

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

**Отчет**

по лабораторной работе №6 «Процедуры, функции, триггеры в PostgreSQL»

по дисциплине «Работа с БД в СУБД MongoDB»

Автор: Хурс П.И..

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

## Оглавление

Вывод.....	18
------------	----

**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).

Создайте базу данных learn.

```
Atlas atlas-g7sk72-shard-0 [primary] learn> use learn
already on db learn
```

*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'});
```

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

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

```
Atlas atlas-g7sk72-shard-0 [primary] test> db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'],
weight: 600, gender: 'm', vampires: 63});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('658b346f45c85cd9a290cc82') }
}
```

*Проверьте содержимое коллекции с помощью метода find.*

```

Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find()
[
  {
    _id: ObjectId('658b362845c85cd9a290cc8d'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('658b362845c85cd9a290cc8e'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('658b362845c85cd9a290cc8f'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('658b362845c85cd9a290cc90'),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('658b362945c85cd9a290cc91'),

```

## 2.2 ВЫБОРКА ДАННЫХ ИЗ БД

### **Практическое задание 2.2.1:**

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

```

Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({gender: 'm'}).sort({name: 1})
[
  {
    _id: ObjectId('658b366e45c85cd9a290cc9a'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('658b362845c85cd9a290cc8d'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('658b362945c85cd9a290cc93'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('658b363045c85cd9a290cc96'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('658b363045c85cd9a290cc94'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {

```

```

Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({gender: 'f'}).sort({name: 1
})
[
  {
    _id: ObjectId('658b362845c85cd9a290cc8e'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('658b362945c85cd9a290cc92'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('658b368b45c85cd9a290cc9b'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('658b363045c85cd9a290cc95'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('658b363045c85cd9a290cc97'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('658b363245c85cd9a290cc98'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('658b362945c85cd9a290cc91'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  }
]

```

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

```

Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({gender: 'f', loves:'carrot'
})
[
  {
    _id: ObjectId('658b362845c85cd9a290cc8e'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('658b362945c85cd9a290cc91'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('658b363045c85cd9a290cc97'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('658b363245c85cd9a290cc98'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]

```

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({gender: 'f', loves: 'carrot'
}).limit(1)
[
  {
    _id: ObjectId('658b362845c85cd9a290cc8e'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
```

### **Практическое задание 2.2.2:**

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

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({gender: "m"}, {gender: 0, lov
es: 0})
[
  {
    _id: ObjectId('658b362845c85cd9a290cc8d'),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId('658b362845c85cd9a290cc8f'),
    name: 'Unicrom',
    weight: 984,
    vampires: 182
  },
  {
    _id: ObjectId('658b362845c85cd9a290cc90'),
    name: 'Roooooodles',
    weight: 575,
    vampires: 99
  },
  {
    _id: ObjectId('658b362945c85cd9a290cc93'),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  },
  {
    _id: ObjectId('658b363045c85cd9a290cc94'),
    name: 'Raleigh',
    weight: 421,
    vampires: 2
  },
  {
    _id: ObjectId('658b363045c85cd9a290cc96'),
    name: 'Pilot',
    weight: 650,
    vampires: 54
  },
  {
    _id: ObjectId('658b366e45c85cd9a290cc9a'),
    name: 'Dunx',
    weight: 704,
    vampires: 165
  }
]
```

### **Практическое задание 2.2.3:**

*Вывести список единорогов в обратном порядке добавления.*

```

Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find().sort({$natural:-1})
[
  {
    _id: ObjectId('658b368b45c85cd9a290cc9b'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('658b366e45c85cd9a290cc9a'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('658b363245c85cd9a290cc98'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('658b363045c85cd9a290cc97'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('658b363045c85cd9a290cc96'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('658b363045c85cd9a290cc95'),

```

#### **Практическое задание 2.1.4:**

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



```

Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({}, {_id:0, loves:{$slice:[0,
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: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    name: 'Solnara',
    loves: [ 'apple' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    name: 'Ayna',
    loves: [ 'strawberry' ],
    weight: 733,

```

## 2.3 ЛОГИЧЕСКИЕ ОПЕРАТОРЫ

### Практическое задание 2.3.1:

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

```

Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({weight:{$lt:700, $gt:500},gender:'f'}, {_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'
  },
  {
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]

```

### Практическое задание 2.3.2:

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

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({weight:{$gt:500},loves:{$all:["grape", "lemon"]}}, {_id:0, })
[
  {
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
]
```

