

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
по теме: Работа с БД в СУБД MongoDB

Специальность:
09.03.03 Мобильные и сетевые технологии

Проверил:
Говорова М.М. _____

Выполнил:
студент группы К3240 Ковалев В.М.

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 5.0.8.

Практическое задание 8.1.1:

1. *Создайте базу данных `learn`.*
2. *Заполните коллекцию единорогов `unicorns`*
3. *Проверьте содержимое коллекции с помощью метода `find`*

```
> db.unicorns.find()
< { _id: ObjectId("628b6c61b5d6d7412ce6e797"),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63 }
{ _id: ObjectId("628b6c61b5d6d7412ce6e798"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43 }
{ _id: ObjectId("628b6c61b5d6d7412ce6e799"),
  name: 'Unicrom',
  loves: [ 'energon', 'redbull' ],
  weight: 984,
  gender: 'm',
  vampires: 182 }
{ _id: ObjectId("628b6c61b5d6d7412ce6e79a"),
  name: 'Roooooodles',
  loves: [ 'apple' ],
```

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}).limit(3)
< { _id: ObjectId("628b6cdeb5d6d7412ce6e7a2"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165 }
{ _id: ObjectId("628b6c61b5d6d7412ce6e797"),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63 }
{ _id: ObjectId("628b6c61b5d6d7412ce6e79d"),
  name: 'Kenny',
  loves: [ 'grape', 'lemon' ],
  weight: 690,
  gender: 'm',
  vampires: 39 }
db.unicorns.find({gender:'f'}).sort({name:1}).limit(3)
```

```
> db.unicorns.find({gender:'f'}).sort({name:1}).limit(3)
< { _id: ObjectId("628b6c61b5d6d7412ce6e798"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43 }
{ _id: ObjectId("628b6c61b5d6d7412ce6e79c"),
  name: 'Ayna',
  loves: [ 'strawberry', 'lemon' ],
  weight: 733,
  gender: 'f',
  vampires: 40 }
{ _id: ObjectId("628b6c61b5d6d7412ce6e79f"),
  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"}).limit(1)
< { _id: ObjectId("628b6c61b5d6d7412ce6e798"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43 }
> db.unicorns.findOne({gender:'f', loves:"carrot"})
< { _id: ObjectId("628b6c61b5d6d7412ce6e798"),
  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({name:1}).limit(3)
< { _id: ObjectId("628b6cdeb5d6d7412ce6e7a2"),
  name: 'Dunx',
  weight: 704,
  vampires: 165 }
{ _id: ObjectId("628b6c61b5d6d7412ce6e797"),
  name: 'Horny',
  weight: 600,
  vampires: 63 }
{ _id: ObjectId("628b6c61b5d6d7412ce6e79d"),
  name: 'Kenny',
  weight: 690,
  vampires: 39 }
```

Практическое задание 8.1.4:

1. Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({$natural:-1})
< { _id: ObjectId("628b6cdeb5d6d7412ce6e7a2"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165 }
{ _id: ObjectId("628b6c80b5d6d7412ce6e7a1"),
  name: 'Nimue',
  loves: [ 'grape', 'carrot' ],
  weight: 540,
  gender: 'f' }
{ _id: ObjectId("628b6c61b5d6d7412ce6e7a0"),
  name: 'Pilot',
  loves: [ 'apple', 'watermelon' ],
  weight: 650,
  gender: 'm',
  vampires: 54 }
```

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' ],
```

Практическое задание 8.1.6:

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

```
> db.unicorns.find({weight:{$gt:500, $lt:700}}, {_id:0})
< { name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63 }
{ name: 'Roooooodles',
  loves: [ 'apple' ],
  weight: 575,
  gender: 'm',
  vampires: 99 }
{ name: 'Solnara',
  loves: [ 'apple', 'carrot', 'chocolate' ],
  weight: 550,
  gender: 'f',
  vampires: 80 }
{ name: 'Kenny',
  loves: [ 'grape', 'lemon' ],
  weight: 690,
```

Практическое задание 8.1.7:

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

```
> db.unicorns.find({weight:{$gt:500, $lt:700}, loves:{$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("628b6c80b5d6d7412ce6e7a1"),
  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}, name:1, _id:0}).sort({name:1})
< { name: 'Dunx', loves: [ 'grape' ] }
  { name: 'Horny', loves: [ 'carrot' ] }
  { name: 'Kenny', loves: [ 'grape' ] }
  { name: 'Pilot', loves: [ 'apple' ] }
  { name: 'Raleigh', loves: [ 'apple' ] }
  { name: 'Roooooodles', loves: [ 'apple' ] }
  { name: 'Unicrom', loves: [ 'energon' ] }
```

Практическое задание 8.2.1:

1. 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"}, {_id:0, name:1, mayor:1})
< { name: 'New York',
  mayor: { name: 'Michael Bloomberg', party: 'I' } }
```

