

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего  
образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе «ВВЕДЕНИЕ В СУБД MONGODB.  
УСТАНОВКА MONGODB. НАЧАЛО РАБОТЫ С БД»

по дисциплине «Базы данных»

Автор: Акулов Алексей

Факультет: ФИКТ

Группа: K32391

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

**Цель:** овладеть практическими навыками установки СУБД MongoDB.

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД MongoDB 4+, 6.0.6 (текущая)

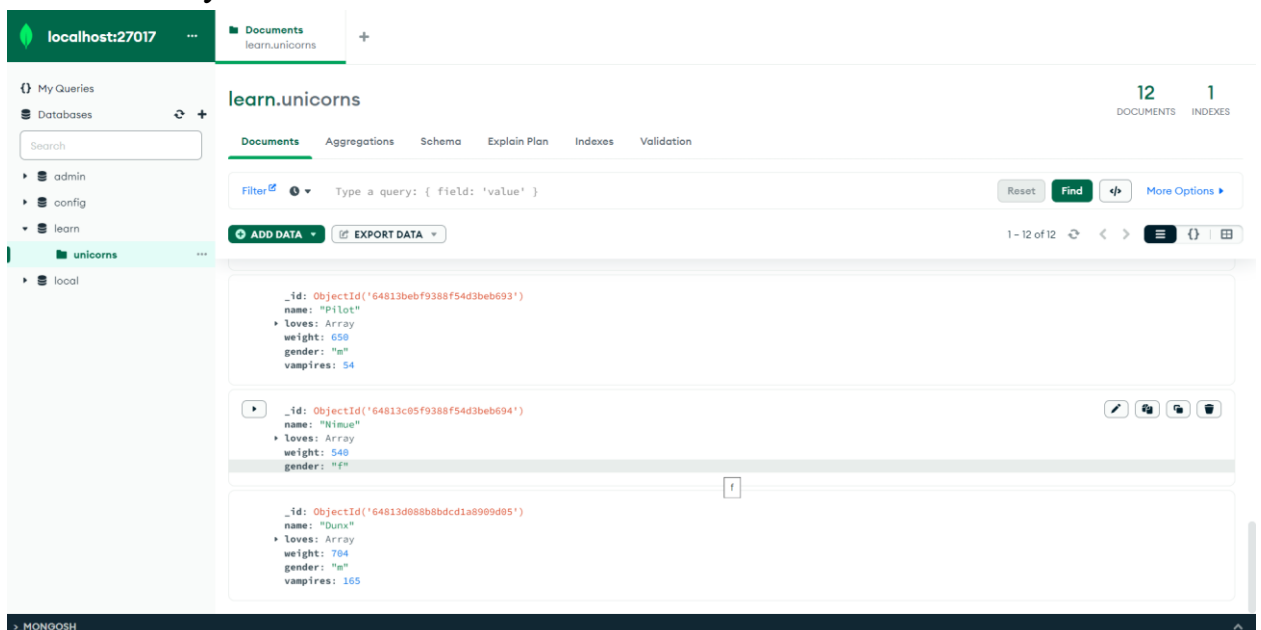
Выполнение:

## 1. Задание 8.1.1

### а. Вставка через консоль а затем через документ

```
mongosh mongodb://127.0.0.1:27020
learn> db.unicorns.insert({name: 'Menny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("64813ba6f9388f54d3beb690") }
}
learn> db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("64813bc3f9388f54d3beb691") }
}
learn> db.unicorns.find().count()
8
learn> db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("64813bdcf9388f54d3beb692") }
}
learn> db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("64813bebf9388f54d3beb693") }
}
learn> db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("64813c05f9388f54d3beb694") }
}
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.find().count()
11
learn> db.unicorns.find().count()
12
learn>
```

### б. Результат find



## 2. Задание 8.1.2

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

```
learn> db.unicorns.find({gender : 'f'}).sort({name : 1}).limit(3)
[
  {
    _id: ObjectId("64813b12f9388f54d3beb68b"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("64813b94f9388f54d3beb68f"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId("64813bdcf9388f54d3beb692"),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  }
]
learn>
```

- б. Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций `findOne` и `limit`.

```
learn> db.unicorns.find({gender:'f', loves:'carrot'}, {_id:0, name:1}).sort({name:1}).limit(1)
[ { name: 'Aurora' } ]
learn>
```

с.

