

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

## **Лабораторная работа №5**

### **Работа с БД в СУБД MongoDB**

Выполнил:

студент 2 курса ИКТ группа К32392,  
Елистратов Владимир Дмитриевич

Проверила:

Горова Марина Михайловна

Санкт-Петербург  
2023

## 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:

### 8.1.1

- Создание базы данных learn:

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

- Заполнение коллекции единорогов unicorns:

```
> 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: 'Rooooooodles', 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 })
>
```

- Вставка коллекции единорогов в документ вторым способом:

```
> 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 })
```

- Проверка содержимого коллекции:

```
> db.unicorns.find()
{ "_id" : ObjectId("6477441f175f9190acd5e0a4"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a5"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a8"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a9"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6477441f175f9190acd5e0aa"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ab"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ac"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ad"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ae"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6477441f175f9190acd5e0af"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>
```

### 8.1.2

Запросы для вывода списков самцов и самок единорогов. Отсортируйте списки по имени:

Код запроса:

```
db.unicorns.find({"gender": "m"}).limit(3).sort({"name": -1})
```

```
db.unicorns.find({"gender": "f"}).limit(3).sort({"name": -1})
```

Результат выполнения запроса

```
> db.unicorns.find({"gender": "m"}).limit(3).sort({"name": -1})
{ "_id" : ObjectId("6477441f175f9190acd5e0a6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ab"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
> db.unicorns.find({"gender": "f"}).limit(3).sort({"name": -1})
{ "_id" : ObjectId("6477441f175f9190acd5e0a8"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ae"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6477441f175f9190acd5e0ac"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
>
```

Запрос для поиска всех самок, которые любят carrot. Ограниченный первой особью с помощью функций findOne и limit.

Код запроса:

```
db.unicorns.find({"gender": "f", "loves": "carrot"}).limit(1)
```

Результат выполнения запроса

```
> db.unicorns.find({"gender": "f", "loves": "carrot"}).limit(1)
{ "_id" : ObjectId("6477441f175f9190acd5e0a5"), "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

Код запроса:

```
db.unicorns.find({"gender": "m"}, {"gender": 0, "loves": 0}).sort({"name": -1})
```

Результат выполнения запроса

```
> db.unicorns.find({"gender": "m"}, {"gender": 0, "loves": 0}).sort({"name": -1})
{ "_id" : ObjectId("6477441f175f9190acd5e0a6"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a7"), "name" : "Roooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ab"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ad"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("6477441f175f9190acd5e0aa"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a4"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("647744d8175f9190acd5e0af"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
```

### 8.1.4

Вывести список единорогов в обратном порядке добавления:

Код запроса:

```
db.unicorns.find().sort({ $natural: -1 })
```

Результат выполнения запроса

```
{ "_id" : ObjectId("647744d8175f9190acd5e0af"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ae"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6477441f175f9190acd5e0ad"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ac"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ab"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6477441f175f9190acd5e0aa"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a9"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a8"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a5"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a4"), "name" : "Horny", "loves" : [ "carrot", "grape" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

### 8.1.5

Запрос выводящий список единорогов с названием первого любимого предпочтения, исключая идентификатор:

Код запроса:

```
db.unicorns.find({}, {"_id": 0, "loves": {$slice: 1}})
```

Результат выполнения запроса

```
{ "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" : "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 }
```

### 8.1.6

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

Код запроса:

```
db.unicorns.find({"weight": {$gte: 500, $lt: 700}}, {"_id": 0})
```

Результат выполнения запроса

```
{ "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
> |
```

### 8.1.7

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

Код запроса:

```
db.unicorns.find({"weight": {$gte: 500, $lt: 700}, "loves": {$all: ["grape", "lemon"]}}, {"_id": 0})
```

Результат выполнения запроса

```
> db.unicorns.find({"weight": {$gte: 500, $lt: 700}, "loves": {$all: ["grape", "lemon"]}}, {"_id": 0})
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
```

## 8.1.8

Найти всех единорогов, не имеющих ключ `vampires`:

Код запроса:

```
db.unicorns.find({"vampires": {$exists: false}})
```

Результат выполнения запроса

```
> db.unicorns.find({"vampires": {$exists: false}})
{ "_id" : ObjectId("6477441f175f9190acd5e0ae"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

## 8.1.9

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

Код запроса:

```
db.unicorns.find({"gender": "m"}, {"_id": 0, "loves": {$slice: 1}, "name": 1}).sort({"name": 1})
```

Результат выполнения запроса

```
> db.unicorns.find({"gender": "m"}, {"_id": 0, "loves": {$slice: 1}, "name": 1}).sort({"name": 1});
{ "name" : "Dunx", "loves" : [ "grape" ] }
{ "name" : "Horny", "loves" : [ "carrot" ] }
{ "name" : "Kenny", "loves" : [ "grape" ] }
{ "name" : "Pilot", "loves" : [ "apple" ] }
{ "name" : "Raleigh", "loves" : [ "apple" ] }
{ "name" : "Roooooodles", "loves" : [ "apple" ] }
{ "name" : "Unicrom", "loves" : [ "energon" ] }
```

## 8.2.1

- Создание коллекции `towns`

Код запроса:

```
db.towns.insert({name: "Punxsutawney ", populatiuon: 6200, last_sensus: ISODate("2008-01-31"), famous_for: [""], mayor: {name: "Jim Wehrle" }})
```

```
db.towns.insert({name: "New York", populatiuon: 22200000, last_sensus: ISODate("2009-07-31"), famous_for: ["status of liberty", "food"], mayor: { name: "Michael Bloomberg", party: "I"}})
```

```
db.towns.insert({name: "Portland", populatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous_for: ["beer", "food"], mayor: { name: "Sam Adams", party: "D"}})
```

Результат выполнения запроса

```
> db.towns.insert({name: "Punxsutawney ", population: 6200, last_sensus: ISODate("2008-01-31"), famous_for: [""], mayor: {name: "Jim Wehrle" }})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "New York", population: 22200000, last_sensus: ISODate("2009-07-31"), famous_for: ["status of liberty", "food"], mayor: { name: "Michael Bloomberg", party: "I"}})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "Portland", population: 528000, last_sensus: ISODate("2009-07-20"), famous_for: ["beer", "food"], mayor: { name: "Sam Adams", party: "D"}})
WriteResult({ "nInserted" : 1 })
> db.towns.find()
{ "_id" : ObjectId("64776f63175f9190acd5e0b5"), "name" : "Punxsutawney ", "population" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("64776f63175f9190acd5e0b6"), "name" : "New York", "population" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("64776f63175f9190acd5e0b7"), "name" : "Portland", "population" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
```

- Сформировать запрос, который возвращает список городов с независимыми мэрами. Вывести только название города и информацию о мэре.

Код запроса:

```
db.towns.find({"mayor.party": "I"}, {"name": 1, "_id": 0, "mayor": 1})
```

Результат выполнения запроса

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

- Сформировать запрос, который возвращает список беспартийных мэров. Вывести только название города и информацию о мэре и

Код запроса:

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

Результат выполнения запроса

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

### 8.2.2

- Сформировать функцию для вывода списка самцов единорогов

Код запроса:

```
fn = function(){
  return this.gender == "m"
}
```

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

Код запроса:

```
var cursor = db.unicorns.find(); null;
cursor.limit(2); null;
cursor.sort("name": 1); null;
```

- Сформировать запрос, который возвращает список беспартийных мэров. Вывести только название города и информацию о мэре

Код запроса:

```
cursor.forEach(printjson);
```

Результат выполнения запроса

```
> cursor.forEach(printjson);
{
  "_id" : ObjectId("6477441f175f9190acd5e0a4"),
  "name" : "Horny",
  "loves" : [
    "carrot",
    "papaya"
  ],
  "weight" : 600,
  "gender" : "m",
  "vampires" : 63
}
{
  "_id" : ObjectId("6477441f175f9190acd5e0a5"),
  "name" : "Aurora",
  "loves" : [
    "carrot",
    "grape"
  ],
  "weight" : 450,
  "gender" : "f",
  "vampires" : 43
}
>
```

### 8.2.3

Вывести количество самок единорогов весом от полутонны до 600 кг

Код запроса:

```
db.unicorns.find({"gender": "f","weight": {$gte: 500, $lt: 600}}).count()
```

Результат выполнения запроса

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

### 8.2.4

Вывести список предпочтений

Код запроса:

```
db.unicorns.distinct("loves")
```

Результат выполнения запроса

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

### 8.2.5

Посчитать количество особей единорогов обоих полов

Код запроса:

```
db.unicorns.aggregate({$group: {_id: "$gender", count:{$sum: 1}}})
```

Результат выполнения запроса

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



## 8.2.6

- Выполнить команду

Код запроса:

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

Результат выполнения запроса

```
> db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
```

- Проверить содержимое коллекции unicorns .

Результат выполнения запроса

```
> db.unicorns.find()
{ "_id" : ObjectId("6477441f175f9190acd5e0a4"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a5"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a7"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a8"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a9"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6477441f175f9190acd5e0aa"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ab"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ac"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ad"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ae"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("647744d8175f9190acd5e0af"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("64778502bdb069e3a5e8b56b"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

## 8.2.7

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

Код запроса:

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

Результат выполнения запроса

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

- Проверить содержимое коллекции unicorns

Результат выполнения запроса

```
> db.unicorns.find()
{ "_id" : ObjectId("6477441f175f9190acd5e0a4"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a5"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a8"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a9"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("6477441f175f9190acd5e0aa"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ab"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ac"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ad"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ae"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("647744d8175f9190acd5e0af"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("64778502bdb069e3a5e8b56b"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

## 8.2.8

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

Код запроса:

```
db.unicorns.update({"name": "Raleigh", "gender": "m"}, {$addToSet: {loves: "redbull"}})
```

Результат выполнения запроса

```
> db.unicorns.update({"name": "Raleigh", "gender": "m"}, {$addToSet: {loves: "redbull"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

- Проверить содержимое коллекции unicorns

Результат выполнения запроса

```
> db.unicorns.find()
{ "_id" : ObjectId("6477441f175f9190acd5e0a4"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a5"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a8"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a9"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("6477441f175f9190acd5e0aa"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ab"), "name" : "Raleigh", "loves" : [ "apple", "sugar", "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ac"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ad"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ae"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("647744d8175f9190acd5e0af"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("64778502bdb069e3a5e8b56b"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

## 8.2.9

- Всем самцам единорогов увеличить количество убитых вампиров на 5

Код запроса:

```
db.unicorns.updateMany({"gender": "m"}, {$inc: {"vampires": 5}})
```

Результат выполнения запроса

```
> db.unicorns.updateMany({"gender": "m"}, {$inc: {"vampires": 5}})
{ "acknowledged" : true, "matchedCount" : 8, "modifiedCount" : 8 }
> |
```

- Проверить содержимое коллекции unicorns .

Результат выполнения запроса

```
> db.unicorns.find()
{ "_id" : ObjectId("6477441f175f9190acd5e0a4"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a5"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a8"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6477441f175f9190acd5e0a9"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("6477441f175f9190acd5e0aa"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ab"), "name" : "Raleigh", "loves" : [ "apple", "sugar", "red bull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ac"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ad"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("6477441f175f9190acd5e0ae"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("647744d8175f9190acd5e0af"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("64778502bdb069e3a5e8b56b"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
>
```

## 8.2.10

- Изменить информацию о городе Портланд: мэр этого города теперь беспартийный

Код запроса:

```
db.town.update({"name": "Portland"}, {$unset: {"mayor.party": 1}})
```

Результат выполнения запроса

```
> db.town.update({"name": "Portland"}, {$unset: {"mayor.party": 1}})
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> |
```

- Проверить содержимое коллекции towns .

Результат выполнения запроса

