

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Лабораторная работа № 5.2
«Работа с БД в СУБД MongoDB »

Выполнил: Галиновский Роман Андреевич
Группа: К3240
Преподаватель: 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: компьютерный класс.

Программное обеспечение: СУБД MongoDB 5.08

Выполнение:

1. Практическое задание 8.1.1:

1.1 Создайте базу данных *learn*

1.2 Заполните таблицу единорогов *unicorns*

1.3. Используя второй способ, вставьте в коллекцию единорогов документ.

1.4. Проверьте содержимое коллекции с помощью метода *find*:

```
> db.unicorns.find()
{ "_id" : ObjectId("628e8dc73de3ad2a6d139ae8"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628e8dce3de3ad2a6d139ae9"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628e8dd83de3ad2a6d139aea"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628e8dde3de3ad2a6d139aeb"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628e8de43de3ad2a6d139aec"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("628e8de73de3ad2a6d139aed"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628e8ded3de3ad2a6d139aee"), "name" : "Kennny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628e8df13de3ad2a6d139aef"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628e8df83de3ad2a6d139af0"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628e8dfd3de3ad2a6d139af1"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628e8e023de3ad2a6d139af2"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("628e8e683de3ad2a6d139af3"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>
```

2. Практическое задание 8.1.2:

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

1)Самки:

```
> db.unicorns.find({gender: 'f'}).sort({name:1}).limit(3)
{ "_id" : ObjectId("628e8dce3de3ad2a6d139ae9"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628e8de73de3ad2a6d139aed"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628e8df83de3ad2a6d139af0"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
>
```

2)Самцы

```
> db.unicorns.find({gender: 'm'}).sort({name:1})
{ "_id" : ObjectId("628e8dce3de3ad2a6d139ae9"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628e8de73de3ad2a6d139aed"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628e8df83de3ad2a6d139af0"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628e8e023de3ad2a6d139af2"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("628e8de43de3ad2a6d139aec"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
```

2.2. 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("628e8dce3de3ad2a6d139ae9"),
  "name" : "Aurora",
  "loves" : [
    "carrot",
    "grape"
  ],
  "weight" : 450,
  "gender" : "f",
  "vampires" : 43
}

> db.unicorns.find({gender: 'f', loves:"carrot"}).limit(1)
{ "_id" : ObjectId("628e8dce3de3ad2a6d139ae9"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

3. Практическое задание 8.1.3:

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

```
> db.unicorns.find({gender: "f"}, {loves: 0, gender: 0}).sort({name: 1})
{ "_id" : ObjectId("628e8dce3de3ad2a6d139ae9"), "name" : "Aurora", "weight" : 450, "vampires" : 43 }
{ "_id" : ObjectId("628e8de73de3ad2a6d139aed"), "name" : "Ayna", "weight" : 733, "vampires" : 40 }
{ "_id" : ObjectId("628e8df83de3ad2a6d139af0"), "name" : "Leia", "weight" : 601, "vampires" : 33 }
{ "_id" : ObjectId("628e8e023de3ad2a6d139af2"), "name" : "Nimue", "weight" : 540 }
{ "_id" : ObjectId("628e8de43de3ad2a6d139aec"), "name" : "Solnara", "weight" : 550, "vampires" : 80 }
```

4. Практическое задание 8.1.4:

4.1. Вывести список единорогов в обратном порядке добавления:

```
> db.unicorns.find().sort({$natural: -1})
{ "_id" : ObjectId("628e8e683de3ad2a6d139af3"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628e8e023de3ad2a6d139af2"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("628e8dfd3de3ad2a6d139af1"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628e8df83de3ad2a6d139af0"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("628e8df13de3ad2a6d139aef"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
```

5. Практическое задание 8.1.5:

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" : "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 }
```


6. **Практическое задание 8.1.6:**

6.1. 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" }
```

7. **Практическое задание 8.1.7:**

7.1. 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 }
```

8. **Практическое задание 8.1.8:**

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

```
{ "_id" : ObjectId("628e8e023de3ad2a6d139af2"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

9. **Практическое задание 8.1.9:**

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("628e8e683de3ad2a6d139af3"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("628e8dc73de3ad2a6d139ae8"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628e8ded3de3ad2a6d139aee"), "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("628e8dfd3de3ad2a6d139af1"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628e8df13de3ad2a6d139aef"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("628e8dde3de3ad2a6d139aeb"), "name" : "Rooodoodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628e8dd83de3ad2a6d139aea"), "name" : "Unicrom", "loves" : [ "energion" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
```

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

9.3. 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" } }
```

9.4. 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" } }
```

10. Практическое задание 8.2.2.

10.1. Сформировать функцию для вывода списка самцов единорогов.

```
> fn = function() {return this.gender=="m"}
function() {return this.gender=="m"}
> db.unicorns.find(fn)
{ "_id" : ObjectId("628e8dc73de3ad2a6d139ae8"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("628e8dd83de3ad2a6d139aea"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628e8dde3de3ad2a6d139aeb"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("628e8ded3de3ad2a6d139aee"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 450, "gender" : "m", "vampires" : 120 }
```

10.2. Создать курсор для этого списка из первых двух особей с

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

10.3. Вывести результат, используя forEach.

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

11. Практическое задание 8.2.3.

11.1. Вывести количество самок единорогов весом от полутонны до 600 кг.

```
> db.unicorns.find({gender: "f", weight: {$gt: 500, $lt: 600}}).count()
4
>
```

12. Практическое задание 8.2.4.

12.1. Вывести список предпочтений.

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

13. Практическое задание 8.2.5.

13.1. Посчитать количество особей единорогов обоих полов.

