

Министерство науки и высшего образования Российской
Федерации

федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Лабораторная работа №5
«Работа с БД в СУБД MongoDB»

Выполнил:

Чухонин Иван Андреевич
Группа К32421

Преподаватель:

Говорова Мария Михайловна

Санкт-Петербург
2023

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.

Практическое задание:

1. Установите MongoDB для обеих типов систем (32/64 бита).
2. Проверьте работоспособность системы запуском клиента mongo.
3. Выполните методы:
 - a) db.help()
 - b) db.help
 - c) db.stats()
4. Создайте БД learn.
5. Получите список доступных БД.
6. Создайте коллекцию unicorns, вставив в нее документ {name: 'Aurora', gender: 'f', weight: 450}.
7. Просмотрите список текущих коллекций.
8. Переименуйте коллекцию unicorns.
9. Просмотрите статистику коллекции.
10. Удалите коллекцию.
11. Удалите БД learn.

Практическая часть

8.1.1

- 1) создайте базу данных *learn*.
use learn
- 2) Заполните коллекцию единорогов *unicorns*:

```
learn> db.unicorns.drop()
true
learn> db.unicorns.stats()
MongoServerError: PlanExecutor error during aggregation :: caused by :: Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [learn.unicorns] not found.
... {name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 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, gender: 'f', vampires: 80},
... {name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40},
... {name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39},
... {name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2},
... {name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33},
... {name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54},
... {name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 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, gender: 'f', vampires: 80},
... {name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40},
... {name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39},
... {name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2},
... {name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33},
... {name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54},
... {name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'}}]
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6475207c59cf2c6f9b510a0b"),
    '1': ObjectId("6475207c59cf2c6f9b510a0c"),
    '2': ObjectId("6475207c59cf2c6f9b510a0d"),
    '3': ObjectId("6475207c59cf2c6f9b510a0e"),
    '4': ObjectId("6475207c59cf2c6f9b510a0f"),
    '5': ObjectId("6475207c59cf2c6f9b510a10"),
    '6': ObjectId("6475207c59cf2c6f9b510a11"),
    '7': ObjectId("6475207c59cf2c6f9b510a12"),
    '8': ObjectId("6475207c59cf2c6f9b510a13"),
    '9': ObjectId("6475207c59cf2c6f9b510a14"),
    '10': ObjectId("6475207c59cf2c6f9b510a15")
  }
}
```

рисунок №1 - заполнение коллекции единорогов

3) *Используя второй способ, вставьте в коллекцию единорогов документ:*

```
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("6475230a59cf2c6f9b510a16") }
}
learn>
```

рисунок №2 - заполнение коллекции единорогов вторым способом

4) Проверьте содержимое коллекции с помощью метода *find*.

```
learn> db.unicorns.find()
[
  {
    _id: ObjectId("6475207c59cf2c6f9b510a0b"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a0c"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a0d"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a0e"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a0f"),
    name: 'Solnara', internal pages': 0,
    loves: [ 'apple', 'carrot', 'chocolate' ], values': 0,
    weight: 550, ore variable-size deleted values': 0,
    gender: 'f', ore variable-size leaf pages': 0,
    vampires: 80 ord size': 0,
    'maximum internal page size': 4096,
  },
]
```

рисунок №3 - содержимое коллекции единорогов

8.1.2

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

```
learn> db.unicorns.find({gender: 'm'}).limit(3).sort({name: 1})
```

	_id ObjectId	name String	loves Array	weight Int32	gender String
1	ObjectId('6475207c59cf2c6f9b5...	"Horny"	[] 2 elements	600	"m"
2	ObjectId('6475207c59cf2c6f9b5...	"Unicrom"	[] 2 elements	984	"m"
3	ObjectId('6475207c59cf2c6f9b5...	"Rooooooodles"	[] 1 elements	575	"m"

рисунок №4 - запрос выдаёт трёх самцов отсортированных по имени

```
learn> db.unicorns.find({gender: 'f'}).limit(3).sort({name: 1})
```

