

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 5

по теме:

«Работа с БД в СУБД MongoDB»

по дисциплине: Проектирование и реализация баз данных

Специальность:

45.03.04 Интеллектуальные системы в гуманитарной сфере

Проверила:

Говорова М.М.

Дата: «...» ... 2023 г.

Оценка _____

Выполнила:

студентка группы К32421

Милька К.А.

Санкт-Петербург 2022/2023

Цель работы: овладеть практическими навыками установки СУБД MongoDB, работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.

Выполнение:

Практическая работа 5.2

Практическое задание 8.1.1:

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'});
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

learn> db.createCollection("unicorns")
{ ok: 1 }
learn> db.unicorns.insertMany([
...   {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', 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("649be50d1ab36f842ac742c4"),
    '1': ObjectId("649be50d1ab36f842ac742c5"),
    '2': ObjectId("649be50d1ab36f842ac742c6"),
    '3': ObjectId("649be50d1ab36f842ac742c7"),
    '4': ObjectId("649be50d1ab36f842ac742c8"),
    '5': ObjectId("649be50d1ab36f842ac742c9"),
    '6': ObjectId("649be50d1ab36f842ac742ca"),
    '7': ObjectId("649be50d1ab36f842ac742cb"),
    '8': ObjectId("649be50d1ab36f842ac742cc"),
    '9': ObjectId("649be50d1ab36f842ac742cd"),
    '10': ObjectId("649be50d1ab36f842ac742ce")
  }
}
learn>
```

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.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

learn> db.createCollection("unicorns")
{ ok: 1 }

learn> db.unicorns.insertOne(
...   {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}
... );
{
  acknowledged: true,
  insertedId: ObjectId("649be5761ab36f842ac742cf")
}
learn> db.unicorns.find();
[
  {
    _id: ObjectId("649be50d1ab36f842ac742c4"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742c5"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742c6"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742c7"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742c8"),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742c9"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 500,
    gender: 'f',
    vampires: 70
  }
]
```

Практическое задание 8.1.2:

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`.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

learn> db.unicorns.find({gender: 'm'}, {_id: 0, name: 1}).sort({name: 1})
[
  { name: 'Dunx' },
  { name: 'Horny' },
  { name: 'Kenny' },
  { name: 'Pilot' },
  { name: 'Raleigh' },
  { name: 'Roooooodles' },
  { name: 'Unicrom' }
]
learn> db.unicorns.find({gender: 'f'}, {_id: 0, name: 1}).sort({name: 1}).limit(3)
[ { name: 'Aurora' }, { name: 'Ayna' }, { name: 'Leia' } ]
learn> db.unicorns.findOne({gender: 'f', loves: 'carrot'}, {_id: 0, name: 1})
{ name: 'Aurora' }
learn> db.unicorns.find({gender: 'f', loves: 'carrot'}, {_id: 0, name: 1}).limit(1)
[ { name: 'Aurora' } ]
learn>
```

Практическое задание 8.1.3:

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

learn> db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0});
[
  {
    _id: ObjectId("649be50d1ab36f842ac742c4"),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742c6"),
    name: 'Unicrom',
    weight: 984,
    vampires: 182
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742c7"),
    name: 'Roooooodles',
    weight: 575,
    vampires: 99
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742ca"),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742cb"),
    name: 'Raleigh',
    weight: 421,
    vampires: 2
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742cd"),
    name: 'Pilot',
    weight: 650,
    vampires: 54
  },
  {
    _id: ObjectId("649be5761ab36f842ac742cf"),
    name: 'Dunx',
    weight: 704,
    vampires: 165
  }
]
learn>
```

