Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе 5 «Работа с БД в СУБД MongoDB»

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

Автор: Никифорова Анна Дмитриевна

Факультет: Инфокоммуникационные технологии

Группа: K32421

Преподаватель: Говорова Марина Михайловна

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:

Для работы с MongoDB используется программа MongoDB Compass, которая предоставляет как удобный графический пользовательский интерфейс, так и возможность работы через консоль mongosh.

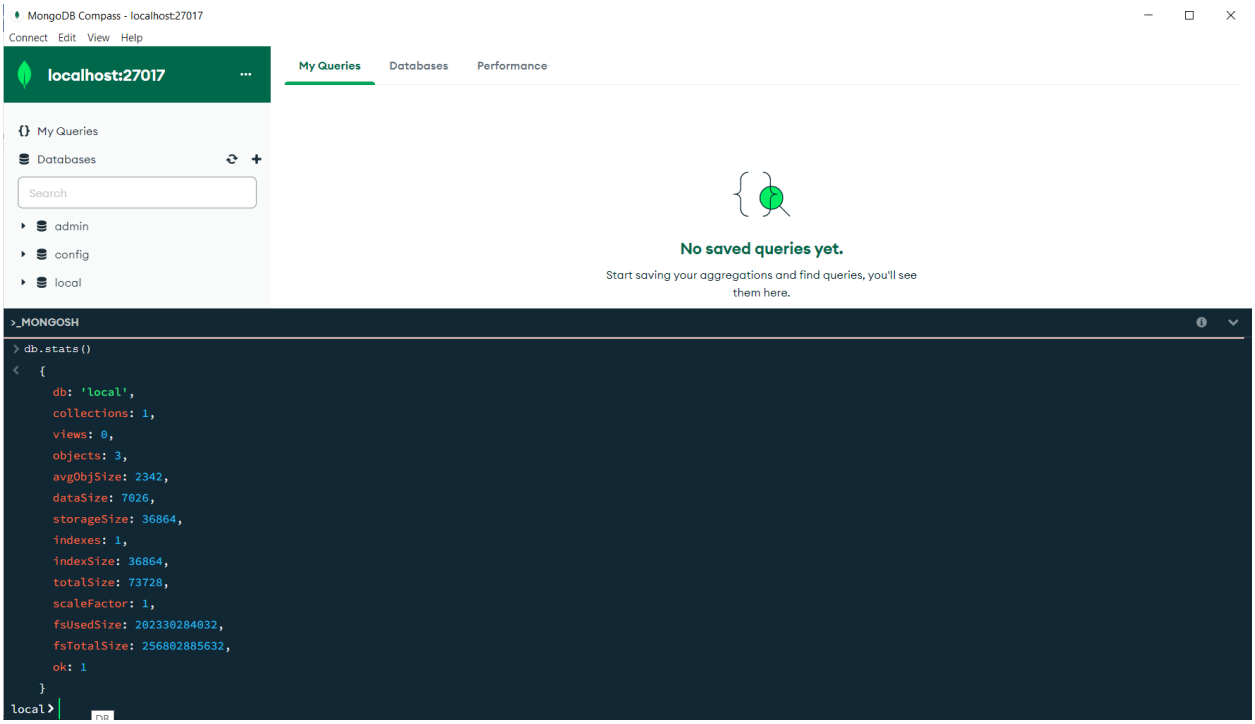

Рисунок 1 – интерфейс MongoDB Compass

Практическое задание 8.1.1:

1. Создайте базу данных learn.

2. 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'});

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

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

4. Проверьте содержимое коллекции с помощью метода find.

```
>_MONGOSH
> use learn
<   'switched to db learn'
> db.createCollection("unicorns")
<   { ok: 1 }
> db.unicorns.insertMany([{name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', var
<   {
      acknowledged: true,
      insertedIds: {
        '0': ObjectId("6464b9087a13e277e7f54efc"),
        '1': ObjectId("6464b9087a13e277e7f54efd"),
        '2': ObjectId("6464b9087a13e277e7f54efe"),
        '3': ObjectId("6464b9087a13e277e7f54eff"),
        '4': ObjectId("6464b9087a13e277e7f54f00"),
        '5': ObjectId("6464b9087a13e277e7f54f01"),
        '6': ObjectId("6464b9087a13e277e7f54f02"),
        '7': ObjectId("6464b9087a13e277e7f54f03"),
        '8': ObjectId("6464b9087a13e277e7f54f04"),
        '9': ObjectId("6464b9087a13e277e7f54f05"),
        '10': ObjectId("6464b9087a13e277e7f54f06")
      }
    }
```

