

Министерство науки и высшего образования Российской Федерации  
федеральное государственное автономное образовательное учреждение  
высшего образования  
“Национальный исследовательский университет ИТМО”

Факультет инфокоммуникационных технологий

## **ЛАБОРАТОРНАЯ РАБОТА №5(7)**

**Процедуры, функции, триггеры в PostgreSQL**  
**по дисциплине:**  
**«Проектирование и реализация баз данных»**

**Выполнил:**

Кондратьев Алексей Алексеевич  
Группа К3241

**Проверила:**

Горова Марина Михайловна

Санкт-Петербург  
2022

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

## Выполнение:

### Практическое задание 8.1.1:

1. Создайте базу данных learn

```
> use learn
switched to db learn
```

2. Заполните коллекцию единорогов unicorns:

```
> db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Rooodooles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
WriteResult({ "nInserted" : 1 })
```

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

```
> document=({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})
{
  "name" : "Dunx",
  "loves" : [
    "grape",
    "watermelon"
  ],
  "weight" : 704,
  "gender" : "m",
  "vampires" : 165
}
> db.unicorns.insertOne(document)
{
  "acknowledged" : true,
  "insertedId" : ObjectId("629670f6b6d2559bca11f384")
}
```

4. Проверьте содержимое коллекции с помощью find.

```
> db.unicorns.find()
{ "_id" : ObjectId("62966fecb6d2559bca11f379"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37a"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37b"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37c"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37d"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37e"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37f"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62966fecb6d2559bca11f380"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62966fecb6d2559bca11f381"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62966fecb6d2559bca11f382"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62966fecb6d2559bca11f383"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629670f6b6d2559bca11f384"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>
```

## 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.

```
> db.unicorns.find({gender: 'm'}).limit(3).sort({name: 1})
{ "_id" : ObjectId("629670f6b6d2559bca11f384"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62966fecb6d2559bca11f379"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37f"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
> db.unicorns.find({gender: 'f'}).limit(3).sort({name: 1})
{ "_id" : ObjectId("62966fecb6d2559bca11f37a"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37e"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62966fecb6d2559bca11f381"), "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.

```
> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
{
  "_id" : ObjectId("62966fecb6d2559bca11f37a"),
  "name" : "Aurora",
  "loves" : [
    "carrot",
    "grape"
  ],
  "weight" : 450,
  "gender" : "f",
  "vampires" : 43
}
> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
{ "_id" : ObjectId("62966fecb6d2559bca11f37a"), "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.

```
> db.unicorns.find({gender: 'm'}, {gender: 0, loves: 0}).limit(3).sort({name: 1})
{ "_id" : ObjectId("629670f6b6d2559bca11f384"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("62966fecb6d2559bca11f379"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37f"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
```

## Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({$natural: -1})
{ "_id" : ObjectId("629670f6b6d2559bca11f384"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62966fecb6d2559bca11f383"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62966fecb6d2559bca11f382"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62966fecb6d2559bca11f381"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62966fecb6d2559bca11f380"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37f"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37e"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37d"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37c"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37b"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37a"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62966fecb6d2559bca11f379"), "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.

```
> 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.

```
> 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.

```
> 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.

```
> db.unicorns.find({vampires: {$exists: false}})
{ "_id" : ObjectId("62966fecb6d2559bca11f383"), "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.

```
> db.unicorns.find({gender: 'm'}, {name: 1, _id: 0, 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" ] }
```

Практическое задание 8.2.1:

1. Создайте коллекцию towns, включающую следующие документы:

```
> db.towns.insertOne({name: "Punxsutawney ",
... populatiuon: 6200,
... last_sensus: ISODate("2008-01-31"),
... famous_for: [""],
... mayor: {
...   name: "Jim Wehrle"
... }}
... )
{
  "acknowledged" : true,
  "insertedId" : ObjectId("62967764b6d2559bca11f385")
}
```

```
> db.towns.insertOne({name: "New York",
... populatiuon: 22200000,
... last_sensus: ISODate("2009-07-31"),
... famous_for: ["status of liberty", "food"],
... mayor: {
...   name: "Michael Bloomberg",
...   party: "I"}}
... )
{
  "acknowledged" : true,
  "insertedId" : ObjectId("62967802b6d2559bca11f386")
}
```

```
> db.towns.insertOne({name: "Portland",
... populatiuon: 528000,
... last_sensus: ISODate("2009-07-20"),
... famous_for: ["beer", "food"],
... mayor: {
...   name: "Sam Adams",
...   party: "D"}}
... )
{
  "acknowledged" : true,
  "insertedId" : ObjectId("62967835b6d2559bca11f387")
}
```

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

```
> db.towns.find({'mayor.party': 'I'}, {_id: 0, name: 1, mayor: 1})
{ "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.

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

## Практическое задание 8.2.2:

### 1. Сформировать функцию для вывода списка самцов единорогов.

