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

Лабораторная работа №5
«Реализация БД с использованием СУБД MongoDB. Запросы к базе данных»

Выполнили:
Смирнов Тимур Олегович
Группа: K3243

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 5.0.8.

Ход работы:

Практическое задание 8.1.1:

1. Создайте базу данных learn;
2. Заполните коллекцию единорогов unicorns:

Листинг 1 – Создание БД и заполнение коллекции

```
> 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: 'Roooooodles', 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 })
```

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

Листинг 2 – Добавление данных 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 })
```

4. Проверьте содержимое коллекции с помощью метода find:

Листинг 3 – Проверка записей

```
> db.unicorns.find()
{ "_id" : ObjectId("628f69b80eac2cc8d95b807d"), "name" : "Horny", "loves" : [
"carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807e"), "name" : "Aurora", "loves" : [
"carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807f"), "name" : "Unicrom", "loves" : [
"energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8080"), "name" : "Roooooodles", "loves" : [
"apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8081"), "name" : "Solnara", "loves" : [
"apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
```

```
{ "_id" : ObjectId("628f69b80eac2cc8d95b8082"), "name" : "Ayna", "loves" : [
"strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8083"), "name" : "Kenny", "loves" : [
"grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8084"), "name" : "Raleigh", "loves" : [
"apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8085"), "name" : "Leia", "loves" : [ "apple",
"watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8086"), "name" : "Pilot", "loves" : [
"apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8087"), "name" : "Nimue", "loves" : [
"grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("628f6a6a0eac2cc8d95b8088"), "name" : "Dunx", "loves" : [ "grape",
"watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

Практическое задание 8.1.2:

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

Листинг 4 – Списки самцов и самок

```
> db.unicorns.find({"gender": "m"}).sort({"name": 1})
{ "_id" : ObjectId("628f6a6a0eac2cc8d95b8088"), "name" : "Dunx", "loves" : [ "grape",
"watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807d"), "name" : "Horny", "loves" : [
"carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8083"), "name" : "Kenny", "loves" : [
"grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8086"), "name" : "Pilot", "loves" : [
"apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8084"), "name" : "Raleigh", "loves" : [
"apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8080"), "name" : "Roooooodles", "loves" : [
"apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807f"), "name" : "Unicrom", "loves" : [
"energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
> db.unicorns.find({"gender": "f"}).sort({"name": 1}).limit(3)
{ "_id" : ObjectId("628f69b80eac2cc8d95b807e"), "name" : "Aurora", "loves" : [
"carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

```
{ "_id" : ObjectId("628f69b80eac2cc8d95b8082"), "name" : "Ayna", "loves" : [
"strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8085"), "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.

Листинг 5 – Список морковных самок

```
> db.unicorns.findOne({"loves": "carrot"})
{
        "_id" : ObjectId("628f69b80eac2cc8d95b807d"),
        "name" : "Horny",
        "loves" : [
                "carrot",
                "papaya"
        ],
        "weight" : 600,
        "gender" : "m",
        "vampires" : 63
}
> db.unicorns.find({"loves": "carrot"}).limit(1)
{ "_id" : ObjectId("628f69b80eac2cc8d95b807d"), "name" : "Horny", "loves" : [
"carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

Практическое задание 8.1.3:

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

Листинг 6 – Список самцов без пола и предпочтений

```
> db.unicorns.find({"gender":"m"}, {"gender": 0, "loves": 0})
{ "_id" : ObjectId("628f69b80eac2cc8d95b807d"), "name" : "Horny", "weight" : 600,
"vampires" : 63 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807f"), "name" : "Unicrom", "weight" : 984,
"vampires" : 182 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8080"), "name" : "Roooooodles", "weight" :
575, "vampires" : 99 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8083"), "name" : "Kenny", "weight" : 690,
"vampires" : 39 }
```