	_id ObjectId	name String	loves Array	weight Int32	gender String
1	ObjectId('6475207c59cf2c6f9b5...	"Aurora"	[] 2 elements	450	"f"
2	ObjectId('6475207c59cf2c6f9b5...	"Solnara"	[] 3 elements	550	"f"
3	ObjectId('6475207c59cf2c6f9b5...	"Ayna"	[] 2 elements	733	"f"

рисунок №5 - запрос выдаёт трёх самок отсортированных по имени

2) Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций findOne и limit.

```
learn> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
{
  _id: ObjectId("6475207c59cf2c6f9b510a0c"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
```

рисунок №6 - результат запроса 8.1.2

8.1.3

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

```
learn> db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0})
[
  {
    _id: ObjectId("6475207c59cf2c6f9b510a0b"),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a0d"),
    name: 'Unicrom',
    weight: 984,
    vampires: 182
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a0e"),
    name: 'Roooooodles',
    weight: 575,
    vampires: 99
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a11"),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a12"),
    name: 'Raleigh',
    weight: 421,
    vampires: 2
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a14"),
    name: 'Pilot',
    weight: 650,
    vampires: 54
  },
  {
    _id: ObjectId("6475230a59cf2c6f9b510a16"),
    name: 'Dunx',
    weight: 704,
    vampires: 165
  }
]
```

рисунок №7 - результат запроса 8.1.3

8.1.4

1) Вывести список единорогов в обратном порядке добавления.

```
learn> db.unicorns.find().sort({$natural: -1})
[
  {
    _id: ObjectId("6475230a59cf2c6f9b510a16"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a15"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a14"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a13"),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a12"),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId("6475207c59cf2c6f9b510a11"),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
```

рисунок №8 - список единорогов в обратном порядке добавления

8.1.5

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

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

рисунок №9 - выполнение запроса

8.1.6

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

```
learn> 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'
  }
]
```

рисунок №10 - выполнение запроса

8.1.7

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

```
learn> 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
  }
]
```

рисунок №11 - выполнение запроса

8.1.8

1) Найти всех единорогов, не имеющих ключ *vampires*.

```
learn> db.unicorns.find({vampires: {$exists:0}})
[
  {
    _id: ObjectId("6475207c59cf2c6f9b510a15"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
```

рисунок №12 - выполнение запроса

8.1.9

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

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

рисунок №13 - выполнение запроса

8.2.1

1) Создайте коллекцию `towns`, включающую следующие документы:

```
learn> db.towns.insertMany([
...     name: "Portland",
...     populatiuon: 528000,
...     last_sensus: ISODate("2009-07-20"),
...     famous_for: ["beer", "food"],
...     mayor: {
...         name: "Sam Adams",
...         party: "D"
...     }
... }, {
...     name: "Punxsutawney ",
...     populatiuon: 6200,
...     last_sensus: ISODate("2008-01-31"),
...     famous_for: [""],
...     mayor: {
...         name: "Jim Wehrle"
...     }
... }, {
...     name: "New York",
...     populatiuon: 22200000,
...     last_sensus: ISODate("2009-07-31"),
...     famous_for: ["status of liberty", "food"],
...     mayor: {
...         name: "Michael Bloomberg",
...         party: "I"
...     }
... })
```

рисунок №14 - выполнение запроса

2) Сформировать запрос, который возвращает список городов с независимыми мэрами (`party="I"`) . Вывести только название города и информацию о мэре.

```
learn> db.towns.find({"mayor.party": 'I'}, {mayor:1, name:1})
[
  {
    _id: ObjectId("6477bc8eb63b716a4410ad32"),
    name: 'New York',
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  }
]
```

рисунок №15 - выполнение запроса

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

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

рисунок №16 - выполнение запроса

8.2.2

1) Сформировать функцию для вывода списка самцов единорогов.

```
learn> male = function() {return this.gender == 'm'}
[Function: male]
```

рисунок №17 - выполнение запроса

2) Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.