Рисунок 2 – Создание коллекции

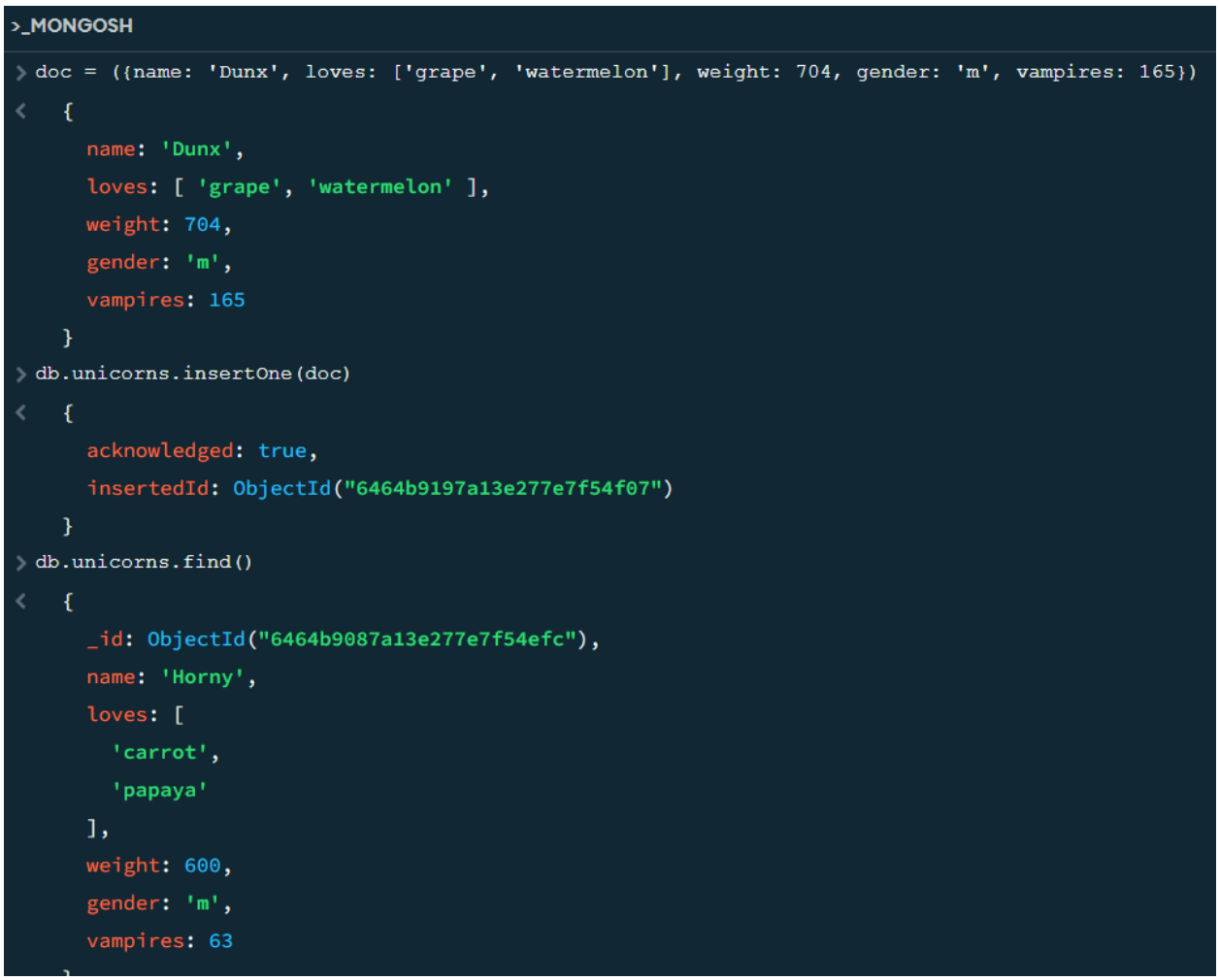

Рисунок 3 – Второй метод вставки

Микровывод: синтаксис db.insert() устаревший, нужно использовать insertOne() либо insertMany().

Практическое задание 8.1.2:

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

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

```
db.unicorns.find({gender: 'm'}, {_id: 0, name: 1}).sort({name: 1}).toArray()
  [
    { name: 'Dunx' },
    { name: 'Horny' },
    { name: 'Kenny' },
    { name: 'Pilot' },
    { name: 'Raleigh' },
    { name: 'Roooooodles' },
    { name: 'Unicrom' }
  ]
db.unicorns.find({gender: 'f'}, {_id: 0, name: 1}).sort({name: 1}).limit(3).toArray()
  [ { name: 'Aurora' }, { name: 'Ayna' }, { name: 'Leia' } ]
```

