**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе №6 «Процедуры, функции, триггеры в PostgreSQL**»**

по дисциплине **«Работа с БД в СУБД MongoDB»**

Автор: Бунос М.В.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

**Оглавление**

[Вывод 19](#_Toc154184643)

**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).

Создайте базу данных learn.



*Заполните коллекцию единорогов unicorns:*

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

db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690,  gender: 'm', vampires: 39});

db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

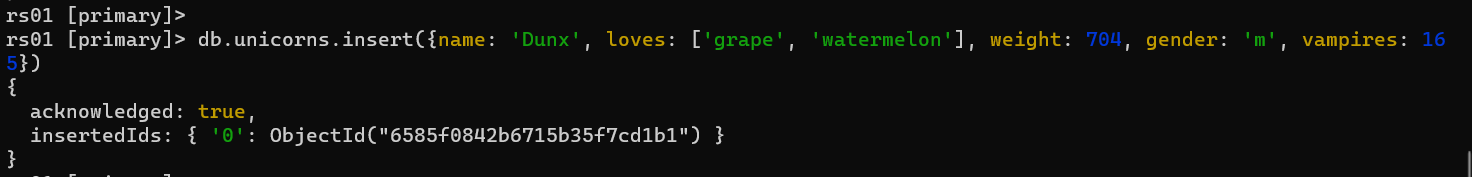
db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

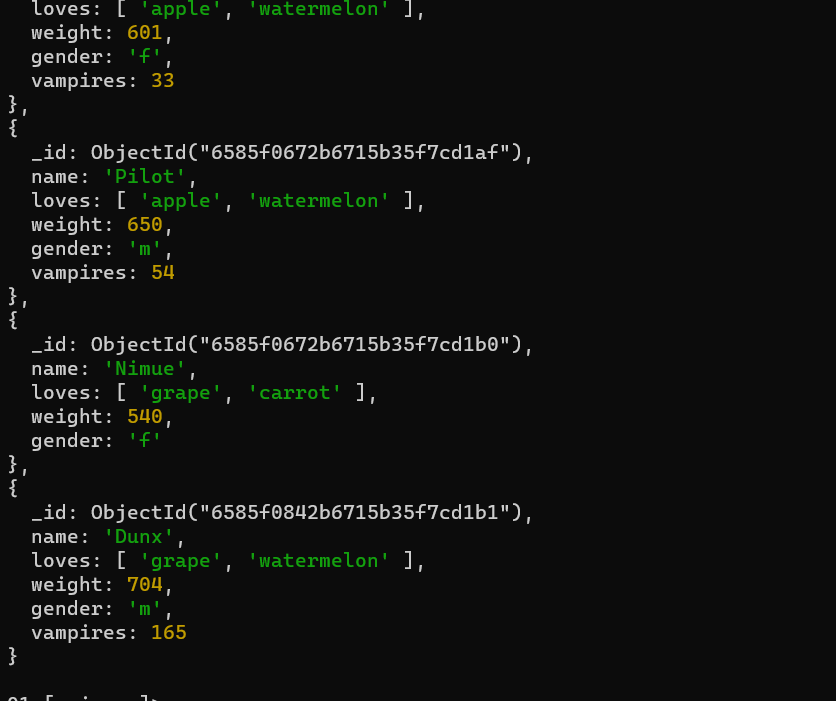
db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

