

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»
по дисциплине:
«Проектирование и реализация баз данных»

Выполнила:
студентка 2 курса ФИКТ
группы K3241
Барышева З.А.

Проверила:
Говорова М.В.

Санкт-Петербург
2022

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.

1 CRUD-ОПЕРАЦИИ В СУБД MONGODB. ВСТАВКА ДАННЫХ. ВЫБОРКА ДАННЫХ

1.1 ВСТАВКА ДОКУМЕНТОВ В КОЛЛЕКЦИЮ

1) Создайте базу данных learn.

2) Заполните коллекцию единорогов unicorns:

```
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: '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'});
```

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

```
{name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}
```

4) Проверьте содержимое коллекции с помощью метода find.

1-2:

```
test> use learn
switched to db learn
learn> db.unicorns.insert({name: 'Horny', loves: ['carrot', 'papaya'], weight: 600, gender: 'm', vampires: 63});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63457b3c8cbce4d96074e563") }
}
learn> db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63457b3c8cbce4d96074e564") }
}
learn> db.unicorns.insert({name: 'Uncrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63457b3c8cbce4d96074e565") }
}
learn> db.unicorns.insert({name: 'Rooooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63457b3c8cbce4d96074e566") }
}
learn> db.unicorns.insert({name: 'Solnara', loves: ['apple', 'carrot', 'chocolate'], weight: 550, gender: 'f', vampires: 80});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63457b3c8cbce4d96074e567") }
}
learn> db.unicorns.insert({name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63457b3c8cbce4d96074e568") }
}
learn> db.unicorns.insert({name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63457b3c8cbce4d96074e569") }
}
learn> db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63457b3c8cbce4d96074e56a") }
}
learn> db.unicorns.insert({name: 'Lela', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63457b3c8cbce4d96074e56b") }
}
learn> db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63457b3c8cbce4d96074e56c") }
}
learn> db.unicorns.insert({name: 'Ninue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63457b3c8cbce4d96074e56d") }
}
```

3:

```
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.insertOne(document)
{
  acknowledged: true,
  insertedId: ObjectId("63457ef2c7f6a39990990fa4")
}
```

4:

```
learn> db.unicorns.find()
[
  {
    _id: ObjectId("63457b3b8cbce4d96074e563"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e564"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e565"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e566"),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e567"),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e568"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
]
```

```

},
{
  _id: ObjectId("63457b3c8cbce4d96074e569"),
  name: 'Kenny',
  loves: [ 'grape', 'lemon' ],
  weight: 690,
  gender: 'm',
  vampires: 39
},
{
  _id: ObjectId("63457b3c8cbce4d96074e56a"),
  name: 'Raleigh',
  loves: [ 'apple', 'sugar' ],
  weight: 421,
  gender: 'm',
  vampires: 2
},
{
  _id: ObjectId("63457b3c8cbce4d96074e56b"),
  name: 'Leia',
  loves: [ 'apple', 'watermelon' ],
  weight: 601,
  gender: 'f',
  vampires: 33
},
{
  _id: ObjectId("63457b3c8cbce4d96074e56c"),
  name: 'Pilot',
  loves: [ 'apple', 'watermelon' ],
  weight: 650,
  gender: 'm',
  vampires: 54
},
{
  _id: ObjectId("63457b3c8cbce4d96074e56d"),
  name: 'Nimue',
  loves: [ 'grape', 'carrot' ],
  weight: 540,
  gender: 'f'
},
{
  _id: ObjectId("63457ef2c7f6a39990990fa4"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
]
learn>

```

1.2 ВЫБОРКА ДАННЫХ ИЗ БД

1.2

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

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

1:

Список самцов, отсортированный по имени:

```
]
learn> db.unicorns.find({gender: 'm'}).sort({name: 1})
[
  {
    _id: ObjectId("63457ef2c7f6a39990990fa4"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId("63457b3b8cbce4d96074e563"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e569"),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e56c"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e56a"),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e566"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e565"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  }
]
```

Первые три самки списка:

```
learn> db.unicorns.find({gender: 'f'}).sort({name: 1}).limit(3)
[
  {
    _id: ObjectId("63457b3c8cbce4d96074e564"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e568"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e56b"),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  }
]
```

