

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Отчет по лабораторной работе № 5

«Работа с БД в СУБД MongoDB»

по дисциплине:

«Проектирование и реализация баз данных»

Выполнила:

студент II курса ИКТ

группы

K3241

Селезнев Илья Олегович

Проверила:

Говорова Марина Михайловна

Санкт-Петербург 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.

Оборудование: компьютер.

Программное обеспечение: СУБД MongoDB 5.0.8.

Практическое задание и выполнение:

Практическое задание 8.1.1:

1) Создайте базу данных learn:

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

2) Заполните коллекцию 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 })
```

- 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("628f8ee71859c56e0983a3c6"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c7"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c8"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c9"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3ca"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cb"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cc"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cd"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3ce"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cf"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3d0"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("628f90811859c56e0983a3d1"), "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.

```
> db.unicorns.find({gender:'f'}).sort({name: 1}).limit(3)
{ "_id" : ObjectId("628f8ee71859c56e0983a3c7"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cb"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3ce"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
```

```
> db.unicorns.find({gender:'m'}).sort({name: 1})
{ "_id" : ObjectId("628f90811859c56e0983a3d1"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c6"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cc"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cf"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cd"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c9"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c8"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
```

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

```
> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
{
  "_id" : ObjectId("628f8ee71859c56e0983a3c7"),
  "name" : "Aurora",
  "loves" : [
    "carrot",
    "grape"
  ],
  "weight" : 450,
  "gender" : "f",
  "vampires" : 43
}
```

```
> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
{ "_id" : ObjectId("628f8ee71859c56e0983a3c7"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

Практическое задание 8.1.3:

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

```
> db.unicorns.find({gender: "m"}, {loves: 0, gender: 0}).sort({name: 1});
{ "_id" : ObjectId("628f90811859c56e0983a3d1"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c6"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cc"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cf"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cd"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c9"), "name" : "Rooooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c8"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
```


Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({$natural: -1});
{ "_id" : ObjectId("628f90811859c56e0983a3d1"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3d0"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cf"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3ce"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cd"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cc"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cb"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3ca"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c9"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c8"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c7"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c6"), "name" : "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.

```
> 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" : "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 }
```

Практическое задание 8.1.6:

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

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

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

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.

```
> db.unicorns.find({vampires:{$exists:false}});
{ "_id" : ObjectId("628f8ee71859c56e0983a3d0"), "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.

```
> db.unicorns.find({gender: 'm'}, {loves: {$slice:1}}).sort({name:1});
{ "_id" : ObjectId("628f90811859c56e0983a3d1"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c6"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cc"), "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cf"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cd"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c9"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c8"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
```

Практическое задание 8.2.1:

1) Создайте коллекцию towns, включающую следующие документы:


```
> db.towns.insert({name: "Punxsutawney ",
... populatiuon: 6200,
... last_sensus: ISODate("2008-01-31"),
... famous_for: [""],
... mayor: {
... name: "Jim Wehrle"
... }})
WriteResult({ "nInserted" : 1 })
> 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"}})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "Portland",
... populatiuon: 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("628f9a5c1859c56e0983a3d2"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("628f9a5c1859c56e0983a3d3"), "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("628f9a5c1859c56e0983a3d4"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "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.

```
> 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" } }
```

Практическое задание 8.2.2:

- 1) Сформировать функцию для вывода списка самцов единорогов.

```
> fn = function() {return this.gender == "m"}
function() {return this.gender == "m"}
> db.unicorns.find(fn)
{ "_id" : ObjectId("628f8ee71859c56e0983a3c6"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c8"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3c9"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cc"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cd"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628f8ee71859c56e0983a3cf"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628f90811859c56e0983a3d1"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

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

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

- 3) Вывести результат, используя forEach.

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

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

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

Практическое задание 8.2.4:

Вывести список предпочтений.

```
> 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.

