

Министерство науки и высшего образования Российской Федерации  
федеральное государственное автономное образовательное учреждение  
высшего образования  
«Национальный исследовательский университет ИТМО»  
Факультет инфокоммуникационных технологий

**Лабораторная работа №5.2**  
**«Работа с БД в СУБД MongoDB»**  
**по дисциплине:**  
**«Базы данных»**

**Выполнила:**

студентка II курса ИКТ

группы К3242

Липина Ольга Андреевна

**Проверил:**

*Говорова Марина Михайловна*

Санкт-Петербург

2022

**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.

## Задания

### Практическое задание 8.1.1:

Создать базу, вставить документы, проверить содержимое.

```
A> use learn
switched to db learn
> db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Rooodooles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
WriteResult({ "nInserted" : 1 })
> document={name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}
{
  "name" : "Dunx",
  "loves" : [
    "grape",
    "watermelon"
  ],
  "weight" : 704,
  "gender" : "m",
  "vampires" : 165
}
> db.un
db.unicorns
db.unicorns.in
db.unicorns.initializeOrderedBulkOp( db.unicorns.insert( db.unicorns.insertOne(
db.unicorns.initializeUnorderedBulkOp( db.unicorns.insertMany(
> db.unicorns.insert(document)
WriteResult({ "nInserted" : 1 })
> df.unicorns.find()
uncaught exception: ReferenceError: df is not defined :
@(shell):1:1
> db.unicorns.find()
{ "_id" : ObjectId("62a4917a52f0c7e0d0e77655"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77656"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77657"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77658"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
```

Рисунок 1 – Результат выполнения запросов задания 8.1.1

### 8.1.2 Выборка данных из базы

1. Сформируйте запросы для вывода списков самцов и самок единорогов, ограничьте список самок первыми тремя особями и 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().sort({name: 1}).limit(3)
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77656"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765a"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 400, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62a4927852f0c7e0d0e77660"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
{
  "_id" : ObjectId("62a4917b52f0c7e0d0e77656"),
  "name" : "Aurora",
  "loves" : [
    "carrot",
    "grape"
  ],
  "weight" : 450,
  "gender" : "f",
  "vampires" : 43
}
> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77656"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
>
```

Рисунок 2 – Результат выполнения запросов задания 8.1.2

### 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}).sort({name: 1})
{ "_id" : ObjectId("62a4927852f0c7e0d0e77660"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("62a4917a52f0c7e0d0e77655"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765b"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765e"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765c"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77658"), "name" : "Rooooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77657"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
```

Рисунок 3 – Результат выполнения запросов задания 8.1.3

### 8.1.4

Вывести список единорогов в обратном порядке добавления

```
> db.unicorns.find().sort({ $natural: -1})
{ "_id" : ObjectId("62a4927852f0c7e0d0e77660"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765f"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "gender" : "f" }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765e"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765d"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765c"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765b"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765a"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77659"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77658"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77657"), "name" : "Unicrom", "loves" : [ "energion", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77656"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a4917a52f0c7e0d0e77655"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

Рисунок 4 – Результат выполнения запросов задания 8.1.4

#### 8.1.5

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

```
> db.unicorns.find( {}, {loves: {$slice: 1}, _id: 0})
{ "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Aurora", "loves" : [ "carrot" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "name" : "Unicrom", "loves" : [ "energion" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Solnara", "loves" : [ "apple" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "name" : "Ayna", "loves" : [ "strawberry" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "name" : "Leia", "loves" : [ "apple" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "name" : "Nimue", "loves" : [ "grape" ], "weight" : 540, "gender" : "f" }
{ "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

Рисунок 5 – Результат выполнения запросов задания 8.1.5

#### 8.1.6

Вывести список самок единорогов весом от 500 кг 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 }
{ "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

Рисунок 6 – Результат выполнения запросов задания 8.1.6

#### 8.1.7

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

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

Рисунок 7– Результат выполнения запросов задания 8.1.7

### 8.1.8

Найти всех единорогов, не имеющих ключ vampires.

```
> db.unicorns.find({vampires: {$exists: false}})
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765f"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "vampires" : "f" }
```