2:

Самки, которые любят морковку:

```
learn> db.unicorns.find({gender: 'f', loves: 'carrot'})
[
  {
    _id: ObjectId("63457b3c8cbce4d96074e564"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e567"),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e56d"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
```

Одна самка:

```
learn> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
{
  _id: ObjectId("63457b3c8cbce4d96074e564"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
learn> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
[
  {
    _id: ObjectId("63457b3c8cbce4d96074e564"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
learn>
```

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}).sort({name: 1})
[
  {
    _id: ObjectId("63457ef2c7f6a39990990fa4"),
    name: 'Dunx',
    weight: 704,
    vampires: 165
  },
  {
    _id: ObjectId("63457b3b8cbce4d96074e563"),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e569"),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e56c"),
    name: 'Pilot',
    weight: 650,
    vampires: 54
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e56a"),
    name: 'Raleigh',
    weight: 421,
    vampires: 2
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e566"),
    name: 'Roooooodles',
    weight: 575,
    vampires: 99
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e565"),
    name: 'Unicrom',
    weight: 984,
    vampires: 182
  }
]
```

1.4

- 1) Вывести список единорогов в обратном порядке добавления.

Начало списка:

```
learn> db.unicorns.find().sort({ $natural: -1})
[
  {
    _id: ObjectId("63457ef2c7f6a39990990fa4"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e56d"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },

```

Конец списка:

```

  {
    _id: ObjectId("63457b3c8cbce4d96074e565"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e564"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("63457b3b8cbce4d96074e563"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
]
learn>

```

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

1.6 ЛОГИЧЕСКИЕ ОПЕРАТОРЫ

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

```
learn> db.unicorns.find({gender: 'f', weight: {$gte: 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'
  }
]
```

1.7

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

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

1.8

1) Найти всех единорогов, не имеющих ключ vampires.

```
learn> db.unicorns.find({vampires: {$exists: false}})
[
  {
    _id: ObjectId("63457b3c8cbce4d96074e56d"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
```

1.9

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

```
learn> 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: 'Rooooooodles', loves: [ 'apple' ] },
  { name: 'Unicrom', loves: [ 'energon' ] }
]
```

2 ЗАПРОСЫ К БАЗЕ ДАННЫХ MONGODB. ВЫБОРКА ДАННЫХ. ВЛОЖЕННЫЕ ОБЪЕКТЫ. ИСПОЛЬЗОВАНИЕ КУРСОРОВ. АГРЕГИРОВАННЫЕ ЗАПРОСЫ. ИЗМЕНЕНИЕ ДАННЫХ

2.1 ЗАПРОС К ВЛОЖЕННЫМ ОБЪЕКТАМ

1) Создайте коллекцию towns, включающую следующие документы:

```
{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"}}

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

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

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

2:

```
towns> db.towns.find({'mayor.paty': 'I'}, {name: 1, mayor: 1, _id: 0})
[
  { name: 'New York', mayor: { name: 'Michael Bloomberg', paty: 'I' } }
]
```

3:

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

2.2 КУРСОРЫ И ИСПОЛЬЗОВАНИЕ JAVASCRIPT

- 1) Сформировать функцию для вывода списка самцов единорогов.
- 2) Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.
- 3) Вывести результат, используя forEach.

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

2.3 АГРЕГИРОВАННЫЕ ЗАПРОСЫ

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

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

2.4

- 1) Вывести список предпочтений.

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

2.5

- 1) Посчитать количество особей единорогов обоих полов.

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

2.6 РЕДАКТИРОВАНИЕ ДАННЫХ

- 1) Выполнить команду:
> db.unicorns.save({name: 'Barny', loves: ['grape'],
weight: 340, gender: 'm'})
- 2) Проверить содержимое коллекции unicorns.

```
learn> db.unicorns.save({name: 'Barney', loves: ['grape'],
... weight: 340, gender: 'm'})
TypeError: db.unicorns.save is not a function
learn>
```

2.7

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

```
learn> db.unicorns.updateOne({name: 'Ayna'}, {$set: {weight: 800, vampires: 51}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.findOne({name: 'Ayna'})
{
  _id: ObjectId("63457b3c8cbce4d96074e568"),
  name: 'Ayna',
  loves: [ 'strawberry', 'lemon' ],
  weight: 800,
  gender: 'f',
  vampires: 51
}
```