### **Практическое задание 2.3.3:**

*Найти всех единорогов, не имеющих ключ vampires.*

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({vampires:{$exists:false}})
[
  {
    _id: ObjectId('658b362845c85cd9a290cc8d'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('658b362845c85cd9a290cc8e'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('658b362845c85cd9a290cc8f'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('658b362845c85cd9a290cc90'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('658b362945c85cd9a290cc91'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('658b362945c85cd9a290cc92'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('658b362945c85cd9a290cc93'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('658b363045c85cd9a290cc94'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
  }
]
```

### **Практическое задание 2.3.4:**

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

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({gender:'m'}, {name:1, _id:0, loves:{$slice:[0,1]}})
[
  { name: 'Horny', loves: [ 'carrot' ] },
  { name: 'Unicrom', loves: [ 'energon' ] },
  { name: 'Roooooodles', loves: [ 'apple' ] },
  { name: 'Kenny', loves: [ 'grape' ] },
  { name: 'Raleigh', loves: [ 'apple' ] },
  { name: 'Pilot', loves: [ 'apple' ] },
  { name: 'Dunx', loves: [ 'grape' ] }
]
```

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

### 3.1 ЗАПРОС К ВЛОЖЕННЫМ ОБЪЕКТАМ

#### Практическое задание 3.1.1:

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"}}
```

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.towns.insertMany([
...   {
...     name: "Punxsutawney",
...     population: 6200,
...     last_sensus: ISODate("2008-01-31"),
...     famous_for: [],
...     mayor: {
...       name: "Jim Wehrle"
...     }
...   },
...   {
...     name: "New York",
...     population: 22200000,
...     last_sensus: ISODate("2009-07-31"),
...     famous_for: ["statue of liberty", "food"],
...     mayor: {
...       name: "Michael Bloomberg",
...       party: "I"
...     }
...   },
...   {
...     name: "Portland",
...     population: 528000,
...     last_sensus: ISODate("2009-07-20"),
...     famous_for: ["beer", "food"],
...     mayor: {
...       name: "Sam Adams",
...       party: "D"
...     }
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('658b3e9a45c85cd9a290cca2'),
    '1': ObjectId('658b3e9a45c85cd9a290cca3'),
    '2': ObjectId('658b3e9a45c85cd9a290cca4')
  }
}
```

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

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.towns.find(
...   { "mayor.party": "I" },
...   { name: 1, "mayor.name": 1, _id: 0 }
... );
[ { name: 'New York', mayor: { name: 'Michael Bloomberg' } } ]
```

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

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.towns.find(
...   { "mayor.party": { $exists: false } },
...   { name: 1, "mayor.name": 1, _id: 0 }
... );
[ { name: 'Punxsutawney', mayor: { name: 'Jim Wehrle' } } ]
```

### Практическое задание 3.1.2:

3. Сформировать функцию для вывода списка самцов единорогов.
4. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.

5. Вывести результат, используя forEach.

```
Atlas atlas-g7sk72-shard-0 [primary] learn> function printMaleUnicorns() {
...   var cursor = db.unicorns.find({ gender: 'm' }).sort({ name: 1 }).limit(2);
...   cursor.forEach(function(unicorn) {
...     print('Name: ' + unicorn.name + ', Loves: ' + unicorn.loves.join(', ') + ', Weight: ' + unicorn.weight + ', Vampires: ' + unicorn.vampires);
...   });
... }
[Function: printMaleUnicorns]
Atlas atlas-g7sk72-shard-0 [primary] learn>

Atlas atlas-g7sk72-shard-0 [primary] learn> printMaleUnicorns();
Name: Dunx, Loves: grape, watermelon, Weight: 704, Vampires: 165
Name: Horny, Loves: carrot, papaya, Weight: 600, Vampires: 63
```

### Практическое задание 3.2.1:

Вывести количество самок единорогов весом от полутонны до 600 кг.

### Практическое задание 3.2.2:

Вывести список предпочтений.