Рисунок 4 – Задание 1

```
db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
  {
    _id: ObjectId("6464b9087a13e277e7f54efd"),
    name: 'Aurora',
    loves: [
      'carrot',
      'grape'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
db.unicorns.findOne({gender: 'f', loves: 'carrot'})
  {
    _id: ObjectId("6464b9087a13e277e7f54efd"),
    name: 'Aurora',
    loves: [
      'carrot',
      'grape'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
```

Рисунок 5 – Задание 2

```
db.unicorns.find({}, {_id: 0, name: 1}).limit(2).toArray()
  [ { name: 'Horny' }, { name: 'Aurora' } ]
db.unicorns.find({}, {_id: 0, name: 1}).limit(2).sort({name: 1}).toArray()
  [ { name: 'Aurora' }, { name: 'Ayna' } ]
db.unicorns.find({}, {_id: 0, name: 1}).sort({name: 1}).limit(2).toArray()
  [ { name: 'Aurora' }, { name: 'Ayna' } ]
db.unicorns.aggregate({"$limit": 2}, {"$sort": {name: 1}}, {"$project": {_id: 0, name: 1}}).toArray()
  [ { name: 'Aurora' }, { name: 'Horny' } ]
db.unicorns.aggregate({"$sort": {name: 1}}, {"$limit": 2}, {"$project": {_id: 0, name: 1}}).toArray()
  [ { name: 'Aurora' }, { name: 'Ayna' } ]
```

Рисунок 6 – Разница в казалось бы одинаковых запросах

Микровывод: между цепочками команд find().sort().limit() и find().limit().sort() нет никакой разницы, они обе сначала сортируют всю коллекцию, а потом достают из неё какое-либо число документов. Лучше использовать агрегацию данных.

В limit() можно передать отрицательное значение, минус будет проигнорирован.

Супер нелогично, что запрос типа find({array: “value”}) работает так, как он работает. Было бы логичнее, если бы запрос записывался как find({array: {$any: “value”}}.

Практическое задание 8.1.3:

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

```
db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0})
  {
    _id: ObjectId("6464b9087a13e277e7f54efc"),
    name: 'Horny',
    weight: 600,
    vampires: 63
  }
  {
    _id: ObjectId("6464b9087a13e277e7f54efe"),
    name: 'Unicrom',
    weight: 984,
    vampires: 182
  }
```

Рисунок 7 – Выполнение задания

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```
db.unicorns.find({}, {_id: 0, name: 1}).sort({$natural: -1}).toArray()
  [
    { name: 'Dunx' },
    { name: 'Nimue' },
    { name: 'Pilot' },
    { name: 'Leia' },
    { name: 'Raleigh' },
    { name: 'Kenny' },
    { name: 'Ayna' },
    { name: 'Solnara' },
    { name: 'Roooooodles' },
    { name: 'Unicrom' },
    { name: 'Aurora' },
    { name: 'Horny' }
  ]
```

Рисунок 8 – Выполнение задания

Практическое задание 8.1.5:

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

```
db.unicorns.find({}, {_id: 0, loves: {$slice: 1}})
  {
    name: 'Horny',
    loves: [
      'carrot'
    ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
  {
    name: 'Aurora',
    loves: [
      'carrot'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
  {
    name: 'Unicrom',
```

Рисунок 9 – Выполнение задания

Практическое задание 8.1.6:

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

```
db.unicorns.find({gender: 'f', weight: {$gt : 500, $lt: 700}}, {_id: 0})
  {
    name: 'Solnara',
    loves: [
      'apple',
      'carrot',
      'chocolate'
    ],
    weight: 550,
    gender: 'f',
    vampires: 80
  }
  {
    name: 'Leia',
    loves: [
      'apple',
      'watermelon'
    ],
    weight: 601,
    gender: 'f',
    vampires: 33
  }
  {
```

Рисунок 10 – Выполнение задания

Практическое задание 8.1.7:

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

```
db.unicorns.find({gender: 'm', weight: {$gt: 500}, loves: {$all: ['grape', 'lemon']}}, {_id: 0})
  {
    name: 'Kenny',
    loves: [
      'grape',
      'lemon'
    ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
```

Рисунок 11 – Выполнение задания

Микровыводы: при выборе через $all порядок элементов не влияет.

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.

Рисунок 12 – Выполнение задания

Практическое задание 8.1.9:

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

```
db.unicorns.find({gender: 'm'}, {_id: 0, name: 1, loves: {$slice: 1}}).sort({name: 1})
  {
    name: 'Dunx',
    loves: [
      'grape'
    ]
  }
  {
    name: 'Horny',
    loves: [
      'carrot'
    ]
  }
  {
    name: 'Kenny',
    loves: [
      'grape'
    ]
  }
  {
    name: 'Pilot',
    loves: [
```

Рисунок 13 – Выполнение задания

Практическое задание 8.2.1:

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"}}
```

