

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский университет ИТМО»

Факультет прикладной информатики

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

Выполнил: студент группы К3241

ФИО: Кижваткин Никита Максимович

Проверил: преподаватель М.М. Говорова

Санкт-Петербург
2025

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 4+, 8.0.4 (последняя).

Практическое задание 2.1.1

Создание базы данных и заполнение коллекции

```
test> use learn
switched to db learn
learn> 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: 'Rooooooodles', 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'});
...
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6835cba9e70a9e21f335df72') }
}
learn> document=({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})
...
{
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
learn> db.unicorns.insert(document)
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6835cbe4e70a9e21f335df73') }
}
learn> db.unicorns.find()
[
  {
    _id: ObjectId('6835cba9e70a9e21f335df68'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df69'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df6a'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df6b'),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df6c'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
```

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. Ограничьте этот список первой особью с помощью функций findOne и limit

Команды MongoDB:

```
db.unicorns.find({gender: 'm'}).sort({name: 1})
db.unicorns.find({gender: 'f'}).sort({name: 1}).limit(3)
db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
db.unicorns.findOne({gender: 'f', loves: 'carrot'})
```

Практическое задание 2.2.2

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

Команды MongoDB:

```
db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0})
```

Практическое задание 2.2.3

Вывести список единорогов в обратном порядке добавления.

Команды MongoDB:

```
db.unicorns.find().sort({$natural: -1})
```

```
learn> db.unicorns.find().sort({$natural: -1})
[
  {
    _id: ObjectId('6835cbe4e70a9e21f335df73'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df72'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df71'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df70'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df6f'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
  },
]
```

Практическое задание 2.2.4

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

Команды MongoDB:

```
db.unicorns.find({}, {loves: {$slice: 1}, _id: 0})
```

```
learn> 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: 675,
    gender: 'm',
    vampires: 99
  },
  {
    name: 'Solnara',
    loves: [ 'apple' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    name: 'Ayna',
    loves: [ 'strawberry' ],

```

Практическое задание 2.3.1

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

Команды MongoDB:

```
db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 700}}, {_id: 0})
```

```
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 грапе и lemon, исключив вывод идентификатора.

Команды MongoDB:

```
db.unicorns.find({gender: 'm', weight: {$gte: 500}, loves: {$all: ['grape', 'lemon']}}, {_id: 0})
```

```
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.

Команды MongoDB:

```
db.unicorns.find({vampires: {$exists: false}})
```

```
learn> db.unicorns.find({vampires: {$exists: false}})
[...
[
  {
    _id: ObjectId('6835cba9e70a9e21f335df72'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
```

Практическое задание 2.3.4

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

Команды MongoDB:

```
db.unicorns.find({gender: 'm'}, {name: 1, loves: {$slice: 1}, _id: 0}).sort({name: 1})
```

```
learn> db.unicorns.find({gender: 'm'}, {name: 1, loves: {$slice: 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: 'Rooooooodles', loves: [ 'apple' ] },
  { name: 'Unicrom', loves: [ 'energon' ] }
]
```


Практическое задание 3.1.1

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

Команды MongoDB:

```
db.towns.insert({...})
```

```
db.towns.find({"mayor.party": "I"}, {name: 1, mayor: 1})
```

```
db.towns.find({"mayor.party": {$exists: false}}, {name: 1, mayor: 1})
```

```
learn> db.towns.find({"mayor.party": "I"}, {name: 1, mayor: 1})
...
[
  {
    _id: ObjectId('6835d7fde70a9e21f335df75'),
    name: 'New York',
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  }
]
learn> db.towns.find({"mayor.party": {$exists: false}}, {name: 1, mayor: 1})
...
[
  {
    _id: ObjectId('6835d7f0e70a9e21f335df74'),
    name: 'Punxsutawney ',
    mayor: { name: 'Jim Wehrle' }
  }
]
```

Практическое задание 3.1.2

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

Команды MongoDB:

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

```
cursor.forEach(function(unicorn) { print(unicorn.name); });
```

```
learn> var cursor = db.unicorns.find({gender: 'm'}).sort({name: 1}).limit(2); cursor.forEach(function(unicorn) { print(unicorn.name); });
Dunx
Hozny
```

Практическое задание 3.2.1

Вывести количество самок единорогов весом от полутонны до 600 кг.