```
> fn = function() {return this.gender == 'm'}
function() {return this.gender == 'm'}
> db.unicorns.find(fn)
{ "_id" : ObjectId("62966fecb6d2559bca11f379"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37b"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37c"), "name" : "Roooddles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37f"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62966fecb6d2559bca11f380"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62966fecb6d2559bca11f382"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629670f6b6d2559bca11f384"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

2. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.
3. Вывести результат, используя forEach.

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

## Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

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

## Практическое задание 8.2.4:

Вывести список предпочтений.

```
> 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.

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

## Практическое задание 8.2.6:

Выполнить команду:

Проверить содержимое коллекции unicorns



```
> db.unicorns.save({name: 'Barney', loves: ['grape'],
... weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find({name: "Barney"})
{ "_id" : ObjectId("62967cd5b6d2559bca11f388"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

### Практическое задание 8.2.7:

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

### Проверить содержимое коллекции unicorns

```
> db.unicorns.updateOne({name: "Ayna"}, {$set: {name: "Ayna", loves: ["strawberry", "lemon"], weight: 800, gender: 'f', vampires: 51}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.unicorns.find({name: "Ayna"})
{ "_id" : ObjectId("62966fecb6d2559bca11f37e"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
```

### Практическое задание 8.2.8:

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

### Проверить содержимое коллекции unicorns.

```
> db.unicorns.updateOne({gender: 'm', name: "Raleigh"}, {$set: {loves: ["redbull"]}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.unicorns.find({name: "Raleigh"})
{ "_id" : ObjectId("62966fecb6d2559bca11f380"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
```

### Практическое задание 8.2.9:

Всем самцам единорогов увеличить количество убитых вапмиров на 5.

### Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({gender: "m"})
{ "_id" : ObjectId("62966fecb6d2559bca11f379"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37b"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37c"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37f"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62966fecb6d2559bca11f380"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62966fecb6d2559bca11f382"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629670f6b6d2559bca11f384"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62967cd5b6d2559bca11f388"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
> db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}})
{ "acknowledged" : true, "matchedCount" : 8, "modifiedCount" : 8 }
> db.unicorns.find({gender: "m"})
{ "_id" : ObjectId("62966fecb6d2559bca11f379"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37b"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37c"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37f"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("62966fecb6d2559bca11f380"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("62966fecb6d2559bca11f382"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("629670f6b6d2559bca11f384"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("62967cd5b6d2559bca11f388"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

### Практическое задание 8.2.10:

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

### Проверить содержимое коллекции towns.

```
{ "_id" : ObjectId("62967835b6d2559bca11f387"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
> db.towns.updateOne({name: "Portland"}, {$unset: {mayor.party: 1}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.towns.find({name: "Portland"})
{ "_id" : ObjectId("62967835b6d2559bca11f387"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams" } }
```

### Практическое задание 8.2.11:

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

### Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: "Pilot"})
{ "_id" : ObjectId("62966fecb6d2559bca11f382"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
> db.unicorns.updateOne({name: "Pilot"}, {$push: {loves: 'chocolate'}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.unicorns.find({name: "Pilot"})
{ "_id" : ObjectId("62966fecb6d2559bca11f382"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
```

### Практическое задание 8.2.12:

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

Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: "Aurora"})
{ "_id" : ObjectId("62966fecb6d2559bca11f37a"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
> db.unicorns.updateOne({name: "Aurora"}, {$addToSet: {loves: {$each: ['sugar', 'lemon']}}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.unicorns.find({name: "Aurora"})
{ "_id" : ObjectId("62966fecb6d2559bca11f37a"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

### Практическое задание 8.2.13:

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

Удалите документы с беспартийными мэрами

Проверьте содержание коллекции

```
> db.towns.deleteOne({"mayor.party": {$exists: false}})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.towns.find()
{ "_id" : ObjectId("62967802b6d2559bca11f386"), "name" : "New York", "populatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("62967835b6d2559bca11f387"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams" } }
```

Очистите коллекцию

Просмотрите список доступных коллекций



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

### Практическое задание 8.3.1:

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

