Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной

техники Кафедра вычислительных систем

ОТЧЕТ
по лабораторной работе № 6 на тему:
«Работа с БД в СУБД MongoDB**»**

Выполнил: Губанов Егор Романович
Группа: К3240

Проверил: М.М. Говорова

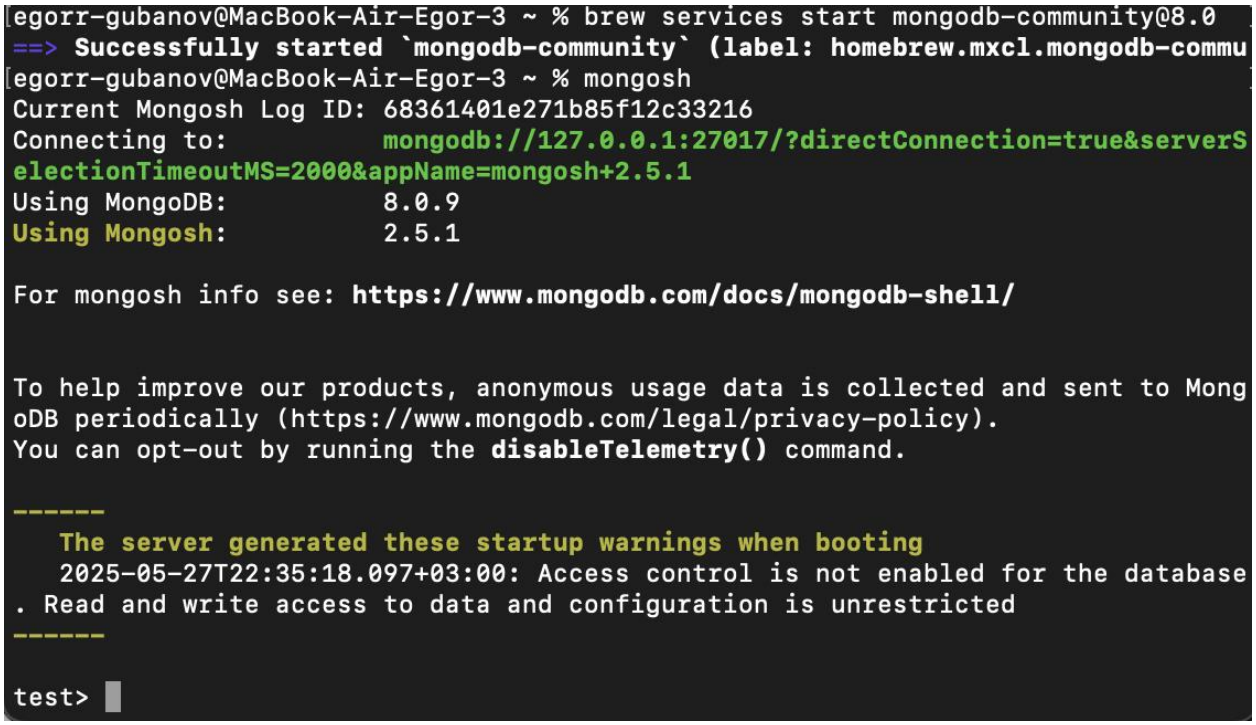
Санкт-Петербург
2025

Цель работы

Овладеть практическими навыками работы с CRUD-операциями, вложенными объектами, агрегацией, изменением данных, ссылками и индексами в MongoDB.

Установка 1

Установил с brew и запустил сервер



Практическое задание 2.1.1

Создание базы данных и коллекции

```
... db.unicorns.insertMany([
...   {name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vamp
ires: 63},
...   {name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vam
pires: 43},
...   {name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm',
 vampires: 182},
...   {name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires
: 99},
...   {name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gend
er:'f', vampires:80},
...   {name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', va
mpires: 40},
...   {name:'Kenny', loves: ['grape', 'lemon'], weight: 690,  gender: 'm', vampi
res: 39},
...   {name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vam
pires: 2},
...   {name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', v
ampires: 33},
...   {name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm',
vampires: 54},
...   {name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'}
... ])
...
... const dunx = {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gen
der: 'm', vampires: 165}
[... db.unicorns.insertOne(dunx)
already on db learn
learn>
```

```
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany
, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('68361508e271b85f12c33221') }
}
```