*Используя второй способ, вставьте в коллекцию единорогов документ:*

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



*Проверьте содержимое коллекции с помощью метода find.*



2.2 ВЫБОРКА ДАННЫХ ИЗ БД

**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.*





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





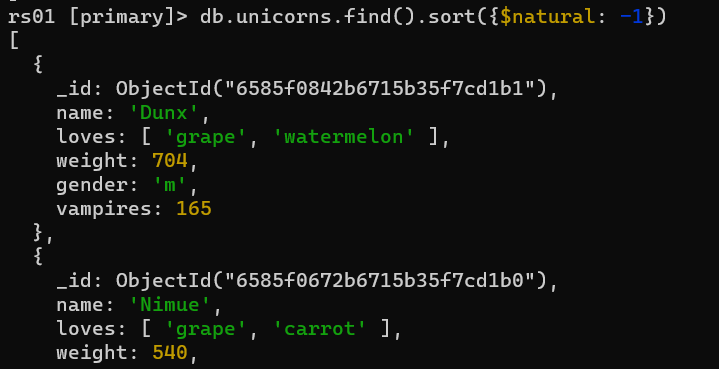
**Практическое задание 2.2.2:**

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



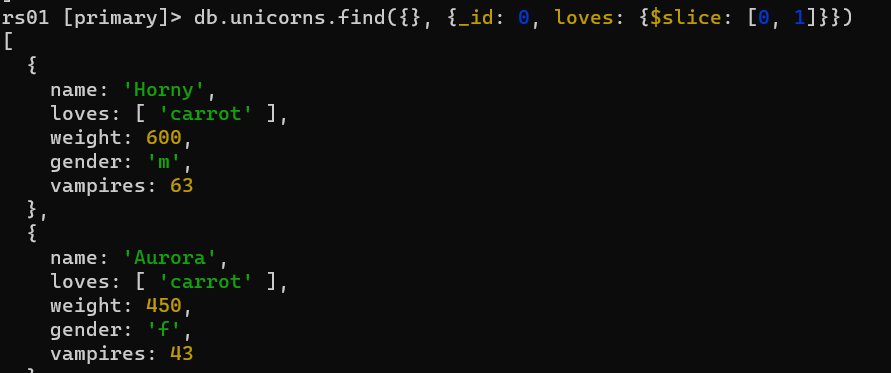
**Практическое задание 2.2.3:**

*Вывести список единорогов в обратном порядке добавления.*



**Практическое задание 2.1.4:**

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



2.3 ЛОГИЧЕСКИЕ ОПЕРАТОРЫ

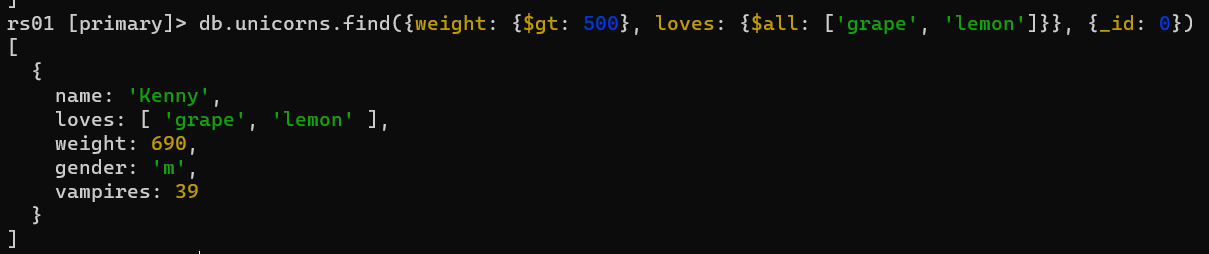
**Практическое задание 2.3.1:**

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



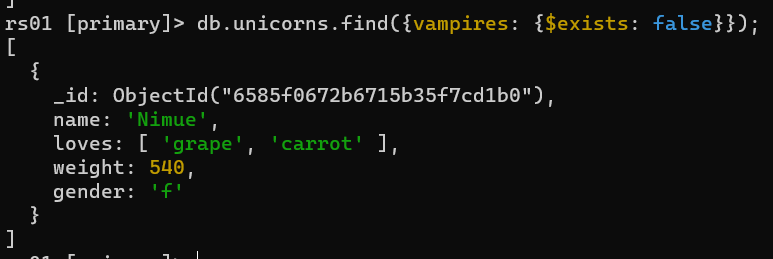
**Практическое задание 2.3.2:**

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



**Практическое задание 2.3.3:**

*Найти всех единорогов, не имеющих ключ vampires.*



**Практическое задание 2.3.4:**

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



3 ЗАПРОСЫ К БАЗЕ ДАННЫХ MONGODB.