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.distinct('loves')
[
  'apple',
  'chocolate',
  'grape',
  'papaya',
  'strawberry',
  'watermelon',
  'carrot',
  'energon',
  'lemon',
  'redbull',
  'sugar'
]
```

### Практическое задание 3.2.3:

Посчитать количество особей единорогов обоих полов.

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.aggregate({$group: {_id: "$gender", count: {$sum: 1}}})
[ { _id: 'm', count: 7 }, { _id: 'f', count: 7 } ]
```

### Практическое задание 3.3.2:

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

```
TypeError: db.unicorns.save is not a function
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.updateOne({name:"Ayna"}, [{set:{weight:500, vampires:51}}])
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.findOne({name:"Ayna"})
{
  _id: ObjectId('658b362945c85cd9a290cc92'),
  name: 'Ayna',
  loves: ['strawberry', 'lemon'],
  weight: 733,
  gender: 'f',
  vampires: 51,
  weight: 500
}
```

2. Проверить содержимое коллекции unicorns

### Практическое задание 3.3.3:

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

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.updateOne({name:"Raleigh"}, [{set:{loves:["redbull"]} }])
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.findOne({name:"Raleigh"})
{
  _id: ObjectId('658b363045c85cd9a290cc94'),
  name: 'Raleigh',
  loves: [ 'redbull' ],
  weight: 421,
  gender: 'm',
  vampires: 2
}
```

### Практическое задание 3.3.4:

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

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.updateMany({gender:"m"}, {$inc:{vampires:5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 7,
  modifiedCount: 7,
  upsertedCount: 0
}
```

2. Проверить содержимое коллекции unicorns.

### Практическое задание 3.3.5:

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

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.towns.updateOne({name: "Portland"}, [{set: "mayor.party"}])
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-g7sk72-shard-0 [primary] learn> db.towns.find({name: "Portland"})
[
  {
    _id: ObjectId('658b3e9a45c85cd9a290cca4'),
    name: 'Portland',
    population: 528000,
    last_sensus: ISODate('2009-07-20T00:00:00.000Z'),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams' }
  }
]
```

2. Проверить содержимое коллекции *towns*.

### **Практическое задание 3.3.6:**

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

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.updateOne({name: "Pilot"}, {$push: {loves: "chocolate"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({name: "Pilot"})
[
  {
    _id: ObjectId('658b363045c85cd9a290cc96'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
```

2. Проверить содержимое коллекции *unicorns*.

### **Практическое задание 3.3.7:**

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

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.updateOne({name: "Aurora"}, {$push: {loves: {$each: ["sugar", "lime"]}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find({name: "Aurora"})
[
  {
    _id: ObjectId('658b362845c85cd9a290cc8e'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lime' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
```

2. Проверить содержимое коллекции *unicorns*.

### **Практическое задание 3.4.1:**

4. 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"}}

```

```

Atlas atlas-g7sk72-shard-0 [primary] learn> db.towns.insertMany([
...   {
...     name: "Punxsutawney",
...     population: 6200,
...     last_sensus: ISODate("2008-01-31"),
...     famous_for: ["phil the groundhog"],
...     mayor: {
...       name: "Jim Wehrle"
...     }
...   },
...   {
...     name: "New York",
...     population: 22200000,
...     last_sensus: ISODate("2009-07-31"),
...     famous_for: ["statue of liberty", "food"],
...     mayor: {
...       name: "Michael Bloomberg",
...       party: "I"
...     }
...   },
...   {
...     name: "Portland",
...     population: 528000,
...     last_sensus: ISODate("2009-07-20"),
...     famous_for: ["beer", "food"],
...     mayor: {
...       name: "Sam Adams",
...       party: "D"
...     }
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('658b43d745c85cd9a290cca5'),
    '1': ObjectId('658b43d745c85cd9a290cca6'),
    '2': ObjectId('658b43d745c85cd9a290cca7')
  }
}

```

5. Удалите документы с беспартийными мэрами.

```

Atlas atlas-g7sk72-shard-0 [primary] learn> db.towns.remove({ "mayor.party": { $exists: false } });
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 3 }

```

6. Проверьте содержание коллекции.
7. Очистите коллекцию.
8. Просмотрите список доступных коллекций.

```

{ acknowledged: true, deletedCount: 5 }
Atlas atlas-g7sk72-shard-0 [primary] learn> db.towns.drop();
true
Atlas atlas-g7sk72-shard-0 [primary] learn> db.towns.find();

Atlas atlas-g7sk72-shard-0 [primary] learn> show collections;
unicorns
users
Atlas atlas-g7sk72-shard-0 [primary] learn>