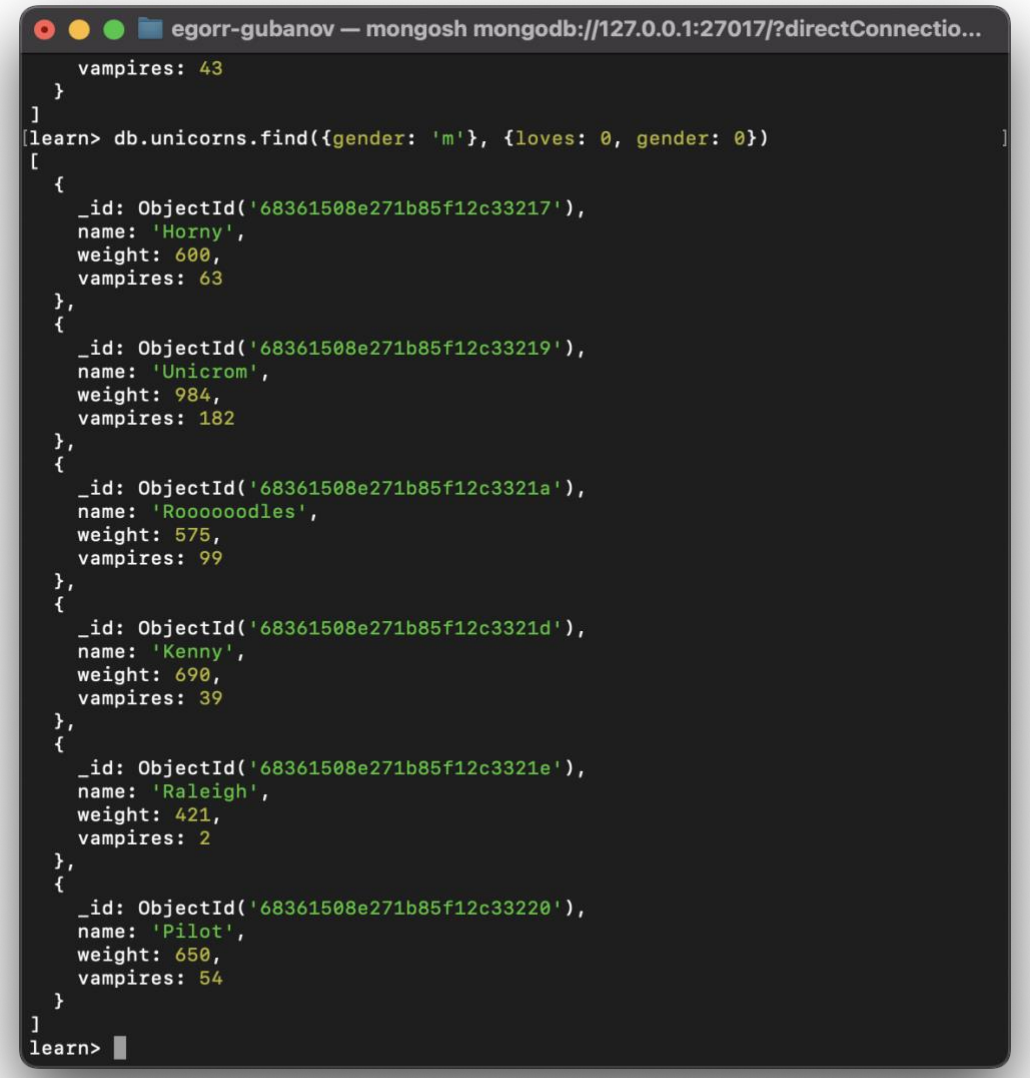
Практическое задание 2.2.1

Сформируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени. Найдите всех самок, которые любят carrot.


egorr-gubanov — mongosh mongodb://127.0.0.1:27017/?directConnectio...
learn> db.unicorns.find({gender: 'm'}).sort({name: 1})
[
{
_id: ObjectId('68361508e271b85f12c33217'),
name: 'Horny',
loves: ['carrot', 'papaya'],
weight: 600,
gender: 'm',
vampires: 63
},
{
_id: ObjectId('68361508e271b85f12c3321d'),
name: 'Kenny',
loves: ['grape', 'lemon'],
weight: 690,
gender: 'm',
vampires: 39
},
{
_id: ObjectId('68361508e271b85f12c33220'),
name: 'Pilot',
loves: ['apple', 'watermelon'],
weight: 650,
gender: 'm',
vampires: 54
},
{
_id: ObjectId('68361508e271b85f12c3321e'),
name: 'Raleigh',
loves: ['apple', 'sugar'],
weight: 421,
gender: 'm',
vampires: 2
},
{
_id: ObjectId('68361508e271b85f12c3321a'),
name: 'Roooooodles',
loves: ['apple'],
weight: 575,
gender: 'm',
vampires: 99
},
{

Практическое задание 2.2.2

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



Практическое задание 2.2.3

Вывести список единорогов в обратном порядке добавления.

