

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИТМО»  
Факультет инфокоммуникационных технологий

**ОТЧЕТ**  
**О ЛАБОРАТОРНОЙ РАБОТЕ № 5.2**  
по теме: Работа с БД в СУБД MongoDB  
по дисциплине: Проектирование и реализация баз данных

Специальность: 09.03.03 Мобильные и сетевые технологии

Проверил:  
Говорова М.М. \_\_\_\_\_  
Дата: «\_\_» \_\_\_\_\_ 20\_\_ г.  
Оценка \_\_\_\_\_

Выполнил:  
студент группы К3240 Кулагина  
С.В.

Санкт-Петербург 2021/2022

## ЦЕЛЬ РАБОТЫ

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.

## ПРАКТИЧЕСКОЕ ЗАДАНИЕ

1. Создайте базу данных learn

```
> use learn
switched to db learn
```

2. Заполните коллекцию единорогов unicorns

```
> db.createCollection("unicorns")
{ "ok" : 1 }
> db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', v
});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f',
});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender:
: 182});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Rooodooles', loves: ['apple'], weight: 575, gender: 'm', vamp
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550,
vampires:80});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f'
0});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', v
;
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm',
});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f'
33});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: '
54});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'})
```

3. Используя второй способ, вставьте в коллекцию единорогов документ

```
> document = ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})
{
  "name" : "Dunx",
  "loves" : [
    "grape",
    "watermelon"
  ],
  "weight" : 704,
  "gender" : "m",
  "vampires" : 165
}
```

```
> db.unicorns.insert(document)
WriteResult({ "nInserted" : 1 })
```

#### 4. Проверьте содержимое коллекции с помощью метода find

```
> db.unicorns.find()
{ "_id" : ObjectId("6294ba2acdd6912ca11ab66"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6294ba43cdd6912ca11ab67"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6294ba61cdd6912ca11ab68"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6294ba6acdd6912ca11ab69"), "name" : "Rooodoodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6294ba74cdd6912ca11ab6a"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6294ba85cdd6912ca11ab6b"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6294ba85cdd6912ca11ab6c"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6294ba85cdd6912ca11ab6d"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6294ba85cdd6912ca11ab6e"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6294ba85cdd6912ca11ab6f"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6294ba85cdd6912ca11ab70"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6294bbc9cdd6912ca11ab71"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

### 8.2.2 Выборка данных из БД

#### 8.1.2

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

Для самцов

```
> db.unicorns.find({gender: "m"}).sort({name:1})
{ "_id" : ObjectId("6294bbc9cdd6912ca11ab71"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6294ba2acdd6912ca11ab66"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6294ba85cdd6912ca11ab6c"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6294ba85cdd6912ca11ab6f"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6294ba85cdd6912ca11ab6d"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6294ba6acdd6912ca11ab69"), "name" : "Rooodoodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6294ba61cdd6912ca11ab68"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
```

Для самок

```
> db.unicorns.find({gender: "f"}).sort({name:1}).limit(3)
{ "_id" : ObjectId("6294ba43cdd6912ca11ab67"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6294ba85cdd6912ca11ab6b"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6294ba85cdd6912ca11ab6e"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
```

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

FindOne

```
> db.unicorns.findOne({gender: "f", loves:"carrot"})
{
  "_id" : ObjectId("6294ba43cdd6912ca11ab67"),
  "name" : "Aurora",
  "loves" : [
    "carrot",
    "grape"
  ],
  "weight" : 450,
  "gender" : "f",
  "vampires" : 43
}
```

## Limit

```
> db.unicorns.find({gender: "f", loves: "carrot"}).limit(1)
{ "_id" : ObjectId("6294ba43cdcd6912ca11ab67"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

### 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("6294bbc9cdcd6912ca11ab71"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("6294ba2acddcd6912ca11ab66"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("6294ba85cdcd6912ca11ab6c"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("6294ba85cdcd6912ca11ab6f"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("6294ba85cdcd6912ca11ab6d"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("6294ba6acddcd6912ca11ab69"), "name" : "Roooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("6294ba61cdcd6912ca11ab68"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
>
```

### 8.1.4

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({ $natural: -1 })
{ "_id" : ObjectId("628b756a3dae894bf27ae69c"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae69b"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("628b74ae3dae894bf27ae69a"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae699"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae698"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae697"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae696"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae695"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae694"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae693"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae692"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae691"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

### 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" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "name" : "Roooooodles", "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 }
```

### 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 }
{ "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

### 8.1.7

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

```
> db.unicorns.find({gender:"m", weight:{$gt : 500}, loves: {$in:["grape","lemon"]},{_id : 0})
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

### 8.1.8

Найти всех единорогов, не имеющих ключ vampires.