```

## 4 ССЫЛКИ И РАБОТА С ИНДЕКСАМИ В БАЗЕ ДАННЫХ MONGODB

### 4.1 ССЫЛКИ В БД

#### Практическое задание 4.1.1:

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

```

users
Atlas atlas-g7sk72-shard-0 [primary] learn> db.habitats.insertMany([
...   { _id: "forest", fullName: "Enchanted Forest", description: "A magical and dense forest with mystical pro
perties." },
...   { _id: "mountain", fullName: "Crystal Mountain", description: "A sparkling mountain range with high peaks
"},
...   { _id: "valley", fullName: "Sunshine Valley", description: "A bright and sunny valley, always in spring."
}
... ]);
{
  acknowledged: true,
  insertedIds: { '0': 'forest', '1': 'mountain', '2': 'valley' }
}

```

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

```

Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.updateOne({ name: "Horny" }, { $set: { habitat: "forest"
} });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.updateOne({ name: "Aurora" }, { $set: { habitat: "mountai
n" } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-g7sk72-shard-0 [primary] learn>

```

9. *Проверьте содержание коллекции единорогов.*

```

Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.find();
[
  {
    _id: ObjectId('658b362845c85cd9a290cc8d'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 68,
    habitat: 'forest'
  },
  {
    _id: ObjectId('658b362845c85cd9a290cc8e'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lime' ],
    weight: 450,
    gender: 'f',
    vampires: 43,
    habitat: 'mountain'
  },
  {

```



## 4.3 УПРАВЛЕНИЕ ИНДЕКСАМИ

### Практическое задание 4.3.1:

11. Получите информацию о всех индексах коллекции *unicorns*.

```
name_1
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_',
    { v: 2, key: { name: 1 }, name: 'name_1', unique: true }
]
```

12. Удалите все индексы, кроме индекса для идентификатора.

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.dropIndex("name_1")
{
  nIndexesWas: 2,
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1703626117, i: 4 }),
    signature: {
      hash: Binary.createFromBase64('Quay5oDQAXZFfLSU0W4abNaGfD9E=', 0),
      keyId: Long('7261632508792406018')
    }
  },
  operationTime: Timestamp({ t: 1703626117, i: 3 })
}
```

13. Попробуйте удалить индекс для идентификатора.

```
Atlas atlas-g7sk72-shard-0 [primary] learn> db.unicorns.dropIndex("_id")
MongoServerError: cannot drop _id index
Atlas atlas-g7sk72-shard-0 [primary] learn>
```

## 4.4 ПЛАН ЗАПРОСА

### Практическое задание 4.4.1:

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 < 100000; i++){db.numbers.insert({value: i})}
;
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('658348f9db12a495f317eabc') }
}
learn> ;

[learn> db.numbers.find().count();
100000
[learn> db.numbers.find({value: {$in: [9996, 9997, 9998, 9999]}})
[
  { _id: ObjectId('658348d2db12a495f3168b29'), value: 9996 },
  { _id: ObjectId('658348d2db12a495f3168b2a'), value: 9997 },
  { _id: ObjectId('658348d2db12a495f3168b2b'), value: 9998 },
  { _id: ObjectId('658348d2db12a495f3168b2c'), value: 9999 }
]
```

3. Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра `executionTimeMillis`)
4. Создайте индекс для ключа `value`.

```
{
  executionStats: {
    executionSuccess: true,
    nReturned: 4,
    executionTimeMillis: 63,
    totalKeysExamined: 0,
    totalDocsExamined: 100000,
  }
}
```

5. Получите информацию о всех индексах коллекции `numbers`.
6. Выполните запрос 2.

```
learn> db.numbers.ensureIndex({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({value: {$in: [9996, 9997, 9998, 9999]}})
[
  { _id: ObjectId('658348d2db12a495f3168b29'), value: 9996 },
  { _id: ObjectId('658348d2db12a495f3168b2a'), value: 9997 },
  { _id: ObjectId('658348d2db12a495f3168b2b'), value: 9998 },
  { _id: ObjectId('658348d2db12a495f3168b2c'), value: 9999 }
]
```

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

```
executionStats: {
  executionSuccess: true,
  nReturned: 4,
  executionTimeMillis: 1,
  totalKeysExamined: 5,
  totalDocsExamined: 4,
}
```

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

Ускорение с индексом очень сильно заметно.

## Вывод

В ходе лабораторной работы была изучена работа с NoSQL БД MongoDB.