```
> db.towns.find()
{ "_id" : ObjectId("64776f5a175f9190acd5e0b5"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("64776f63175f9190acd5e0b6"), "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("64776f69175f9190acd5e0b7"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
>
```

### 8.2.11

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

Код запроса:

```
db.unicorns.update({"gender": "m", "name": "Pilot"}, {$addToSet: {loves: "chocolate"}})
```

Результат выполнения запроса

```
> db.unicorns.update({"gender": "m", "name": "Pilot"}, {$addToSet: {loves: "chocolate"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> |
```

- Проверить содержимое коллекции unicorns .

Результат выполнения запроса

```
{ "_id" : ObjectId("6477441f175f9190acd5e0ad"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
> |
```

### 8.2.12

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

Код запроса:

```
db.unicorns.update({"gender": "f", "name": "Aurora"}, {$addToSet: {"loves": {$each: ["sugar", "lemon"]}}})
```

Результат выполнения запроса

```
> db.unicorns.update({"gender": "f", "name": "Aurora"}, {$addToSet: {"loves": {$each: ["sugar", "lemon"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
>
```

- Проверить содержимое коллекции unicorns .

Результат выполнения запроса

```
> db.unicorns.find({"name": "Aurora"})
{ "_id" : ObjectId("6477441f175f9190acd5e0a5"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
> |
```

## 8.2.13

- Создание коллекции towns

Код запроса:

```
db.towns.insert({name: "Punxsutawney ", popujatiuon: 6200, last_sensus:
ISODate("2008-01-31"), famous_for: ["phil the groundhog"], mayor: { name: "Jim
Wehrle" }})
```

```
db.towns.insert({name: "New York", popujatiuon: 22200000, last_sensus:
ISODate("2009-07-31"), famous_for: ["status of liberty", "food"], mayor: {name:
"Michael Bloomberg", party: "I"}})
```

```
db.towns.insert({name: "Portland", popujatiuon: 528000, last_sensus:
ISODate("2009-07-20"), famous_for: ["beer", "food"], mayor: {name: "Sam
Adams", party: "D"}})
```

- Удалите документы с беспартийными мэрами

Код запроса:

```
db.towns.deleteMany({"mayor.party": {$exists: 0}})
```

Результат выполнения запроса

```
> db.towns.deleteMany({"mayor.party": {$exists: 0}})
{ "acknowledged" : true, "deletedCount" : 0 }
> |
```

- Проверить содержимое коллекции towns .

Результат выполнения запроса

```
> db.towns.find()
{ "_id" : ObjectId("64779fe036e18d83359f9834"), "name" : "New York", "popujatiuon" : 22200000, "last_s
ensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : {
"name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("64779fe436e18d83359f9835"), "name" : "Portland", "popujatiuon" : 528000, "last_sen
sus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam A
dams", "party" : "D" } }
> |
```

- Очистите коллекцию

Код запроса:

```
db.towns.deleteMany({})
```

Результат выполнения запроса

```
> db.towns.deleteMany({})
{ "acknowledged" : true, "deletedCount" : 2 }
> |
```

- Просмотрите список доступных коллекций.

Код запроса:

```
Show collections
```

Результат выполнения запроса

```
> show collections
towns
>
```

### 8.3.1

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

Код запроса:

```
db.habitatAreas.insert({"_id": "SP", "name": "Space", "description": "Бескрайнее ничего, иногда встречаются каменные шары"})
```

```
db.habitatAreas.insert({"_id": "NF", "name": "North Forest", "description": "Замерзший бескрайний лес и 3 метра снега"})
```

```
db.habitatAreas.insert({"_id": "HL", "name": "Hell", "description": "Жарко, вечно пахнет гарью, и соседи шумные"})
```

```
db.habitatAreas.insert({"_id": "HV", "name": "Haven", "description": "Вроде рай, но соседи все еще шумные"})
```

Результат выполнения запроса