```
> db.unicorns.find ({vampires: {$exists:false}}, {_id : 0})
{ "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

### 8.1.9

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

```
> db.unicorns.find({gender:"m"}, {loves: {$slice : 1}}).sort({name : 1})
{ "_id" : ObjectId("628b756a3dae894bf27ae69c"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae691"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae697"), "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae69a"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae698"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae694"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae693"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
```

### 8.2.1

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.

```
> db.towns.find({"mayor.party" : "I"}, {name : 1, mayor : 1, _id : 0})
{ "name" : "New York", "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
```

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

```
> db.towns.find({"mayor.party" : {$exists : false}}, {name : 1, mayor : 1, _id : 0})
{ "name" : "Punxsutawney ", "mayor" : { "name" : "Jim Wehrle" } }
```

### 8.2.2

1. Сформировать функцию для вывода списка самцов единорогов.

```
> f=function(){return this.gender=="m";}
function(){return this.gender=="m";}
```

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

```
> var cursor = db.unicorns.find(f);null;
null
> cursor.sort({name : 1}).limit(2)
{ "_id" : ObjectId("628b756a3dae894bf27ae69c"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628b74ae3dae894bf27ae691"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
>
```

3. Вывести результат, используя `forEach`.

```
> var cursor = db.unicorns.find(fn); null;
null
> cursor.sort({name : 1}).limit(2); null;
null
> cursor.forEach(function(obj){ print(obj.name); })
Dunx
Horny
```

### 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
```

### 8.2.4

Вывести список предпочтений.

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

### 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 }
```

### 8.2.6

1. Выполнить команду:

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

2. Проверить содержимое коллекции `unicorns`.



```

> db.unicorns.save({name: 'Barney', loves: ['grape'],
... weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("6294acc8724e714bd3307b814"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6294acc8724e714bd3307b815"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6294acc8724e714bd3307b816"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6294acc8724e714bd3307b817"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6294acc8724e714bd3307b818"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6294acd0324e714bd3307b819"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6294acd0324e714bd3307b81a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6294acd0324e714bd3307b81c"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6294acd0324e714bd3307b81d"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6294cdf324e714bd3307b81e"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6294cdf324e714bd3307b81f"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6294ce0324e714bd3307b820"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6294ceb624e714bd3307b821"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }

```

## 8.2.7

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

2. Проверить содержимое коллекции unicorns.

```

> db.unicorns.update({name: "Ayna"}, {name: "Ayna", weight: 800, gender: "f", vampires: 51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Ayna"})
{ "_id" : ObjectId("628b74ae3dae894bf27ae696"), "name" : "Ayna", "weight" : 800, "gender" : "f", "vampires" : 51 }

```

## 8.2.8

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

2. Проверить содержимое коллекции unicorns.

```

> db.unicorns.update({name: "Raleigh"}, {$set: {loves: ["redbull"]}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Raleigh"})
{ "_id" : ObjectId("628b74ae3dae894bf27ae698"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

```

## 8.2.9

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

2. Проверить содержимое коллекции unicorns.

```

> db.unicorns.find()
{ "_id" : ObjectId("6294acc8724e714bd3307b814"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6294acc8724e714bd3307b815"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6294acc8724e714bd3307b816"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6294acc8724e714bd3307b817"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6294acc8724e714bd3307b818"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6294acd0324e714bd3307b819"), "name" : "Ayna", "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("6294acd0324e714bd3307b81a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6294acd0324e714bd3307b81c"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6294acd0324e714bd3307b81d"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6294cdf324e714bd3307b81e"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6294cdf324e714bd3307b81f"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6294ce0324e714bd3307b820"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6294ceb624e714bd3307b821"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }

```

```

> db.unicorns.updateMany({gender: "m"}, {$inc: {vampires: 5}})
{ "acknowledged" : true, "matchedCount" : 8, "modifiedCount" : 8 }
> db.unicorns.find({gender: "m"})
{ "_id" : ObjectId("6294acc8724e714bd3307b814"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("6294acc8724e714bd3307b816"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("6294acc8724e714bd3307b817"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("6294acd0324e714bd3307b81a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("6294acd0324e714bd3307b81c"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("6294cdf324e714bd3307b81e"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("6294ce0324e714bd3307b820"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("6294ceb624e714bd3307b821"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
>

```

## 8.2.10

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

## 2. Проверить содержимое коллекции towns.

```
> db.towns.update({name:"Portland"}, {$unset:{mayor.party:1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.towns.find()
{ "_id" : ObjectId("6294c3f624e714bd3307b7fa"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("6294c41424e714bd3307b7fb"), "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("6294c43524e714bd3307b7fc"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams" } }
```