```
{ "_id" : ObjectId("628f69b90eac2cc8d95b8084"), "name" : "Raleigh", "weight" : 421,
"vampires" : 2 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8086"), "name" : "Pilot", "weight" : 650,
"vampires" : 54 }
{ "_id" : ObjectId("628f6a6a0eac2cc8d95b8088"), "name" : "Dunx", "weight" : 704,
"vampires" : 165 }
```

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

Листинг 7 – Задание 8.1.4

```
> db.unicorns.find().sort({$natural: -1})
{ "_id" : ObjectId("628f6a6a0eac2cc8d95b8088"), "name" : "Dunx", "loves" : [ "grape",
"watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8087"), "name" : "Nimue", "loves" : [
"grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8086"), "name" : "Pilot", "loves" : [
"apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8085"), "name" : "Leia", "loves" : [ "apple",
"watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8084"), "name" : "Raleigh", "loves" : [
"apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8083"), "name" : "Kenny", "loves" : [
"grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8082"), "name" : "Ayna", "loves" : [
"strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8081"), "name" : "Solnara", "loves" : [
"apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8080"), "name" : "Roooooodles", "loves" : [
"apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807f"), "name" : "Unicrom", "loves" : [
"energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807e"), "name" : "Aurora", "loves" : [
"carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807d"), "na8me" : "Horny", "loves" : [
"carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

Практическое задание 8.1.5:

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

Листинг 8 – Задание 8.1.5

```
> 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 }
```

Практическое задание 8.1.6:

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

Листинг 9 – Задание 8.1.6

```
>db.unicorns.find({"gender":"f", "weight": {$gte: 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" }
```

Практическое задание 8.1.7:

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

Листинг 10 – Задание 8.1.7

```
> db.unicorns.find({"gender":"m", "weight": {$gte: 500}, "loves": {$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 }
```

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.

Листинг 11 – Задание 8.1.8

```
> db.unicorns.find({"vampires": {$exists: false}})
{ "_id" : ObjectId("628f69b90eac2cc8d95b8087"), "name" : "Nimue", "loves" : [
"grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

Практическое задание 8.1.9:

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

Листинг 12 – Задание 8.1.9

```
> db.unicorns.find({}, {loves: {$slice: 1}, name:1, _id:0}).sort({name: 1})
{ "name" : "Aurora", "loves" : [ "carrot" ] }
{ "name" : "Ayna", "loves" : [ "strawberry" ] }
{ "name" : "Dunx", "loves" : [ "grape" ] }
{ "name" : "Horny", "loves" : [ "carrot" ] }
{ "name" : "Kenny", "loves" : [ "grape" ] }
{ "name" : "Leia", "loves" : [ "apple" ] }
{ "name" : "Nimue", "loves" : [ "grape" ] }
{ "name" : "Pilot", "loves" : [ "apple" ] }
{ "name" : "Raleigh", "loves" : [ "apple" ] }
{ "name" : "Roooooodles", "loves" : [ "apple" ] }
{ "name" : "Solnara", "loves" : [ "apple" ] }
{ "name" : "Unicrom", "loves" : [ "energon" ] }
```

Практическое задание 8.2.1:

1. Создайте коллекцию towns, включающую следующие документы:

Листинг 13 – Документы коллекции towns

```
{name: "Punxsutawney ",
populatiuon: 6200,
last_sensus: ISODate("2008-01-31"),
famous_for: [""],
mayor: {
   name: "Jim Wehrle"
}},
{name: "New York",
populatiuon: 22200000,
last_sensus: ISODate("2009-07-31"),
famous_for: ["status of liberty", "food"],
mayor: {
   name: "Michael Bloomberg",
party: "I"}},
{name: "Portland",
populatiuon: 528000,
last_sensus: ISODate("2009-07-20"),
famous_for: ["beer", "food"],
mayor: {
   name: "Sam Adams",
party: "D"}}
```

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

Листинг 14 – Задание 8.2.1

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

Практическое задание 8.2.2:

1. Сформировать функцию для вывода списка самцов единорогов.
2. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.
3. Вывести результат, используя forEach.
4. Содержание коллекции единорогов unicorns:

Листинг 15 – Коллекции единорогов