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

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

```
db.towns.find({"mayor.party": "I"}, {_id: 0, name: 1, mayor: 1})
  {
    name: 'New York',
    mayor: {
      name: 'Michael Bloomberg',
      party: 'I'
    }
  }
db.towns.find({"mayor.party": {$exists: false}}, {_id: 0, name: 1, mayor: 1})
  {
    name: 'Punxsutawney ',
    mayor: {
      name: 'Jim Wehrle'
    }
  }
```

Рисунок 14 – Выполнение задания

Практическое задание 8.2.2:

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.

```
learn> fn = function() {return this.gender == 'm';}
[Function: fn]
learn> var cursor = db.unicorns.find({"$where": fn}).sort({name: 1}).limit(2); null;
null
learn> cursor.forEach(function(obj) {print(obj) });
{
  _id: ObjectId("6464b9197a13e277e7f54f07"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
{
  _id: ObjectId("6464b9087a13e277e7f54efc"),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
```

Рисунок 15 – Выполнение задания

Микровыводы: код из ТЗ не работает, мне удалось найти только такое решение через $where. А еще у MongoDB Compass какие-то проблемы с функциями, код виснет и работает бесконечно.

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
db.unicorns.find({gender: 'f', weight: {$gt : 500, $lt: 600}}).count()
  2
```

Рисунок 16 – Выполнение задания

Практическое задание 8.2.4:

Вывести список предпочтений.

Рисунок 17 – Выполнение задания

Практическое задание 8.2.5:

Посчитать количество особей единорогов обоих полов.

Рисунок 18 – Выполнение задания

Практическое задание 8.2.6:

1. Выполнить команду:

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

2. Проверить содержимое коллекции unicorns.

Рисунок 19 – Выполнение задания

Микровыводы: метод save() является устаревшим. Из документации:

Starting in MongoDB 4.2, the `db.collection.save()` method is deprecated.
Use `db.collection.insertOne()` or `db.collection.replaceOne()` instead.

Практическое задание 8.2.7:

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

2. Проверить содержимое коллекции unicorns.

