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

### «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

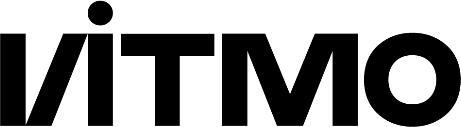
### Отчет

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

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

Автор: Кадникова Е.М. 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.

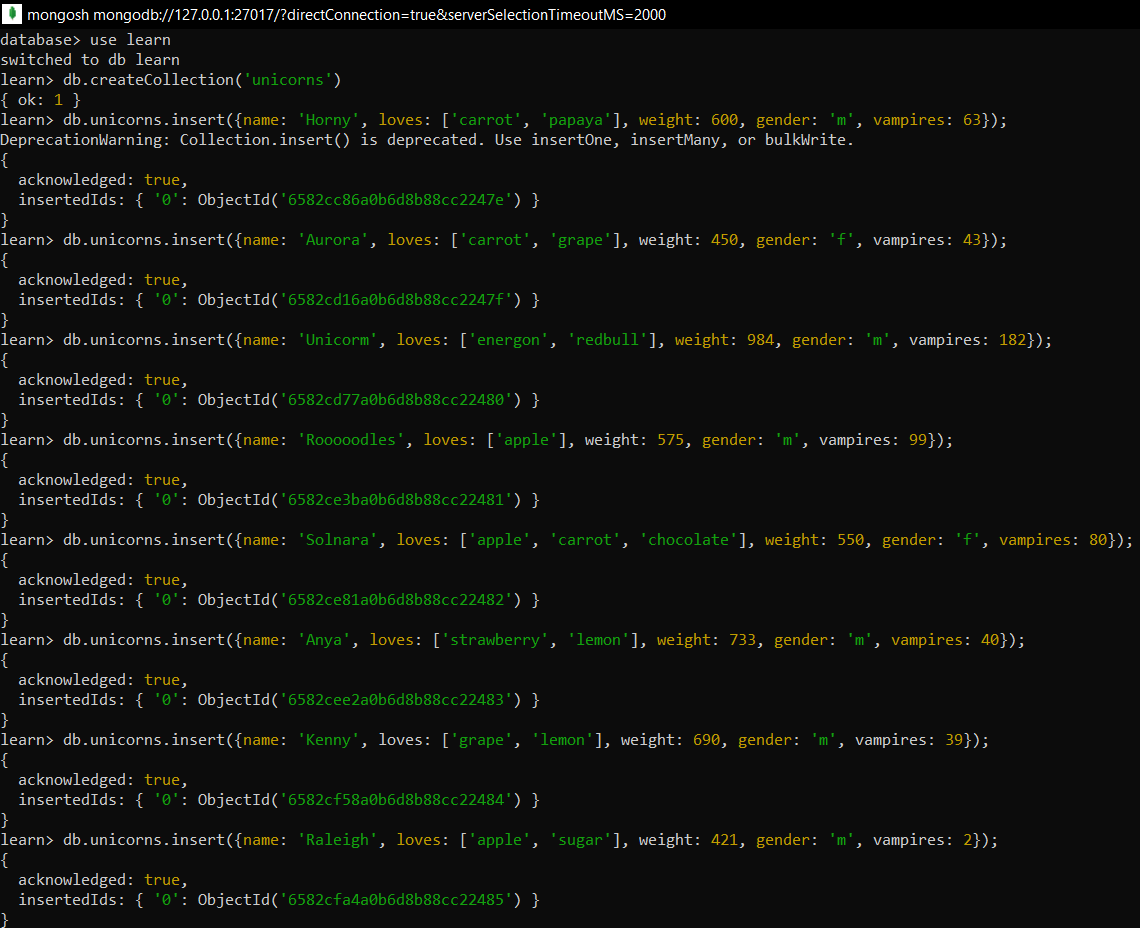
**Оборудование**: компьютерный класс.