2. Сформировать запрос, который возвращает список беспартийных мэров (`party` отсутствует). Вывести только название города и информацию о мэре.

```
> db.towns.find({"mayor.party":{$exists:false}}, {_id:0, name:1, mayor:1})
< { name: 'Punxsutawney ', mayor: { name: 'Jim Wehrle' } }
```

Практическое задание 8.2.2:

1. Сформировать функцию для вывода списка самцов единорогов.

```
> get_males = function() {return this.gender=="m";}
function() {return this.gender=="m";}
> db.unicorns.find(get_males)
{ "_id" : ObjectId("628b6c61b5d6d7412ce6e797"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 575,
  "_id" : ObjectId("628b6c61b5d6d7412ce6e799"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 575,
  "_id" : ObjectId("628b6c61b5d6d7412ce6e79a"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575,
  "_id" : ObjectId("628b6c61b5d6d7412ce6e79d"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 600,
  "_id" : ObjectId("628b6c61b5d6d7412ce6e79e"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 575,
  "_id" : ObjectId("628b6c61b5d6d7412ce6e7a0"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 575 }
```

2. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.
3. Вывести результат, используя `forEach`.

```
> var cursor = db.unicorns.find(get_males).sort({name:1}).limit(2)
> cursor.forEach(function(obj){print(obj.name)})
Dunx
Horny
> █
```

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

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

Практическое задание 8.2.4:

Вывести список предпочтений.

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

Практическое задание 8.2.5:

Посчитать количество особей единорогов обоих полов.

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

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.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("628b6c61b5d6d7412ce6e797"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" :
{ "_id" : ObjectId("628b6c61b5d6d7412ce6e798"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" :
{ "_id" : ObjectId("628b6c61b5d6d7412ce6e799"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weigh
{ "_id" : ObjectId("628b6c61b5d6d7412ce6e79a"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575,
{ "_id" : ObjectId("628b6c61b5d6d7412ce6e79b"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate"
{ "_id" : ObjectId("628b6c61b5d6d7412ce6e79c"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight"
{ "_id" : ObjectId("628b6c61b5d6d7412ce6e79d"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 6
{ "_id" : ObjectId("628b6c61b5d6d7412ce6e79e"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" :
{ "_id" : ObjectId("628b6c61b5d6d7412ce6e79f"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight"
{ "_id" : ObjectId("628b6c61b5d6d7412ce6e7a0"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight"
{ "_id" : ObjectId("628b6c80b5d6d7412ce6e7a1"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" :
{ "_id" : ObjectId("628b6cdeb5d6d7412ce6e7a2"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight"
{ "_id" : ObjectId("628cd8e1e812acba27bae86"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gend
>
```

Практическое задание 8.2.7:

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

```
> db.unicorns.updateOne({name: 'Ayna'}, {$set: {weight:800, vampires:51}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0 }
> db.unicorns.find({name:'Ayna'})
< { _id: ObjectId("628b6c61b5d6d7412ce6e79c"),
  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.updateOne({name: 'Raleigh'}, {$set: {loves:["redbull"]}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.unicorns.find({name:'Raleigh'})
< { _id: ObjectId("628b6c61b5d6d7412ce6e79e"),
  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.
2. Проверить содержимое коллекции `unicorns`.

```
> db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires:5}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 8,
  modifiedCount: 8,
  upsertedCount: 0 }
```

У единорога Raleigh было 2 убитых вампира, стало 5

```
> db.unicorns.find({name:'Raleigh'})
< { _id: ObjectId("628b6c61b5d6d7412ce6e79e"),
  name: 'Raleigh',
  loves: [ 'redbull' ],
  weight: 421,
  gender: 'm',
  vampires: 7 }
learn>
```

Практическое задание 8.2.10:

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

```
> db.towns.updateOne({name: "Portland"}, {$unset:{'mayor.party':1}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.towns.find()
< { _id: ObjectId("628cb7e8cd1e73c046d32d4e"),
  name: 'Punxsutawney ',
  populatiuon: 6200,
  last_sensus: 2008-01-31T00:00:00.000Z,
  famous_for: [ '' ],
  mayor: { name: 'Jim Wehrle' } }
{ _id: ObjectId("628cb844cd1e73c046d32d4f"),
  name: 'New York',
  populatiuon: 22200000,
  last_sensus: 2009-07-31T00:00:00.000Z,
  famous_for: [ 'status of liberty', 'food' ],
  mayor: { name: 'Michael Bloomberg', party: 'I' } }
{ _id: ObjectId("628cb86dcd1e73c046d32d50"),
  name: 'Portland',
  populatiuon: 528000,
  last_sensus: 2009-07-20T00:00:00.000Z,
  famous_for: [ 'beer', 'food' ],
  mayor: { name: 'Sam Adams' } }
learn>
```


Практическое задание 8.2.11:

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

```
> db.unicorns.updateOne({name: "Pilot"}, {$push:{loves:"chocolate"}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.unicorns.find({name:'Pilot'})
< { _id: ObjectId("628b6c61b5d6d7412ce6e7a0"),
  name: 'Pilot',
  loves: [ 'apple', 'watermelon', 'chocolate' ],
  weight: 650,
  gender: 'm',
  vampires: 59 }
learn>
```

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.updateOne({name: "Aurora"}, {$addToSet:{loves:{$each:['sugar', 'lemon']}}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.unicorns.find({name:'Aurora'})
< { _id: ObjectId("628b6c61b5d6d7412ce6e798"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
  weight: 450,
  gender: 'f',
  vampires: 43 }
learn> |
```

Практическое задание 8.2.13:

1. Удалите документы с беспартийными мэрами.