```
db.unicorns.update({name : "Ayna"}, {weight: 800, vampires: 51})
```

Рисунок 20 – Выполнение задания

Микровыводы: метод update() является устаревшим. Из терминала:

'DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.'

Практическое задание 8.2.8:

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

2. Проверить содержимое коллекции unicorns.

Рисунок 21 – Выполнение задания

Практическое задание 8.2.9:

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

2. Проверить содержимое коллекции unicorns.

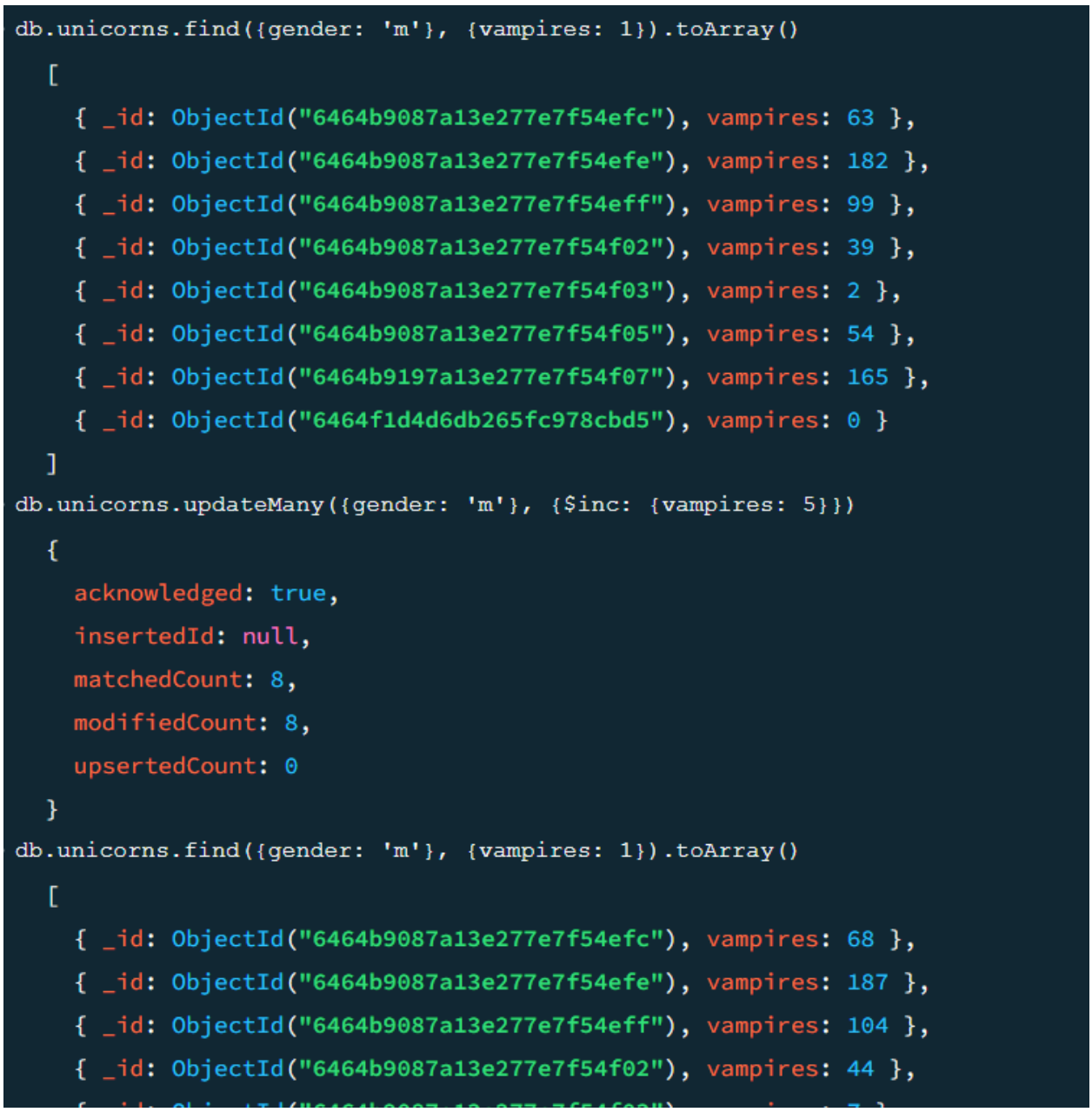

Рисунок 22 – Выполнение задания

Практическое задание 8.2.10:

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

2. Проверить содержимое коллекции towns.

Рисунок 23 – Выполнение задания

Практическое задание 8.2.11:

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

2. Проверить содержимое коллекции unicorns.

Рисунок 24 – Выполнение задания

Практическое задание 8.2.12:

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

2. Проверить содержимое коллекции unicorns.

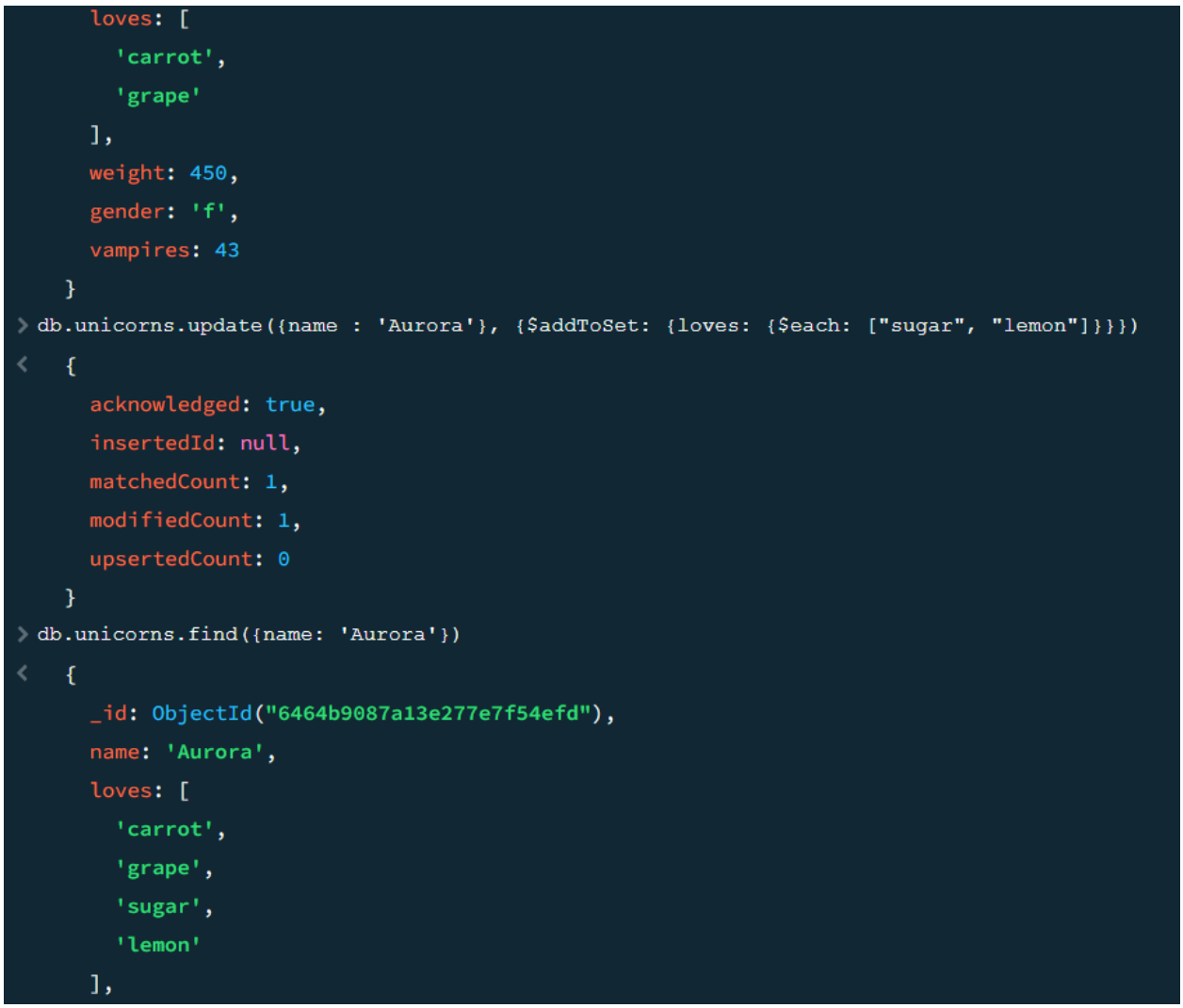

Рисунок 25 – Выполнение задания

Практическое задание 8.2.13:

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"}}
```

