Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Лабораторная работа №5.2

НАЧАЛО РАБОТЫ С БД В СУБД MONGODB

По дисциплине: «Проектирование и реализация баз данных»

Выполнила:

студентка II курса ИКТ группы К3241

Коник Анастасия Александровна

Проверила:

Говорова Марина Михайловна

Санкт-Петербург

22.05.22

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.

Практическое задание 8.1.1:

1. *Создайте базу данных learn.*
2. *Заполните коллекцию 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: 'Roooooodles', 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'});
```

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

```
document = ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})
db.unicorns.insert(document)
```

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

```
> db.unicorns.find()
{ "_id" : ObjectId("628a8178470c78154d9d7bfb"), "name" : "Horny", "loves" :
ght" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628a8182470c78154d9d7bfc"), "name" : "Aurora", "loves" :
ght" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628a8182470c78154d9d7bfd"), "name" : "Unicrom", "loves"
"weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628a8182470c78154d9d7bfe"), "name" : "Roooooodles", "lov
: 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628a8182470c78154d9d7bff"), "name" : "Solnara", "loves"
olate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("628a8182470c78154d9d7c00"), "name" : "Ayna", "loves" : [
eight" : 733, "gender" : "f", "vampires" : 40 }
```

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("628a8182470c78154d9d7bfc"), "name" : "Aurora", "loves" : [ "carrot", "gra
ght" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628a8182470c78154d9d7c00"), "name" : "Ayna", "loves" : [ "strawberry", "
eight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628a8182470c78154d9d7c03"), "name" : "Leia", "loves" : [ "apple", "waterm
eight" : 601, "gender" : "f", "vampires" : 33 }
> db.unicorns.find({gender:'m'}).sort({name: 1})
{ "_id" : ObjectId("628a84be470c78154d9d7c06"), "name" : "Dunx", "loves" : [ "grape", "waterm
eight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628a8178470c78154d9d7bfb"), "name" : "Horny", "loves" : [ "carrot", "papa
ght" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628a8182470c78154d9d7c01"), "name" : "Kenny", "loves" : [ "grape", "lemon
t" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628a8182470c78154d9d7c04"), "name" : "Pilot", "loves" : [ "apple", "water
```

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("628a8182470c78154d9d7bfc"),
        "name" : "Aurora",
        "loves" : [
                "carrot",
                "grape"
        ],
        "weight" : 450,
        "gender" : "f",
        "vampires" : 43
}
> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
{ "_id" : ObjectId("628a8182470c78154d9d7bfc"), "name" : "Aurora", "loves" : [ "carrot", "grape" ],
ght" : 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("628a84be470c78154d9d7c06"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("628a8178470c78154d9d7bfb"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("628a8182470c78154d9d7c01"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("628a8182470c78154d9d7c04"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("628a8182470c78154d9d7c02"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("628a8182470c78154d9d7bfe"), "name" : "Roooooodles", "weight" : 575, "vampires" : 9
}
{ "_id" : ObjectId("628a8182470c78154d9d7bfd"), "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("628a84be470c78154d9d7c06"), "name" : "Dunx", "loves" :
eight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628a8182470c78154d9d7c05"), "name" : "Nimue", "loves" :
ht" : 540, "gender" : "f" }
{ "_id" : ObjectId("628a8182470c78154d9d7c04"), "name" : "Pilot", "loves" :
weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628a8182470c78154d9d7c03"), "name" : "Leia", "loves" :
eight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628a8182470c78154d9d7c02"), "name" : "Raleigh", "loves"
ght" : 421, "gender" : "m", "vampires" : 2 }
```

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" : "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 }
```

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("628a8182470c78154d9d7c05"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "
er" : "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("628a84be470c78154d9d7c06"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m",
"vampires" : 165 }
{ "_id" : ObjectId("628a8178470c78154d9d7bfb"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m
, "vampires" : 63 }
{ "_id" : ObjectId("628a8182470c78154d9d7c01"), "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m"
 "vampires" : 39 }
{ "_id" : ObjectId("628a8182470c78154d9d7c04"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m"
```

Практическое задание 8.2.1:

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

{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.*

```
> 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("628a8178470c78154d9d7bfb"), "name" : "Horny", "loves" :
der" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628a8182470c78154d9d7bfd"), "name" : "Unicrom", "loves"
"gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628a8182470c78154d9d7bfe"), "name" : "Roooooodles", "love
: "m", "vampires" : 99 }
{ "_id" : ObjectId("628a8182470c78154d9d7c01"), "name" : "Kenny", "loves" :
```

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

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

3. *Вывести результат, используя forEach.*

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

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: '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})

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()
4
```

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" : "f", "count" : 10 }
{ "_id" : "m", "count" : 13 }
```

Практическое задание 8.2.6:

1. *Выполнить команду:*

> db.unicorns.save({name: 'Barny', loves: ['grape'],

weight: 340, gender: 'm'})

2. *Проверить содержимое коллекции unicorns.*

