

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

Отчет

по лабораторной работе «Работа с БД в СУБД MongoDB»
по дисциплине «**Проектирование и реализация баз данных**»

Автор: Жигалова Анастасия Евгеньевна

Факультет: ИКТ

Группа: K32392

Преподаватель: Говорова М. М.

Дата: 31.05.2023



Санкт-Петербург 2023

Цель: 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.

Оборудование: компьютерный класс.

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

Выполнение:

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

Практическое задание 8.1.1:

- 1) *Создайте базу данных learn.*
- 2) *Заполните коллекцию единорогов unicorns*

```
learn> db.unicorns.find()
[
  {
    _id: ObjectId("6477b4a2eb41e284e324e49e"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("6477b4afeb41e284e324e49f"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("6477b4b9eb41e284e324e4a0"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId("6477b51aeb41e284e324e4a1"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId("6477b524eb41e284e324e4a2"),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
]
```

```

{
  _id: ObjectId("6477b557eb41e284e324e4a3"),
  name: 'Ayna',
  loves: [ 'strawberry', 'lemon' ],
  weight: 733,
  gender: 'f',
  vampires: 40
},
{
  _id: ObjectId("6477b562eb41e284e324e4a4"),
  name: 'Kenny',
  loves: [ 'grape', 'lemon' ],
  weight: 690,
  gender: 'm',
  vampires: 39
},
{
  _id: ObjectId("6477b578eb41e284e324e4a5"),
  name: 'Raleigh',
  loves: [ 'apple', 'sugar' ],
  weight: 421,
  gender: 'm',
  vampires: 2
},
{
  _id: ObjectId("6477b59deb41e284e324e4a6"),
  name: 'Leia',
  loves: [ 'apple', 'watermelon' ],
  weight: 601,
  gender: 'f',
  vampires: 33
},
{
  _id: ObjectId("6477b5aceb41e284e324e4a7"),
  name: 'Pilot',
  loves: [ 'apple', 'watermelon' ],
  weight: 650,
  gender: 'm',
  vampires: 54
},
{
  _id: ObjectId("6477b66deb41e284e324e4a9"),
  name: 'Nimue',
  loves: [ 'grape', 'carrot' ],
  weight: 540,
  gender: 'f'
}
]

```

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.insertOne(document)
{
  acknowledged: true,
  insertedId: ObjectId("6477bdc3651c4760106c4617")
}

```

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

```
{
  _id: ObjectId("6477b59deb41e284e324e4a6"),
  name: 'Leia',
  loves: [ 'apple', 'watermelon' ],
  weight: 601,
  gender: 'f',
  vampires: 33
},
{
  _id: ObjectId("6477b5aceb41e284e324e4a7"),
  name: 'Pilot',
  loves: [ 'apple', 'watermelon' ],
  weight: 650,
  gender: 'm',
  vampires: 54
},
{
  _id: ObjectId("6477b66deb41e284e324e4a9"),
  name: 'Nimue',
  loves: [ 'grape', 'carrot' ],
  weight: 540,
  gender: 'f'
},
{
  _id: ObjectId("6477bdc3651c4760106c4617"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
]
```

8.2.2 ВЫБОРКА ДАННЫХ ИЗ БД

Практическое задание 8.1.2:

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

```
learn> {}db.unicorns.find({gender: 'f'}) .sort({name: 1}) . limit (3);
[
  {
    _id: ObjectId("6477b4afeb41e284e324e49f"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("6477b557eb41e284e324e4a3"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId("6477b59deb41e284e324e4a6"),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  }
]
```

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("6477b4afeb41e284e324e49f"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
```

Практическое задание 8.1.3:

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("6477b4a2eb41e284e324e49e"),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId("6477b4b9eb41e284e324e4a0"),
    name: 'Unicrom',
    weight: 984,
    vampires: 182
  },
  {
    _id: ObjectId("6477b51aeb41e284e324e4a1"),
    name: 'Rooooooodles',
    weight: 575,
    vampires: 99
  },
  {
    _id: ObjectId("6477b562eb41e284e324e4a4"),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  },
  {
    _id: ObjectId("6477b578eb41e284e324e4a5"),
    name: 'Raleigh',
    weight: 421,
    vampires: 2
  },
  {
    _id: ObjectId("6477b5aceb41e284e324e4a7"),
    name: 'Pilot',
    weight: 650,
    vampires: 54
  },
  {
    _id: ObjectId("6477bdc3651c4760106c4617"),
    name: 'Dunx',
    weight: 704,
    vampires: 165
  }
]
```

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```
learn> db.unicorns. find() .sort ({$natural: -1});
[
  {
    _id: ObjectId("6477bdc3651c4760106c4617"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId("6477b66deb41e284e324e4a9"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId("6477b5aceb41e284e324e4a7"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId("6477b59deb41e284e324e4a6"),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId("6477b578eb41e284e324e4a5"),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId("6477b56deb41e284e324e4a4"),
    name: 'Merlin',
    loves: [ 'apple', 'watermelon' ],
    weight: 500,
    gender: 'm',
    vampires: 10
  },
  {
    _id: ObjectId("6477b55deb41e284e324e4a3"),
    name: 'Galahad',
    loves: [ 'apple', 'watermelon' ],
    weight: 400,
    gender: 'm',
    vampires: 5
  },
  {
    _id: ObjectId("6477b54deb41e284e324e4a2"),
    name: 'Lancelot',
    loves: [ 'apple', 'watermelon' ],
    weight: 300,
    gender: 'm',
    vampires: 1
  },
  {
    _id: ObjectId("6477b53deb41e284e324e4a1"),
    name: 'Arthur',
    loves: [ 'apple', 'watermelon' ],
    weight: 200,
    gender: 'm',
    vampires: 0
  }
]
```

```
{
  _id: ObjectId("6477b562eb41e284e324e4a4"),
  name: 'Kenny',
  loves: [ 'grape', 'lemon' ],
  weight: 690,
  gender: 'm',
  vampires: 39
},
{
  _id: ObjectId("6477b557eb41e284e324e4a3"),
  name: 'Ayna',
  loves: [ 'strawberry', 'lemon' ],
  weight: 733,
  gender: 'f',
  vampires: 40
},
{
  _id: ObjectId("6477b524eb41e284e324e4a2"),
  name: 'Solnara',
  loves: [ 'apple', 'carrot', 'chocolate' ],
  weight: 550,
  gender: 'f',
  vampires: 80
},
{
  _id: ObjectId("6477b51aeb41e284e324e4a1"),
  name: 'Rooooooodles',
  loves: [ 'apple' ],
  weight: 575,
  gender: 'm',
  vampires: 99
},
{
  _id: ObjectId("6477b4b9eb41e284e324e4a0"),
  name: 'Unicrom',
  loves: [ 'energon', 'redbull' ],
  weight: 984,
  gender: 'm',
  vampires: 182
},
{
  _id: ObjectId("6477b4afeb41e284e324e49f"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43
},
}
```



```

{
  _id: ObjectId("6477b4a2eb41e284e324e49e"),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
]

```

Практическое задание 8.1.5:

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
  },
]

```

```

{
  name: 'Kenny',
  loves: [ 'grape' ],
  weight: 690,
  gender: 'm',
  vampires: 39
},
{
  name: 'Raleigh',
  loves: [ 'apple' ],
  weight: 421,
  gender: 'm',
  vampires: 2
},
{
  name: 'Leia',
  loves: [ 'apple' ],
  weight: 601,
  gender: 'f',
  vampires: 33
},
{
  name: 'Pilot',
  loves: [ 'apple' ],
  weight: 650,
  gender: 'm',
  vampires: 54
},
{ name: 'Nimue', loves: [ 'grape' ], weight: 540, gender: 'f' },
{
  name: 'Dunx',
  loves: [ 'grape' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
]

```

Практическое задание 8.1.6:

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

```

learn> db.unicorns. find( {gender: 'f' , weight: { $gte: 500, $lte: 700 } }, { _id: 0});
[
  {
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]

```

Практическое 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:{$gte:500}, loves:{$all: ['grape', 'lemon']}}, {_id: 0})
[
  {
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
]
```

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.

```
learn> db.unicorns.find({ vampires: { $exists: false } });
[
  {
    _id: ObjectId("6477b66deb41e284e324e4a9"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
```

Практическое задание 8.1.9:

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' ] }
]
```

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

8.2.1 ЗАПРОС К ВЛОЖЕННЫМ ОБЪЕКТАМ

Практическое задание 8.2.1:

1) Создайте коллекцию `towns`, включающую следующие документы

```
learn> db.towns.find()
[
  {
    _id: ObjectId("647836f7651c4760106c4618"),
    name: 'Punxsutawney ',
    populatiuon: 6200,
    last_sensus: ISODate("2008-01-31T00:00:00.000Z"),
    famous_for: [ '' ],
    mayor: { name: 'Jim Wehrle' }
  },
  {
    _id: ObjectId("64783732651c4760106c4619"),
    name: 'New York',
    populatiuon: 22200000,
    last_sensus: ISODate("2009-07-31T00:00:00.000Z"),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  },
  {
    _id: ObjectId("6478375d651c4760106c461a"),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    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.

