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
по дисциплине:
«Проектирование и реализация баз данных»

Выполнил:
студент 2 курса ИКТ
группы К3241
Попов Ньургун

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.

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: '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});
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'});
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.*

```
{ "_id" : ObjectId("629d092a0173a91f7e8f1352"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1353"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1354"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1355"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1356"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1357"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1358"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1359"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135a"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135b"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135c"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629d0c270173a91f7e8f1360"), "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:'m'}).sort({name: 1})
{ "_id" : ObjectId("629d0c270173a91f7e8f1360"), "name" : "Dunx", "loves" : [ "grape", "watermelon
" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629d092a0173a91f7e8f1352"), "name" : "Horny", "loves" : [ "carrot", "papaya"
], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1358"), "name" : "Kenny", "loves" : [ "grape", "lemon" ],
 "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135b"), "name" : "Pilot", "loves" : [ "apple", "watermelo
n" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1359"), "name" : "Raleigh", "loves" : [ "apple", "sugar"
], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1355"), "name" : "Roooooodles", "loves" : [ "apple" ], "w
eight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1354"), "name" : "Unicrom", "loves" : [ "energon", "redbu
ll" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
```

```
> db.unicorns.find({gender:'f'}).sort({name: 1}).limit(3)
{ "_id" : ObjectId("629d092b0173a91f7e8f1353"), "name" : "Aurora", "loves" : [ "carrot", "grape"
], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1357"), "name" : "Ayna", "loves" : [ "strawberry", "lemon
" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135a"), "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.*

```
> db.unicorns.find({gender:'f', loves:'carrot'})
{ "_id" : ObjectId("629d092b0173a91f7e8f1353"), "name" : "Aurora", "loves" : [ "carrot", "grape"
], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1356"), "name" : "Solnara", "loves" : [ "apple", "carrot"
, "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135c"), "name" : "Nimue", "loves" : [ "grape", "carrot" ]
, "weight" : 540, "gender" : "f" }
```

```
> db.unicorns.findOne({gender:'f', loves:'carrot'})
{
        "_id" : ObjectId("629d092b0173a91f7e8f1353"),
        "name" : "Aurora",
        "loves" : [
                "carrot",
                "grape"
        ],
        "weight" : 450,
        "gender" : "f",
        "vampires" : 43
}
```

```
> db.unicorns.find({gender:'f', loves:'carrot'}).limit(1)
{ "_id" : ObjectId("629d092b0173a91f7e8f1353"), "name" : "Aurora", "loves" : [ "carrot", "grape"
], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

Практическое задание 8.1.3:

1. *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({$natural:-1})
{ "_id" : ObjectId("629d0c270173a91f7e8f1360"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135b"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1359"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1358"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1355"), "name" : "Roooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1354"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
{ "_id" : ObjectId("629d092a0173a91f7e8f1352"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
```

Практическое задание 8.1.4:

1. *Вывести список единорогов в обратном порядке добавления.*

```
> db.unicorns.find().sort({$natural:-1})
{ "_id" : ObjectId("629d0c270173a91f7e8f1360"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weigh
t" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135c"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight"
: 540, "gender" : "f" }
{ "_id" : ObjectId("629d092b0173a91f7e8f135b"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weig
ht" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135a"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weigh
t" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1359"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight"
 : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1358"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" :
 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1357"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weigh
t" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1356"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolat
e" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1355"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 57
5, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1354"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "wei
ght" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1353"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight"
 : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629d092a0173a91f7e8f1352"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight"
 : 600, "gender" : "m", "vampires" : 63 }
```

Практическое задание 8.1.5:

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

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

Практическое задание 8.1.6:

1. *Вывести список самок единорогов весом от полтонны 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", "vampire
s" : 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:

1. *Вывести список самцов единорогов весом от полтонны 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:

1. *Найти всех единорогов, не имеющих ключ vampires.*

```
> db.unicorns.find({vampires:{$exists:false}})
{ "_id" : ObjectId("629d092b0173a91f7e8f135c"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "
weight" : 540, "gender" : "f" }
```

Практическое задание 8.1.9:

1. *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("629d0c270173a91f7e8f1360"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 7
04, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629d092a0173a91f7e8f1352"), "name" : "Horny", "loves" : [ "carrot" ], "weight" :
 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1358"), "name" : "Kenny", "loves" : [ "grape" ], "weight" :
690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135b"), "name" : "Pilot", "loves" : [ "apple" ], "weight" :
650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1359"), "name" : "Raleigh", "loves" : [ "apple" ], "weight"
: 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1355"), "name" : "Roooooodles", "loves" : [ "apple" ], "weig
ht" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1354"), "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"), fam
ous_for: [""], mayor: {name: "Jim Wehrle"}})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "New York", populatiuon: 22200000, last_sensus: ISODate("2009-07-31"), famo
us_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 })
```