ВЫБОРКА ДАННЫХ. ВЛОЖЕННЫЕ ОБЪЕКТЫ. ИСПОЛЬЗОВАНИЕ КУРСОРОВ. АГРЕГИРОВАННЫЕ ЗАПРОСЫ. ИЗМЕНЕНИЕ ДАННЫХ

3.1 ЗАПРОС К ВЛОЖЕННЫМ ОБЪЕКТАМ

**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"),

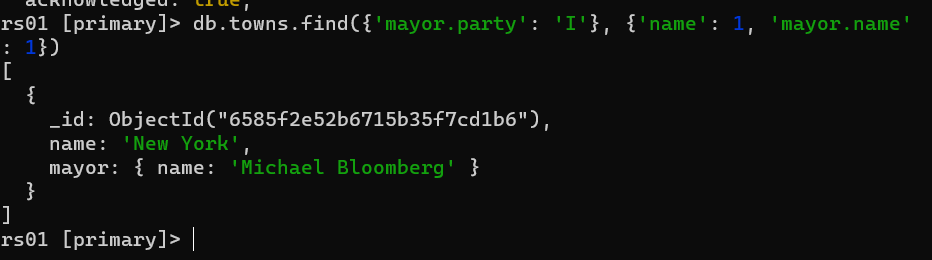
famous\_for: ["beer", "food"],

mayor: {

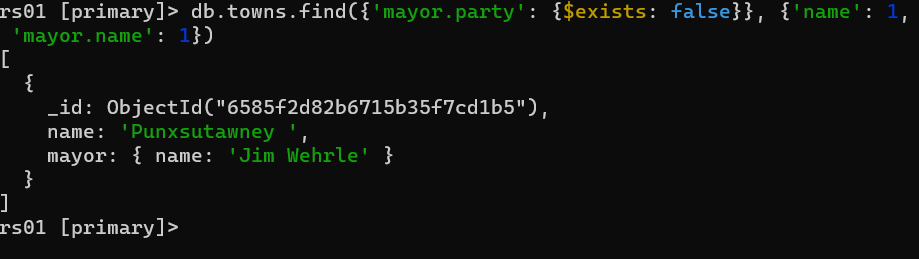
   name: "Sam Adams",

party: "D"}}

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

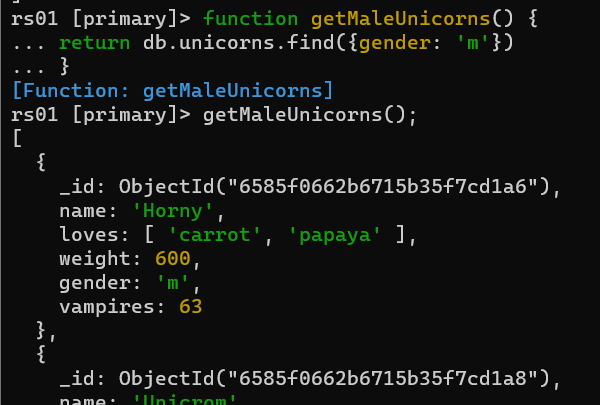


1. *Сформировать запрос, который возвращает список беспартийных мэров (*party *отсутствует*)*. Вывести только название города и информацию о мэре.*