2.8

- 1) Для самца единорога Raleigh внести изменения в БД: теперь он любит рэдбул.
- 2) Проверить содержимое коллекции unicorns.

```
learn> db.unicorns.updateOne({name: 'Raleigh'}, {$set: {loves: ['redbull']}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.findOne({name: 'Raleigh'})
{
  _id: ObjectId("63457b3c8cbce4d96074e56a"),
  name: 'Raleigh',
  loves: [ 'redbull' ],
  weight: 421,
  gender: 'm',
  vampires: 2
}
```

2.9

- 1) Всем самцам единорогов увеличить количество убитых вампиров на 5.
- 2) Проверить содержимое коллекции unicorns.

```

learn> db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 7,
  modifiedCount: 7,
  upsertedCount: 0
}
learn> db.unicorns.find({gender: 'm'})
[
  {
    _id: ObjectId("63457b3b8cbce4d96074e563"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 68
  },
  {

```

2.10

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

```

towns> db.towns.updateOne({name: 'Portland'}, {$unset: {'mayor.paty': 1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
towns> db.towns.find()
[
  {
    _id: ObjectId("6345c2f4c7f6a39990990fa5"),
    name: 'Punxsutawney ',
    populatiuon: 6200,
    last_sensus: ISODate("2008-01-31T00:00:00.000Z"),
    famous_for: [ '' ],
    mayor: { name: 'Jim Wehrle' }
  },
  {
    _id: ObjectId("6345c38dc7f6a39990990fa6"),
    name: 'New York',
    populatiuon: 22200000,
    last_sensus: ISODate("2009-07-31T00:00:00.000Z"),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', paty: 'I' }
  },
  {
    _id: ObjectId("6345c3f8c7f6a39990990fa7"),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams' }
  }
]

```


2.11

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

```
learn> db.unicorns.updateOne({name: 'Pilot'}, {$push: {loves: ['chocolate']}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.findOne({name: 'Pilot'})
{
  _id: ObjectId("63457b3c8cbce4d96074e56c"),
  name: 'Pilot',
  loves: [ 'apple', 'watermelon', [ 'chocolate' ] ],
  weight: 650,
  gender: 'm',
  vampires: 59
}
```

2.12

- 1) Изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и лимоны.
- 2) Проверить содержимое коллекции unicorns.

```
learn> db.unicorns.updateOne({name: 'Aurora'}, {$addToSet: {loves: {$each: ['sugar', 'lemon']}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.findOne({name: 'Aurora'})
{
  _id: ObjectId("63457b3c8cbce4d96074e564"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
```

2.13 УДАЛЕНИЕ ДАННЫХ ИЗ КОЛЛЕКЦИИ

- 1) Создайте коллекцию towns, включающую следующие документы:

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

```
{name: "New York",
popujatiuon: 22200000,
```

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

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

- 2) Удалите документы с беспартийными мэрами.
- 3) Проверьте содержание коллекции.
- 4) Очистите коллекцию.
- 5) Просмотрите список доступных коллекций.

1-3:

```
towns> db.towns.remove({'mayor.paty': {$exists: false}})
{ acknowledged: true, deletedCount: 1 }
towns> db.towns.find()
[
  {
    _id: ObjectId("6345d15dc7f6a39990990fa9"),
    name: 'New York',
    populatiuon: 22200000,
    last_sensus: ISODate("2009-07-31T00:00:00.000Z"),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', paty: 'I' }
  },
  {
    _id: ObjectId("6345d16ac7f6a39990990faa"),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', paty: 'D' }
  }
]
```

4-5:

```
towns> db.towns.remove({})
{ acknowledged: true, deletedCount: 2 }
towns> show collections
towns
```

3 ССЫЛКИ И РАБОТА С ИНДЕКСАМИ В БАЗЕ ДАННЫХ MONGODB

3.1 ССЫЛКИ В БД

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

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

3) Проверьте содержание коллекции единорогов.