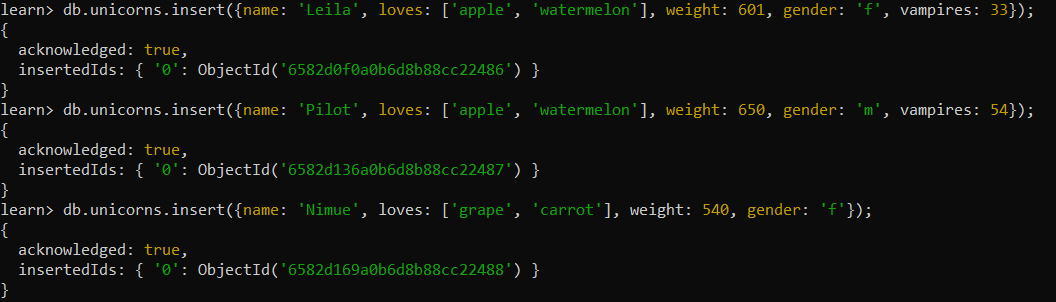
**Программное обеспечение**: СУБД MongoDB 4+, 7.0.1 (текущая), MongoShell.

**Выполнение**

**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

**

**Задание 2.2.1:**

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

Листинг запроса для получения всех самцов:

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

**

Листинг запроса для получения всех самок:

*db.unicorns.find({gender: 'f'}).sort({name: 1});*

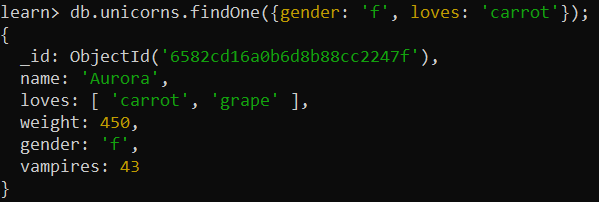
**

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

Листинг запроса с использованием limit: *db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1);*

**

Листинг запроса через findOne: *db.unicorns.findOne({gender: 'f', loves: 'carrot'});*



**Задание 2.2.2:**

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

Листинг запроса: *db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0}).sort({name: 1});*



**Задание 2.2.3:**

Вывести список единорогов в обратном порядке добавления.

Листинг запроса: *db.unicorns.find().sort({$natural: -1});*



**Задание 2.1.4:**

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

Листинг запроса: *db.unicorns.find({}, {loves: {$slice: 1}, \_id: 0});*



**Задание 2.3.1:**

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

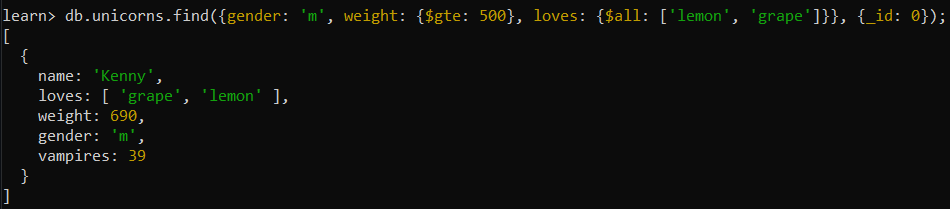
Листинг запроса: *db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 700}}, {\_id: 0});*



**Задание 2.3.2:**

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

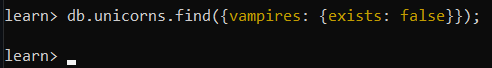
Листинг запроса: *db.unicorns.find({gender: 'm', weight: {$gte: 500}, loves: {$all: ['lemon', 'grape']}}, {\_id: 0});*



**Задание 2.3.3:**

Найти всех единорогов, не имеющих ключ vampires.

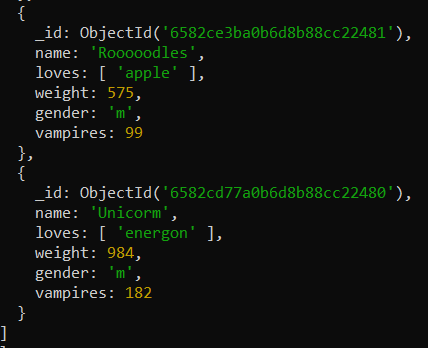
Листинг запроса: *db.unicorns.find({vampires: {exists: false}});*



**Задание 2.3.4:**

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

Листинг запроса: *db.unicorns.find({gender: 'm'}, {loves: {$slice: 1}}).sort({name: 1});*