Рисунок 8 – Результат выполнения запросов задания 8.1.8

### 8.1.9

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

```
> db.unicorns.find({gender: 'm'}, {name: 1, loves: {$slice: 1}}).sort({name: 1})
{ "_id" : ObjectId("62a4927852f0c7e0d0e77660"), "name" : "Dunx", "loves" : [ "grape" ] }
{ "_id" : ObjectId("62a4917a52f0c7e0d0e77655"), "name" : "Horny", "loves" : [ "carrot" ] }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765b"), "name" : "Kenny", "loves" : [ "grape" ] }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765e"), "name" : "Pilot", "loves" : [ "apple" ] }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765c"), "name" : "Raleigh", "loves" : [ "apple" ] }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77658"), "name" : "Rooooooodles", "loves" : [ "apple" ] }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77657"), "name" : "Unicrom", "loves" : [ "energon" ] }
>
```

Рисунок 9 – Результат выполнения запросов задания 8.1.9

## 8.2 Выборка данных. Агрегированные запросы. Использование курсоров. Изменение данных. Вложенные объекты

### 8.2.1. Вложенные объекты

1. Создайте коллекцию towns, включающую следующие документы.
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'}, {name: 1, mayor: 1})
{ "_id" : ObjectId("62a49c7d8b7ccb118e36b1cc"), "name" : "New York", "mayor" : { "name" : "Michael Bloomberg",
"I" } }

> db.towns.find({"mayor.party": {$exists: false}}, {name: 1, mayor: 1})
{ "_id" : ObjectId("62a49c688b7ccb118e36b1cb"), "name" : "Punxsutawney ", "mayor" : { "name" : "Jim Wehrle" } }
```

Рисунок 10 – Результат выполнения запросов задания 8.2.1

### 8.2.2. Функции

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.

```
> func = function() {return this.gender == 'm'}
function() {return this.gender == 'm'}
> db.unicorns.find(func)
{ "_id" : ObjectId("62a4917a52f0c7e0d0e77655"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 45,
"gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77657"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 45,
"gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77658"), "name" : "Rooodoodles", "loves" : [ "apple" ], "weight" : 575,
"gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765b"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 69,
"gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765c"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 45,
"gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765e"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 45,
"gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a4927852f0c7e0d0e77660"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 45,
"gender" : "m", "vampires" : 165 }
```

```
> var cursor = db.unicorns.find({gender: "m"}); null;
null
> cursor.sort({name: 1}).limit(2); null;
null
```

```
> cursor.forEach(function(fn){print (fn.name);})
Dunx
Horny
```

Рисунок 11 – Результат выполнения запросов задания 8.2.2

### 8.2.3. Агрегатные функции

Вывести количество самок единорогов весом от 500 кг до 600 кг

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

Рисунок 12 – Результат выполнения запросов задания 8.2.3

### 8.2.4.

Вывести список предпочтений.

```
> db.unicorns.distinct('loves')  
[  
  "apple",  
  "carrot",  
  "chocolate",  
  "energon",  
  "grape",  
  "lemon",  
  "papaya",  
  "redbull",  
  "strawberry",  
  "sugar",  
  "watermelon"  
]
```

Рисунок 13 – Результат выполнения запросов задания 8.2.4

### 8.2.5.

Посчитать количество особей единорогов обоих полов

```
> db.unicorns.aggregate({$group: {_id: '$gender', count: {$sum: 1}}})  
{ "_id" : "f", "count" : 5 }  
{ "_id" : "m", "count" : 7 }
```

Рисунок 14 – Результат выполнения запросов задания 8.2.5

### 8.2.6. Обновление данных

1. Выполнить указанную в задании команду.
2. Проверить содержимое коллекции unicorns