**Практическое задание 3.1.2:**

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



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

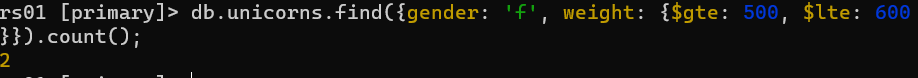




1. *Вывести результат, используя forEach.*

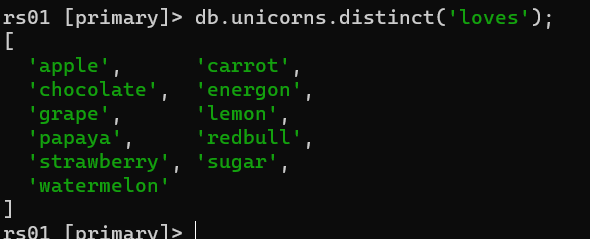
**Практическое задание 3.2.1:**

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



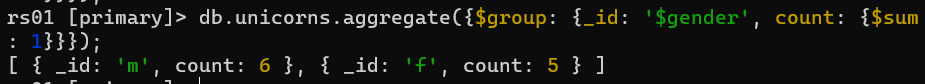
**Практическое задание 3.2.2:**

*Вывести список предпочтений.*



**Практическое задание 3.2.3:**

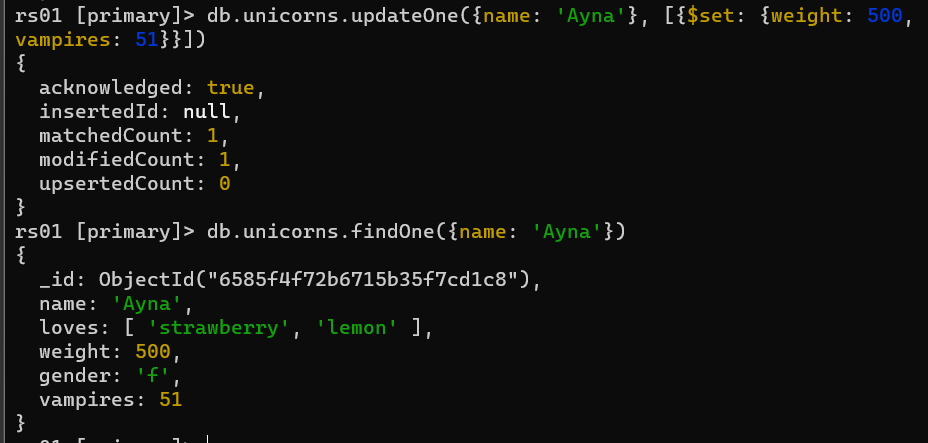
*Посчитать количество особей единорогов обоих полов.*



|  |  |
| --- | --- |
|  | **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:**

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

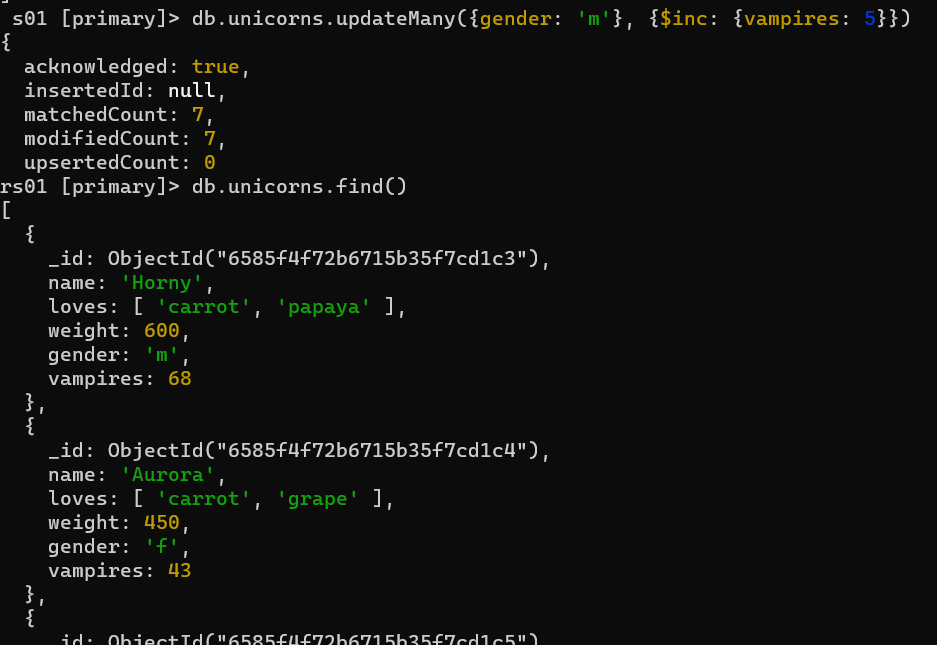


1. *Проверить содержимое коллекции unicorns*

|  |  |
| --- | --- |
|  | **Практическое задание 3.3.3:**   1. *Для самца единорога* Raleigh *внести изменения в БД: теперь он любит рэдбул.* 2. *Проверить содержимое коллекции unicorns.* |

