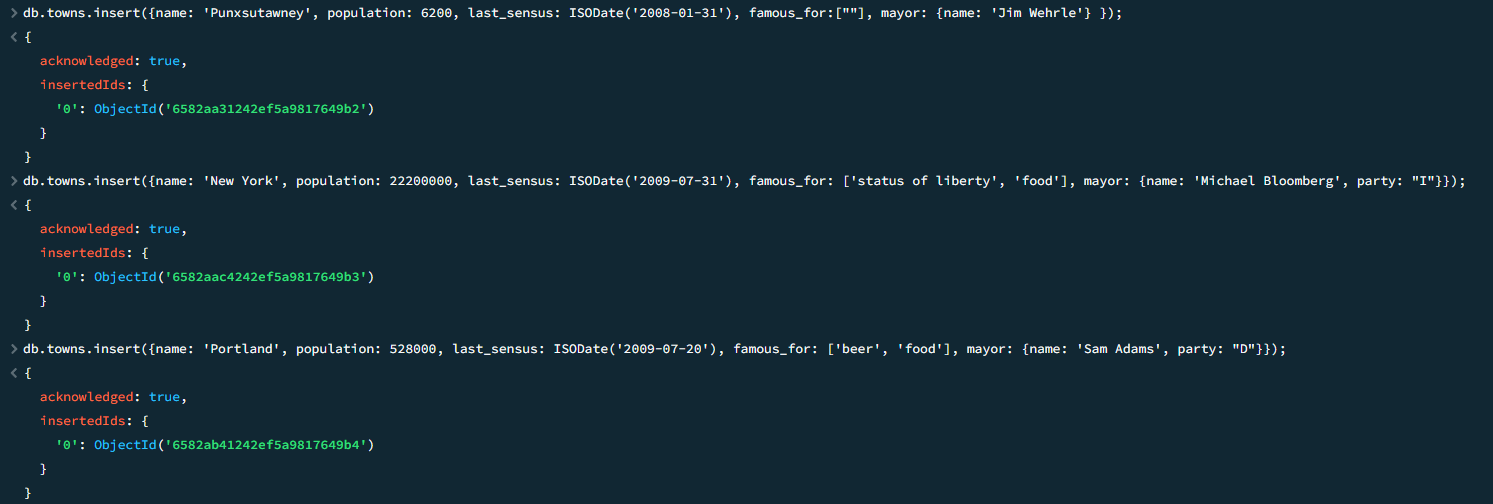
last\_sensus: ISODate("2009-07-31"), famous\_for: ["status of liberty", "food"], mayor: {

name: "Michael Bloomberg", party: "I"}},

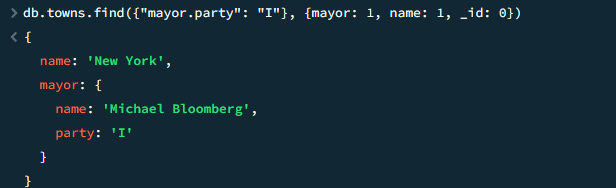
{name: "Portland", populatiuon: 528000,

last\_sensus: ISODate("2009-07-20"), famous\_for: ["beer", "food"], mayor: {

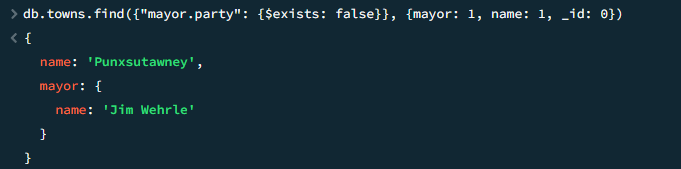
name: "Sam Adams", party: "D"}}



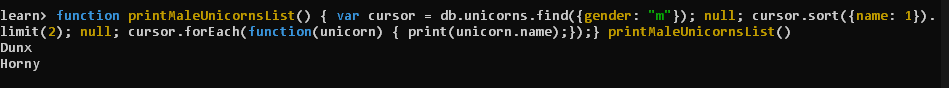
1. Сформировать запрос, который возвращает список городов с независимыми мэрами (party="I"). Вывести только название города и информацию о мэре.



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



# Практическое задание 3.1.2

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

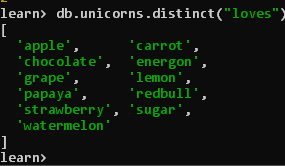
# Практическое задание 3.2.1

Вывести количество самок единорогов весом от полутонны до 600 кг.



# Практическое задание 3.2.2

Вывести список предпочтений.