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.*

```
> fun_male = function() {return this.gender == 'm'}
function() {return this.gender == 'm'}
> db.unicorns.find(fun_male)
{ "_id" : ObjectId("629d092a0173a91f7e8f1352"), "name" : "Horny", "loves" : [ "carrot", "papaya" ],
"weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1354"), "name" : "Unicrom", "loves" : [ "energon", "redbull"
 ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1355"), "name" : "Roooooodles", "loves" : [ "apple" ], "weig
ht" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1358"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "w
eight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1359"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ],
"weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135b"), "name" : "Pilot", "loves" : [ "apple", "watermelon"
], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629d0c270173a91f7e8f1360"), "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(fun_male) {print (fun_male.name)})
Dunx
Horny
```

4. *Содержание коллекции единорогов unicorns:*

```
> db.unicorns.find()
{ "_id" : ObjectId("629d092a0173a91f7e8f1352"), "name" : "Horny", "loves" : [ "carrot", "papaya" ],
"weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1353"), "name" : "Aurora", "loves" : [ "carrot", "grape" ],
"weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1354"), "name" : "Unicrom", "loves" : [ "energon", "redbull"
 ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1355"), "name" : "Roooooodles", "loves" : [ "apple" ], "weig
ht" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1356"), "name" : "Solnara", "loves" : [ "apple", "carrot", "
chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1357"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ]
, "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1358"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "w
eight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1359"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ],
"weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135a"), "name" : "Leia", "loves" : [ "apple", "watermelon" ]
, "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135b"), "name" : "Pilot", "loves" : [ "apple", "watermelon"
], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135c"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "
weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629d0c270173a91f7e8f1360"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ]
, "weight" : 704, "gender" : "m", "vampires" : 165 }
```

Практическое задание 8.2.3:

1. *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:

1. *Вывести список предпочтений.*

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

Практическое задание 8.2.5:

1. *Посчитать количество особей единорогов обоих полов.*

```
> 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'})
WriteResult({ "nInserted" : 1 })
```

2. *Проверить содержимое коллекции unicorns.*

```
> db.unicorns.find({name:'Barny'})
{ "_id" : ObjectId("629d256a0173a91f7e8f136b"), "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 вампира.*

```
> 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("629d092b0173a91f7e8f1357"), "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("629d092b0173a91f7e8f1359"), "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.*

```
> 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("629d092a0173a91f7e8f1352"), "name" : "Horny", "loves" : [ "carrot", "papaya" ],
"weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1354"), "name" : "Unicrom", "loves" : [ "energon", "redbull"
 ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1355"), "name" : "Roooooodles", "loves" : [ "apple" ], "weig
ht" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1358"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "w
eight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1359"), "name" : "Raleigh", "loves" : "redbull", "weight" :
421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629d092b0173a91f7e8f135b"), "name" : "Pilot", "loves" : [ "apple", "watermelon"
], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629d0c270173a91f7e8f1360"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ]
, "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629d256a0173a91f7e8f136b"), "name" : "Barny", "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({name:'Portland'})
{ "_id" : ObjectId("629d21470173a91f7e8f136a"), "name" : "Portland", "populatiuon" : 528000, "last_s
ensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "S
am Adams" } }
```

Практическое задание 8.2.11:

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

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

2. *Проверить содержимое коллекции unicorns.*

```
> db.unicorns.find({name:'Pilot'})
{ "_id" : ObjectId("629d092b0173a91f7e8f135b"), "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.*

```
{ "_id" : ObjectId("629d092b0173a91f7e8f1353"), "name" : "Aurora", "loves" : [ "carrot", "grape", "s
ugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

Практическое задание 8.2.13:

1. *Создайте коллекцию towns, включающую следующие документы:*

```
> db.towns.insert({name: "Punxsutawney ", popujatiuon: 6200, last_sensus: ISODate("2008-01-31"), fam
ous_for: ["phil the groundhog"], mayor: {name: "Jim Wehrle"}})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "New York", popujatiuon: 22200000, last_sensus: ISODate("2009-07-31"), famo
us_for: ["status of liberty", "food"], mayor: {name: "Michael Bloomberg", party: "I"}})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "Portland", popujatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous
_for: ["beer", "food"], mayor: {name: "Sam Adams", party: "D"}})
WriteResult({ "nInserted" : 1 })
```