**Практическое задание 3.3.4:**

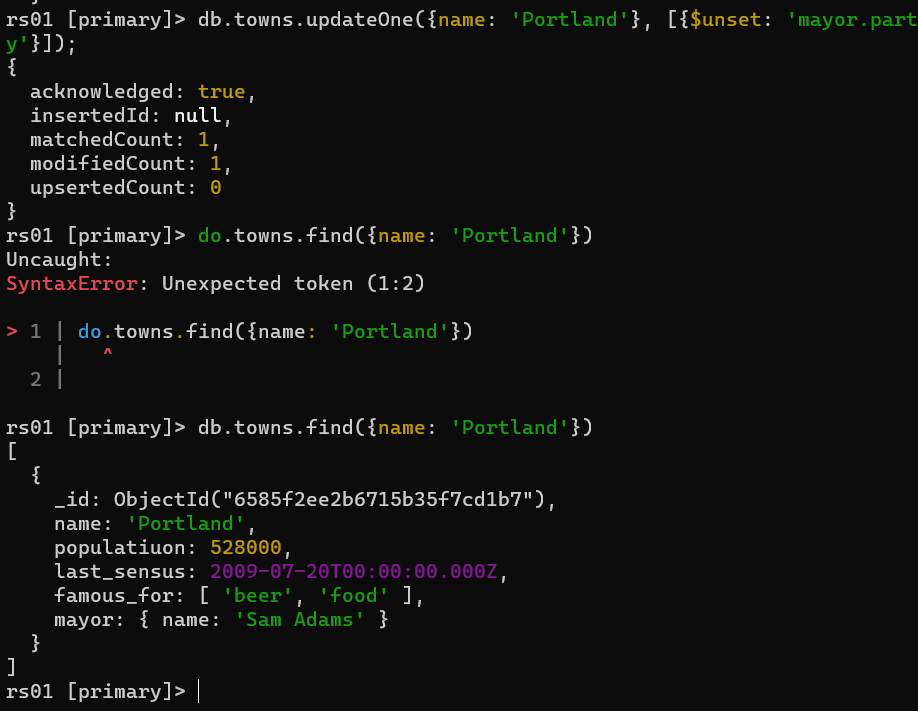
1. *Всем самцам единорогов увеличить количество убитых вапмиров на 5.*



1. *Проверить содержимое коллекции unicorns.*

**Практическое задание 3.3.5:**

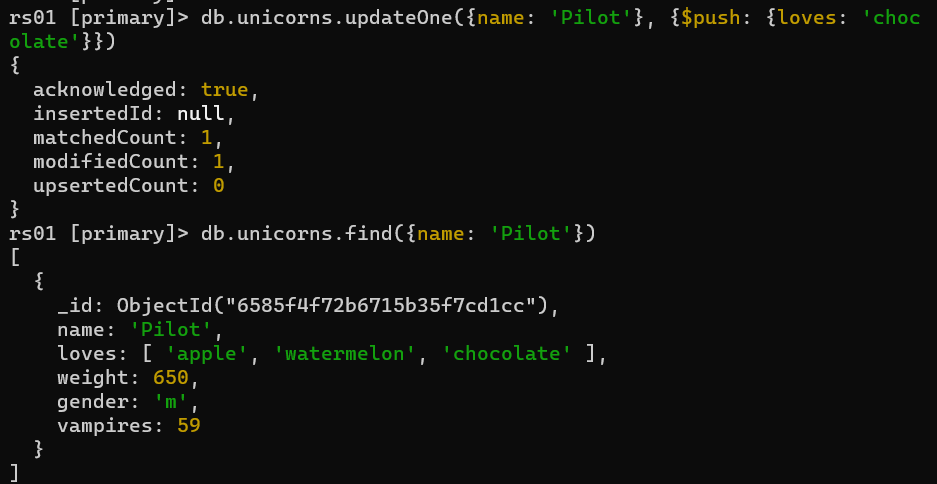
1. *Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.*