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

Практическое задание 8.2.6:

1) Выполнить команду:

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

2) Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: "Barney"})
{ "_id" : ObjectId("628fa1571859c56e0983a3d5"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

Практическое задание 8.2.7:

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

```
> db.unicorns.update({name: "Ayna"}, {name:"Ayna", loves: ['strawberry', 'lemon'], weight: 800, gender: "f", vampires: 51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

2) Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: "Ayna"})
{ "_id" : ObjectId("629134b759044d05ae1a2c6f"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
```

Практическое задание 8.2.8:

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

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

2) Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: "Raleigh"})
{ "_id" : ObjectId("629134b759044d05ae1a2c71"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421, "gender" : "m", "vampires" : 2 }
```

Практическое задание 8.2.9:

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

```
> db.unicorns.update({gender: "m"}, {$inc: {vampires: 5}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

- 2) Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({gender: "m"})
{ "_id" : ObjectId("629134b759044d05ae1a2c6a"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("629134b759044d05ae1a2c6c"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629134b759044d05ae1a2c6d"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629134b759044d05ae1a2c70"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629134b759044d05ae1a2c71"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629134b759044d05ae1a2c73"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629134ce59044d05ae1a2c75"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629137b959044d05ae1a2c76"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

Практическое задание 8.2.10:

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

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

- 2) Проверить содержимое коллекции towns.

```
> db.towns.find()
{ "_id" : ObjectId("628f9a5c1859c56e0983a3d2"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("628f9a5c1859c56e0983a3d3"), "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("628f9a5c1859c56e0983a3d4"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams" } }
```

Практическое задание 8.2.11:

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

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

- 2) Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: "Pilot"})
{ "_id" : ObjectId("629134b759044d05ae1a2c73"), "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.

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

- 2) Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: "Aurora"})
{ "_id" : ObjectId("629134b759044d05ae1a2c6b"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

Практическое задание 8.2.13:

- 1) Создайте коллекцию towns, включающую следующие документы:

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

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

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

- 2) Удалите документы с беспартийными мэрами.

```
> db.towns.remove({"mayor.party": {$exists: false}})
WriteResult({ "nRemoved" : 3 })
```

- 3) Проверьте содержание коллекции.


```
> db.towns.find()
{ "_id" : ObjectId("628f9a5c1859c56e0983a3d3"), "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("629142df59044d05ae1a2c78"), "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("6291431a59044d05ae1a2c79"), "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.

```
> db.towns.remove({})
WriteResult({ "nRemoved" : 3 })
```

5) Просмотрите список доступных коллекций.

```
> show collections
towns
unicorns
> db.towns.find()
```

Практическое задание 8.3.1:

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

```
> db.zones.insert({_id:"GF", name: "Gravity Falls", description: "Gravity Falls is the best cartoon"})
WriteResult({ "nInserted" : 1 })
> db.zones.insert({_id:"SU", name: "Steven Universe", description: " Steven Universe is the second best cartoon"})
WriteResult({ "nInserted" : 1 })
> db.zones.insert({_id:"AL", name: "Alfea", description: " Winx forever in my heart "})
WriteResult({ "nInserted" : 1 })
```

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", $id: "GF"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Aurora"}, {$set: {zone: {$ref:"zones", $id: "SU"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Ayna"}, {$set: {zone: {$ref:"zones", $id: "AL"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

3) Проверьте содержание коллекции единорогов.

```
> db.unicorns.find()
{ "_id" : ObjectId("629134b759044d05ae1a2c6a"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68, "zone" : DBRef("zones", "GF") }
{ "_id" : ObjectId("629134b759044d05ae1a2c6b"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "zone" : DBRef("zones", "SU") }
{ "_id" : ObjectId("629134b759044d05ae1a2c6c"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629134b759044d05ae1a2c6d"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629134b759044d05ae1a2c6e"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629134b759044d05ae1a2c6f"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51, "zone" : DBRef("zones", "AL") }
```

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}
```

Практическое задание 8.3.2:

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

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

- 2) Содержание коллекции единорогов unicorns:

```
db.unicorns.insert({name: 'Horny', dob: new Date(1992,2,13,7,47), loves:
['carrot','papaya'], weight: 600, gender: 'm', vampires: 63}); db.unicorns.insert({name:
'Aurora', dob: new Date(1991, 0, 24, 13, 0), loves: ['carrot', 'grape'], weight: 450, gender: 'f',
vampires: 43});

db.unicorns.insert({name: 'Unicrom', dob: new Date(1973, 1, 9, 22, 10), loves: ['energon',
'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', dob: new Date(1979, 7, 18, 18, 44), loves: ['apple'],
weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', dob: new Date(1985, 6, 4, 2, 1), loves:['apple', 'carrot',
'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', dob: new Date(1998, 2, 7, 8, 30), loves: ['strawberry',
'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', dob: new Date(1997, 6, 1, 10, 42), loves: ['grape', 'lemon'],
weight: 690, gender: 'm', vampires: 39}); db.unicorns.insert({name: 'Raleigh', dob: new
Date(2005, 4, 3, 0, 57), loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
db.unicorns.insert({name: 'Leia', dob: new Date(2001, 9, 8, 14, 53), loves: ['apple',
'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', dob: new Date(1997, 2, 1, 5, 3), loves: ['apple', 'watermelon'],
weight: 650, gender: 'm', vampires: 54}); db.unicorns.insert ({name: 'Nimue', dob: new
Date(1999, 11, 20, 16, 15), loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
db.unicorns.insert {name: 'Dunx', dob: new Date(1976, 6, 18, 18, 18), loves: ['grape',
'watermelon'], weight: 704, gender: 'm', vampires: 165
```

Практическое задание 8.3.3:

- 1) Получите информацию о всех индексах коллекции unicorns .

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


2) Удалите все индексы, кроме индекса для идентификатора.

```
> db.unicorns.dropIndexes()
{
  "nIndexesWas" : 1,
  "msg" : "non-_id indexes dropped for collection",
  "ok" : 1
}
```

3) Попробуйте удалить индекс для идентификатора.

```
> db.unicorns.dropIndex({"_id":1})
{
  "ok" : 0,
  "errmsg" : "cannot drop _id index",
  "code" : 72,
  "codeName" : "InvalidOptions"
}
```

Практическое задание 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("62929c4a3fd518afbd9f7cb7"), "value" : 99999 }
{ "_id" : ObjectId("62929c4a3fd518afbd9f7cb6"), "value" : 99998 }
{ "_id" : ObjectId("62929c4a3fd518afbd9f7cb5"), "value" : 99997 }
{ "_id" : ObjectId("62929c4a3fd518afbd9f7cb4"), "value" : 99996 }
```

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

```

> 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" : "LIMIT",
      "limitAmount" : 4,
      "inputStage" : {
        "stage" : "COLLSCAN",
        "direction" : "backward"
      }
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 4,
    "executionTimeMillis" : 3,
  }
}

```

4) Создайте индекс для ключа value.

```

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

```

5) Получите информацию о всех индексах коллекции numbers.

```

> db.numbers.getIndexes()
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_"
  },
  {
    "v" : 2,
    "key" : {
      "value" : 1
    },
    "name" : "value_1"
  }
]

```

6) Выполните запрос 2.

```
> db.numbers.find().sort({ $natural: -1 }).limit(4)
{ "_id" : ObjectId("62929c4a3fd518afbd9f7cb7"), "value" : 99999 }
{ "_id" : ObjectId("62929c4a3fd518afbd9f7cb6"), "value" : 99998 }
{ "_id" : ObjectId("62929c4a3fd518afbd9f7cb5"), "value" : 99997 }
{ "_id" : ObjectId("62929c4a3fd518afbd9f7cb4"), "value" : 99996 }
```

7) Проанализируйте план выполнения запроса с установленным индексом.

Сколько потребовалось времени на выполнение запроса?

```
> 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" : "LIMIT",
      "limitAmount" : 4,
      "inputStage" : {
        "stage" : "COLLSCAN",
        "direction" : "backward"
      }
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 4,
    "executionTimeMillis" : 0,
```

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

С индексами выполнение запросов оказалось быстрее на 3 мс, чем без индексов. Поэтому можно утверждать, что с индексами запрос будет работать эффективнее.

Выводы:

В ходе лабораторной работы были изучены и поняты методы работы с CRUD-операциями, изучена работа с вложенными объектами MongoDB, изучена работа с ссылками и индексами MongoDB