```
db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender:
'm', vampires: 63});
db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender:
'f', vampires: 43});
db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984,
gender: 'm', vampires: 182});
db.unicorns.insert({name: 'Roooooodles', 44), loves: ['apple'], weight: 575, gender:
'm', vampires: 99});
db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'],
weight:550, gender:'f', vampires:80});
db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender:
'f', vampires: 40});
db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690,  gender:
'm', vampires: 39});
db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender:
'm', vampires: 2});
db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601,
gender: 'f', vampires: 33});
db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650,
gender: 'm', vampires: 54});
db.unicorns.insert ({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender:
'f'});
db.unicorns.insert ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704,
gender: 'm', vampires: 165})
```

Листинг 16 – Задание 8.2.2

```
> function manpower() {return this.gender=="m"}
> db.unicorns.find(manpower)
{ "_id" : ObjectId("628f69b80eac2cc8d95b807d"), "name" : "Horny", "loves" : [
"carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

```
{ "_id" : ObjectId("628f69b80eac2cc8d95b807f"), "name" : "Unicrom", "loves" : [
"energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8080"), "name" : "Roooooodles", "loves" : [
"apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8083"), "name" : "Kenny", "loves" : [
"grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8084"), "name" : "Raleigh", "loves" : [
"apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8086"), "name" : "Pilot", "loves" : [
"apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628f6a6a0eac2cc8d95b8088"), "name" : "Dunx", "loves" : [ "grape",
"watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
> var cursor = db.unicorns.find(manpower);null;
null
> cursor.limit(2);null;
null
> cursor.forEach(function(obj){print(obj.name);})
Horny
Unicrom
```

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

Листинг 17 – Задание 8.2.3

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

Практическое задание 8.2.4:

Вывести список предпочтений.

Листинг 18 – Задание 8.2.4

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

```
        "strawberry",
        "sugar",
        "watermelon"
]
```

Практическое задание 8.2.5:

Посчитать количество особей единорогов обоих полов.

Листинг 19 – Задание 8.2.5

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

Практическое задание 8.2.6:

1. Выполнить команду:

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

2. Проверить содержимое коллекции unicorns.

Листинг 20 – Задание 8.2.6

```
> db.unicorns.find()
{ "_id" : ObjectId("628f69b80eac2cc8d95b807d"), "name" : "Horny", "loves" : [
"carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807e"), "name" : "Aurora", "loves" : [
"carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807f"), "name" : "Unicrom", "loves" : [
"energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8080"), "name" : "Roooooodles", "loves" : [
"apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8081"), "name" : "Solnara", "loves" : [
"apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8082"), "name" : "Ayna", "loves" : [
"strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8083"), "name" : "Kenny", "loves" : [
"grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8084"), "name" : "Raleigh", "loves" : [
"apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8085"), "name" : "Leia", "loves" : [ "apple",
"watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8086"), "name" : "Pilot", "loves" : [
"apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
```

```
{ "_id" : ObjectId("628f69b90eac2cc8d95b8087"), "name" : "Nimue", "loves" : [
"grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("628f6a6a0eac2cc8d95b8088"), "name" : "Dunx", "loves" : [ "grape",
"watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628f8d740eac2cc8d95b808f"), "name" : "Barny", "loves" : [ "grape"
], "weight" : 340, "gender" : "m" }
```

Практическое задание 8.2.7:

1. Для самки единорога Ayna внести изменения в БД: теперь ее вес 800, она убила 51 вапмира.
2. Проверить содержимое коллекции unicorns.

Листинг 21 – Задание 8.2.7

```
> db.unicorns.update({name: "Ayna"}, {name: "Ayna", weight: 800, vampires: 51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({"name": "Ayna"})
{ "_id" : ObjectId("628f69b80eac2cc8d95b8082"), "name" : "Ayna", "weight" : 800,
"vampires" : 51 }
```

Практическое задание 8.2.8:

1. Для самца единорога Raleigh внести изменения в БД: теперь он любит рэдбул.
2. Проверить содержимое коллекции unicorns.