# Практическое задание 3.2.3

Посчитать количество особей единорогов обоих полов

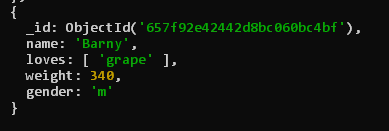


# Практическое задание 3.3.1

1. Выполнить команду:

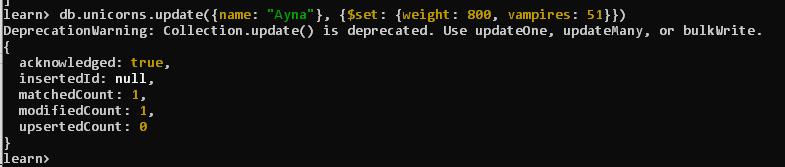
> db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})

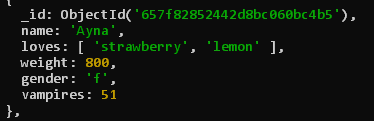
1. Проверить содержимое коллекции unicorns.



# Практическое задание 3.3.2

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

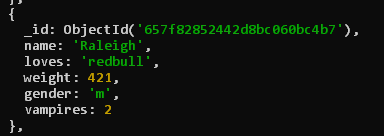




# Практическое задание 3.3.3

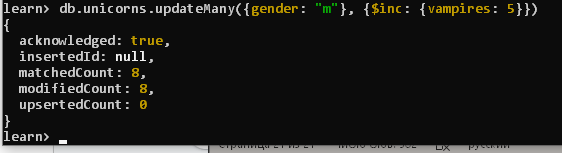
Для самца единорога Raleigh внести изменения в БД: теперь он любит рэдбул.





# Практическое задание 3.3.4

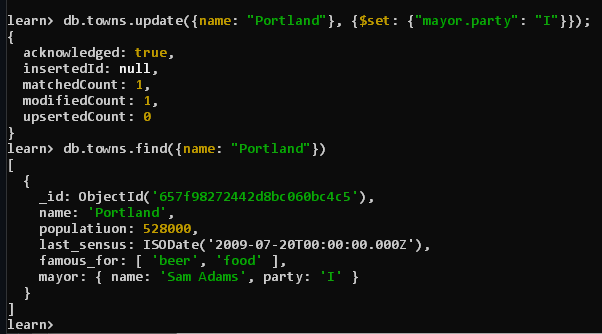
Всем самцам единорогов увеличить количество убитых вапмиров на 5.





# Практическое задание 3.3.5

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



# Практическое задание 3.3.6

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



# Практическое задание 3.3.7

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



# Практическое задание 3.4.1

* 1. *Создайте коллекцию towns, включающую следующие документы:*

{name: "Punxsutawney ", popujatiuon: 6200,

last\_sensus: ISODate("2008-01-31"), famous\_for: ["phil the groundhog"], mayor: {

name: "Jim Wehrle"

}}

{name: "New York", popujatiuon: 22200000,

last\_sensus: ISODate("2009-07-31"), famous\_for: ["status of liberty", "food"], mayor: {

name: "Michael Bloomberg", party: "I"}}

{name: "Portland", popujatiuon: 528000,

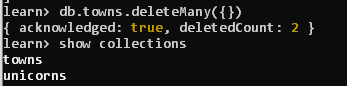
last\_sensus: ISODate("2009-07-20"), famous\_for: ["beer", "food"], mayor: {

name: "Sam Adams", party: "D"}}

* 1. *Удалите документы с беспартийными мэрами.*

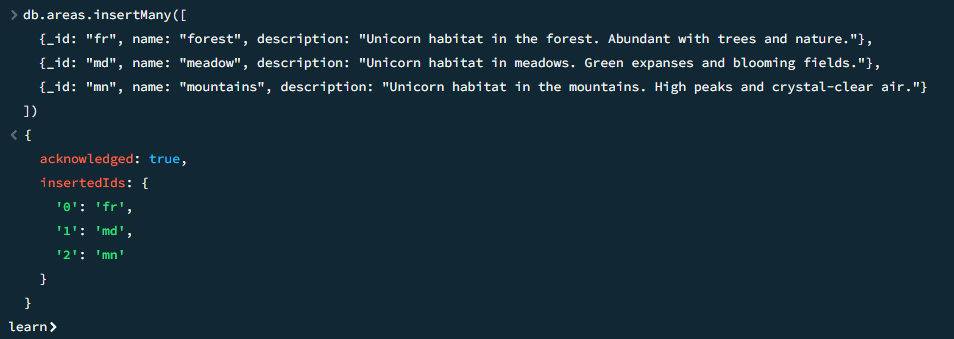
1. *Проверьте содержание коллекции.*
2. *Очистите коллекцию.*
3. *Просмотрите список доступных коллекций.*