2. Удалите документы с беспартийными мэрами.

3. Проверьте содержание коллекции.

4. Очистите коллекцию.

5. Просмотрите список доступных коллекций.

Коллекция towns уже была создана в рамках данной лабораторной, по итогу модификаций в ней два беспартийных мэра.

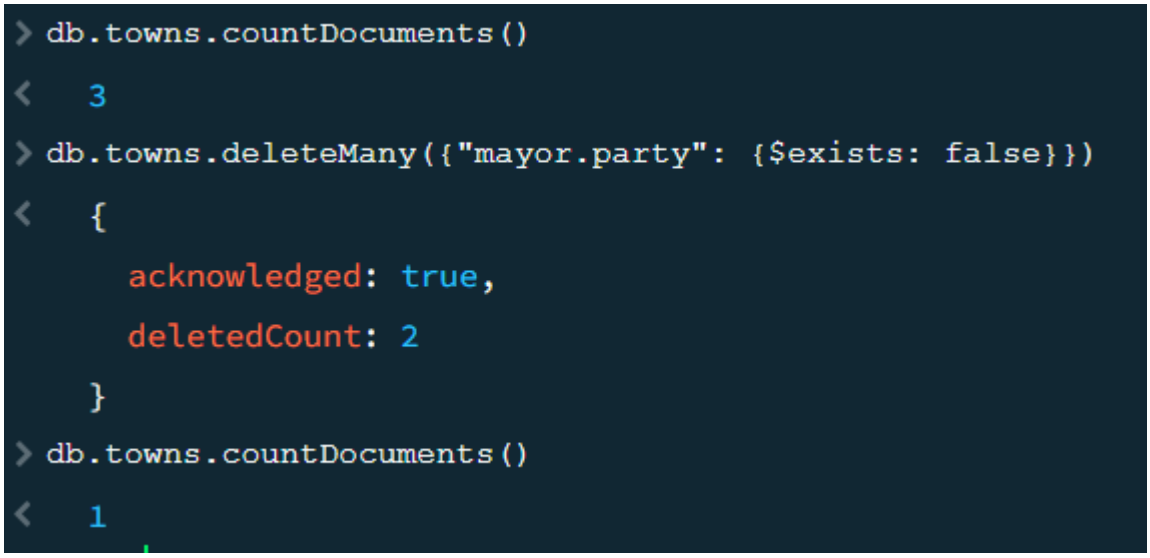

Рисунок 26 – Выполнение первой части задания

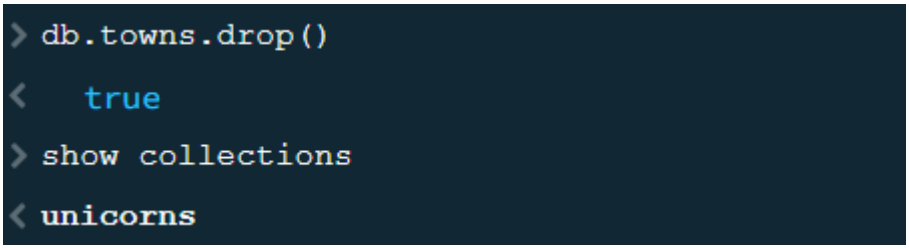

Рисунок 27 – Выполнение второй части задания

Практическое задание 8.3.1:

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 единорогов.

