

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: студент группы К3239

ФИО: Кабдулвахитов Эмир Ержанович

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.

Практическое задание 2.1.1

Создание базы данных и заполнение коллекции

```
Atlas atlas-j2z2wp-shard-0 [primary] test> use learn
...
switched to db learn
Atlas atlas-j2z2wp-shard-0 [primary] 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: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
... db.unicorns.insert({name: 'Solnere', 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('682de546e66eb3d029c52b79') }
}
Atlas atlas-j2z2wp-shard-0 [primary] learn> var document = {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165};
... db.unicorns.insert(document);
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('682de59ce66eb3d029c52b7a') }
}
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.find()
...
[
  {
    _id: ObjectId('682de546e66eb3d029c52b79'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b70'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b71'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b72'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b73'),
    name: 'Solnere',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b74'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b75'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b76'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b77'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b78'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b79'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f',
    vampires: 54
  },
  {
    _id: ObjectId('682de59ce66eb3d029c52b7a'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }
]
```

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'})
```

```
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})
```

```

Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.find({gender: 'f'}).sort({name: 1}).limit(3)
[...
[
  {
    _id: ObjectId('682de545e66eb3d029c52b70'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b74'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b77'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  }
]
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
[...
{
  _id: ObjectId('682de545e66eb3d029c52b70'),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0})
[...
[
  {
    _id: ObjectId('682de545e66eb3d029c52b6f'),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b71'),
    name: 'Unicrom',
    weight: 984,
    vampires: 182
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b72'),
    name: 'Rooooooodles',
    weight: 575,
    vampires: 99
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b75'),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b76'),
    name: 'Raleigh',
    weight: 421,
    vampires: 2
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b78'),
    name: 'Pilot',
    weight: 650,
    vampires: 54
  },
]

```

Практическое задание 2.2.3

Вывести список единорогов в обратном порядке добавления.

Команды MongoDB:

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

```
weight: 704,
vampires: 165
}
]
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.find().sort({$natural: -1})
[
  {
    _id: ObjectId('682de59ce66eb3d029c52b7a'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b79'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b78'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b77'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b76'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b75'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('682de546e66eb3d029c52b74'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b73'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 650,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b72'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 675,
    gender: 'm',
    vampires: 99
  },
  {
    {
```

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})
```

```
weight: 600,  
gender: 'm',  
vampires: 63  
}  
}  
Atlas atlas-j2z2wp-shard-0 [primary] 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: 'Unicorn',  
    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' },  
]
```

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})
```

Практическое задание 2.3.2

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

Команды MongoDB:

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

Практическое задание 2.3.3

Найти всех единорогов, не имеющих ключ vampires.

Команды MongoDB:

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

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})
```

```

Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 700}}, {_id: 0})
...
[
  {
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.find({gender: 'm', weight: {$gte: 500}, loves: {$all: ['grape', 'lemon']}}, {_id: 0})
[
  {
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
]
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.find({vampires: {$exists: false}})
[
  {
    _id: ObjectId('682de546e66eb3d029c52b79'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
Atlas atlas-j2z2wp-shard-0 [primary] 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})
```



```

Atlas atlas-j2z2wp-shard-0 [primary] learn> db.towns.insert({name: "Punxsutawney", populatiuon: 6200, last_sensus:
... db.towns.insert({name: "New York", populatiuon: 22200000, last_sensus: ISODate("2009-07-31"), famous_for: ["st
... db.towns.insert({name: "Portland", populatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous_for: ["beer
{...
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('682dee6be66eb3d029c52b7d') }
}
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.towns.find({"mayor.party": "I"}, {name: 1, mayor: 1})
{...
[
  {
    _id: ObjectId('682dee6be66eb3d029c52b7c'),
    name: 'New York',
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  }
]
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.towns.find({"mayor.party": {$exists: false}}, {name: 1, mayor: 1})
{...
[
  {
    _id: ObjectId('682dee6be66eb3d029c52b7b'),
    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); });

```

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()

```

Практическое задание 3.2.2

Вывести список предпочтений.

Команды MongoDB:

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

Практическое задание 3.2.3

Посчитать количество особей единорогов обоих полов.

Команды MongoDB:

```
db.unicorns.aggregate([{$group: {_id: "$gender", count: {$sum: 1}}}}])
```

```
Atlas atlas-j2z2wp-shard-0 [primary] learn> var cursor = db.unicorns.find({gender: 'm'}).sort({name: 1}).limit(2);
... cursor.forEach(function(unicorn) { print(unicorn.name); });
[...
Dunx
Horny

Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 600}}).count()
(node:25851) [MONGODB DRIVER] Warning: cursor.count is deprecated and will be removed in the next major version, please
(Use `node --trace-warnings ...` to show where the warning was created)
2
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.distinct("loves")
[...
[
  'apple',      'carrot',
  'chocolate', 'energon',
  'grape',      'lemon',
  'papaya',     'redbull',
  'strawberry', 'sugar',
  'watermelon'
]
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.aggregate([{$group: {_id: "$gender", count: {$sum: 1}}}}])
[...
[ { _id: 'f', count: 5 }, { _id: 'm', count: 7 } ]
```

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"})
```

```
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.save({name: "Barny", loves: ["grape"], weight: 340, gender: "m"})
TypeError: db.unicorns.save is not a function
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.insertOne({
...   name: "Barny",
...   loves: ["grape"],
...   weight: 340,
...   gender: "m"
... })
[...
{
  acknowledged: true,
  insertedId: ObjectId('682e04c2e66eb3d029c52b7e')
}
Atlas atlas-j2z2wp-shard-0 [primary] learn> █
```

Практическое задание 3.3.2

Обновить Айна: вес 800, вампиры 51.