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

рисунок №18 - выполнение запроса

3) Вывести результат, используя forEach.

```
learn> cursor.forEach(function (obj) {print(obj.name);})
Dunx
Horny
```

рисунок №19 - выполнение запроса

8.2.3

1) Вывести количество самок единорогов весом от полутонны до 600 кг.

```
learn> db.unicorns.find({weight: {$gt: 500, $lt: 600}}).count()
3
```

рисунок №20 - выполнение запроса

8.2.4

1) Вывести список предпочтений.

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

рисунок №21 - выполнение запрос

8.2.5

1) Подсчитать количество особей единорогов обоих полов.

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

8.2.6

1. Выполнить команду:

```
learn> db.unicorns.save({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})
TypeError: db.unicorns.save is not a function
learn> db.unicorns.insertOne({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})
{
  acknowledged: true,
  insertedId: ObjectId("647a0b4ef7b7ab35d86174ed")
}
```

```
{
  _id: ObjectId("647a0b4ef7b7ab35d86174ed"),
  name: 'Barney',
  loves: [ 'grape' ],
  weight: 340,
  gender: 'm'
}
```

рисунок №22 - выполнение запроса

2. Проверить содержимое коллекции *unicorns*.

	_id ObjectId	name String	loves Array	weight Int32	gender String
1	ObjectId('6475207c59cf2c6f9b5...	"Horny"	[] 2 elements	600	"m"
2	ObjectId('6475207c59cf2c6f9b5...	"Aurora"	[] 2 elements	450	"f"
3	ObjectId('6475207c59cf2c6f9b5...	"Unicrom"	[] 2 elements	984	"m"
4	ObjectId('6475207c59cf2c6f9b5...	"Roooooodles"	[] 1 elements	575	"m"
5	ObjectId('6475207c59cf2c6f9b5...	"Solnara"	[] 3 elements	550	"f"
6	ObjectId('6475207c59cf2c6f9b5...	"Ayna"	[] 2 elements	733	"f"
7	ObjectId('6475207c59cf2c6f9b5...	"Kenny"	[] 2 elements	690	"m"
8	ObjectId('6475207c59cf2c6f9b5...	"Raleigh"	[] 2 elements	421	"m"
9	ObjectId('6475207c59cf2c6f9b5...	"Leia"	[] 2 elements	601	"f"
10	ObjectId('6475207c59cf2c6f9b5...	"Pilot"	[] 2 elements	650	"m"
11	ObjectId('6475207c59cf2c6f9b5...	"Nimue"	[] 2 elements	540	"f"
12	ObjectId('6475230a59cf2c6f9b5...	"Dunx"	[] 2 elements	704	"m"
13	ObjectId('647a0b4ef7b7ab35d86...	"Barny"	[] 1 elements	340	"m"

рисунок №23 - выполнение запроса в MongoDB Compass

8.2.7

1. Для самки единорога *Ayna* внести изменения в БД: теперь ее вес 800, она убила 51 вампира.

```
learn> db.unicorns.updateOne({name: 'Ayna'}, {$set: {weight: 800, vampires: 51}}, {upsert: 0})
```

2. Проверить содержимое коллекции *unicorns*.

```
_id: ObjectId('6475207c59cf2c6f9b510a10')
name: "Ayna"
▶ loves: Array
weight: 800
gender: "f"
vampires: 51
```

рисунок №24 - выполнение запроса

8.2.8

- 1) Для самца единорога *Raleigh* внести изменения в БД: теперь он любит рэдбул.

```
learn> db.unicorns.update({name: 'Raleigh'}, {$set: {loves: "redbull"}}, {upsert: 0})
```

2) Проверить содержимое коллекции *unicorns*.

```
_id: ObjectId('6475207c59cf2c6f9b510a12')
name: "Raleigh"
loves: "redbull"
weight: 421
gender: "m"
vampires: 2
```

8.2.9

- 1) Всем самцам единорогов увеличить количество убитых вампиров на 5.

```
learn> db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}})
```

рисунок №25 - выполнение запроса

- 2) Проверить содержимое коллекции *unicorns*.