```
> db.unicorns.aggregate({"$group":{_id:"$gender", count:{$sum:1}}})
{ "_id" : "m", "count" : 15 }
{ "_id" : "f", "count" : 9 }
>
```

14. Практическое задание 8.2.6.

14.1. Выполнить команду

14.2. Проверить содержимое коллекции *unic*.

```
> db.unicorns.find().sort({name: 1})
{ "_id" : ObjectId("628e8dce3de3ad2a6d139ae9"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("628e8de73de3ad2a6d139aed"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628ea0b83de3ad2a6d139afc"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("628ea27b3de3ad2a6d139b03"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 346, "gender" : "m" }
{ "_id" : ObjectId("628e8e683de3ad2a6d139af3"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

15. Практическое задание 8.2.7.

15.1. Для самки единорога Ауна 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 })
>
```

15.6. Проверить содержимое коллекции *unicorns*

```
> db.unicorns.find({name: "Ayna"})
{ "_id" : ObjectId("628e8de73de3ad2a6d139aed"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
```

16. Практическое задание 8.2.8.

16.1. 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 })
```

16.2. Проверить содержимое коллекции *unicorns*.

```
> db.unicorns.find({name: "Raleigh"})
{ "_id" : ObjectId("628e8df13de3ad2a6d139aef"), "name" : "Raleigh", "loves" : "Redbull", "weight" : 421, "gender" : "m", "vampires" : 2 }
```

17. Практическое задание 8.2.9.

17.1. Всем самцам единорогов увеличить количество убитых вампиров на 5.

17.2. 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("628e8dc73de3ad2a6d139ae8"), "name" : "H", "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("628e8dd83de3ad2a6d139aea"), "name" : "U", "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("628e8dde3de3ad2a6d139aeb"), "name" : "R", "weight" : 575, "gender" : "m", "vampires" : 99 }
```

18. Практическое задание 8.2.10.

18.1. 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 })
```

18.2. Проверить содержимое коллекции *towns*.

```
> db.towns.find({name: "Portland"})
{ "_id" : ObjectId("628e9acb3de3ad2a6d139af6"), "name" : "Portland", "population" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams" } }
>
```

19. Практическое задание 8.2.11.

19.1. 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 })
```

19.2. Проверить содержимое коллекции *unicorns*.

```
> db.unicorns.find({name: "Pilot"})
{ "_id" : ObjectId("628e8dfd3de3ad2a6d139af1"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("628ea0ca3de3ad2a6d139b00"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
>
```


20. Практическое задание 8.2.12.

20.1. Изменить информацию о самке единорога Ауруса: теперь она любит еще и сахар, и лимоны.

20.2. Проверить содержимое коллекции *unicorns*.

```
> db.unicorns.update({name: "Aurora", gender: "f"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Aurora"})
{ "_id" : ObjectId("628e8dce3de3ad2a6d139ae9"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

21. Практическое задание 8.2.13.

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

21.2. Удалите документы с беспартийными мэрами.

21.3. Проверьте содержание коллекции.

21.4. Очистите коллекцию.

21.5. 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()
>
```

22. Практическое задание 8.3.1.

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

```
> db.zones.insert({_id: "78", short: "ars", full: "Arsenev", descr: "shithole"})
WriteResult({ "nInserted" : 1 })
> db.zones.insert({_id: "64", short: "vdk", full: "Vladivostok", descr: "VLADIVOSTOK 2000"})
WriteResult({ "nInserted" : 1 })
```

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

```
> db.unicorns.update({name: "Aurora"}, {$set: {"city": {$ref: "zones", $id: "78"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Horny"}, {$set: {"city": {$ref: "zones", $id: "64"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

22.3. Проверьте содержание коллекции единорогов.

```
> db.unicorns.find()
{ "_id" : ObjectId("628e8dc73de3ad2a6d139ae8"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68, "city" : DBRef("zones", "64") }
{ "_id" : ObjectId("628e8dce3de3ad2a6d139ae9"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "city" : DBRef("zones", "78") }
```

23. Практическое задание 8.3.2.

23.1. Проверьте, можно ли задать для коллекции *unicorns* индекс для ключа *name* с флагом *unique*.

24. Практическое задание 8.3.3.

24.1. Получите информацию о всех индексах коллекции *unicorns*.

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

24.2. Удалите все индексы, кроме индекса для идентификатора.

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

24.3. Попробуйте удалить индекс для идентификатора.

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

25. Практическое задание 8.3.4.

25.1. 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 })
>
```

25.2. Выберите последних четыре документа

```
> var cursor = db.numbers.find().sort({$natural: -1}).limit(4)
> db.numbers.find().sort({$natural: -1}).limit(4)
{ "_id" : ObjectId("628eb02e3de3ad2a6d1521be"), "value" : 99999 }
{ "_id" : ObjectId("628eb02e3de3ad2a6d1521bd"), "value" : 99998 }
{ "_id" : ObjectId("628eb02e3de3ad2a6d1521bc"), "value" : 99997 }
{ "_id" : ObjectId("628eb02e3de3ad2a6d1521bb"), "value" : 99996 }
>
```

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

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 4,
  "executionTimeMillis" : 47,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 100000,
  "executionStages" : {
```

25.4. Создайте индекс для ключа *value*

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

25.5. Получите информацию обо всех индексах коллекции *numbers*

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

25.6. Выполните запрос 2

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

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

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

25.9. - *Запрос с использования индекса оказался гораздо быстрее, чем без него. Использование индекса ради ускорения возвращения результатов запроса оправдано.*

Выводы:

В ходе работы были получены практические навыки работы с CRUD - операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, с ссылками и индексами в базе данных MongoDB. MongoDB предоставляет мощный CLI интерфейс для выполнения CRUD операций, отличительной особенностью является интеграция полноценного языка программирования: Javascript.