Практическое задание 2.2.4

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

```
learn> db.unicorns.find({}, {_id: 0, name: 1, firstLove: {$arrayElemAt: ['$loves
',
[
 mongosh mongodb://127.0.0.1:27017/?
 directConnection=true&serverSelectionTimeout
 MS=2000 — ?directConnection=true&
 __CFBundleIdentifier=com.apple.Terminal
  { name: 'Unicrom', firstLove: 'energon' },
  { name: 'Roooooodles', firstLove: 'apple' },
  { name: 'Solnara', firstLove: 'apple' },
  { name: 'Ayna', firstLove: 'strawberry' },
  { name: 'Kenny', firstLove: 'grape' },
  { name: 'Raleigh', firstLove: 'apple' },
  { name: 'Leia', firstLove: 'apple' },
  { name: 'Pilot', firstLove: 'apple' },
  { name: 'Nimue', firstLove: 'grape' }
]
```

Практическое задание 2.3.1

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

```
learn> db.unicorns.find({
...   gender: 'f',
...   weight: {$gte: 500, $lte: 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 2.3.2

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

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

Практическое задание 2.3.3

Найти всех единорогов, не имеющих ключ vampires.

Практическое задание 2.3.4

Вывести упорядоченный список имен самцов единорогов с информацией об их первом предпочтении.

```
learn> db.unicorns.find({vampires: {$exists: false}})
[
  {
    _id: ObjectId('68361508e271b85f12c33221'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
learn> db.unicorns.find(
...   {gender: 'm'},
...   {_id: 0, name: 1, firstLove: {$arrayElemAt: ['$loves', 0]}}
... ).sort({name: 1})
[
  { name: 'Horny', firstLove: 'carrot' },
  { name: 'Kenny', firstLove: 'grape' },
  { name: 'Pilot', firstLove: 'apple' },
  { name: 'Raleigh', firstLove: 'apple' },
  { name: 'Roooooodles', firstLove: 'apple' },
  { name: 'Unicrom', firstLove: 'energon' }
]
```

Практическое задание 3.1.1

Создайте коллекцию towns и выполните выборки по мэрам с party="I" и без party.

Практическое задание 3.1.2

Сформировать функцию для вывода списка самцов единорогов и вывести первых двух.

```
learn> function printMaleUnicorns() {
...   var cursor = db.unicorns.find({gender: 'm'}).sort({name: 1}).limit(2)
...   cursor.forEach(function(unicorn) {
...     print(unicorn.name)
...   })
... }
[Function: printMaleUnicorns]
learn> printMaleUnicorns()
Horny
Kenny

learn> var maleUnicornsCursor = db.unicorns.find({gender: 'm'}).sort({name: 1}).
limit(2)
... maleUnicornsCursor.forEach(function(unicorn) {
...   printjson(unicorn.name)
... })
Horny
Kenny
```

Практическое задание 3.2.1

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
learn> db.unicorns.countDocuments({
...   gender: 'f',
...   weight: {$gte: 500, $lte: 600}
... })
2
```

Практическое задание 3.2.2

Вывести список предпочтений.

```
learn> db.unicorns.aggregate([
...   {$unwind: "$loves"},
...   {$group: {_id: "$loves"}},
...   {$sort: {_id: 1}}
... ])
[
  { _id: 'apple' },
  { _id: 'carrot' },
  { _id: 'chocolate' },
  { _id: 'energon' },
  { _id: 'grape' },
  { _id: 'lemon' },
  { _id: 'papaya' },
  { _id: 'redbull' },
  { _id: 'strawberry' },
  { _id: 'sugar' },
  { _id: 'watermelon' }
]
```

Практическое задание 3.2.3

Посчитать количество особей единорогов обоих полов.

```
learn> db.unicorns.aggregate([
...   {$group: {
...     _id: "$gender",
...     count: {$sum: 1}
...   }}
... ])
[ { _id: 'm', count: 6 }, { _id: 'f', count: 5 } ]
```