1. *Проверить содержимое коллекции towns.*

**Практическое задание 3.3.6:**

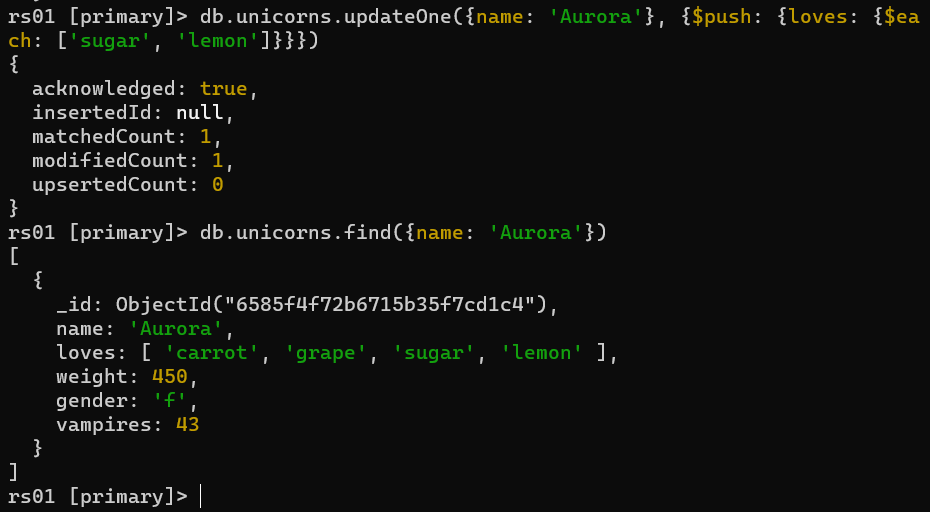
1. *Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.*