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
learn> db.unicorns.find({}, {_id: 0, name: 1}).sort({$natural: -1})
[
  { name: 'Dunx' },
  { name: 'Nimue' },
  { name: 'Pilot' },
  { name: 'Leia' },
  { name: 'Raleigh' },
  { name: 'Kenny' },
  { name: 'Ayna' },
  { name: 'Solnara' },
  { name: 'Rooooooodles' },
  { name: 'Unicrom' },
  { name: 'Aurora' },
  { name: 'Horny' }
]
learn> 
```

Практическое задание 8.1.5:

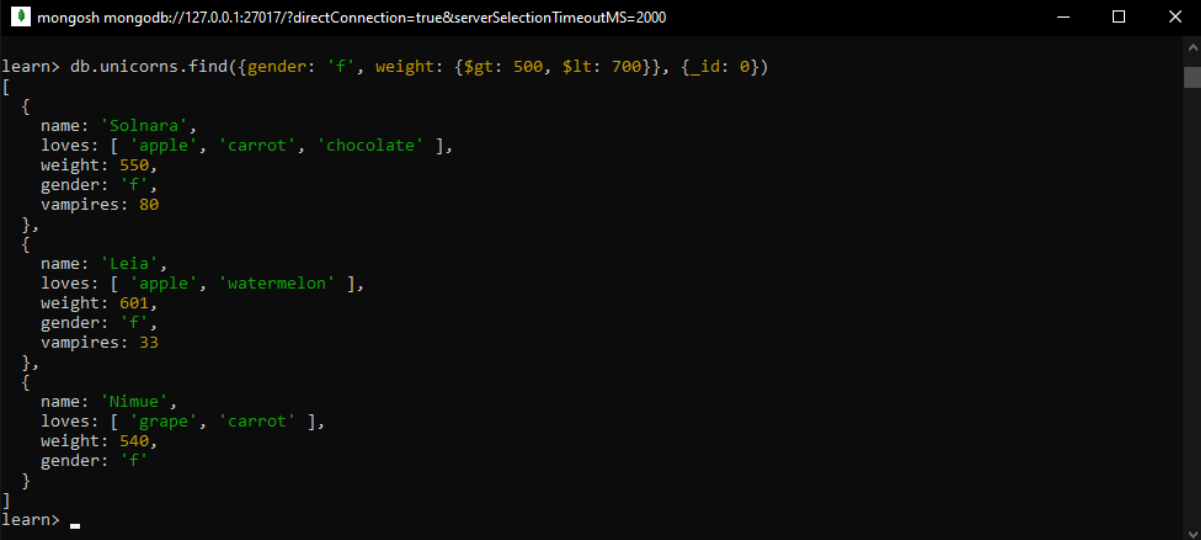
Вывести список единорогов с названием первого любимого предпочтения, исключив идентификатор.

```
mongosh mongod://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
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
  },
  {
    name: 'Kenny',
    loves: [ 'grape' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {

```

Практическое задание 8.1.6:

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



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

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'
  }
]
learn> _
```


Практическое задание 8.1.7:

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
learn> db.unicorns.find({gender: 'm', loves: {$all: ['grape', 'lemon']}, weight: {$gt: 500}}, {_id: 0})
[
  {
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
]
learn> _
```

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
learn> db.unicorns.find({vampires: {$exists: 0}})
[
  {
    _id: ObjectId("649be50d1ab36f842ac742ce"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
learn> _
```

Практическое задание 8.1.9:

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
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' ] }
]
learn> _
```

Практическое задание 8.2.1:

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"
  }
}
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
learn> db.createCollection('towns')
{ ok: 1 }
learn> db.towns.insertMany([
...   {
...     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"
...     }
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649bea391ab36f842ac742d0"),
    '1': ObjectId("649bea391ab36f842ac742d1"),
    '2': ObjectId("649bea391ab36f842ac742d2")
  }
}
learn>
```

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
learn> db.towns.find({'mayor.party': 'I'}, {_id: 0, name: 1, mayor: 1});
[[
  {
    name: 'New York',
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  }
]]
learn> db.towns.find({'mayor.party': {'$exists': 0}}, {_id: 0, name: 1, mayor: 1});
[ { name: 'Punxsutawney ', mayor: { name: 'Jim Wehrle' } } ]
learn>
```

Практическое задание 8.2.2:

1. Сформировать функцию для вывода списка самцов единорогов.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

learn> function males() {
...   db.unicorns.find({gender: 'm'}).forEach(function(x) { print(x.name) })
... };

learn> males();
Horny
Unicrom
Rooooooodles
Kenny
Raleigh
Pilot
Dunx
learn>
```