1:

```
learn> db.areas.insertOne({_id: 'w', name: 'Winterfell', description: 'Winter is coming'})
{ acknowledged: true, insertedId: 'w' }
learn> db.areas.insertOne({_id: 'kl', name: "King's Landing", description: 'Flames and blood'})
{ acknowledged: true, insertedId: 'kl' }
learn> db.areas.insertOne({_id: 'cc', name: "Casterly Cliff", description: 'Hear my roar'})
{ acknowledged: true, insertedId: 'cc' }
learn> db.areas.find()
[
  { _id: 'w', name: 'Winterfell', description: 'Winter is coming' },
  {
    _id: 'kl',
    name: "King's Landing",
    description: 'Flames and blood'
  },
  { _id: 'cc', name: 'Casterly Cliff', description: 'Hear my roar' }
]
```

2-3:

```
learn> db.unicorns.updateOne({name: 'Aurora'}, {$set: {area: {$ref: "areas", $id: 'w'}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({name: 'Nimue'}, {$set: {area: {$ref: "areas", $id: 'kl'}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({name: 'Raleigh'}, {$set: {area: {$ref: "areas", $id: 'cc'}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,

```

```

}
learn> db.unicorns.find({area: {$exists: true}})
[
  {
    _id: ObjectId("63457b3c8cbce4d96074e564"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43,
    area: DBRef("areas", 'w')
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e56a"),
    name: 'Raleigh',
    loves: [ 'redbull' ],
    weight: 421,
    gender: 'm',
    vampires: 7,
    area: DBRef("areas", 'cc')
  },
  {
    _id: ObjectId("63457b3c8cbce4d96074e56d"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f',
    area: DBRef("areas", 'kl')
  }
]

```

3.2 НАСТРОЙКА ИНДЕКСОВ

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

```

learn> db.unicorns.ensureIndex({'name': 1}, {'unique': true})
[ 'name_1' ]

```

3.3 УПРАВЛЕНИЕ ИНДЕКСАМИ

- 1) Получите информацию о всех индексах коллекции unicorns .
- 2) Удалите все индексы, кроме индекса для идентификатора.
- 3) Попытайтесь удалить индекс для идентификатора.

```

learn> db.unicorns.ensureIndex({'name': 1}, {'unique': true})
[ 'name_1' ]
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')
{ nIndexesWas: 2, ok: 1 }
learn> db.unicorns.dropIndex('_id_')
MongoServerError: cannot drop _id index

```

3.4 ПЛАН ЗАПРОСА

- 1) Создайте объемную коллекцию numbers, задействовав курсор:

```
for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
```
- 2) Выберите последних четыре документа.
- 3) Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)
- 4) Создайте индекс для ключа value.
- 5) Получите информацию о всех индексах коллекции numbers.
- 6) Выполните запрос 2.
- 7) Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?
- 8) Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?

1-2:

```

numbers> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6345dc55c7f6a399909a964a") }
}
numbers> db.numbers.find().limit(10)
[
  { _id: ObjectId("6345da5cc7f6a39990990fab"), value: 0 },
  { _id: ObjectId("6345da5cc7f6a39990990fac"), value: 1 },
  { _id: ObjectId("6345da5cc7f6a39990990fad"), value: 2 },
  { _id: ObjectId("6345da5cc7f6a39990990fae"), value: 3 },
  { _id: ObjectId("6345da5cc7f6a39990990faf"), value: 4 },
  { _id: ObjectId("6345da5cc7f6a39990990fb0"), value: 5 },
  { _id: ObjectId("6345da5cc7f6a39990990fb1"), value: 6 },
  { _id: ObjectId("6345da5cc7f6a39990990fb2"), value: 7 },
  { _id: ObjectId("6345da5cc7f6a39990990fb3"), value: 8 },
  { _id: ObjectId("6345da5cc7f6a39990990fb4"), value: 9 }
]
numbers> db.numbers.explain('executionStats').find().sort({value: -1}).limit(4)

```

3:

```

executionTimeMillis: 174,

```

4-6:

```
numbers> db.numbers.ensureIndex({'value': 1})
[ 'value_1' ]
numbers> db.numbers.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { value: 1 }, name: 'value_1' }
]
numbers> db.numbers.explain('executionStats').find().sort({value: -1}).limit(4)
{
```

7:

```
executionTimeMillis: 10,
```

Выводы: выполнена работа с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.