		name String	loves Mixed	weight Int32	gender String	vampires
1	7c59cf2c6f9b5...	"Horny"	[] 2 elements	600	"m"	68
2	7c59cf2c6f9b5...	"Unicrom"	[] 2 elements	984	"m"	187
3	7c59cf2c6f9b5...	"Roooooodles"	[] 1 elements	575	"m"	104
4	7c59cf2c6f9b5...	"Kenny"	[] 2 elements	690	"m"	44
5	7c59cf2c6f9b5...	"Raleigh"	"redbull"	421	"m"	7
6	7c59cf2c6f9b5...	"Pilot"	[] 2 elements	650	"m"	59
7	0a59cf2c6f9b5...	"Dunx"	[] 2 elements	704	"m"	170
8	4ef7b7ab35d86...	"Barny"	[] 1 elements	340	"m"	5

рисунок №26 - выполнение запроса в MongoDB Compass

8.2.10

- 1) Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.

```
learn> db.towns.updateOne({name: 'Portland'}, {$unset: {"mayor.party": 1}}, {upsert: 0})
```

рисунок №27 - выполнение запроса

- 2) Проверить содержимое коллекции *towns*.

```
_id: ObjectId('6477bc8eb63b716a4410ad33')
name: "Portland"
populatiuon: 528000
last_sensus: 2009-07-20T00:00:00.000+00:00
▶ famous_for: Array
▼ mayor: Object
  name: "Sam Adams"
```

рисунок №28 - выполнение запроса в MongoDB Compass

8.2.11

- 1) Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.

```
learn> db.unicorns.updateOne({name: 'Pilot'}, {$push: {loves: 'chocolate'}})
```

рисунок №29 - выполнение запроса

2) Проверить содержимое коллекции *unicorns*.

```
_id: ObjectId('6475207c59cf2c6f9b510a14')
name: "Pilot"
▼ loves: Array
  0: "apple"
  1: "watermelon"
  2: "chocolate"
weight: 650
gender: "m"
vampires: 59
```

рисунок №30 - выполнение запроса в MongoDB Compass

8.2.12

1) Изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и лимоны.

```
learn> db.unicorns.updateOne({name: 'Aurora'}, {$push: {loves: {$each: ['sugar', 'lemons']}}})
```

2) Проверить содержимое коллекции *unicorns*.

```
_id: ObjectId('6475207c59cf2c6f9b510a0c')
name: "Aurora"
▼ loves: Array
  0: "carrot"
  1: "grape"
  2: "sugar"
  3: "lemons"
weight: 450
gender: "f"
vampires: 43
```

рисунок №31 - выполнение запроса в MongoDB Compass

8.2.13

1) Создайте коллекцию *towns*, включающую следующие документы:

```
db.towns.insertMany([
  {name: "Punxsutawney ",
  population: 6200,
  last_sensus: ISODate("2008-01-31"),
  famous_for: ["phil the groundhog"],
  mayor: {name: "Jim Wehrle"}},
```



```
{name: "New York", population: 22200000,
last_sensus: ISODate("2009-07-31"),
famous_for: ["statue of liberty", "food"],
mayor: {name: "Michael Bloomberg", party: "I"}},

{name: "Portland", population: 528000,
last_sensus: ISODate("2009-07-20"),
famous_for: ["beer", "food"],
mayor: {name: "Sam Adams", party: "D"}}})
```

2) Удалите документы с беспартийными мэрами.

```
learn> db.towns.deleteMany({"mayor.party": {$exists: 0}})
{ acknowledged: true, deletedCount: 1 }
```

рисунок №32 - выполнение запроса

3) Проверьте содержание коллекции.

```
learn> db.towns.find({}, {mayor: 1})
[
  {
    _id: ObjectId("647c4f4416928386cbf82aee"),
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  },
  {
    _id: ObjectId("647c4f4416928386cbf82aef"),
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
```

рисунок №33 - выполнение запроса

4) Очистите коллекцию.