```

> db.unicorns.save({name: 'Barney', loves: ['grape'],
... weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("62a4917a52f0c7e0d0e77655"), "name" : "Horny", "loves" : [ "carrot", "pap
der" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77656"), "name" : "Aurora", "loves" : [ "carrot", "gr
der" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77657"), "name" : "Unicrom", "loves" : [ "enegon", "
"gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77658"), "name" : "Roooooodles", "loves" : [ "apple"
: "m", "vampires" : 99 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77659"), "name" : "Solnara", "loves" : [ "apple", "ca
t" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765a"), "name" : "Ayna", "loves" : [ "strawberry", "
ender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765b"), "name" : "Kenny", "loves" : [ "grape", "lemd
r" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765c"), "name" : "Raleigh", "loves" : [ "apple", "su
der" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765d"), "name" : "Leia", "loves" : [ "apple", "water
ender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765e"), "name" : "Pilot", "loves" : [ "apple", "wate
gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765f"), "name" : "Nimue", "loves" : [ "grape", "carr
er" : "f" }
{ "_id" : ObjectId("62a4927852f0c7e0d0e77660"), "name" : "Dunx", "loves" : [ "grape", "water
ender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a4a1408b7ccb118e36b1ce"), "name" : "Barney", "loves" : [ "grape" ], "we
}

```

Рисунок 14 – Результат выполнения запросов задания 8.2.6

### 8.2.7.

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

```

> db.unicorns.update({name: 'Ayna', gender: 'f'}, {weight: 800, 'vampires': 51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: 'Ayna'})

{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765a"), "name" : "Ayna", "gender" : "f", "weight" : 800, "vampires" : 51 }
>

```

Рисунок 15 – Результат выполнения запросов задания 8.2.7

### 8.2.8. Установка значения поля

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

```

> db.unicorns.update({name: 'Raleigh'}, {$set: {loves: 'redbull'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: 'Raleigh'})
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765c"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421, "gender" : "m",
"vampires" : 2 }
>

```

Рисунок 16 – Результат выполнения запросов задания 8.2.8

### 8.2.9. Увеличение значения поля для опр записей таблицы

Всем самцам единорогов увеличить количество убитых вампиров на 5.



```

> db.unicorns.update({gender: 'm'}, {$inc: {vampires: 5}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorn.find({gender: 'm'})
> db.unicorns.find({gender: 'm'})
{ "_id" : ObjectId("62a4917a52f0c7e0d0e77655"), "name" : "Horny", "loves" : [ "carr
t", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77657"), "name" : "Unicrom", "loves" : [ "er
rgon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77658"), "name" : "Roooooodles", "loves" : [
"apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765b"), "name" : "Kenny", "loves" : [ "grap
", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765c"), "name" : "Raleigh", "loves" : "redb
ll", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765e"), "name" : "Pilot", "loves" : [ "appl
", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a4927852f0c7e0d0e77660"), "name" : "Dunx", "loves" : [ "grape
", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a4a1408b7ccb118e36b1ce"), "name" : "Barney", "loves" : [ "grap
" ], "weight" : 340, "gender" : "m" }
>

```

Рисунок 17 – Результат выполнения запросов задания 8.2.9

#### 8.2.10. Удаление поля для записи

Изменить информацию о городе Portland: мэр этого города теперь беспартийный

```

> db.towns.update({name: 'Portland'}, {$unset: {'mayor.party': 1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.towns.find({name: 'Portland'})
{ "_id" : ObjectId("62a49c988b7ccb118e36b1cd"), "name" : "Portland", "populatiuon" :
528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "
food" ], "mayor" : { "name" : "Sam Adams" } }
>

```

Рисунок 18 – Результат выполнения запросов задания 8.2.10

#### 8.2.11.

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

```

> db.unicorns.update({name: 'Pilot'}, {$push: {loves: 'chocolate'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: 'Pilot'})
{ "_id" : ObjectId("62a4917b52f0c7e0d0e7765e"), "name" : "Pilot", "loves" : [ "apple
", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
>

```

Рисунок 19 – Результат выполнения запросов задания 8.2.11

#### 8.2.12.

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

```

> db.unicorns.update({name: 'Aurora'}, {$addToSet: {loves: {$each: ['sugar', 'lemon'
]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: 'Aurora'})
{ "_id" : ObjectId("62a4917b52f0c7e0d0e77656"), "name" : "Aurora", "loves" : [ "carr
ot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
>

```

Рисунок 20 – Результат выполнения запросов задания 8.2.12

### 8.2.13. Удаление элементов из коллекции

Удалите документы с беспартийными мэрами. Удалите коллекцию. Посмотрите список доступных коллекций.