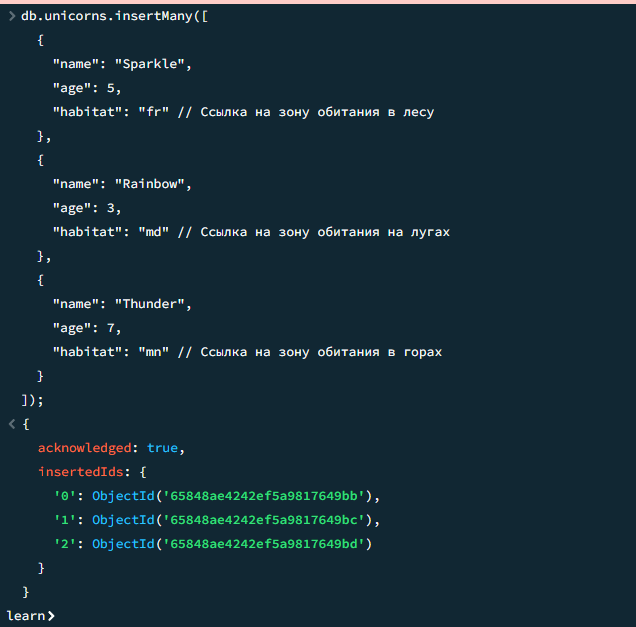




# Практическое задание 4.1.1

1. Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.
2. Включите для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания.
3. Проверьте содержание коллекции едиорогов.
4. Содержание коллекции единорогов unicorns:





# 

# Практическое задание 4.2.1

1. Проверьте, можно ли задать для коллекции unicorns индекс для ключа name с флагом unique.
2. Содержание коллекции единорогов unicorns:

db.unicorns.insert({name: 'Horny', dob: new Date(1992,2,13,7,47), loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', dob: new Date(1991, 0, 24, 13, 0), loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', dob: new Date(1973, 1, 9, 22, 10), loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', dob: new Date(1979, 7, 18, 18, 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', dob: new Date(1985, 6, 4, 2, 1), loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', dob: new Date(1998, 2, 7, 8, 30), loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', dob: new Date(1997, 6, 1, 10, 42), loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

db.unicorns.insert({name: 'Raleigh', dob: new Date(2005, 4, 3, 0, 57), loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', dob: new Date(2001, 9, 8, 14, 53), loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', dob: new Date(1997, 2, 1, 5, 3), loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

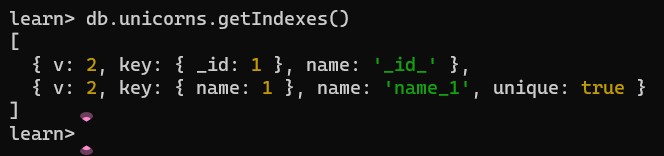
db.unicorns.insert ({name: 'Nimue', dob: new Date(1999, 11, 20, 16, 15), loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

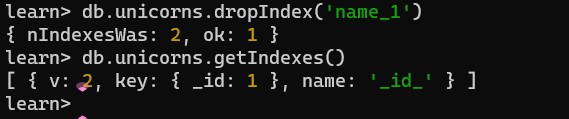
db.unicorns.insert ({name: 'Dunx', dob: new Date(1976, 6, 18, 18, 18), loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165});



# Практическое задание 4.3.1

* 1. *Получите информацию о всех индексах коллекции unicorns .*



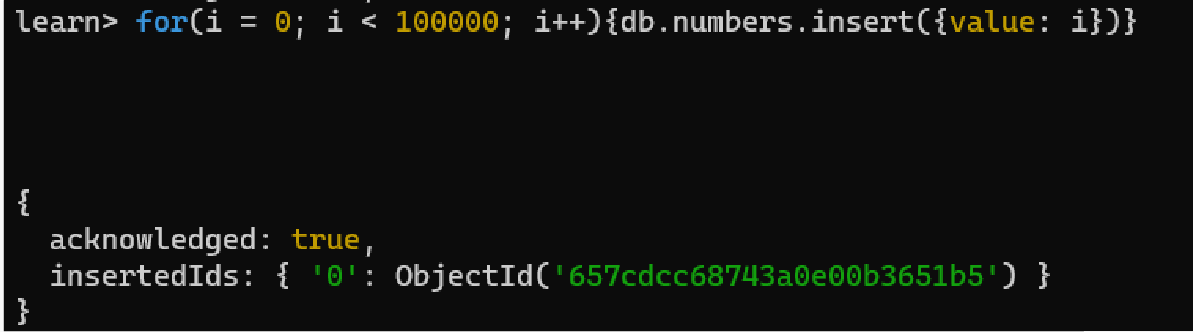
* 1. *Удалите все индексы, кроме индекса для идентификатора.*
  2. *Попытайтесь удалить индекс для идентификатора.*