Практическое задание 3.3.1

Добавить самца Barny

```
learn> db.unicorns.insertOne({name: "Barny", loves: ["grape"], weight: 340, gend
er: "m"})
...
{
  acknowledged: true,
  insertedId: ObjectId('68361bd0e271b85f12c33225')
}
learn> db.unicorns.find({name: 'Barny'})
[
  {
    _id: ObjectId('68361bd0e271b85f12c33225'),
    name: 'Barny',
    loves: [ 'grape' ],
    weight: 340,
    gender: 'm'
  }
]
```

Практическое задание 3.3.2

Обновить Ayna: вес 800, вампиры 51.

```
learn> db.unicorns.update(
...   {name: 'Ayna'},
...   {$set: {weight: 800, vampires: 51}}
... )
```

Практическое задание 3.3.3

Обновить Raleigh: добавить redbull в loves.

```
learn> db.unicorns.update(
...   {name: 'Raleigh'},
...   {$set: {loves: ['redbull']}}
... )
```

Практическое задание 3.3.4

Увеличить количество убитых вампиров у всех самцов на 5.

```
learn> db.unicorns.updateMany(
...   {gender: 'm'},
...   {$inc: {vampires: 5}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 7,
  modifiedCount: 7,
  upsertedCount: 0
}
```

Практическое задание 3.3.5

Убрать партию у мэра Портланда

```
learn> db.towns.update(
...   {name: 'Portland'},
...   {$unset: {'mayor.party': 1}}
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Практическое задание 3.3.6

Обновить Pilot: добавить chocolate в loves.

```
learn> db.unicorns.update({name: "Pilot"}, {$push: {loves: "chocolate"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Практическое задание 3.3.7

Обновить Aurora: добавить sugar и lemon в loves.

```
learn> db.unicorns.update(
...   {name: 'Aurora'},
...   {$addToSet: {loves: {$each: ['sugar', 'lemon']}}}
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Практическое задание 3.4.1

Удалить беспартийных мэров, очистить коллекцию, просмотреть коллекции.

```
learn> db.towns.deleteMany({"mayor.party": {$exists: false}})
{ acknowledged: true, deletedCount: 3 }
```

```
learn> db.towns.deleteMany({})
{ acknowledged: true, deletedCount: 3 }
learn> show collections
towns
unicorns
```

Практическое задание 4.1.1

Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание. Включите для нескольких единорогов в документы ссылку на зону обитания, используя второй способ автоматического связывания. Проверьте содержание коллекции единорогов.

```
learn> db.habitats.insertMany([
...   {
...     _id: "forest",
...     fullName: "Волшебный лес",
...     description: "Густой лес с древними деревьями и светящимися грибами"
...   },
...   {
...     _id: "mountains",
...     fullName: "Хрустальные горы",
...     description: "Высокие горы с кристальными пещерами"
...   },
...   {
...     _id: "meadow",
...     fullName: "Радужные луга",
...     description: "Просторные луга с цветами всех цветов радуги"
...   }
... ])
...
{
  acknowledged: true,
  insertedIds: { '0': 'forest', '1': 'mountains', '2': 'meadow' }
}
learn> db.unicorns.update(
...   {name: "Horny"},
...   {$set: {habitat: new DBRef('habitats', 'forest')}}
... )
...
... db.unicorns.update(
...   {name: "Aurora"},
...   {$set: {habitat: new DBRef('habitats', 'meadow')}}
... )
...
... db.unicorns.update(
...   {name: "Unicrom"},
...   {$set: {habitat: new DBRef('habitats', 'mountains')}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
learn> db.unicorns.find({}, {name: 1, habitat: 1, _id: 0})
[
  { name: 'Horny', habitat: DBRef('habitats', 'forest') },
  { name: 'Aurora', habitat: DBRef('habitats', 'meadow') },
  { name: 'Unicrom', habitat: DBRef('habitats', 'mountains') },
  { name: 'Roooooodles' },
  { name: 'Solnara' },
  { name: 'Ayna' },
  { name: 'Kenny' },
  { name: 'Raleigh' },
  { name: 'Leia' },
  { name: 'Pilot' },
  { name: 'Nimue' },
  { name: 'Barny' }
]
learn> var unicorn = db.unicorns.findOne({name: "Horny"})
... db[unicorn.habitat.$ref].findOne({_id: unicorn.habitat.$id})
```

Практическое задание 4.2.1

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