**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,

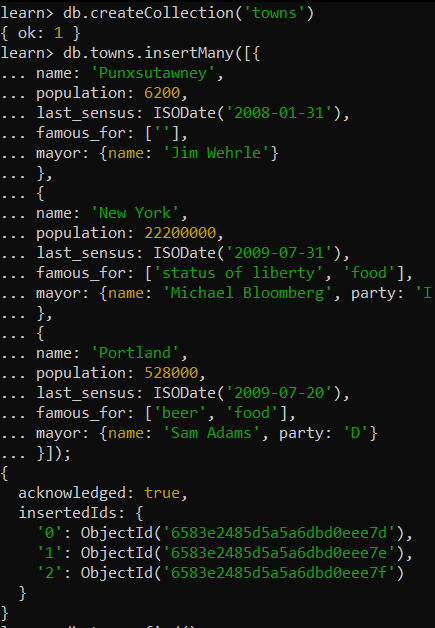
last\_sensus: ISODate("2009-07-20"),

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.

Листинг запроса: *db.towns.find({'mayor.party': 'I'}, {name: 1,'mayor.name': 1, \_id: 0})*



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

Листинг запроса:

*db.towns.find({'mayor.party': {$exists: false}}, {name: 1, 'mayor.name': 1, \_id: 0})*

****

**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.

Листинг запроса:

*function printMaleUnicorns()*

*... {*

*... var cursor = db.unicorns.find({gender: 'm'});*

*... null;*

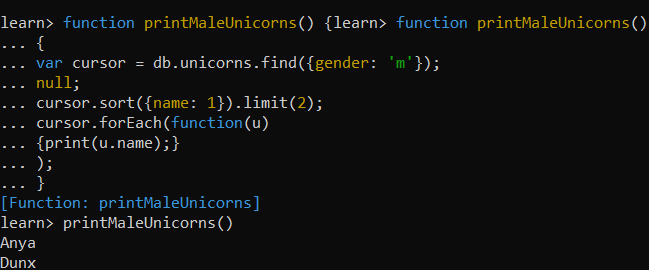
*... cursor.sort({name: 1}).limit(2);*

*... cursor.forEach(function(u)*

*... {print(u.name);}*

*... );*

*... }*

****

**Задание 3.2.1:**

Вывести количество самок единорогов весом от полутонны до 600 кг.

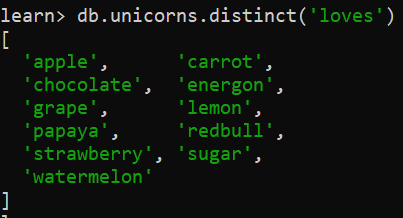
Листинг запроса: *db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 600}}).count()*

****

**Задание 3.2.2:**

Вывести список предпочтений.

Листинг запроса: *db.unicorns.distinct('loves')*

****

**Задание 3.2.3:**

Посчитать количество особей единорогов обоих полов.

Листинг запроса: *db.unicorns.aggregate({'$group': {\_id: '$gender', count: {$sum: 1}}})*



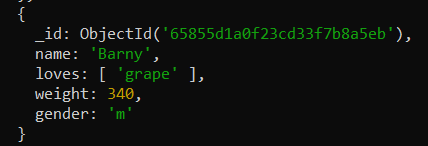
**Задание 3.3.1:**

1. Выполнить команду:

> db.unicorns.save({name: 'Barny', loves: ['grape'],

weight: 340, gender: 'm'})

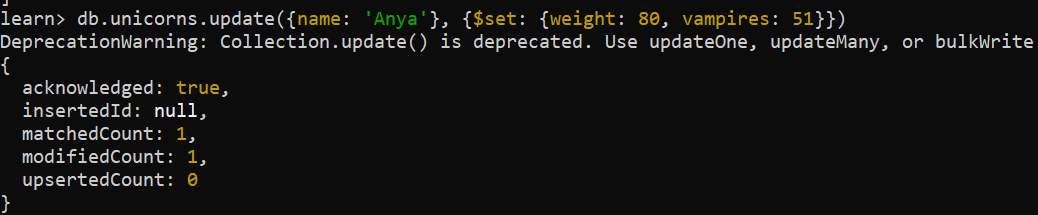
1. Проверить содержимое коллекции unicorns.