1. *Проверить содержимое коллекции unicorns.*

**Практическое задание 3.3.7:**

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



1. *Проверить содержимое коллекции unicorns.*

**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"),

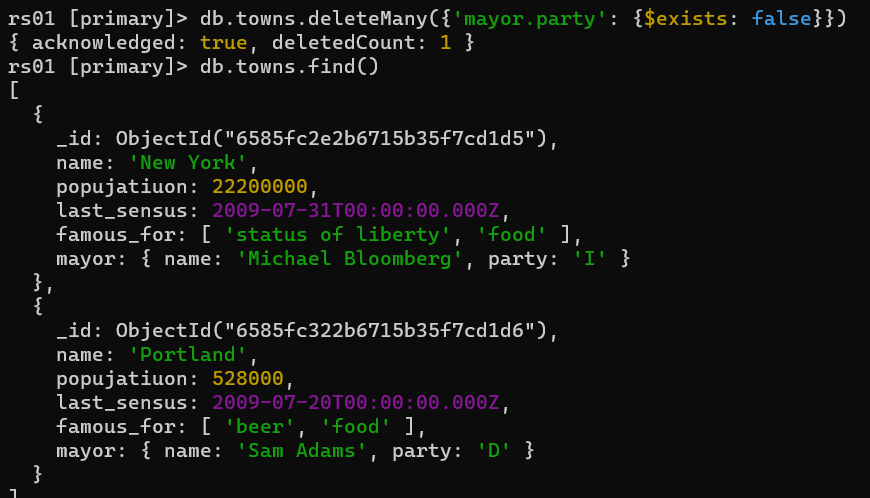
famous\_for: ["beer", "food"],

mayor: {

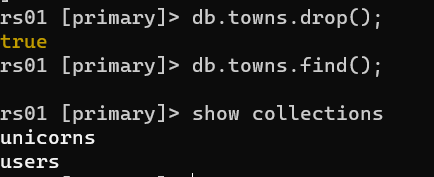
   name: "Sam Adams",

party: "D"}}

1. *Удалите документы с беспартийными мэрами.*



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

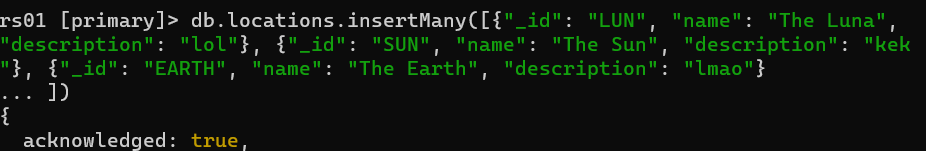


4 ССЫЛКИ И РАБОТА С ИНДЕКСАМИ В БАЗЕ ДАННЫХ MONGODB

4.1 ССЫЛКИ В БД

**Практическое задание 4.1.1:**

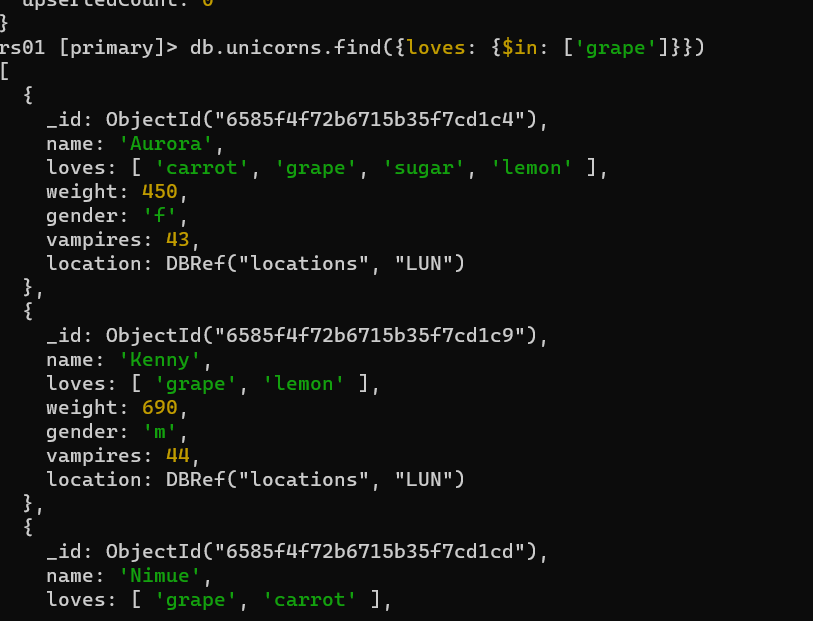
1. *Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.*



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



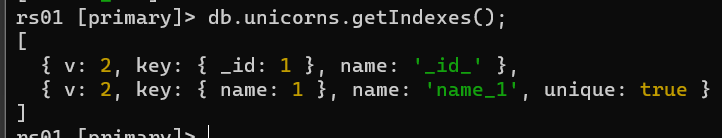
1. *Проверьте содержание коллекции едиорогов.*



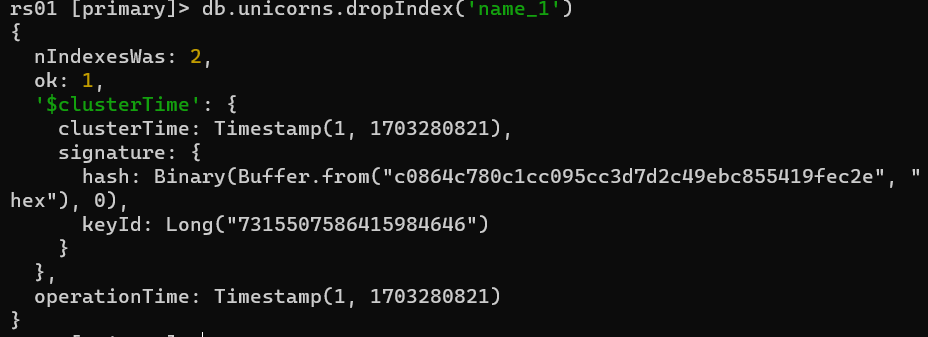
4.3 УПРАВЛЕНИЕ ИНДЕКСАМИ

**Практическое задание 4.3.1:**

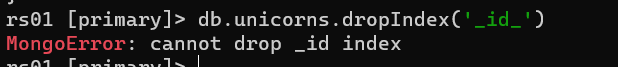
1. *Получите информацию о всех индексах коллекции unicorns .*