Листинг 22 – Задание 8.2.8

```
> db.unicorns.update({name: "Raleigh"}, {$set: {loves: "RedBull"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({"name": "Raleigh"})
{ "_id" : ObjectId("628f69b90eac2cc8d95b8084"), "name" : "Raleigh", "loves" :
"RedBull", "weight" : 421, "gender" : "m", "vampires" : 2 }
```

Практическое задание 8.2.9:

1. Всем самцам единорогов увеличить количество убитых вапмиров на 5.
2. Проверить содержимое коллекции unicorns.

Листинг 23 – Задание 8.2.9

```
> db.unicorns.update({gender: "f"}, {$inc: {vampires: 5}})
```

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("628f69b80eac2cc8d95b807d"), "name" : "Horny", "loves" : [
"carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807e"), "name" : "Aurora", "loves" : [
"carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 48 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b807f"), "name" : "Unicrom", "loves" : [
"energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8080"), "name" : "Roooooodles", "loves" : [
"apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8081"), "name" : "Solnara", "loves" : [
"apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("628f69b80eac2cc8d95b8082"), "name" : "Ayna", "weight" : 800,
"vampires" : 51 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8083"), "name" : "Kenny", "loves" : [
"grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8084"), "name" : "Raleigh", "loves" :
"RedBull", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8085"), "name" : "Leia", "loves" : [ "apple",
"watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8086"), "name" : "Pilot", "loves" : [
"apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628f69b90eac2cc8d95b8087"), "name" : "Nimue", "loves" : [
"grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("628f6a6a0eac2cc8d95b8088"), "name" : "Dunx", "loves" : [ "grape",
"watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628f8d740eac2cc8d95b808f"), "name" : "Barny", "loves" : [ "grape"
], "weight" : 340, "gender" : "m" }
```

Практическое задание 8.2.10:

1. Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.
2. Проверить содержимое коллекции towns.

Листинг 24 – Задание 8.2.10

```
> db.towns.update({name: "Portland"}, {$unset: {mayor: 1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.towns.find({name: "Portland"})
{ "_id" : ObjectId("628f71d2ff32c7ae5c154822"), "name" :
"Portland",
```

```
"populatiuon" : 528000, "last_sensus" : ISODate("2009-07-
20T00:00:00Z"),
"famous_for" : [ "beer", "food" ] }
```

Практическое задание 8.2.11:

1. Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.
2. Проверить содержимое коллекции unicorns.

Листинг 25 – Задание 8.2.11

```
> db.unicorns.update({name: "Pilot"}, {$push: {loves: "chocolate"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Pilot"})
{ "_id" : ObjectId("628f640afc4ad54c59cbd348"), "name" : "Pilot", "loves" : [
"apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires"
: 54 }
```

Практическое задание 8.2.12:

1. Изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и лимоны.
2. Проверить содержимое коллекции unicorns.

Листинг 26 – Задание 8.2.12

```
> db.unicorns.update({name: "Horny"}, {$addToSet: {loves: {$each: ["sugar",
"lemon"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Horny"})
{ "_id" : ObjectId("628f640afc4ad54c59cbd33f"), "name" : "Horny", "loves" : [
"carrot", "papaya", "sugar", "lemon" ], "weight" : 600, "gender" : "m",
"vampires" : 63 }
```

Практическое задание 8.2.13:

1. Создайте коллекцию towns, включающую следующие документы:

Листинг 27 – Документы для 8.2.13

```
{name: "Punxsutawney ",
popujatiuon: 6200,
last_sensus: ISODate("2008-01-31"),
```

```
famous_for: ["phil the groundhog"],
mayor: {
   name: "Jim Wehrle"
}}
{name: "New York",
popujatiuon: 22200000,
last_sensus: ISODate("2009-07-31"),
famous_for: ["status of liberty", "food"],
mayor: {
   name: "Michael Bloomberg",
party: "I"}}
{name: "Portland",
popujatiuon: 528000,
last_sensus: ISODate("2009-07-20"),
famous_for: ["beer", "food"],
mayor: {
   name: "Sam Adams",
party: "D"}}
```

2. Удалите документы с беспартийными мэрами.
3. Проверьте содержание коллекции.
4. Очистите коллекцию.
5. Просмотрите список доступных коллекций.

Листинг 28 – Задание 8.2.13