Содержание коллекции 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', 44), 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'});
db.unicorns.insert {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704,
gender: 'm', vampires: 165}
```

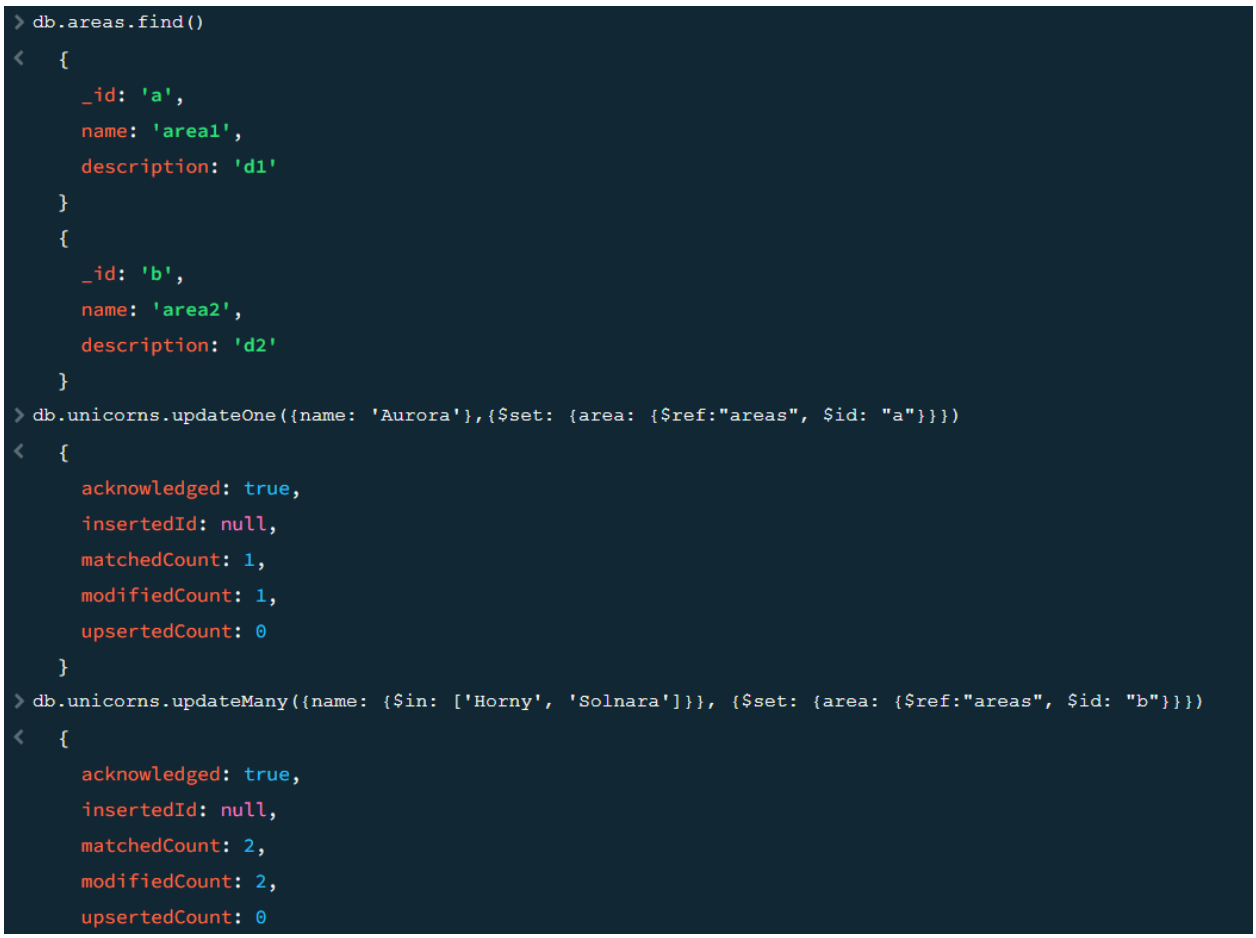

Рисунок 28 – Создание новой коллекции и связей