```
> db.areas.insertMany([{"_id": 1, name: 'Sky', desc: 'Light-Blue area with white chocolate milk'}, {"_id": 2, name: 'Rainbow field', desc: 'Bright area with flowers and the waterfall'}])
{ "acknowledged" : true, "insertedIds" : [ 1, 2 ] }
```

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

```
> db.unicorns.updateOne({name: 'Horny'}, {$set: {areas: {$ref: "areas", $id: "1"}}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.unicorns.updateOne({name: 'Aurora'}, {$set: {areas: {$ref: "areas", $id: "2"}}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

### Проверьте содержание коллекции единорогов

```
> db.unicorns.find()
{ "_id" : ObjectId("62966fecb6d2559bca11f379"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68, "areas" : DBRef("areas", "1") }
{ "_id" : ObjectId("62966fecb6d2559bca11f37a"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "areas" : DBRef("areas", "2") }
{ "_id" : ObjectId("62966fecb6d2559bca11f37b"), "name" : "Unicrom", "loves" : [ "enengon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37c"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37d"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37e"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("62966fecb6d2559bca11f37f"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("62966fecb6d2559bca11f380"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("62966fecb6d2559bca11f381"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62966fecb6d2559bca11f382"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("62966fecb6d2559bca11f383"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629670f6b6d2559bca11f384"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("62967cd5b6d2559bca11f388"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

### Практическое задание 8.3.2:

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

```
> db.unicorns.ensureIndex({name: 1}, {unique: true})
uncaught exception: TypeError: db.unicorns.ensureIndex is not a function :
```

Нельзя, имена могут совпадать

### Практическое задание 8.3.3:

Получите информацию о всех индексах коллекции unicorns .

Удалите все индексы, кроме индекса для идентификатора.

Попытайтесь удалить индекс для идентификатора.

```
> db.unicorns.getIndexes()
[ { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_" } ]
> db.unicorns.dropIndex("_id_")
{
  "ok" : 0,
  "errmsg" : "cannot drop _id index",
  "code" : 72,
  "codeName" : "InvalidOptions"
}
```

### Практическое задание 8.3.4:

Создайте объемную коллекцию numbers, задействовав курсор:

```
> for(i = 0; i < 100000; i++){db.numbers.insertOne({value: i})}
{
  "acknowledged" : true,
  "insertedId" : ObjectId("62968d64b6d2559bca137a28")
}
```

Выберите последних четыре документа.

```
> db.numbers.find().sort({$natural: -1}).limit(4)
{ "_id" : ObjectId("62968d64b6d2559bca137a28"), "value" : 99999 }
{ "_id" : ObjectId("62968d64b6d2559bca137a27"), "value" : 99998 }
{ "_id" : ObjectId("62968d64b6d2559bca137a26"), "value" : 99997 }
{ "_id" : ObjectId("62968d64b6d2559bca137a25"), "value" : 99996 }
```

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

С курсором

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 4,
  "executionTimeMillis" : 0,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 4,
  "executionStages" : {
    "stage" : "LIMIT",
```

Без курсора

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 4,
  "executionTimeMillis" : 2,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 4,
  "executionStages" : {
    "stage" : "LIMIT",
    "nReturned" : 4,
    "executionTimeMillisEstimate" : 0,
    "works" : 6,
```

Создайте индекс для ключа value.

```
> db.numbers.createIndex({"value": 1})
{
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "createdCollectionAutomatically" : false,
  "ok" : 1
}
```

Получите информацию о всех индексах коллекции numbers

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

Выполните запрос 2

```
> db.numbers.find().sort({$natural: -1}).limit(4)
{ "_id" : ObjectId("62968d64b6d2559bca137a28"), "value" : 99999 }
{ "_id" : ObjectId("62968d64b6d2559bca137a27"), "value" : 99998 }
{ "_id" : ObjectId("62968d64b6d2559bca137a26"), "value" : 99997 }
{ "_id" : ObjectId("62968d64b6d2559bca137a25"), "value" : 99996 }
```

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

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 4,
  "executionTimeMillis" : 0,
  "totalKeysExamined" : 0,
```

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

В данной работе время выполнения запроса с индексом и без одинаково, думаю при большом количестве данных оно будет заметнее

**Вывод:** в ходе работы были получены практические работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.