## 3. Задание 8.1.3

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

```
{
  _id: ObjectId("64813b01f9388f54d3beb68a"),
  name: 'Horny',
  weight: 600,
  vampires: 63
},
{
  _id: ObjectId("64813b5cf9388f54d3beb68c"),
  name: 'Unicorn',
  weight: 984,
  vampires: 182
},
{
  _id: ObjectId("64813b6cf9388f54d3beb68d"),
  name: 'Roooooodles',
  weight: 575,
  vampires: 99
},
{
  _id: ObjectId("64813ba6f9388f54d3beb690"),
  name: 'Kenny',
  weight: 690,
  vampires: 39
},
{
  _id: ObjectId("64813bc3f9388f54d3beb691"),
  name: 'Raleigh',
  weight: 421,
  vampires: 2
},
{
  _id: ObjectId("64813bef9388f54d3beb693"),
  name: 'Pilot',
  weight: 650,
  vampires: 54
},
{
  _id: ObjectId("64813d088b8bdcd1a8909d05"),
  name: 'Dunx',
  weight: 704,
```

## 4. Задание 8.1.4

Вывести список единорогов в обратном порядке добавления.

```
mongosh mongodb://127.0.0.1:27000
learn> db.unicorns.find({}).sort({$natural : -1})
[
  {
    _id: ObjectId("64813d088b8bdc1a8909d05"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId("64813c05f9388f54d3beb694"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId("64813beb9388f54d3beb693"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId("64813bdcf9388f54d3beb692"),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId("64813bc3f9388f54d3beb691"),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  }
]
```

## 5. Задание 8.1.5

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

```
learn> db.unicorns.find({}, {_id : 0, love : {$slice : 1}})
[
  {
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    name: 'Ayna',
  }
]
```

## 6. Задание 8.1.6

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'
  }
]
```

#### 7. Задание 8.1.7

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
  }
]
learn> |
```

#### 8. Задание 8.1.8

*Найти всех единорогов, не имеющих ключ vampires.*

```
learn> db.unicorns.find({gender:'f', vampires:{$exists:false}}, {_id:0, name:1})
[ { name: 'Nimue' } ]
```

#### 9. Задание 8.1.9

*Вывести 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: 'Horny', loves: [ 'carrot' ] },
  { name: 'Kenny', loves: [ 'grape' ] },
  { name: 'Pilot', loves: [ 'apple' ] },
  { name: 'Raleigh', loves: [ 'apple' ] },
  { name: 'Rooooooodles', loves: [ 'apple' ] },
  { name: 'Unicrom', loves: [ 'energon' ] }
]
```

#### 10. Задание 8.2.1

- Создайте коллекцию towns, включающую следующие документы:
- 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"}, {_id:0, name:1, mayor:1}).sort({name:1})
[
  {
    name: 'New York',
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  }
]
```

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

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

#### 11.Задание 8.2.2

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

```

learn> un_male = function() {return this.gender == 'm';}
[Function: un_male]
learn> var cursor = db.unicorns.find({'$where' : males}).sort({name : 1}).limit(2)
ReferenceError: males is not defined
learn> var cursor = db.unicorns.find({'$where' : un_male}).sort({name : 1}).limit(2)

learn> cursor.forEach(function(obj) {print(obj)})
{
  _id: ObjectId("64813d088b8bdcd1a8909d05"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
{
  _id: ObjectId("64813b01f9388f54d3beb68a"),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63
}

learn>

```

#### 12. Задание 8.2.3

Вывести количество самок единорогов весом от полутонны до 600 кг.

```

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

```

#### 13. Задание 8.2.4

Вывести список предпочтений

```

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

```

#### 14. Задание 8.2.5

Подсчитать количество особей единорогов обоих полов.

```

learn> db.unicorns.aggregate({'$group':{'_id':'$gender', count:{$sum:1}}})
[ { _id: 'm', count: 8 }, { _id: 'f', count: 5 } ]