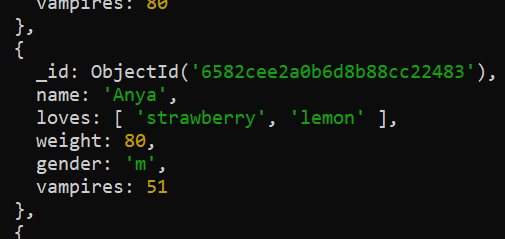
**Задание 3.3.2:**

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

Листинг запроса: *db.unicorns.update({name: 'Anya'}, {$set: {weight: 80, vampires: 51}})*



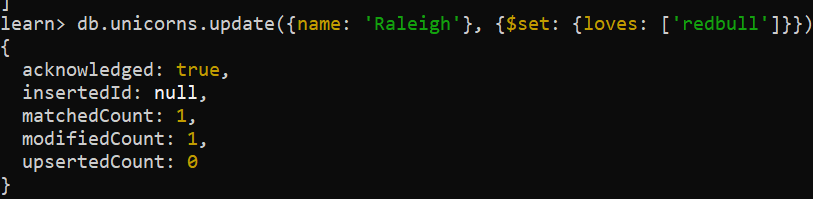
1. Проверить содержимое коллекции unicorns.



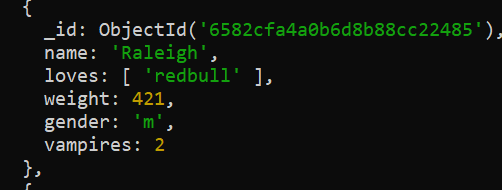
**Задание 3.3.3:**

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

Листинг запроса: *db.unicorns.update({name: 'Raleigh'}, {$set: {loves: ['redbull']}})*

**

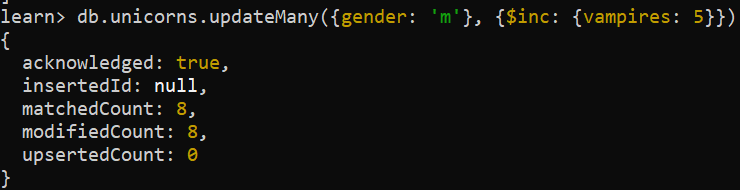
1. Проверить содержимое коллекции unicorns.



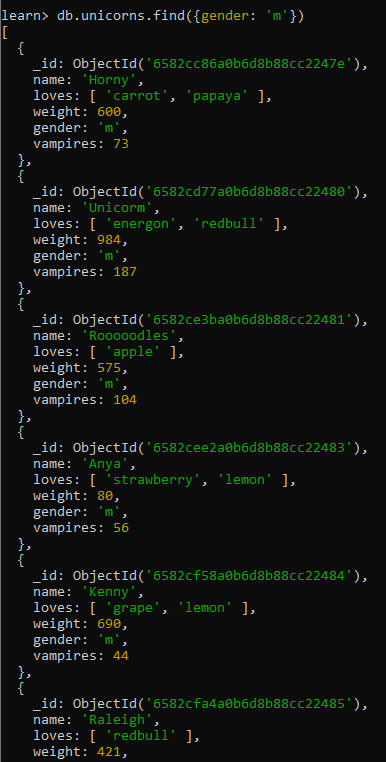
**Задание 3.3.4:**

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

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

**

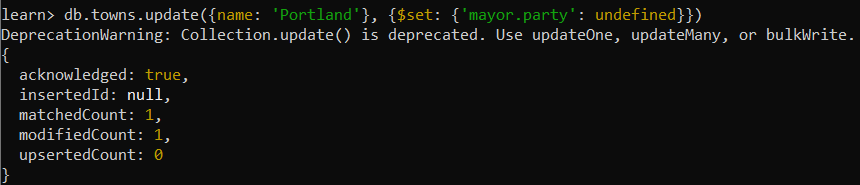
1. Проверить содержимое коллекции unicorns.



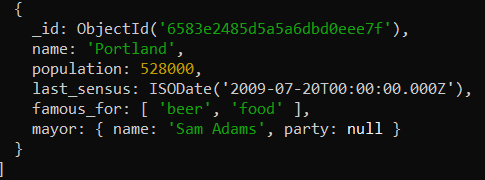
**Задание 3.3.5:**

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

Листинг запроса: *db.towns.update({name: 'Portland'}, {$set: {'mayor.party': undefined}})*



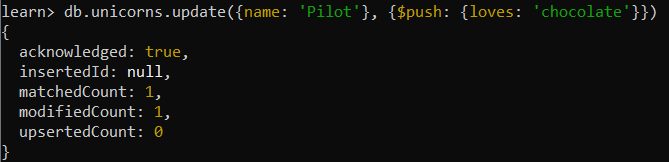
1. Проверить содержимое коллекции towns.