```
> db.towns.remove({"mayor.party": null}, true)
WriteResult({ "nRemoved" : 1 })
> db.towns.find()
{ "_id" : ObjectId("628f71bbff32c7ae5c154821"), "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("628f71d2ff32c7ae5c154822"), "name" : "Portland",
"populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"),
"famous_for" : [ "beer", "food" ] }
> db.towns.remove({})
WriteResult({ "nRemoved" : 2 })
> show collections
towns
unicorns
```

Практическое задание 8.3.1:

1. Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.
2. Включите для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания.
3. Проверьте содержание коллекции едиорогов.
4. Содержание коллекции единорогов unicorns:

Листинг 29 – Документы для 8.3.1

```
db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender:
'm', vampires: 63});
db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender:
'f', vampires: 43});
db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984,
gender: 'm', vampires: 182});
db.unicorns.insert({name: 'Roooooodles', 44), loves: ['apple'], weight: 575, gender:
'm', vampires: 99});
db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'],
weight:550, gender:'f', vampires:80});
db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender:
'f', vampires: 40});
db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690,  gender:
'm', vampires: 39});
db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender:
'm', vampires: 2});
db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601,
gender: 'f', vampires: 33});
db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650,
gender: 'm', vampires: 54});
db.unicorns.insert ({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender:
'f'});
db.unicorns.insert {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704,
gender: 'm', vampires: 165}
```

Листинг 30 – Задание 8.3.1

```
> db.createCollection("envs")
```

```
{ "ok" : 1 }
> db.envs.insert(_id: "w", name: "water"})
uncaught exception: SyntaxError: missing ) after argument list :
@(shell):1:18
> db.envs.insert({_id: "w", name: "water"})
WriteResult({ "nInserted" : 1 })
> db.envs.insert({_id: "e", name: "earth"})
WriteResult({ "nInserted" : 1 })
> db.envs.find()
{ "_id" : "w", "name" : "water" }
{ "_id" : "e", "name" : "earth" }
> db.unicorns.update({name: "Ayna"}, {$set:{env{$ref:"env", id: "w"}}})
uncaught exception: SyntaxError: missing : after property id :
@(shell):1:45
> db.unicorns.update({name: "Ayna"}, {$set:{env:{$ref:"env", id: "w"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Horny"}, {$set:{env:{$ref:"env", id: "e"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("628f6314fc4ad54c59cbd33e"), "name" : "Aurora", "gender" :
"f", "weight" : 450, "vampires" : 5, "loves" : [ "sugar", "lemon" ] }
{ "_id" : ObjectId("628f640afc4ad54c59cbd33f"), "name" : "Horny", "loves" : [
"carrot", "papaya", "sugar", "lemon" ], "weight" : 600, "gender" : "m",
"vampires" : 63, "env" : { "$ref" : "env", "id" : "e" } }
{ "_id" : ObjectId("628f640afc4ad54c59cbd340"), "name" : "Aurora", "loves" : [
"carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628f640afc4ad54c59cbd341"), "name" : "Unicrom", "loves" : [
"energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628f640afc4ad54c59cbd342"), "name" : "Roooooodles", "loves"
: [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f640afc4ad54c59cbd343"), "name" : "Solnara", "loves" : [
"apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" :
80 }
{ "_id" : ObjectId("628f640afc4ad54c59cbd344"), "weight" : 800, "vampires" : 51
}
{ "_id" : ObjectId("628f640afc4ad54c59cbd345"), "name" : "Kenny", "loves" : [
"grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f640afc4ad54c59cbd346"), "name" : "Raleigh", "loves" :
"RedBull", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f640afc4ad54c59cbd347"), "name" : "Leia", "loves" : [
"apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628f640afc4ad54c59cbd348"), "name" : "Pilot", "loves" : [
```