Команды MongoDB:

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

Практическое задание 3.3.3

Обновить Raleigh: добавить redbull в loves.

Команды MongoDB:

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

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})
```

Практическое задание 3.3.5

Убрать партию у мэра Портланда.

Команды MongoDB:

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

Практическое задание 3.3.6

Обновить Pilot: добавить chocolate в loves.

Команды MongoDB:

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

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"]}}})
```

```
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.update({name: "Ayna"}, {$set: {weight: 800, vampires: 51}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.update({name: "Raleigh"}, {$push: {loves: "redbull"}})
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.update({gender: "m"}, {$inc: {vampires: 5}}, {multi: true})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 8,
  modifiedCount: 8,
  upsertedCount: 0
}
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.towns.update({name: "Portland"}, {$unset: {"mayor.party": 1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.update({name: "Pilot"}, {$push: {loves: "chocolate"}})
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.update({name: "Aurora"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}})
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-j2z2wp-shard-0 [primary] learn> █
```

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
```

```
[Atlas atlas-j2z2wp-shard-0 [primary] learn> db.towns.remove({"mayor.party": {$exists: false}})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 2 }
[Atlas atlas-j2z2wp-shard-0 [primary] learn> db.towns.find()
[
  {
    _id: ObjectId('682dee6be66eb3d029c52b7c'),
    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' }
  }
]
[Atlas atlas-j2z2wp-shard-0 [primary] learn> db.towns.remove({})
{ acknowledged: true, deletedCount: 1 }
[Atlas atlas-j2z2wp-shard-0 [primary] learn> show towns
MongoshInvalidInputError: [COMMON-10001] 'towns' is not a valid argument for "show".
[Atlas atlas-j2z2wp-shard-0 [primary] learn> show collections
towns
unicorns
[Atlas atlas-j2z2wp-shard-0 [primary] learn> db.towns.find()
Atlas atlas-j2z2wp-shard-0 [primary] 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: "nw", name: "Northwest", desc: "Forests and rivers"});
```

```
db.habitats.insert({_id: "desert", name: "Desert", desc: "Hot and sandy"});
```

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

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

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

```

Atlas atlas-j2z2wp-shard-0 [primary] learn> db.habitats.insert({_id: "nw", name: "Northwest", desc: "Forests and rivers"});
... db.habitats.insert({_id: "desert", name: "Desert", desc: "Hot and sandy"});
... db.unicorns.update({name: "Horny"}, {$set: {habitat: {$ref: "habitats", $id: "nw"}}});
... db.unicorns.update({name: "Aurora"}, {$set: {habitat: {$ref: "habitats", $id: "desert"}}});
... db.unicorns.find();
[...
[
  {
    _id: ObjectId('682de545e66eb3d029c52b6f'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 68,
    habitat: DBRef('habitats', 'nw')
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b70'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43,
    habitat: DBRef('habitats', 'desert')
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b71'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 187
  },
  {
    _id: ObjectId('682de545e66eb3d029c52b72'),
    name: 'Rooooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 104
  },
]
]

```

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})
```

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_");
```

```

Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.createIndex({name: 1}, {unique: true})
name_1
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.getIndexes();
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1', unique: true }
]
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.dropIndexes();
[...
{
  nIndexesWas: 2,
  msg: 'non-_id indexes dropped for collection',
  ok: 1,
  'clusterTime': {
    clusterTime: Timestamp({ t: 1747847966, i: 5 }),
    signature: {
      hash: Binary.createFromBase64('2a1c0xF+HTd8CZFdwzshN8gTPq4=', 0),
      keyId: Long('7475079124839563267')
    }
  },
  operationTime: Timestamp({ t: 1747847966, i: 5 })
}
]
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.getIndexes();
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.unicorns.dropIndex("_id_")
MongoServerError[InvalidOptions]: cannot drop _id index
Atlas atlas-j2z2wp-shard-0 [primary] 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:

```
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.numbers.explain("executionStats").find().sort({value: -1}).limit(4)
{
  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,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      limitAmount: 4,
      inputStage: {
        stage: 'FETCH',
        nReturned: 4,
        executionTimeMillisEstimate: 0,
        works: 4,
        advanced: 4,
        needTime: 0,
        needYield: 0,
        saveState: 0,
        restoreState: 0,
        isEOF: 0,
        docsExamined: 4,
        alreadyHasObj: 0,
        inputStage: {
          stage: 'IXSCAN',
          nReturned: 4,
          executionTimeMillisEstimate: 0,
```


Результат без индекса:

```
Atlas atlas-j2z2wp-shard-0 [primary] learn> db.numbers.explain("executionStats").find().sort({ value: -1 }).limit(4)
{
  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: 2,
        works: 4043,
        advanced: 4042,
        needTime: 0,
        needYield: 0,
        saveState: 0,
        restoreState: 0,
        isEOF: 1,
        direction: 'forward',
        docsExamined: 4042
      }
    }
  },
  queryShapeHash: '1A6A6507B0E0A6B2042E58131791FAF9815BD75FD975CD5C1A95CD9B0CF568B8',
  command: {
    find: 'numbers',
    filter: {},
    sort: { value: -1 },
    limit: 4,
  }
}
```

Без индекса происходит полный просмотр коллекции, который занял 3мс executionTimeMillis, а с индексом происходит индексный поиск, который занял менее 1мс.

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.