### 8.2.11

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

2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.update({name:"Pilot"}, {$push:{loves:"chocolate"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("6294cc8724e714bd3307b814"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("6294cc8f24e714bd3307b815"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6294cc9724e714bd3307b816"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("6294cce824e714bd3307b817"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("6294ccf324e714bd3307b818"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6294cd0324e714bd3307b819"), "name" : "Ayna", "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("6294cd0c24e714bd3307b81a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("6294cde324e714bd3307b81c"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("6294cdea24e714bd3307b81d"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6294cdf324e714bd3307b81e"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("6294cdfc24e714bd3307b81f"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6294ce0324e714bd3307b820"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("6294ceb624e714bd3307b821"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

### 8.2.12

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

2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.update({name:"Aurora"}, {$addToSet:{loves:{each:["sugar", "lemon"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("6294cc8724e714bd3307b814"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("6294cc8f24e714bd3307b815"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6294cc9724e714bd3307b816"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("6294cce824e714bd3307b817"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("6294ccf324e714bd3307b818"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6294cd0324e714bd3307b819"), "name" : "Ayna", "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("6294cd0c24e714bd3307b81a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("6294cde324e714bd3307b81c"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("6294cdea24e714bd3307b81d"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6294cdf324e714bd3307b81e"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("6294cdfc24e714bd3307b81f"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6294ce0324e714bd3307b820"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("6294ceb624e714bd3307b821"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

### 8.2.13

1. Создайте коллекцию towns, включающую следующие документы

```
> db.towns.find()
{ "_id" : ObjectId("6294da4524e714bd3307b822"), "name" : "Punxsutawney ", "popujatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "phil the groundhog" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("6294da8124e714bd3307b823"), "name" : "New York", "popujatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("6294daae24e714bd3307b824"), "name" : "Portland", "popujatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
```

2. Удалите документы с беспартийными мэрами

3. Проверьте содержание коллекции.



```
> db.towns.remove({"mayor.party":{"exists:false"}})
WriteResult({ "nRemoved" : 1 })
> db.towns.find()
{ "_id" : ObjectId("6294da8124e714bd3307b823"), "name" : "New York", "popujatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("6294daae24e714bd3307b824"), "name" : "Portland", "popujatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
```

4. Очистите коллекцию.

5. Просмотрите список доступных коллекций.

```
> db.towns.drop()
true
> show collections
unicorns
>
```

### 8.3.1

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

```
> db.zones.insertMany([{"id":"forest", name:"big forest", description:"a very big and huge forest"}, {"id":"jungle", name:"funny jungle", description:"a very funny jungle"}, {"id":"home", name:"my home", description:"my lovelu home"}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("6294ddca24e714bd3307b825"),
    ObjectId("6294ddca24e714bd3307b826"),
    ObjectId("6294ddca24e714bd3307b827")
  ]
}
```

```
> db.zones.find()
{ "_id" : ObjectId("6294ddca24e714bd3307b825"), "id" : "forest", "name" : "big forest", "description" : "a very big and huge forest" }
{ "_id" : ObjectId("6294ddca24e714bd3307b826"), "id" : "jungle", "name" : "funny jungle", "description" : "a very funny jungle" }
{ "_id" : ObjectId("6294ddca24e714bd3307b827"), "id" : "home", "name" : "my home", "description" : "my lovelu home" }
```

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

```
> db.unicorns.update({name:"Horny"},{$set:{zone:{$ref:"zones",$id:"forest"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name:"Aurora"},{$set:{zone:{$ref:"zones",$id:"jungle"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name:"Unicrom"},{$set:{zone:{$ref:"zones",$id:"home"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

3. Проверьте содержание коллекции единорогов

```
> db.unicorns.find()
{ "_id" : ObjectId("6294cc8724e714bd3307b814"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68, "zone" : DBRef("zones", "forest" ) }
{ "_id" : ObjectId("6294cc8f24e714bd3307b815"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "zone" : DBRef("zones", "jungle" ) }
{ "_id" : ObjectId("6294cc9724e714bd3307b816"), "name" : "Unicrom", "loves" : [ "energion", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187, "zone" : DBRef("zones", "home" ) : "I" ) }
{ "_id" : ObjectId("6294cc9824e714bd3307b817"), "name" : "Roocoodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104, "zone" : "Jim Wehrle" ) }
{ "_id" : ObjectId("6294ccf324e714bd3307b818"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6294cd0324e714bd3307b819"), "name" : "Ayna", "weight" : 800, "gender" : "f", "vampires" : 51, "zone" : "il the groundhog", "mayor" : { name: "Jim Wehrle" } }
{ "_id" : ObjectId("6294cd0c24e714bd3307b81a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44, "zone" : "Jim Wehrle" ) }
{ "_id" : ObjectId("6294cde324e714bd3307b81c"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7, "zone" : "mayor" : { name: "Michael Bloomberg", party: "I" ) } }
{ "_id" : ObjectId("6294cdea24e714bd3307b81d"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33, "zone" : "Adams", party: "D" ) }
{ "_id" : ObjectId("6294cdf324e714bd3307b81e"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59, "zone" : "D" ) }
{ "_id" : ObjectId("6294cdfc24e714bd3307b81f"), "name" : "Nimuc", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6294ceb324e714bd3307b820"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("6294ceb624e714bd3307b821"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