```

#### 15.Задание 8.2.6

- a. Выполнить команду: `db.unicorns.save({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})`
- b. 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> db.unicorns.insert({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6481862af9388f54d3beb695") }
}
```

#### 16.Задание 8.2.7

- a. Для самки единорога Айна внести изменения в БД: теперь ее вес 800, она убила 51 вампира.
- b. 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.find({name : 'Ayna'})
[
  {
    _id: ObjectId("64813b94f9388f54d3beb68f"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 800,
    gender: 'f',
    vampires: 51
  }
]
learn> |
```

#### 17.Задание 8.2.8

- a. Для самца единорога Raleigh внести изменения в БД: теперь он любит рэдбул.
- b. 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.find({name : 'Raleigh'})
[
  {
    _id: ObjectId("64813bc3f9388f54d3beb691"),
    name: 'Raleigh',
    loves: [ 'redbull' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  }
]
learn> |
```

#### 18.Задание 8.2.9

- a. Всем самцам единорогов увеличить количество убитых вампиров на



## б. Проверить содержимое коллекции unicorns.

```
learn> db.unicorns.updateMany({gender: 'm'}, {$inc : {vampires : 5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 8,
  learn> db.unicorns.find().count()
learn> db.unicorns.find({}, {name : 1, vampires : 1})
[
  {
    _id: ObjectId("64813b01f9388f54d3beb68a"),
    name: 'Hozny',
    vampires: 68
  },
  {
    _id: ObjectId("64813b12f9388f54d3beb68b"),
    name: 'Aurora',
    vampires: 43
  },
  {
    _id: ObjectId("64813b5cf9388f54d3beb68c"),
    name: 'Unicrom',
    vampires: 187
  },
  {
    _id: ObjectId("64813b6cf9388f54d3beb68d"),
    name: 'Rooooooodles',
    vampires: 104
  },
  {
    _id: ObjectId("64813b87f9388f54d3beb68e"),
    name: 'Solnara',
    vampires: 80
  },
  {
    _id: ObjectId("64813b94f9388f54d3beb68f"),
    name: 'Ayna',
    vampires: 51
  },
  {
  }
```

## 19.Задание 8.2.10

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

```
learn> db.towns.updateOne({name: 'Portland'}, {$unset : {'mayor.party' : 1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.towns.find({name : 'Portland'})
[
  {
    _id: ObjectId("648182818b8bdc1a8909d0f"),
    name: 'Portland',
    population: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'Food' ],
    mayor: { name: 'Sam Adams' }
  }
]
learn> |
```

## 20.Задание 8.2.11

Изменить информацию о самце единорога Pilot: теперь он любит и шоколад

```
learn> db.unicorns.updateOne({name: 'Pilot'}, {$push : {loves : ['chocolat']}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.find({name : 'Pilot'})
[
  {
    _id: ObjectId("64813beb9388f54d3beb693"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', [ 'chocolat' ] ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
learn> |
```

## 21.Задание 8.2.12

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

```
learn> db.unicorns.updateOne({name: 'Aurora'}, {$addToSet: {loves: {$each: ['sugar', 'lemon']}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.find({name: 'Aurora'})
[
  {
    _id: ObjectId("64813b12f9388f54d3beb68b"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
learn> |
```

## 22. Задание 8.2.13

- Удалите документы с беспартийными мэрами.
- Проверьте содержание коллекции.
- Очистите коллекцию.
- Просмотрите список доступных коллекций

```
learn> db.towns.deleteMany({'mayor.party': {'$exists': false}})
{ acknowledged: true, deletedCount: 1 }
learn> db.towns.find()
[
  {
    _id: ObjectId("648182818b8bdc1a8909d0e"),
    name: 'New York',
    population: 22200000,
    last_sensus: ISODate("2009-07-31T00:00:00.000Z"),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  },
  {
    _id: ObjectId("648182818b8bdc1a8909d0f"),
    name: 'Portland',
    population: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
learn> db.towns.drop()
true
learn> show collections
unicorns
learn> |
```

## 23. Задание 8.3.1

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