Команды MongoDB:

```
db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 600}}).count()
```

```
learn> db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 600}}).count()
...
(node:3633) [MONGODB DRIVER] Warning: cursor.count is deprecated and will be removed in a future version. (Use `node --trace-warnings ...` to show where the warning was created)
2
```

Практическое задание 3.2.2

Вывести список предпочтений.

Команды MongoDB:

```
db.unicorns.distinct("loves")
```

```

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

```

Практическое задание 3.2.3

Посчитать количество особей единорогов обоих полов.

Команды MongoDB:

```

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

```

Практическое задание 3.3.1

Добавить самца Barny.

Команды MongoDB:

```

db.unicorns.save({name: "Barny", loves: ["grape"], weight: 340, gender:
"m"})

```

В моей версии (8.09) команда save была удалена, поэтому:

```

db.unicorns.insertOne({name: "Barny", loves: ["grape"], weight: 340,
gender: "m"})

```

```

learn> db.unicorns.insertOne({name: "Barny", loves: ["grape"], weight: 340,
... gender: "m"})
...
{
  acknowledged: true,
  insertedId: ObjectId('683dee65e70a9e21f335df77')
}
learn>

```

Практическое задание 3.3.2

Обновить Ayna: вес 800, вампиры 51.

Команды MongoDB:

```

db.unicorns.update({name: "Ayna"}, {$set: {weight: 800, vampires:
51}})

```

```
learn> db.unicorns.update({name: "Ayna"}, {$set: {weight: 800, vampires: 51}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany,
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Практическое задание 3.3.3

Обновить Raleigh: добавить redbull в loves.

Команды MongoDB:

```
db.unicorns.update({name: "Raleigh"}, {$push: {loves: "redbull"}})
```

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

Практическое задание 3.3.4

Увеличить количество убитых вампиров у всех самцов на 5.

Команды MongoDB:

```
db.unicorns.update({gender: "m"}, {$inc: {vampires: 5}}, {multi: true})
```

```
learn> db.unicorns.update({gender: "m"}, {$inc: {vampires: 5}}, {multi: true})
[...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 8,
  modifiedCount: 8,
  upsertedCount: 0
}
```

Практическое задание 3.3.5

Убрать партию у мэра Портланда.

Команды MongoDB:


```
db.towns.update({name: "Portland"}, {$unset: {"mayor.party": 1}})
```

```
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.

Команды MongoDB:

```
db.unicorns.update({name: "Pilot"}, {$push: {loves: "chocolate"}})
```

```
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.

Команды MongoDB:

```
db.unicorns.update({name: "Aurora"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}})
```

```
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.

Команды MongoDB:

```
db.towns.remove({"mayor.party": {$exists: false}})
```

```
db.towns.remove({})
```

```
show collections
```

```
learn> db.towns.remove({"mayor.party": {$exists: false}})
[...
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 2 }
learn> db.towns.find()
[
  {
    _id: ObjectId('6835d7fde70a9e21f335df75'),
    name: 'New York',
    populatiuon: 22200000,
    last_sensus: ISODate('2009-07-31T00:00:00.000Z'),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  }
]
learn> db.towns.remove({})
[...
{ acknowledged: true, deletedCount: 1 }
learn> show collections
[...
towns
unicorns
learn> db.towns.find()

learn> █
```

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.

Команды MongoDB:

```
db.habitats.insert({_id: "lakes", name: "Lakes"});
```

```
db.habitats.insert({_id: "hills", name: "Hills"});
```

```
db.unicorns.update({name: "Horny"}, {$set: {habitat: {$ref: "habitats", $id: "lakes"}}});
```

```
db.unicorns.update({name: "Kenny"}, {$set: {habitat: {$ref: "habitats", $id: "hills"}}});
```

```
db.unicorns.find();
```

```
learn> db.habitats.insert({_id: "lakes", name: "Lakes"});
{ acknowledged: true, insertedIds: { '0': 'lakes' } }
learn> db.habitats.insert({_id: "hills", name: "Hills"});
{ acknowledged: true, insertedIds: { '0': 'hills' } }
```

```
learn> db.unicorns.update({name: "Horny"}, {$set: {habitat: {$ref: "habitats",
... $id: "lakes"}}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.update({ name: "Kenny" }, { $set: { habitat: { $ref: "habitats", $id: "hills" } } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
learn> db.unicorns.find()
[
  {
    _id: ObjectId('6835cba9e70a9e21f335df68'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 68,
    habitat: DBRef('habitats', 'lakes')
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df69'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df6a'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 187
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df6b'),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 104
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df6c'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df6d'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 800,
    gender: 'f',
    vampires: 51
  },
  {
    _id: ObjectId('6835cba9e70a9e21f335df6e'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 44,
    habitat: DBRef('habitats', 'hills')
  },
]
```