4. 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: 'Rooooooodles', 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}

```

### 8.3.2

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

```

> db.unicorns.createIndex({"name" : 1}, {"unique" : true})
{
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "createdCollectionAutomatically" : false,
  "ok" : 1
}
> db.unicorns.insert({name:"Pilot"})
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: learn.unicorns index: name_1 dup key: { name: \"Pilot\" }"
  }
})

```

### 8.3.3

1. Получите информацию о всех индексах коллекции unicorns.

```

> db.unicorns.getIndexes()
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_"
  },
  {
    "v" : 2,
    "key" : {
      "name" : 1
    },
    "name" : "name_1",
    "unique" : true
  }
]
>

```

2. Удалите все индексы, кроме индекса для идентификатора.

```
> db.unicorns.dropIndex("name_1")
{ "nIndexesWas" : 2, "ok" : 1 }
> db.unicorns.getIndexes()
[ { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_" } ]
```

3. Попробуйте удалить индекс для идентификатора

```
> db.unicorns.dropIndex("_id_")
{
  "ok" : 0,
  "errmsg" : "cannot drop _id index",
  "code" : 72,
  "codeName" : "InvalidOptions"
}
```

### 8.3.4

1. Создайте объемную коллекцию numbers, задействовав курсор:

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

```
> db.createCollection("numbers")
{ "ok" : 1 }
> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
WriteResult({ "nInserted" : 1 })
```

2. Выберите последних четыре документа.

```
> db.numbers.find().sort({$natural:-1}).limit(4)
{ "_id" : ObjectId("6294e48524e714bd330ac573"), "value" : 99999 }
{ "_id" : ObjectId("6294e48524e714bd330ac572"), "value" : 99998 }
{ "_id" : ObjectId("6294e48524e714bd330ac571"), "value" : 99997 }
{ "_id" : ObjectId("6294e48524e714bd330ac570"), "value" : 99996 }
> █
```

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

```

> db.users.explain("executionStats").find().sort({value:-1}).limit(4)
{
  "explainVersion" : "1",
  "queryPlanner" : {
    "namespace" : "test.users",
    "indexFilterSet" : false,
    "parsedQuery" : {
      },
    "maxIndexedOrSolutionsReached" : false,
    "maxIndexedAndSolutionsReached" : false,
    "maxScansToExplodeReached" : false,
    "winningPlan" : {
      "stage" : "EOF"
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 0,
    "executionTimeMillis" : 3,
    "totalKeysExamined" : 0,
    "totalDocsExamined" : 0,
    "executionStages" : {
      "stage" : "EOF",
      "nReturned" : 0,
      "executionTimeMillisEstimate" : 0,
      "works" : 1,
      "advanced" : 0,
      "needTime" : 0,

```

4.Создайте индекс для ключа value.

5. Получите информацию о всех индексах коллекции nombres

```

> db.numbers.createIndex({"value":1})
{
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "createdCollectionAutomatically" : false,
  "ok" : 1
}
> db.numbers.getIndexes()
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_"
  },
  {
    "v" : 2,
    "key" : {
      "value" : 1
    },
    "name" : "value_1"
  }
]

```

6.Выполните запрос 2.

```
> db.users.explain("executionStats").find().sort({value:-1}).limit(4)
{
  "explainVersion" : "1",
  "queryPlanner" : {
    "namespace" : "test.users",
    "indexFilterSet" : false,
    "parsedQuery" : {

    },
    "maxIndexedOrSolutionsReached" : false,
    "maxIndexedAndSolutionsReached" : false,
    "maxScansToExplodeReached" : false,
    "winningPlan" : {
      "stage" : "EOF"
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 0,
    "executionTimeMillis" : 0,
    "totalKeysExamined" : 0,
    "totalDocsExamined" : 0,
    "executionStages" : {
      "stage" : "EOF",
      "nReturned" : 0,
      "executionTimeMillisEstimate" : 0,
      "works" : 1,
      "advanced" : 0,
      "needTime" : 0,
      "needYield" : 0,
      "saveState" : 0,
      "restoreState" : 0,
      "isEOF" : 1
    }
  },
  "command" : {
    "find" : "users",

```

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?

*ВЫВОД:* индексами запрос сработал. Разница между запросами с индексами и без небольшая, но в дальнейшей работе это могло бы сильно повлиять на скорость работы. Поэтому можно утверждать, что с индексами запрос будет работать быстрее и эффективнее.



**ВЫВОД по работе:** В ходе работы были получены практические навыки работы с CRUD - операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, с ссылками и индексами в базе данных MongoDB.