```
> db.unicorns.find().sort({name: 1})
{ "_id" : ObjectId("628a8182470c78154d9d7bfc"), "name" : "Aurora",
der" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628a8fe623937b321fc1f540"), "name" : "Aurora",
der" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628a8182470c78154d9d7c00"), "name" : "Ayna", "
ender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628a8fe623937b321fc1f543"), "name" : "Ayna", "
ender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628aaebc23937b321fc1f54a"), "name" : "Barny", "
```

Практическое задание 8.2.7:

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

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

2. *Проверить содержимое коллекции unicorns.*

```
> db.unicorns.find({name: "Ayna"})
{ "_id" : ObjectId("6287fa7daf33bf760f5a4b12"), "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.*
2. *Проверить содержимое коллекции unicorns.*

```
> db.unicorns.update({name: "Raleigh", gender: "m"}, {$set: {loves: "redbull"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Raleigh"})
{ "_id" : ObjectId("628a8182470c78154d9d7c02"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421,
 "vampires" : 2 }
```

Практическое задание 8.2.9:

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

```
> db.unicorns.update({gender: "m"}, {$inc: {vampires: 5}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({gender: "m"})
{ "_id" : ObjectId("628a8178470c78154d9d7bfb"), "name" : "Horny", "lov
der" : "m", "vampires" : 68 }
{ "_id" : ObjectId("628a8182470c78154d9d7bfd"), "name" : "Unicrom", "l
"gender" : "m", "vampires" : 182 }
```

Практическое задание 8.2.10:

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

```
 db.towns.update({name: "Portland"}, {$unset: {"mayor.party": "D"}})
riteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
 db.towns.find({name: "Portland"})
 "_id" : ObjectId("628a8d9623937b321fc1f53e"), "name" : "Portland", "populatiuon" : 528000,
9-07-20T00:00:00Z"), "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.*
2. *Проверить содержимое коллекции unicorns.*

```
> db.unicorns.update({name: "Pilot", gender: "m"}, {$push: {loves: "chocolate"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name:"Pilot"})
{ "_id" : ObjectId("628a8182470c78154d9d7c04"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ],
ght" : 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.*

```
> db.unicorns.update({name: "Aurora", gender: "f"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
> db.unicorns.find({name: "Aurora"})
{ "_id" : ObjectId("628a8182470c78154d9d7bfc"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ],
eight" : 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.*

3. *Проверьте содержание коллекции.*

4. *Очистите коллекцию.*

5. *Просмотрите список доступных коллекций.*

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

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

3. *Проверьте содержание коллекции единорогов.*

```
> db.zones.insert({_id:"64", short:"sar", full: "Saratov", descr: "hometown"})
WriteResult({ "nInserted" : 1 })
> db.zones.insert({_id:"78", short:"spb", full: "Saint-Petersburg", descr: "nice city"})
WriteResult({ "nInserted" : 1 })
> db.unicorns.update({name: "Horny"}, {$set: {"city": {$ref:"zones", $id: "64"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Aurora"}, {$set: {"city": {$ref:"zones", $id: "78"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("628a8178470c78154d9d7bfb"), "name" : "Horny", "loves" : [ "carrot", "papaya" ]
der" : "m", "vampires" : 68, "city" : DBRef("zones", "64") }
{ "_id" : ObjectId("628a8182470c78154d9d7bfc"), "name" : "Aurora", "loves" : [ "carrot", "grape",
eight" : 450, "gender" : "f", "vampires" : 43, "city" : DBRef("zones", "78") }
```

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: 'Roooooodles', 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. *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:*

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.*

```
> var cursor=db.numbers.find().sort({ $natural: -1 }).limit(4)
```

```
> db.numbers.find().sort({ $natural: -1 }).limit(4)
{ "_id" : ObjectId("628bc4ab7bdfabac509cd11f"), "value" : 99999 }
{ "_id" : ObjectId("628bc4ab7bdfabac509cd11e"), "value" : 99998 }
{ "_id" : ObjectId("628bc4ab7bdfabac509cd11d"), "value" : 99997 }
{ "_id" : ObjectId("628bc4ab7bdfabac509cd11c"), "value" : 99996 }
```

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

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

```
"executionStats" : {
        "executionSuccess" : true,
        "nReturned" : 4,
        "executionTimeMillis" : 7,
```

(без курсора)

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 numbres.*

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

6. *Выполните запрос 2.*

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

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

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

Время выполнения запросов с индексом и без одинаковое (0 мс), если задействован курсор. Без курсора время больше (7 мс). С курсором невозможно оценить оптимизацию индексов по времени. На больших объемах данных разница должна быть заметнее (с индексами быстрее).

Вывод: в ходе работы были получены практические навыки работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.