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

3. Вывести результат, используя forEach.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

learn> var curs = db.unicorns.find({gender: 'm'}).limit(2).sort({name: 1});
learn> curs.forEach(print)
{
  _id: ObjectId("649be5761ab36f842ac742cf"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
{
  _id: ObjectId("649be50d1ab36f842ac742c4"),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
learn> _
```

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

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

Практическое задание 8.2.4:

Вывести список предпочтений.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
learn> db.unicorns.distinct('loves')
[
  'apple',      'carrot',
  'chocolate', 'energon',
  'grape',      'lemon',
  'papaya',     'redbull',
  'strawberry', 'sugar',
  'watermelon'
]
learn>
```

Практическое задание 8.2.5:

Подсчитать количество особей единорогов обоих полов.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
learn> db.unicorns.aggregate(
...   {$group: {_id: '$gender', count: {$sum: 1}}}
... )
[ { _id: 'm', count: 7 }, { _id: 'f', count: 5 } ]
learn>
```

Практическое задание 8.2.6:

1. Выполнить команду:

```
> db.unicorns.save({name: 'Barney', loves: ['grape'], weight: 340,
gender: 'm'})
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
learn> db.unicorns.save({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})
TypeError: db.unicorns.save is not a function
learn> _
```

Данный метод уже устарел. Вместо него лучше использовать либо методы *insertOne* или *insertMany*.

2. Проверить содержимое коллекции unicorns.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
learn> db.unicorns.find()
[
  {
    _id: ObjectId("649be50d1ab36f842ac742c4"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742c5"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742c6"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId("649be50d1ab36f842ac742c7"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
  }
]
```

Практическое задание 8.2.7:

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
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("649be50d1ab36f842ac742c9"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 800,
    gender: 'f',
    vampires: 51
  }
]
learn>
```

Практическое задание 8.2.8:

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

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("649be50d1ab36f842ac742cb"),
    name: 'Raleigh',
    loves: [ 'redbull' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  }
]
learn> _
```

Практическое задание 8.2.9:

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

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: 0, name: 1, vampires: 1})
[
  { name: 'Horny', vampires: 68 },
  { name: 'Unicrom', vampires: 187 },
  { name: 'Rooodoodles', vampires: 104 },
  { name: 'Kenny', vampires: 44 },
  { name: 'Raleigh', vampires: 7 },
  { name: 'Pilot', vampires: 59 },
  { name: 'Dunx', vampires: 170 }
]
learn> _
```

Практическое задание 8.2.10:

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
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("649bea391ab36f842ac742d2"),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams' }
  }
]
learn> _
```

Практическое задание 8.2.11:

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
learn> db.unicorns.updateOne({name: 'Pilot'}, {$addToSet: {loves: 'chocolate'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.find({name: 'Pilot'})
[
  {
    _id: ObjectId("649be50d1ab36f842ac742cd"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
learn>
```


Практическое задание 8.2.12:

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



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

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("649be50d1ab36f842ac742c5"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
learn>
```

Практическое задание 8.2.13:

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"
  }
}
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

learn> db.towns.insertMany([
...   {
...     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"
...     }
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649bf1621ab36f842ac742d3"),
    '1': ObjectId("649bf1621ab36f842ac742d4"),
    '2': ObjectId("649bf1621ab36f842ac742d5")
  }
}
learn>
```

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

learn> db.towns.deleteMany({'mayor.party': {'$exists': 0}})
{ acknowledged: true, deletedCount: 1 }
learn> db.towns.find({})
[
  {
    _id: ObjectId("649bf1621ab36f842ac742d4"),
    name: 'New York',
    popujatiuon: 22200000,
    last_sensus: ISODate("2009-07-31T00:00:00.000Z"),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  },
  {
    _id: ObjectId("649bf1621ab36f842ac742d5"),
    name: 'Portland',
    popujatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
learn> db.towns.deleteMany({})
{ acknowledged: true, deletedCount: 2 }
learn> db.towns.find()
learn>
```

Практическое задание 8.3.1:

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
learn> db.createCollection('areals')
{ ok: 1 }
learn> db.areals.insertMany([
...   {
...     _id: 'T1',
...     name: 'Test 1',
...     description: 'test 1'
...   },
...   {
...     _id: 'T2',
...     name: 'Test 2',
...     description: 'test 2'
...   },
...   {
...     _id: 'T3',
...     name: 'Test 3',
...     description: 'test 3'
...   },
...   {
...     _id: 'T4',
...     name: 'Test 4',
...     description: 'test 4'
...   },
...   {
...     _id: 'T5',
...     name: 'Test 5',
...     description: 'test 5'
...   },
... ])
{
  acknowledged: true,
  insertedIds: { '0': 'T1', '1': 'T2', '2': 'T3', '3': 'T4', '4': 'T5' }
}
learn>
```

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

3. Проверьте содержание коллекции единорогов.

```
Выбрать mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

learn> db.unicorns.updateMany({gender: 'm'}, {$set: {areal: {$ref: 'areal', $id: 'T1'}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 7,
  modifiedCount: 7,
  upsertedCount: 0
}
learn> db.unicorns.updateMany({gender: 'f'}, {$set: {areal: {$ref: 'areal', $id: 'T2'}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 5,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({name: 'Ayna'}, {$set: {areal: {$ref: 'areal', $id: 'T3'}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.find({}, {_id: 0, name: 1, gender: 1, areal: 1})
[
  { name: 'Horny', gender: 'm', areal: DBRef("areal", 'T1') },
  { name: 'Aurora', gender: 'f', areal: DBRef("areal", 'T2') },
  { name: 'Unicrom', gender: 'm', areal: DBRef("areal", 'T1') },
  { name: 'Rooooooodles', gender: 'm', areal: DBRef("areal", 'T1') },
  { name: 'Solnara', gender: 'f', areal: DBRef("areal", 'T2') },
  { name: 'Ayna', gender: 'f', areal: DBRef("areal", 'T3') },
  { name: 'Kenny', gender: 'm', areal: DBRef("areal", 'T1') },
  { name: 'Raleigh', gender: 'm', areal: DBRef("areal", 'T1') },
  { name: 'Leia', gender: 'f', areal: DBRef("areal", 'T2') },
  { name: 'Pilot', gender: 'm', areal: DBRef("areal", 'T1') },
  { name: 'Nimue', gender: 'f', areal: DBRef("areal", 'T2') },
  { name: 'Dunx', gender: 'm', areal: DBRef("areal", 'T1') }
]
learn> _
```

Практическое задание 8.3.2:

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

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

Да, можно

Практическое задание 8.3.3:

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

2. Попытайтесь удалить индекс для идентификатора

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

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
learn> _
```

Практическое задание 8.3.4:

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 (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?

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

learn> db.createCollection('numbers')
{ ok: 1 }
learn> for (i = 0; i < 100000; i++) {db.numbers.insertOne({value: i})}
{
  acknowledged: true,
  insertedId: ObjectId("649bf6ce1ab36f842ac8c975")
}
learn> db.numbers.find().sort({value: -1}).limit(4).explain('executionStats').executionStats.executionTimeMillis
76
learn>
learn> db.numbers.createIndex({value: 1})
value_1
learn> db.numbers.find().sort({value: -1}).limit(4).explain('executionStats').executionStats.executionTimeMillis
3
learn> _
```

Времени на выполнение запроса без индекса: 76 мс

С индексом: 3 мс

Заключение:

В процессе выполнения лабораторной задачи был произведен анализ и практическое применение ключевых команд MongoDB. Полученный опыт дает необходимые предпосылки для успешного взаимодействия с указанной системой управления базами данных.

Были изучены основы структурирования данных в MongoDB и ее подходы к управлению информацией. В отличие от классических реляционных баз данных, MongoDB сохраняет данные в гибкой структуре, напоминающей формат JSON.

Проанализированы команды вставки MongoDB для добавления новых записей в коллекцию, включая функции `insertOne`, `insertMany` и `save`. Изучены методы извлечения данных из MongoDB, что включает в себя использование функции `find`, применение различных фильтров, а также методы сортировки и ограничения.

В итоге выполнения этой работы получены значимые инструменты и знания для эффективного управления информацией и создания масштабируемых приложений с помощью MongoDB.