```
learn> try {
...   db.unicorns.createIndex({name: 1}, {unique: true})
... } catch (e) {
...   print(e)
... }
...
MongoServerError: An existing index has the same name as the requested index. Wh
en index names are not specified, they are auto generated and can cause conflict
s. Please refer to our documentation. Requested index: { v: 2, unique: true, key
: { name: 1 }, name: "name_1" }, existing index: { v: 2, key: { name: 1 }, name:
 "name_1" }
    at Connection.sendCommand (/opt/homebrew/Cellar/mongosh/2.5.1/libexec/lib/no
de_modules/@mongosh/cli-repl/node_modules/mongodb/lib/cmap/connection.js:299:27)
    at process.processTicksAndRejections (node:internal/process/task_queues:105:
5)
```

Ошибка из-за того что был дубликат

```
learn> try {
...   db.unicorns.createIndex({name: 3}, {unique: true})
... } catch (e) {
...   print(e)
... }
...
name_3
```

Практическое задание 4.3.1

Получите информацию о всех индексах коллекции unicorns. Удалите все индексы, кроме индекса для идентификатора. Попытайтесь удалить индекс для идентификатора.

```
learn> db.unicorns.getIndexes()
...
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1' },
  { v: 2, key: { name: 3 }, name: 'name_3', unique: true }
]
learn> db.unicorns.dropIndexes()
{
  nIndexesWas: 3,
  msg: 'non-_id indexes dropped for collection',
  ok: 1
}
learn> try {
...   db.unicorns.dropIndex("_id_")
... } catch (e) {
...   print(e)
... }
MongoServerError: cannot drop _id index
    at Connection.sendCommand (/opt/homebrew/Cellar/mongosh/2.5.1/libexec/lib/no
de_modules/@mongosh/cli-repl/node_modules/mongodb/lib/cmap/connection.js:299:27)
    at process.processTicksAndRejections (node:internal/process/task_queues:105:
5)
```

Практическое задание 4.4.1

Создайте объемную коллекцию numbers, задействовав курсор. Выберите последние четыре документа. Проанализируйте план выполнения запроса. Сколько потребовалось времени на выполнение запроса? Создайте индекс для ключа value. Получите информацию о всех индексах коллекции numbers. Выполните запрос 2. Проанализируйте план выполнения запроса с установленным индексом. Сравните время выполнения запросов с индексом и без.

```
test> let Data = [];
... for(let i = 0; i < 100000; i++) {
...   Data.push({value: i});
... }
... db.numbers.insertMany(Data);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6836207dbb2363f1822781e3'),
    '1': ObjectId('6836207dbb2363f1822781e4'),
    '2': ObjectId('6836207dbb2363f1822781e5'),
    '3': ObjectId('6836207dbb2363f1822781e6'),
    '4': ObjectId('6836207dbb2363f1822781e7'),
    '5': ObjectId('6836207dbb2363f1822781e8'),
    '6': ObjectId('6836207dbb2363f1822781e9'),
    '7': ObjectId('6836207dbb2363f1822781ea'),
    '8': ObjectId('6836207dbb2363f1822781eb'),
    '9': ObjectId('6836207dbb2363f1822781ec'),
```

```
test> db.numbers.find().sort({_id: -1}).limit(4);
[
  { _id: ObjectId('6836207dbb2363f182290882'), value: 99999 },
  { _id: ObjectId('6836207dbb2363f182290881'), value: 99998 },
  { _id: ObjectId('6836207dbb2363f182290880'), value: 99997 },
  { _id: ObjectId('6836207dbb2363f18229087f'), value: 99996 }
]
test> let explain1 = db.numbers.find().sort({_id: -1}).limit(4).explain("executi
onStats");
... print("без индекса: " + explain1.executionStats.executionTimeMillis + "ms");

без индекса: 2ms

test> db.numbers.createIndex({value: 1});
value_1
test> db.numbers.getIndexes();
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { value: 1 }, name: 'value_1' }
]
test> let explain2 = db.numbers.find().sort({_id: -1}).limit(4).explain("executi
onStats");
... print("с индексом: " + explain2.executionStats.executionTimeMillis + "ms");
...
с индексом: 1ms
```

В ходе сравнения видно что с индексом выполняется в два раза быстрее. Индекс ускоряется в том случае если мы его используем для перебора.

Выводы

В ходе работы я освоил базовые операции MongoDB: создание, чтение, обновление и удаление данных. Научился работать с вложенными документами и сложными запросами, включая агрегацию. Особенно полезным оказалось изучение индексов для ускорения поиска. Полученные навыки помогут мне в разработке приложений с NoSQL-базами данных.