```
> db.unicorns.find({}, {_id: 0, name: 1, area: 1})
<   {
      name: 'Horny',
      area: DBRef("areas", 'b')
    }
    {
      name: 'Aurora',
      area: DBRef("areas", 'a')
    }
    {
      name: 'Unicrom'
    }
    {
      name: 'Roooooodles'
    }
    {
      name: 'Solnara',
      area: DBRef("areas", 'b')
    }
    {
      name: 'Ayna'
    }
    {
      name: 'Kenny'
    }
```

Рисунок 29 – Результат

Практическое задание 8.3.2:

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

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

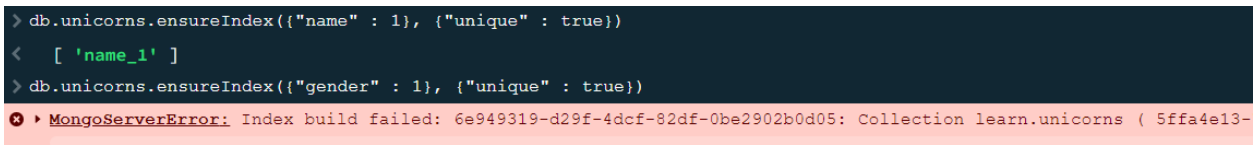

Рисунок 30 – Создание индексов

Практическое задание 8.3.3:

1. Получите информацию о всех индексах коллекции unicorns .

2. Удалите все индексы, кроме индекса для идентификатора.

3. Попытайтесь удалить индекс для идентификатора.

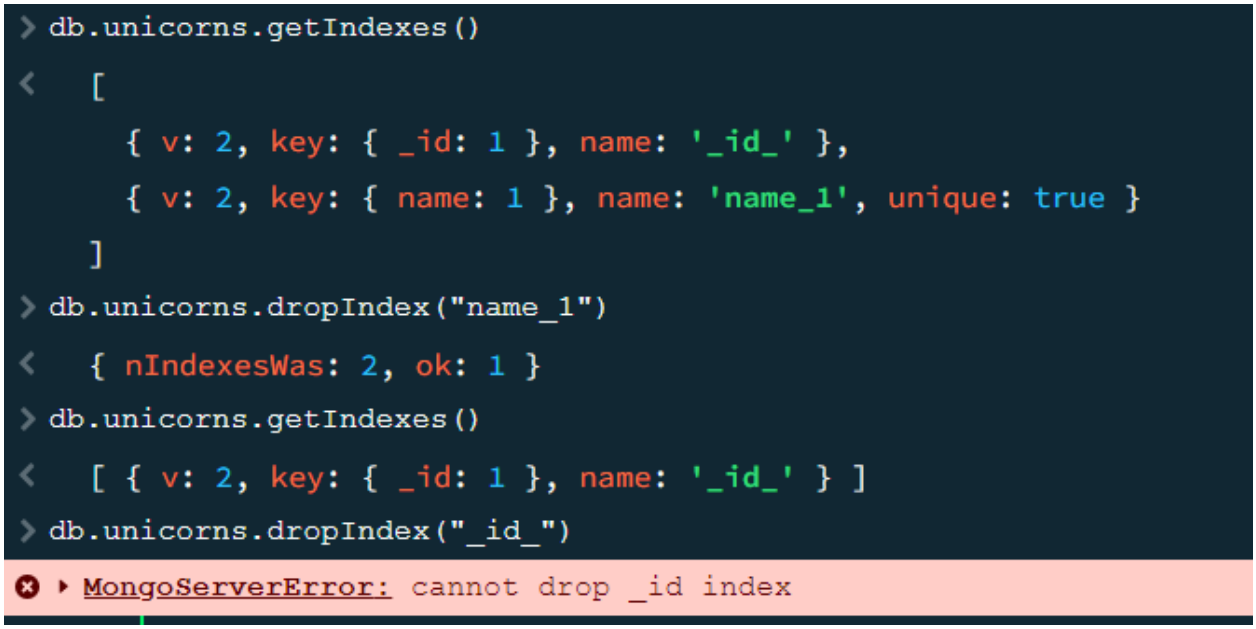

Рисунок 31 – Выполнение задания

Практическое задание 8.3.4:

1. Создайте объемную коллекцию numbers, задействовав курсор:

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

2. Выберите последних четыре документа.

3. Проанализируйте план выполнения запроса 2. Сколько потребовалось executionTimeMillis)

4. Создайте индекс для ключа value.

5. Получите информацию о всех индексах коллекции numbres.

6. Выполните запрос 2.

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

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

Без индекса запрос db.numbers.find().sort({ value: 1 }).limit(4) выполнялся 4 миллисекунды, с индексом – ноль. Правильно созданные индексы могут существенно ускорить работу запроса.

Выводы:

В данной работе я овладела 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.

Обнаружила, что Javascript-операции не выполняются в программе MongoDB Compass. Для работы с ними пришлось использовать mongosh, запущенный из обычной командной строки.

Многие методы из указанных в ТЗ к лабораторной работе являются устаревшими, поэтому в процессе выполнения пришлось заменить их на альтернативные.