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

5) Просмотрите список доступных коллекций.

```
learn> show collections
towns
unicorns
```

8.3.1

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

```
1 db.towns.insertMany([
2   |   |   { _id: 'ru', name: "Russia", description: 'cold place'},
3   |   |   { _id: 'us', name: "USA", description: 'comfortable weather'},
4   |   |   { _id: 'aus', name: "Australia", description: 'hot, dangerous place'}
5   |   | ])
```

рисунок №34 - запрос

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

```
learn> db.unicorns.updateOne({name: 'Pilot'}, {$set: {habitat: {$ref:"habitat", $id: 'ru'}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({name: 'Dunx'}, {$set: {habitat: {$ref:"habitat", $id: 'aus'}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

рисунок №35 - выполнение запроса

3) Проверьте содержание коллекции единорогов.

```
learn> db.unicorns.find({$or: [{name: 'Pilot'}, {name: 'Dunx'}]})
[
  {
    _id: ObjectId("6475207c59cf2c6f9b510a14"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 650,
    gender: 'm',
    vampires: 59,
    habitat: DBRef("habitat", 'ru')
  },
  {
    _id: ObjectId("6475230a59cf2c6f9b510a16"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 170,
    habitat: DBRef("habitat", 'aus')
  }
]
```

рисунок №37 - выполнение запроса

8.3.2

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

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

рисунок №38 - выполнение запроса

8.3.3

- 1) *Получите информацию и удалите все индексы, кроме индекса для идентификатора.*

```
learn> db.unicorns.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1', unique: true }
]
```

рисунок №39 - выполнение запроса

- 2) *Попытайтесь удалить индекс для идентификатора.*

```
learn> db.unicorns.dropIndexes("name_1")
{ nIndexesWas: 2, ok: 1 }
learn> db.unicorns.dropIndexes("_id_")
MongoServerError: cannot drop _id index
```

рисунок №40 - выполнение запроса

8.3.4

- 1) *Создайте объемную коллекцию `numbers`, задействовав курсор:*

```
learn> arr = []; for(i = 0; i < 100000; i++){arr.push({value: i})} db.numbers.insertMany(arr)
```

- 2) *Выберите последних четыре документа.*

```
learn> db.numbers.find().sort({value: -1}).limit(4)
[
  { _id: ObjectId("647cc09f16928386cbfb382f"), value: 99999 },
  { _id: ObjectId("647cc09f16928386cbfb382e"), value: 99998 },
  { _id: ObjectId("647cc09f16928386cbfb382d"), value: 99997 },
  { _id: ObjectId("647cc09f16928386cbfb382c"), value: 99996 }
]
```

рисунок №41 - выполнение запроса

- 3) *Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? По значению параметра `executionTimeMillis` потребовалось 117 секунд.*

```
executionStats: {
  executionSuccess: true,
  nReturned: 4,
  executionTimeMillis: 117,
  totalKeysExamined: 0,
  totalDocsExamined: 100000,
```

рисунок №42 - выполнение запроса

4) Создайте индекс для ключа *value*.

```
learn> db.numbers.ensureIndex({value:1})
[ 'value_1' ]
```

рисунок №43 - выполнение запроса

5) Получите информацию о всех индексах коллекции *numbers*.

Name and Definition	Type	Size
> _id_	REGULAR ⓘ	1.0 MB
> value_1	REGULAR ⓘ	983.0 KB

рисунок №44 - список индексов в mongodb compass

6) Выполните запрос 2.

```
learn> db.numbers.find().sort({value: -1}).limit(4).explain("executionStats")
```

```
executionStats: {
  executionSuccess: true,
  nReturned: 4,
  executionTimeMillis: 2,
  totalKeysExamined: 4,
  totalDocsExamined: 4,
```

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

executionTimeMillis: 2

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

Запрос с индексом оказался значительно эффективнее, чем запрос без индекса(2:117)

Выводы: В данной лабораторной работе была освоена работа с MongoDB, изучены основные методы и функции: добавления и удаления элементов в коллекции, обновление, выборка по количеству и другим параметрам, операторы, анализ индексов. Работа была выполнена при помощи mongosh.