```
> db.habitatAreas.insert({"_id": "SP", "name": "Space", "description": "Бескрайнее ничего, иногда встречаются каменные шары"})
WriteResult({"nInserted" : 1 })
> db.habitatAreas.insert({"_id": "NF", "name": "North Forest", "description": "Замерзший бескрайний лес и 3 метра снега"})
WriteResult({"nInserted" : 1 })
> db.habitatAreas.insert({"_id": "HL", "name": "Hell", "description": "Жарко, вечно пахнет гарью, и соседи шумные"})
WriteResult({"nInserted" : 1 })
> db.habitatAreas.insert({"_id": "HV", "name": "Haven", "description": "Вроде рай, но соседи все еще шумные"})
WriteResult({"nInserted" : 1 })
```

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

Код запроса:

```
db.unicorns.update({"name": "Leia"}, {$set: {habitatAreas: {$ref: "habitatAreas", $id: "SP"}}})
```

```
db.unicorns.update({"name": "Rooodooles"}, {$set: {habitatAreas: {$ref: "habitatAreas", $id: "NF"}}})
```

```
db.unicorns.update({"name": "Horny"}, {$set: {habitatAreas: {$ref: "habitatAreas", $id: "HL"}}})
```

```
db.unicorns.update({"name": "Solnara"}, {$set: {habitatAreas: {$ref: "habitatAreas", $id: "HV"}}})
```

Результат выполнения запроса

```
> db.unicorns.update({"name": "Leia"}, {$set: {habitatAreas: {$ref: "habitatAreas", $id: "SP"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({"name": "Roooooodles"}, {$set: {habitatAreas: {$ref: "habitatAreas", $id: "NF"}}})
uncaught exception: SyntaxError: missing ) after argument list :
@(shell):1:101
> db.unicorns.update({"name": "Roooooodles"}, {$set: {habitatAreas: {$ref: "habitatAreas", $id: "NF"}}})
uncaught exception: SyntaxError: missing ) after argument list :
@(shell):1:101
> db.unicorns.update({"name": "Roooooodles"}, {$set: {habitatAreas: {$ref: "habitatAreas", $id: "NF"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({"name": "Horny"}, {$set: {habitatAreas: {$ref: "habitatAreas", $id: "HL"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({"name": "Solnara"}, {$set: {habitatAreas: {$ref: "habitatAreas", $id: "HV"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

- Проверить содержимое коллекции towns .

Результат выполнения запроса

```
> db.towns.find()
{ "_id" : ObjectId("64779fe036e18d83359f9834"), "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("64779fe436e18d83359f9835"), "name" : "Portland", "popujatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
> |
```

- Проверить содержимое коллекции unicorns .

Результат выполнения запроса

```
> db.unicorns.find()
{ "_id" : ObjectId("64779a1f175f9198acd5e8a5"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 60, "habitatAreas" : DBRef("habitatAreas", "HL") }
{ "_id" : ObjectId("64779a1f175f9198acd5e8a5"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("64779a1f175f9198acd5e8a5"), "name" : "Unicorn", "loves" : [ "energy", "redbull" ], "weight" : 980, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("64779a1f175f9198acd5e8a7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 570, "gender" : "m", "vampires" : 180, "habitatAreas" : DBRef("habitatAreas", "NF") }
{ "_id" : ObjectId("64779a1f175f9198acd5e8a8"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 850, "gender" : "f", "vampires" : 80, "habitatAreas" : DBRef("habitatAreas", "HV") }
{ "_id" : ObjectId("64779a1f175f9198acd5e8a9"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("64779a1f175f9198acd5e8aa"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 40 }
{ "_id" : ObjectId("64779a1f175f9198acd5e8ab"), "name" : "Raleigh", "loves" : [ "apple", "sugar", "redbull" ], "weight" : 820, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("64779a1f175f9198acd5e8ac"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 600, "gender" : "f", "vampires" : 33, "habitatAreas" : DBRef("habitatAreas", "SP") }
{ "_id" : ObjectId("64779a1f175f9198acd5e8ad"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("64779a1f175f9198acd5e8ae"), "name" : "Blaise", "loves" : [ "grape", "carrot" ], "weight" : 500, "gender" : "f" }
{ "_id" : ObjectId("64779a1f175f9198acd5e8af"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 700, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("64779a1f175f9198acd5e8b0"), "name" : "Barvy", "loves" : [ "grape" ], "weight" : 300, "gender" : "m", "vampires" : 5 }
```