****

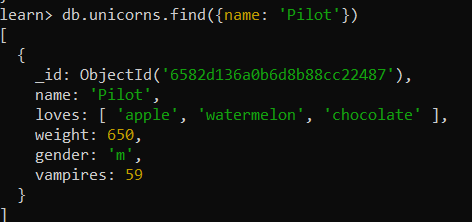
**Задание 3.3.6:**

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

Листинг запроса: *db.unicorns.update({name: 'Pilot'}, {$push: {loves: 'chocolate'}})*



1. Проверить содержимое коллекции unicorns.

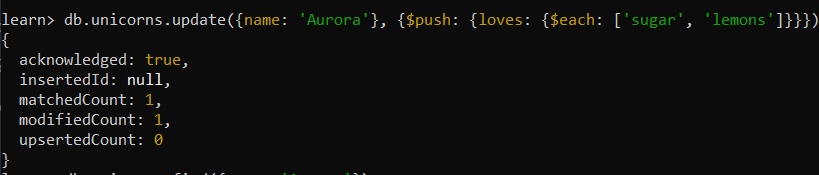
****

**Задание 3.3.7:**

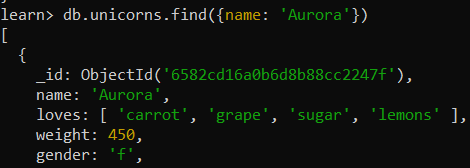
1. Изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и лимоны.

Листинг запроса:

*db.unicorns.update({name: 'Aurora'}, {$push: {loves: {$each: ['sugar', 'lemons']}}})*



1. Проверить содержимое коллекции unicorns.



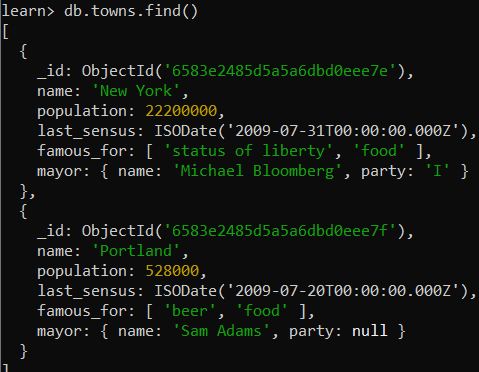
**Задание 3.4.1:**

1. Создайте коллекцию towns, включающую следующие документы:
2. Удалите документы с беспартийными мэрами.

Листинг запроса: *db.towns.deleteMany({'mayor.party': {$exists: false}})*



1. Проверьте содержание коллекции.



1. Очистите коллекцию.

Листинг запроса: *db.towns.deleteMany({})*



1. Просмотрите список доступных коллекций.



**Задание 4.1.1:**

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

Листинг запроса:

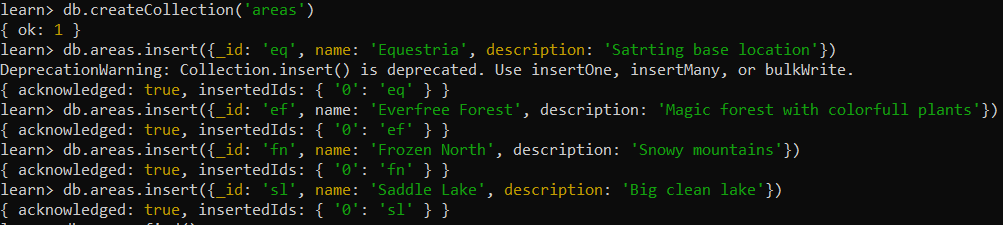
*db.createCollection('areas')*

*db.areas.insert({\_id: 'eq', name: 'Equestria', description: 'Satrting base location'})*

*db.areas.insert({\_id: 'ef', name: 'Everfree Forest', description: 'Magic forest with colorfull plants'})*

*db.areas.insert({\_id: 'fn', name: 'Frozen North', description: 'Snowy mountains'})*