```
"apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires"
: 54 }
{ "_id" : ObjectId("628f640afc4ad54c59cbd349"), "name" : "Nimue", "loves" : [
"grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("628f64a9fc4ad54c59cbd34a"), "name" : "Dunx", "loves" : [
"grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628f788dff32c7ae5c154823"), "name" : "Barny", "loves" : [
"grape" ], "weight" : 340, "gender" : "m" }
{ "_id" : ObjectId("628f7b35ff32c7ae5c154824"), "name" : "Aurora", "gender" :
"f", "weight" : 450 }
{ "_id" : ObjectId("628f7b5eff32c7ae5c154825"), "name" : "Ayna", "weight" : 800,
"vampires" : 51, "env" : { "$ref" : "env", "id" : "w" } }
```

Практическое задание 8.3.2:

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

Листинг 31 – Документы для 8.3.2

```
db.unicorns.insert({name: 'Horny', dob: new Date(1992,2,13,7,47), loves:
['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});
db.unicorns.insert({name: 'Aurora', dob: new Date(1991, 0, 24, 13, 0), loves:
['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
db.unicorns.insert({name: 'Unicrom', dob: new Date(1973, 1, 9, 22, 10), loves:
['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
db.unicorns.insert({name: 'Roooooodles', dob: new Date(1979, 7, 18, 18, 44), loves:
['apple'], weight: 575, gender: 'm', vampires: 99});
db.unicorns.insert({name: 'Solnara', dob: new Date(1985, 6, 4, 2, 1), loves:['apple',
'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});
db.unicorns.insert({name:'Ayna', dob: new Date(1998, 2, 7, 8, 30), loves:
['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
db.unicorns.insert({name:'Kenny', dob: new Date(1997, 6, 1, 10, 42), loves: ['grape',
'lemon'], weight: 690,  gender: 'm', vampires: 39});
db.unicorns.insert({name: 'Raleigh', dob: new Date(2005, 4, 3, 0, 57), loves:
['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
db.unicorns.insert({name: 'Leia', dob: new Date(2001, 9, 8, 14, 53), loves: ['apple',
'watermelon'], weight: 601, gender: 'f', vampires: 33});
db.unicorns.insert({name: 'Pilot', dob: new Date(1997, 2, 1, 5, 3), loves: ['apple',
'watermelon'], weight: 650, gender: 'm', vampires: 54});
```

```
db.unicorns.insert ({name: 'Nimue', dob: new Date(1999, 11, 20, 16, 15), loves:
['grape', 'carrot'], weight: 540, gender: 'f'});
db.unicorns.insert {name: 'Dunx', dob: new Date(1976, 6, 18, 18, 18), loves:
['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165});
```

Листинг 32 – НЕЛЬЗЯ!

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

Практическое задание 8.3.3:

1. Получите информацию о всех индексах коллекции unicorns .
2. Удалите все индексы, кроме индекса для идентификатора.
3. Попытайтесь удалить индекс для идентификатора.

Листинг 33 – Получается не получается

```
> db.unicorns.getIndexes()
[ { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_" } ]
> db.users.dropIndex("name")
{
"ok" : 0,
"errmsg" : "ns not found learn.users",
"code" : 26,
"codeName" : "NamespaceNotFound"
}
> db.users.dropIndex("key")
{
"ok" : 0,
"errmsg" : "ns not found learn.users",
"code" : 26,
"codeName" : "NamespaceNotFound"
}
> db.users.dropIndex("_id")
{
"ok" : 0,
"errmsg" : "ns not found learn.users",
"code" : 26,
"codeName" : "NamespaceNotFound"
}
```

Практическое задание 8.3.4:

1. Создайте объемную коллекцию numbers, задействовав курсор: for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
2. Выберите последних четыре документа.
3. Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)
4. Создайте индекс для ключа value.
5. Получите информацию о всех индексах коллекции numbres.
6. Выполните запрос 2.
7. Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?
8. Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?

Листинг 34 – Задание 8.3.4