```
> db.towns.remove({"mayor.party":{"$exists:0}})
< { acknowledged: true, deletedCount: 1 }
> db.towns.find()
< { _id: ObjectId("628ce78fcd1e73c046d32d55"),
  name: 'New York',
  popujatiuon: 22200000,
  last_sensus: 2009-07-31T00:00:00.000Z,
  famous_for: [ 'status of liberty', 'food' ],
  mayor: { name: 'Michael Bloomberg', party: 'I' } }
{ _id: ObjectId("628ce78fcd1e73c046d32d56"),
  name: 'Portland',
  popujatiuon: 528000,
  last_sensus: 2009-07-20T00:00:00.000Z,
  famous_for: [ 'beer', 'food' ],
  mayor: { name: 'Sam Adams', party: 'D' } }
learn>
```

2. Очистите коллекцию.
3. Просмотрите список доступных коллекций.

```
> db.towns.remove({})
< { acknowledged: true, deletedCount: 2 }
> show collections
< towns
   unicorns
   system.js
> db.towns.find()
<
learn>
```

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.areas.insertOne({_id:"RUS", full_name: "Russia"})
< { acknowledged: true, insertedId: 'RUS' }
> db.areas.insertOne({_id:"USA", full_name: "United States"})
< { acknowledged: true, insertedId: 'USA' }
> db.areas.insertOne({_id:"KZH", full_name: "Kazakhstan"})
< { acknowledged: true, insertedId: 'KZH' }
> db.areas.insertOne({_id:"BEL", full_name: "Belarus"})
< { acknowledged: true, insertedId: 'BEL' }
```

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

```
> db.unicorns.updateOne({name: 'Horny'}, {$set: {"area": {$ref: "areas", $id: "RUS"}}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0 }
> db.unicorns.updateOne({name: 'Aurora'}, {$set: {"area": {$ref: "areas", $id: "USA"}}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.unicorns.updateOne({name: 'Unicrom'}, {$set: {"area": {$ref: "areas", $id: "KZH"}}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.unicorns.updateOne({name: 'Rooooooodles'}, {$set: {"area": {$ref: "areas", $id: "BEL"}}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
```

3. Проверьте содержание коллекции единорогов.

```
> db.unicorns.find()
< { _id: ObjectId("628b6c61b5d6d7412ce6e797"),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 78,
  area: DBRef("areas", 'RUS') }
{ _id: ObjectId("628b6c61b5d6d7412ce6e798"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
  weight: 450,
  gender: 'f',
  vampires: 43,
  area: DBRef("areas", 'USA') }
{ _id: ObjectId("628b6c61b5d6d7412ce6e799"),
  name: 'Unicrom',
  loves: [ 'energon', 'redbull' ],
  weight: 984,
  gender: 'm',
  vampires: 187,
  area: DBRef("areas", 'KZH') }
{ _id: ObjectId("628b6c61b5d6d7412ce6e79a"),
  name: 'Rooooooodles',
  loves: [ 'apple' ],
  weight: 575,
  gender: 'm',
  vampires: 104,
  area: DBRef("areas", 'BEL') }
```

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 о всех индексах коллекции `unicorns`.

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

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})}
```

2. Выберите последних четыре документа.

```
> var last_4 = db.numbers.find().sort({$natural:-1}).limit(4)
> last_4
{ "_id" : ObjectId("628cfa45cd1e73c046d4b3f6"), "value" : 99999 }
{ "_id" : ObjectId("628cfa45cd1e73c046d4b3f5"), "value" : 99998 }
{ "_id" : ObjectId("628cfa45cd1e73c046d4b3f4"), "value" : 99997 }
{ "_id" : ObjectId("628cfa45cd1e73c046d4b3f3"), "value" : 99996 }
```

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

С курсором:

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

Без курсора:

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 4,
  "executionTimeMillis" : 11,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 4,
  "executionStages" : {
```

4. *Создайте индекс для ключа value.*

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

5. *Выполните запрос 2.*

6. *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" : {
    "stage" : "LIMIT"
```

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

Можно увидеть положительный эффект от использования курсора и индексации. Более эффективный запрос после индексации.

Вывод: в ходе работы мною были изучены CRUD-операции в MongoDB, а также другие возможности СУБД.