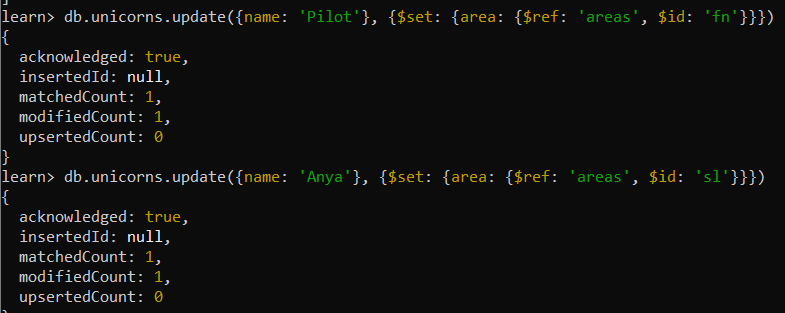
*db.areas.insert({\_id: 'sl', name: 'Saddle Lake', description: 'Big clean lake'})*



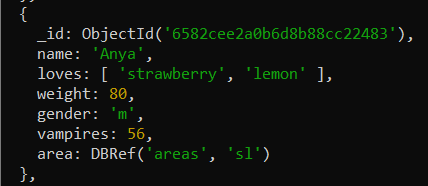
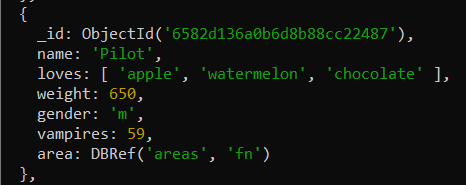
1. Включите для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания.

Листинг запроса: *db.unicorns.update({name: 'Pilot'}, {$set: {area: {$ref: 'areas', $id: 'fn'}}})*

*db.unicorns.update({name: 'Anya'}, {$set: {area: {$ref: 'areas', $id: 'sl'}}})*



1. Проверьте содержание коллекции едиорогов.



**Задание 4.2.1:**

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

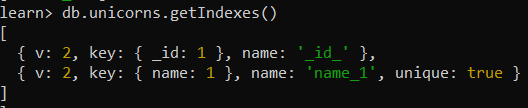
Листинг запроса: *db.unicorns.ensureIndex({'name': 1}, {'unique': true})*



**Задание 4.3.1:**

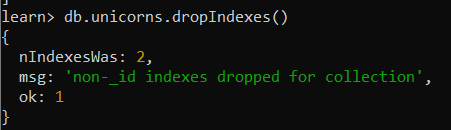
1. Получите информацию о всех индексах коллекции unicorns .

Листинг запроса: *db.unicorns.getIndexes()*



1. Удалите все индексы, кроме индекса для идентификатора.

Листинг запроса: *db.unicorns.dropIndexes()*



1. Попытайтесь удалить индекс для идентификатора.

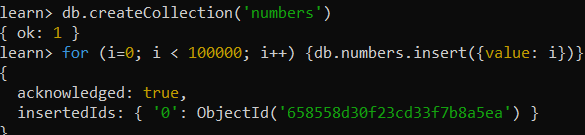
Листинг запроса: *db.unicorns.dropIndex('\_id\_')*



**Задание 4.4.1:**

1. Создайте объемную коллекцию numbers, задействовав курсор:

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

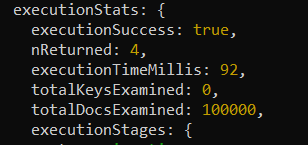


1. Выберите последних четыре документа.\

Листинг запроса: *db.numbers.find().sort({value: -1}).limit(4).explain('executionStats')*



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



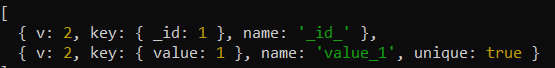
1. Создайте индекс для ключа value.

Листинг запроса: *db.numbers.ensureIndex({'value': 1}, {'unique': true})*

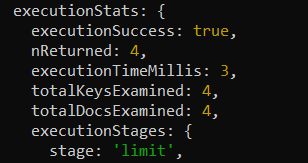


1. Получите информацию о всех индексах коллекции numbres.

Листинг запроса: *db.numbers.getIndexes()*



1. Выполните запрос 2.



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

В первом случае (без использования индексов) для выполнения запроса потребовалось 92 миллисекунды. Во втором случае (с использованием индексов) потребовалось 3 миллисекунды.

**Вывод:** в ходе лабораторной работы овладели 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.