2. *Удалите документы с беспартийными мэрами.*

```
> db.towns.remove({'mayor.party':{$exists:0}})
WriteResult({ "nRemoved" : 1 })
```

3. *Проверьте содержание коллекции.*

```
> db.towns.find()
{ "_id" : ObjectId("629d2b9d0173a91f7e8f136d"), "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("629d2bbe0173a91f7e8f136e"), "name" : "Portland", "popujatiuon" : 528000, "last_s
ensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "S
am Adams", "party" : "D" } }
```

4. *Очистите коллекцию.*

```
> db.towns.remove({})
WriteResult({ "nRemoved" : 2 })
```

```
> db.towns.find()
>
```

5. *Просмотрите список доступных коллекций.*

```
> show collections
towns
unicorns
users
>
```

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.habitat.insert({_id:'HF', name:'Hell', type:'fire'})
WriteResult({ "nInserted" : 1 })
> db.habitat.insert({_id:'EE', name:'Earth', type:'earth'})
WriteResult({ "nInserted" : 1 })
> db.habitat.insert({_id:'PA', name:'Paradise', type:'air'})
WriteResult({ "nInserted" : 1 })
> db.habitat.insert({_id:'OW', name:'Ocean', type:'water'})
WriteResult({ "nInserted" : 1 })
```

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

```
> db.unicorns.update({name:'Horny'}, {$set:{habitate:{$ref:'habitat', $id:'HF'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name:'Aurora'}, {$set:{habitate:{$ref:'habitat', $id:'EE'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name:'Unicorm'}, {$set:{habitate:{$ref:'habitat', $id:'PA'}}})
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> db.unicorns.update({name:'Unicrom'}, {$set:{habitate:{$ref:'habitat', $id:'PA'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name:'Roooooodles'}, {$set:{habitate:{$ref:'habitat', $id:'OW'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

3. *Проверьте содержание коллекции единорогов.*

```
> db.unicorns.find()
{ "_id" : ObjectId("629d092a0173a91f7e8f1352"), "name" : "Horny", "loves" : [ "carrot", "papaya" ],
"weight" : 600, "gender" : "m", "vampires" : 68, "habitate" : DBRef("habitat", "HF") }
{ "_id" : ObjectId("629d092b0173a91f7e8f1353"), "name" : "Aurora", "loves" : [ "carrot", "grape", "s
ugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "habitate" : DBRef("habitat", "EE
") }
{ "_id" : ObjectId("629d092b0173a91f7e8f1354"), "name" : "Unicrom", "loves" : [ "energon", "redbull"
 ], "weight" : 984, "gender" : "m", "vampires" : 182, "habitate" : DBRef("habitat", "PA") }
{ "_id" : ObjectId("629d092b0173a91f7e8f1355"), "name" : "Roooooodles", "loves" : [ "apple" ], "weig
ht" : 575, "gender" : "m", "vampires" : 99, "habitate" : DBRef("habitat", "OW") }
{ "_id" : ObjectId("629d092b0173a91f7e8f1356"), "name" : "Solnara", "loves" : [ "apple", "carrot", "
chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629d092b0173a91f7e8f1357"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ]
```

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
```

Практическое задание 8.3.3:

1. *Получите информацию обо 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. *Попытайтесь 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:*

```
> 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("629d33cf0173a91f7e909a0e"), "value" : 99999 }
{ "_id" : ObjectId("629d33cf0173a91f7e909a0d"), "value" : 99998 }
{ "_id" : ObjectId("629d33cf0173a91f7e909a0c"), "value" : 99997 }
{ "_id" : ObjectId("629d33cf0173a91f7e909a0b"), "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" : "learn.numbers",
                "indexFilterSet" : false,
                "parsedQuery" : {

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

4. *Создайте индекс для ключа value.*

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

5. *Получите информацию обо всех индексах коллекции numbres.*

```
> 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("629d33cf0173a91f7e909a0e"), "value" : 99999 }
{ "_id" : ObjectId("629d33cf0173a91f7e909a0d"), "value" : 99998 }
{ "_id" : ObjectId("629d33cf0173a91f7e909a0c"), "value" : 99997 }
{ "_id" : ObjectId("629d33cf0173a91f7e909a0b"), "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" : "learn.numbers",
                "indexFilterSet" : false,
                "parsedQuery" : {

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

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

Они у меня одинаковые, но могу предположить, что с индексом быстрее.

Выводы:

В результате выполненной работы:

- были выполнены CRUD-операции с вложенными объектами в коллекции базы данных MongoDB;
- и еще много другого.