Практическое задание 4.2.1

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

Команды MongoDB:

```
db.unicorns.createIndex({name: 1}, {unique: true})
```

```
[learn> db.unicorns.createIndex({name: 1}, {unique: true})
name_1
```

Практическое задание 4.3.1

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

Команды MongoDB:

```
db.unicorns.getIndexes();
db.unicorns.dropIndexes();
db.unicorns.dropIndex("_id_");
```

```
[learn> db.unicorns.getIndexes();
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1', unique: true }
]
[learn> db.unicorns.dropIndex("name_1");
{ nIndexWas: 2, ok: 1 }
[learn> db.unicorns.dropIndex("_id_");
MongoServerError[InvalidOptions]: cannot drop _id index
learn> █
```

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.

Команды MongoDB:

```
for(i = 0; i < 100000; i++){ db.numbers.insert({value: i}) }  
db.numbers.find().sort({$natural: -1}).limit(4)  
db.numbers.explain("executionStats").find().sort({$natural: -1}).limit(4)  
db.numbers.createIndex({value: 1})  
db.numbers.getIndexes()  
db.numbers.explain("executionStats").find().sort({$natural: -1}).limit(4)
```


Результат с индексом:

```
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'learn.numbers',
    parsedQuery: {},
    indexFilterSet: false,
    queryHash: 'BA27D965',
    planCacheShapeHash: 'BA27D965',
    planCacheKey: '7A892B81',
    optimizationTimeMillis: 0,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'LIMIT',
      limitAmount: 4,
      inputStage: {
        stage: 'FETCH',
        inputStage: {
          stage: 'IXSCAN',
          keyPattern: { value: 1 },
          indexName: 'value_1',
          isMultiKey: false,
          multiKeyPaths: { value: [] },
          isUnique: false,
          isSparse: false,
          isPartial: false,
          indexVersion: 2,
          direction: 'backward',
          indexBounds: { value: [ '[MaxKey, MinKey]' ] }
        }
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 4,
    executionTimeMillis: 0,
    totalKeysExamined: 4,
    totalDocsExamined: 4,
    executionStages: {
      isCached: false,
      stage: 'LIMIT',
      nReturned: 4,
      executionTimeMillisEstimate: 0,
      works: 5,
      advanced: 4,
      needTime: 0,
      needYield: 0,
      queryStats: {}
    }
  }
}
```

Результат без индекса:

```
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'learn.numbers',
    parsedQuery: {},
    indexFilterSet: false,
    queryHash: 'BA27D965',
    planCacheShapeHash: 'BA27D965',
    planCacheKey: '7A892B81',
    optimizationTimeMillis: 0,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'SORT',
      sortPattern: { value: -1 },
      memLimit: 33554432,
      limitAmount: 4,
      type: 'simple',
      inputStage: { stage: 'COLLSCAN', direction: 'forward' }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 4,
    executionTimeMillis: 3,
    totalKeysExamined: 0,
    totalDocsExamined: 4042,
    executionStages: {
      isCached: false,
      stage: 'SORT',
      nReturned: 4,
      executionTimeMillisEstimate: 2,
      works: 4048,
      advanced: 4,
      needTime: 4043,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      sortPattern: { value: -1 },
      memLimit: 33554432,
      limitAmount: 4,
      type: 'simple',
      totalDataSizeSorted: 260,
      usedDisk: false,
      spills: 0,
      spilledDataStorageSize: 0,
      inputStage: {
        stage: 'COLLSCAN',
        nReturned: 4042,
        executionTimeMillisEstimate: 0
      }
    }
  }
}
```

Без индекса происходит полный просмотр коллекции, который занял 3мс executionTimeMillis снизился с ~1 до <1 мс.

Выводы

В ходе выполнения лабораторной работы были изучены основные возможности MongoDB: вставка, выборка, изменение и удаление документов (CRUD), работа с вложенными документами, агрегация данных, создание и использование индексов, а также связи между коллекциями. Полученные навыки позволяют эффективно использовать MongoDB для хранения и обработки данных в реальных приложениях.