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

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:false"}}, {name:1, mayor:1, _id:0})
[ { name: 'Punxsutawney ', mayor: { name: 'Jim Wehrle' } } ]
```

Практическое задание 8.2.2:

- 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>
[Function: getMaleUnicorns]
learn> function getMaleUnicorns () {return db.unicorns.find ({gender: 'm'});}
[Function: getMaleUnicorns]
learn> var cursor = getMaleUnicorns().sort ({name: 1}).limit (2);cursor.forEach (function (unicorn) {print (unicorn.name); });
Dunx
Horny
```

8.2.2 АГРЕГИРОВАННЫЕ ЗАПРОСЫ

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

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

Практическое задание 8.2.4:

Вывести список предпочтений.

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

Практическое задание 8.2.5:

Посчитать количество особей единорогов обоих полов.

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

8.2.5 РЕДАКТИРОВАНИЕ ДАННЫХ

Практическое задание 8.2.6:

1. Выполнить команду:

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

2. Проверить содержимое коллекции *unicorns*.

```
{  
  _id: ObjectId("6477b5aceb41e284e324e4a7"),  
  name: 'Pilot',  
  loves: [ 'apple', 'watermelon' ],  
  weight: 650,  
  gender: 'm',  
  vampires: 54  
},  
{  
  _id: ObjectId("6477b66deb41e284e324e4a9"),  
  name: 'Nimue',  
  loves: [ 'grape', 'carrot' ],  
  weight: 540,  
  gender: 'f'  
},  
{  
  _id: ObjectId("6477bdc3651c4760106c4617"),  
  name: 'Dunx',  
  loves: [ 'grape', 'watermelon' ],  
  weight: 704,  
  gender: 'm',  
  vampires: 165  
},  
{  
  _id: ObjectId("64783d03651c4760106c461b"),  
  name: 'Barney',  
  loves: [ 'grape' ],  
  weight: 340,  
  gender: 'm'  
}  
]
```

Практическое задание 8.2.7:

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

```
learn> db.unicorns.updateOne ({name: "Ayna" }, {$set: {weight: 800, vampires: 51}});  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

2. Проверить содержимое коллекции *unicorns*.

```
{
  _id: ObjectId("6477b557eb41e284e324e4a3"),
  name: 'Ayna',
  loves: [ 'strawberry', 'lemon' ],
  weight: 800,
  gender: 'f',
  vampires: 51
},
```

Практическое задание 8.2.8:

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

```
learn> db.unicorns.updateOne ({name: "Raleigh" }, {$push: {loves: "redbull" }});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

2. Проверить содержимое коллекции unicorns.

```
{
  _id: ObjectId("64783eda651c4760106c461c"),
  name: 'Raleigh',
  loves: [ 'apple', 'sugar', 'redbull' ],
  weight: 421,
  gender: 'm',
  vampires: 2
}
```

Практическое задание 8.2.9:

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

```

learn> db.unicorns.updateMany({gender:"male"}, {$inc:{vampiresKilled:5}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
learn> db.unicorns.find()
[
  {
    _id: ObjectId("6477b4a2eb41e284e324e49e"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("6477b4afeb41e284e324e49f"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("6477b4b9eb41e284e324e4a0"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId("6477b51aeb41e284e324e4a1"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
  }
]

```

Практическое задание 8.2.10:

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

```

learn> db.towns.updateOne ({name: "Portland"}, {$unset: {"mayor party": " "}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}

```


2. Проверить содержимое коллекции `towns`.

```
{
  _id: ObjectId("6478375d651c4760106c461a"),
  name: 'Portland',
  populatiuon: 528000,
  last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
  famous_for: [ 'beer', 'food' ],
  mayor: { name: 'Sam Adams' }
}
```

Практическое задание 8.2.11:

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

```
learn> db.unicorns.updateOne ({name: "Pilot"}, {$push: {loves: "chocolate"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

2. Проверить содержимое коллекции `unicorns`.