```
> db.createCollection("numbers")
{ "ok" : 1 }
> var curs = db.numbers.find();null;
null
> for(i = 0; i <100000; i++) {db.numbers.insert({value: i})}
WriteResult({ "nInserted" : 1 })
> db.numbers.find()
{ "_id" : ObjectId("628f88ddff32c7ae5c154826"), "value" : 0 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154827"), "value" : 1 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154828"), "value" : 2 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154829"), "value" : 3 }
{ "_id" : ObjectId("628f88ddff32c7ae5c15482a"), "value" : 4 }
{ "_id" : ObjectId("628f88ddff32c7ae5c15482b"), "value" : 5 }
{ "_id" : ObjectId("628f88ddff32c7ae5c15482c"), "value" : 6 }
{ "_id" : ObjectId("628f88ddff32c7ae5c15482d"), "value" : 7 }
{ "_id" : ObjectId("628f88ddff32c7ae5c15482e"), "value" : 8 }
{ "_id" : ObjectId("628f88ddff32c7ae5c15482f"), "value" : 9 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154830"), "value" : 10 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154831"), "value" : 11 }
```

```
{ "_id" : ObjectId("628f88ddff32c7ae5c154832"), "value" : 12 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154833"), "value" : 13 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154834"), "value" : 14 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154835"), "value" : 15 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154836"), "value" : 16 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154837"), "value" : 17 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154838"), "value" : 18 }
{ "_id" : ObjectId("628f88ddff32c7ae5c154839"), "value" : 19 }
Type "it" for more
> db.numbers.find().sort({value: -1}).limit(4)
{ "_id" : ObjectId("628f8901ff32c7ae5c16cec5"), "value" : 99999 }
{ "_id" : ObjectId("628f8901ff32c7ae5c16cec4"), "value" : 99998 }
{ "_id" : ObjectId("628f8901ff32c7ae5c16cec3"), "value" : 99997 }
{ "_id" : ObjectId("628f8901ff32c7ae5c16cec2"), "value" : 99996 }
> db.numbers.getIndexes()
[ { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_" } ]
> db.users.dropIndex("name")
{
"ok" : 0,
"errmsg" : "ns not found learn.users",
"code" : 26,
"codeName" : "NamespaceNotFound"
}
> db.users.dropIndex("key")
{
"ok" : 0,
"errmsg" : "ns not found learn.users",
"code" : 26,
"codeName" : "NamespaceNotFound"
}
> db.users.dropIndex("_id")
{
"ok" : 0,
"errmsg" : "ns not found learn.users",
"code" : 26,
"codeName" : "NamespaceNotFound"
}
> db.numbers.explain("executionStats").find().sort({natural:-1}).limit(4)
{
"explainVersion" : "1",
"queryPlanner" : {
"namespace" : "test.numbers",
```

```
"indexFilterSet" : false,
"parsedQuery" : {

},
"maxIndexedOrSolutionsReached" : false,
"maxIndexedAndSolutionsReached" : false,
"maxScansToExplodeReached" : false,
"winningPlan" : {
"stage" : "EOF"
},
"rejectedPlans" : [ ]
},
"executionStats" : {
"executionSuccess" : true,
"nReturned" : 0,
"executionTimeMillis" : 2
}
> db.numbers.explain("executionStats").find().sort({natural:-1}).limit(4)
{
"explainVersion" : "1",
"queryPlanner" : {
"namespace" : "test.numbers",
"indexFilterSet" : false,
"parsedQuery" : {

},
"maxIndexedOrSolutionsReached" : false,
"maxIndexedAndSolutionsReached" : false,
"maxScansToExplodeReached" : false,
"winningPlan" : {
"stage" : "SORT",
"sortPattern" : {
"natural" : -1
},
"memLimit" : 104857600,
"limitAmount" : 4,
"type" : "simple",
"inputStage" : {
"stage" : "COLLSCAN",
"direction" : "forward"
}
},
```

```
"rejectedPlans" : [ ]
},
"executionStats" : {
"executionSuccess" : true,
"nReturned" : 0,
"executionTimeMillis" : 0
}
```


Вывод:

В данной лабораторной работе была изучена СУБД MongoDB, средства работы с данными.