```
learn> db.createCollection('areals')
{ ok: 1 }
learn> db.areals.insertMany([ { _id: 'station', name: 'Gas Station', description: 'A place to have some fun' }, { _id: 'pomoika', name: 'Dump', description: 'A dirty, smelly place.' }, { _id: 'murino', name: 'Murino', description: 'A small, noisy town.' }, { _id: 'garden', name: 'Crimson Garden', description: 'A place of nature and peace, chill zone' } ])
{ acknowledged: true,
  insertedIds: { '0': 'station', '1': 'pomoika', '2': 'murino', '3': 'garden' } }
learn> db.areals.find()
[
  {
    _id: 'station',
    name: 'Gas Station',
    description: 'A place to have some fun'
  },
  {
    _id: 'pomoika',
    name: 'Dump',
    description: 'A dirty, smelly place.'
  },
  {
    _id: 'murino',
    name: 'Murino',
    description: 'A small, noisy town.'
  },
  {
    _id: 'garden',
    name: 'Crimson Garden',
    description: 'A place of nature and peace, chill zone'
  }
]
learn> |
```

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

```

learn> db.unicorns.updateOne({ name: 'Horny' }, { $set: { habitat: 'station' } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({ name: 'Aurora' }, { $set: { areal: { $ref: 'areals', $id: 'pomoika' } } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({ name: 'Unicrom' }, { $set: { areal: { $ref: 'areals', $id: 'murino' } } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({ name: 'Rooooooodles' }, { $set: { areal: { $ref: 'areals', $id: 'garden' } } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({ name: 'Solnara' }, { $set: { areal: { $ref: 'areals', $id: 'station' } } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

с. Проверьте содержание коллекции едиорогов.

```

learn> db.unicorns.find()
[
  {
    _id: ObjectId("64813b01f9388f54d3beb68a"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 68,
    habitat: 'station',
    areal: DBRef("areals", 'station')
  },
  {
    _id: ObjectId("64813b12f9388f54d3beb68b"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43,
    areal: DBRef("areals", 'pomoika')
  },
  {
    _id: ObjectId("64813b5cf9388f54d3beb68c"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 187,
    areal: DBRef("areals", 'murino')
  },
  {
    _id: ObjectId("64813b6cf9388f54d3beb68d"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 104,
    areal: DBRef("areals", 'garden')
  },
]

```

## 24. Задания 8.3.2. и 8.3.3

- Проверьте, можно ли задать для коллекции unicorns индекс для ключа name с флагом unique.
- Получите информацию о всех индексах коллекции unicorns .
- Попытайтесь удалить индекс id

```

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.dropIndexes('_id_')
MongoServerError: cannot drop _id index
learn>

```

## 25. Задание 8.3.4

- Создайте объемную коллекцию numbers, задействовав курсор:

2. Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра `executionTimeMillis`)
3. Создайте индекс для ключа `value`.
4. Получите информацию о всех индексах коллекции `numbers`.
5. Выполните запрос 2.
6. Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?
7. Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?

```
learn> db.createCollection('numbers')
{ ok: 1 }
learn> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}

learn> db.numbers.find().count()
100000
learn> db.numbers.find().sort({value: -1}).limit(4)
[
  { _id: ObjectId("6481ab516fc51f9c2ca50da8"), value: 99999 },
  { _id: ObjectId("6481ab516fc51f9c2ca50da7"), value: 99998 },
  { _id: ObjectId("6481ab516fc51f9c2ca50da6"), value: 99997 },
  { _id: ObjectId("6481ab516fc51f9c2ca50da5"), value: 99996 }
]
learn> var explain = db.numbers.find().sort({value: -1}).limit(4).explain("executionStats")
learn> printjson(explain.executionStats.executionTimeMillis)
163
learn> db.numbers.createIndex({value: 1})
value_1
learn> db.numbers.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { value: 1 }, name: 'value_1' }
]
learn> db.numbers.find().sort({value: -1}).limit(4)
[
  { _id: ObjectId("6481ab516fc51f9c2ca50da8"), value: 99999 },
  { _id: ObjectId("6481ab516fc51f9c2ca50da7"), value: 99998 },
  { _id: ObjectId("6481ab516fc51f9c2ca50da6"), value: 99997 },
  { _id: ObjectId("6481ab516fc51f9c2ca50da5"), value: 99996 }
]
learn> var explainWithIndex = db.numbers.find().sort({value: -1}).limit(4).explain("executionStats")
learn> printjson(explainWithIndex.executionStats.executionTimeMillis)
3
learn> |
```

Т.е. с индексом сильно быстрее

26.

Вывод:

В процессе работы были получены навыки работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.