```
> db.towns.find()
{ "_id" : ObjectId("62a49c688b7ccb118e36b1cb"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("62a49c7d8b7ccb118e36b1cc"), "name" : "New York", "populatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("62a49c988b7ccb118e36b1cd"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
>
>
> db.towns.remove({'mayor.party': {$exists: false}})
WriteResult({ "nRemoved" : 1 })
> db.towns.find()
{ "_id" : ObjectId("62a49c7d8b7ccb118e36b1cc"), "name" : "New York", "populatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("62a49c988b7ccb118e36b1cd"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
>
```

```
> db.towns.remove({})
WriteResult({ "nRemoved" : 2 })
> show collections
towns
unicorns
> db.towns.find()
>
```

Рисунок 21 – Результат выполнения запросов задания 8.2.13

### 8.3.2. Индексы

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

```
> db.unicorns.ensureIndex({"name" : 1}, {"unique" : true})
uncaught exception: TypeError: db.unicorns.ensureIndex is not a function :
@(shell):1:1
```

Рисунок 22 – Результат выполнения запросов задания 8.3.2

### 8.3.3. Удаление индексов

Получите информацию о всех индексах коллекции unicorns .

Удалите все индексы, кроме индекса для идентификатора.

Попытайтесь удалить индекс для идентификатора.

```

> db.unicorns.getIndexes()
[ { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_" } ]
> db.unicorns.dropIndexes()
{
  "nIndexesWas" : 1,
  "msg" : "non-_id indexes dropped for collection",
  "ok" : 1
}
> db.unicorns.dropIndexes({'_id': 1})
uncaught exception: Error: error dropping indexes : {
  "ok" : 0,
  "errmsg" : "cannot drop _id index",
  "code" : 72,
  "codeName" : "InvalidOptions"
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DBCollection.prototype.dropIndexes@src/mongo/shell/collection.js:704:11
@(shell):1:1

```

Рисунок 23 – Результат выполнения запросов задания 8.3.3

#### 8.3.4.

1. Создайте объемную коллекцию numbers, задействовав курсор.
2. Выберите четыре последних документа.
3. Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? **Менее секунды.**
4. Создайте индекс для ключа value.
5. Получите информацию о всех индексах коллекции numbers.
6. Выполните запрос №2.
7. Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса? **Разницы не обнаружено.**

```

> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
WriteResult({ "nInserted" : 1 })
> db.numbers.find().sort({$natural: -1}).limit(4)
{ "_id" : ObjectId("62a4b2274dd23a7e04cdb6e0"), "value" : 99999 }
{ "_id" : ObjectId("62a4b2274dd23a7e04cdb6df"), "value" : 99998 }
{ "_id" : ObjectId("62a4b2274dd23a7e04cdb6de"), "value" : 99997 }
{ "_id" : ObjectId("62a4b2274dd23a7e04cdb6dd"), "value" : 99996 }
>

```

```

> db.numbers.explain('executionStats').find().sort({$natural: -1}).limit(4)
{
  "explainVersion" : "1",
  "queryPlanner" : {
    "namespace" : "learn.numbers",
    "indexFilterSet" : false,
    "parsedQuery" : {

    },
    "maxIndexedOrSolutionsReached" : false,
    "maxIndexedAndSolutionsReached" : false,
    "maxScansToExplodeReached" : false,
    "winningPlan" : {
      "stage" : "LIMIT",
      "limitAmount" : 4,
      "inputStage" : {
        "stage" : "COLLSCAN",
        "direction" : "backward"
      }
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 4,
    "executionTimeMillis" : 0,

```

```

> db.numbers.createIndex({"value" : 1})
{
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "createdCollectionAutomatically" : false,
  "ok" : 1
}
>

```

```

},
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 4,
  "executionTimeMillis" : 0,

```

Рисунок 24 – Результат выполнения запросов задания 8.3.4

## Выводы:

Во время выполнения лабораторной работы я отработала основные команды манипуляции с данными в СУБД MongoDB – выборка данных, изменение данных, добавление и удаление данных, а также работа с индексами.