### 8.3.2

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

Код запроса:

```
db.unicorns.ensureIndex({"name": 1}, {"unique": true})
```

Результат выполнения запроса

```
learn> db.unicorns.ensureIndex({"name": 1}, {"unique": true})
[ 'name_1' ]
learn> |
```

### 8.3.3

- Получите информацию обо всех индексах коллекции unicorns

Код запроса:

```
db.unicorns.getIndexes()
```

Результат выполнения запроса

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

- Удалите все индексы, кроме индекса для идентификатора

Код запроса:

```
db.unicorns.dropIndex("name_1")
```

Результат выполнения запроса

```
learn> db.unicorns.dropIndex("name_1")
{ nIndexesWas: 2, ok: 1 }
learn> |
```

- Удалите все индексы, кроме индекса для идентификатора

Код запроса:

```
db.unicorns.dropIndex("_id_")
```

Результат выполнения запроса

```
learn> db.unicorns.dropIndex("_id_")
MongoServerError: cannot drop _id index
learn> |
```



### 8.3.4

- Создайте объемную коллекцию numbers, задействовав курсор

Код запроса:

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

Результат выполнения запроса

```
learn> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6477aeaaf94b30b7d39c38eb") }
}
learn>
```

- Выберите последних четыре документа

Код запроса:

```
db.numbers.find({}).skip(99996)
```

Результат выполнения запроса

```
learn> db.numbers.find({}).skip(99996)
[
  { _id: ObjectId("6477aeaaf94b30b7d39c38e8"), value: 99996 },
  { _id: ObjectId("6477aeaaf94b30b7d39c38e9"), value: 99997 },
  { _id: ObjectId("6477aeaaf94b30b7d39c38ea"), value: 99998 },
  { _id: ObjectId("6477aeaaf94b30b7d39c38eb"), value: 99999 }
]
learn> |
```

- Проанализируйте план выполнения запроса

Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)

Код запроса:

```
db.numbers.explain("executionStats").find({}).skip(99996)
```

Результат выполнения запроса

```
executionStats: {
  executionSuccess: true,
  nReturned: 4,
  executionTimeMillis: 35,
  totalKeysExamined: 0,
  totalDocsExamined: 100000,
```

- Создайте индекс для ключа value

Код запроса:

```
db.numbers.ensureIndex({"value": 1}, {"unique": true})
```

Результат выполнения запроса

```
learn> db.numbers.ensureIndex({"value": 1}, {"unique": true})
[ 'value_1' ]
learn> |
```

- Получите информацию обо всех индексах коллекции numbers

Код запроса:

```
db.numbers.getIndexes()
```

Результат выполнения запроса

```
learn> db.numbers.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { value: 1 }, name: 'value_1', unique: true }
]
learn> |
```

- Выберем последние 4 документа

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

Код запроса:

```
db.numbers.explain("executionStats").find({}).skip(99996)
```

Результат выполнения запроса

```
executionStats: {
  executionSuccess: true,
  nReturned: 4,
  executionTimeMillis: 33,
  totalKeysExamined: 0,
  totalDocsExamined: 100000,
  executionStages: {
    $skip: {
      num: 99996
    }
  }
}
```

В целом скорость выполнения запроса с индексом и без в среднем отличается на 5-6 мс. Запросы с индексом работают быстрее, но на незначительное время.

## Выводы:

В ходе лабораторной работы были изучены функции СУБД MongoDB. В том числе возможность создание баз данных и таблиц, работа с запросами на добавление, модификацию и удаление данных, а также индексами.