```
{
  _id: ObjectId("6477b5aceb41e284e324e4a7"),
  name: 'Pilot',
  loves: [ 'apple', 'watermelon', 'chocolate' ],
  weight: 650,
  gender: 'm',
  vampires: 54
},
```

Практическое задание 8.2.12:

1. 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
}
```

2. Проверить содержимое коллекции `unicorns`.

```
{
  _id: ObjectId("6477b4afeb41e284e324e49f"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
  weight: 450,
  gender: 'f',
  vampires: 43
},
```

8.2.6 УДАЛЕНИЕ ДАННЫХ ИЗ КОЛЛЕКЦИИ

Практическое задание 8.2.13:

1) Создайте коллекцию `towns`, включающую следующие документы

```
learn> db.towns.find()
[
  {
    _id: ObjectId("64786538651c4760106c461d"),
    name: 'Punxsutawney ',
    popujatiuon: 6200,
    last_sensus: ISODate("2008-01-31T00:00:00.000Z"),
    famous_for: [ 'phil the groundhog' ],
    mayor: { name: 'Jim Wehrle' }
  },
  {
    _id: ObjectId("6478655e651c4760106c461e"),
    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("64786586651c4760106c461f"),
    name: 'Portland',
    popujatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
```

2) Удалите документы с беспартийными мэрами.

```
learn> db.towns.deleteMany({"mayor.party": {$exists: false}});
{ acknowledged: true, deletedCount: 1 }
learn> db.towns.find()
[
  {
    _id: ObjectId("6478655e651c4760106c461e"),
    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("64786586651c4760106c461f"),
    name: 'Portland',
    popujatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
```

3) Проверьте содержание коллекции.

4) Очистите коллекцию.

```
learn> db.towns.deleteMany({});
```

5) Просмотрите список доступных коллекций.

```
learn> db.getCollectionNames();
[ 'towns', 'unicorns' ]
```

8.3.2 НАСТРОЙКА ИНДЕКСОВ

Практическое задание 8.3.1:

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

```
learn> db.habitat.insertMany([ {_id: "zone1", name: "zone1", description: "Описание зоны 1", }, {_id: "zone2", name: "zone 2", description: "Описание зоны 2", }, {_id: "zone3", name: "zone 3", description: "Описание зоны 3", } ])
learn>
{
  acknowledged: true,
  insertedIds: { '0': 'zone1', '1': 'zone2', '2': 'zone3' }
}
```

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

```
learn> db.unicorns.updateMany({}, {$set: {zone: "zone3"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 13,
  modifiedCount: 13,
  upsertedCount: 0
}
```

3) Проверьте содержание коллекции единорогов.

```
learn> db.unicorns.find()
[
  {
    _id: ObjectId("6477b4a2eb41e284e324e49e"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63,
    zone: 'zone3'
  },
  {
    _id: ObjectId("6477b4afeb41e284e324e49f"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43,
    zone: 'zone3'
  },
  {
    _id: ObjectId("6477b4b9eb41e284e324e4a0"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182,
    zone: 'zone3'
  },
  {
    _id: ObjectId("6477b51aeb41e284e324e4a1"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99,
    zone: 'zone3'
  },
]
```

Практическое задание 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
```

Практическое задание 8.3.3:

- 1) Получите информацию о всех индексах коллекции *unicorns*

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

- 2) Удалите все индексы, кроме индекса для идентификатора.

```
learn> db.unicorns.dropIndexes();
{
  nIndexesWas: 2,
  msg: 'non-_id indexes dropped for collection',
  ok: 1
}
```

- 3) Попробуйте удалить индекс для идентификатора.

```
learn> db.unicorns.dropIndex("_id");
MongoInternalError: index not found with name [_id]
```

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.

```
learn> for (i = 0; i < 10000; i++) {db.numbers.insertOne ({value: i})};
```

```
learn> db.numbers.find().sort({_id: -1}).limit (4);
[
  { _id: ObjectId("64786b79651c4760106c6d2f"), value: 9999 },
  { _id: ObjectId("64786b79651c4760106c6d2e"), value: 9998 },
  { _id: ObjectId("64786b79651c4760106c6d2d"), value: 9997 },
  { _id: ObjectId("64786b79651c4760106c6d2c"), value: 9996 }
]
```

- 3) Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра *executionTimeMillis*)

316 ms

- 4) Создайте индекс для ключа *value*.

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

```
learn> db.numbers.createIndex ({ value: 1 })
```

```
learn> db.numbers.getIndexes();  
[  
  { v: 2, key: { _id: 1 }, name: '_id_' },  
  { v: 2, key: { value: 1 }, name: 'value_1' }  
]
```

6) Выполните запрос 2.

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

```
console 273 ms
```

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

С индексом эффективнее.

ЗАКЛЮЧЕНИЕ

В ходе работы были получены практические навыки работы с CRUD - операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, с ссылками и индексами в базе данных MongoDB. MongoDB предоставляет мощный CLI интерфейс для выполнения CRUD операций, отличительной особенностью является интеграция полноценного языка программирования: Javascript.