1. *Удалите все индексы, кроме индекса для идентификатора.*



1. *Попытайтесь удалить индекс для идентификатора.*



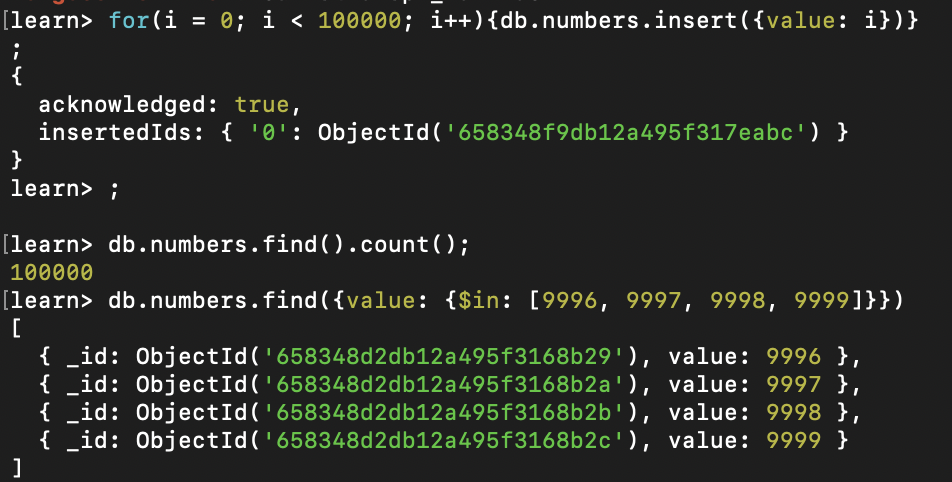
4.4 ПЛАН ЗАПРОСА

**Практическое задание 4.4.1:**

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

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

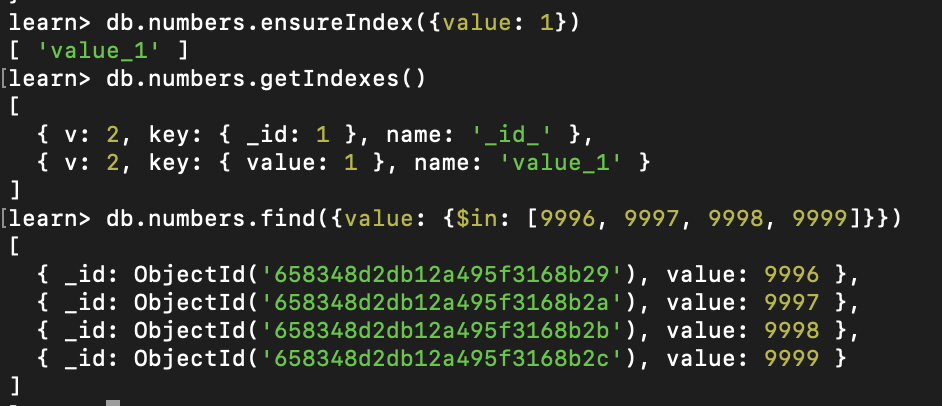
1. *Выберите последних четыре документа*

*.*

1. *Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)*
2. *Создайте индекс для ключа value.*

**

1. *Получите информацию о всех индексах коллекции numbres.*
2. *Выполните запрос 2.*



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

**

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

*Ускорение с индексом очень сильно заметно.*

# Вывод

В ходе лабораторной работы была изучена работа с NoSQL БД MongoDB.