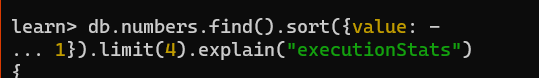
*Изображение выглядит как текст, Шрифт, снимок экрана, Графика  Автоматически созданное описание*

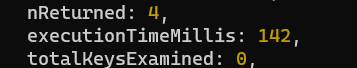
**Практическое задание 4.4.1**

1. *Создайте объемную коллекцию numbers, задействовав курсор:*

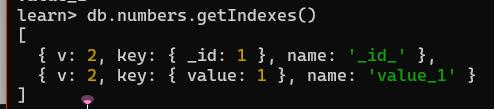
for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}

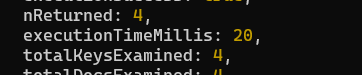


1. *Выберите последних четыре документа.*
2. *Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)*

**

1. Изображение выглядит как текст, Шрифт, снимок экрана  Автоматически созданное описание*Создайте индекс для ключа value.*
2. *Получите информацию о всех индексах коллекции numbres.*

**

1. *Выполните запрос 2.*
2. *Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?*
3. *Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?*

Время выполнения без индекса: 91 мс Время выполнения с индексом: 4 мс

Время выполнения с индексом превосходит в 22.5 раза.

ДОП ЗАДАНИЕ:  
  
  
**1) В чем отличие DBref от FK?**   
DBRef: Использует специальный тип данных DBRef, который представляет собой объект с полями $ref (имя коллекции), $id (идентификатор документа), и $db (необязательное поле с именем базы данных).

FK: Использует обычное поле, содержащее идентификатор связанного документа.  
  
**2) Какими способами можно добавить данные (одну запись или несколько) в массив. Чем они отличаются? Приведите два способа.**   
2 способа: Оператор $push и Оператор $addToSet.

Отличие: Если вам нужно добавить значение в массив вне зависимости от того, есть ли оно уже в массиве или нет, используйте $push. Если вам нужно удостовериться, что значение уникально в массиве, используйте $addToSet.

Пример:

db.students.update(

{ "\_id": ObjectId("идентификатор\_документа") },

{ $push: { "grades": 95 } }

);  
  
Пример:  
db.students.update(

{ "\_id": ObjectId("идентификатор\_документа") },

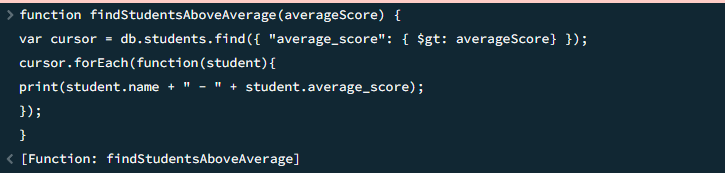
{ $addToSet: { "courses": "Math" } }

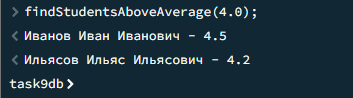
);

**3) Создайте новую базу данных MongoDB с именем "task9db". В этой базе данных создайте коллекцию "students". Добавьте информацию про нескольких студентов: ФИО и средний балл.**



**4) Напишите JavaScript-функцию, которая выполнит выборку всех студентов, у которых средний балл выше определенного значения (например, 4.0). Значение должно передаваться как параметр. Сохраните эту функцию и продемонстрируйте её работоспособность.**





# 

# Вывод

В процессе выполнения данной практической работы были освоены конкретные методы взаимодействия с базой данных в MongoDB, такие как создание, чтение, обновление и удаление данных1. Были изучены фундаментальные команды, позволяющие управлять информацией в коллекциях. В рамках лаборатории были также ознакомлены с использованием вложенных объектов в структуре коллекций, проведением агрегаций данных и методами изменения данных с применением ссылок и индексов. Суммируя опыт работы, можно заключить, что данная лабораторная работа дала уверенность в применении ключевых навыков для взаимодействия с базой данных в MongoDB, а также обогатила практическим опытом выполнения разнообразных операций с данными.