Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

«Работа с БД в СУБД MongoDB»

по дисциплине «Проектирование и реализация баз данных»

Обучающийся Соболев Артём

Факультет прикладной информатики

Группа K3240

Направление подготовки 09.03.03 Прикладная информатика

Образовательная программа Мобильные и сетевые технологии 2023

Преподаватель Говорова Марина Михайловна

Санкт-Петербург

2025

ВВЕДЕНИЕ

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.

ВЫПОЛНЕНИЕ

Практическое задание 2.1.1

Задание:

1. Создать базу данных learn
2. Заполнить коллекцию единорогов unicorns

Задание: создать базу данных learn

Листинг

```
use learn
```

Задание: заполнить коллекцию единорогов unicorns

Листинг

```
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’});

```
test> use learn
switched to db learn
learn> 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'});
...
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('68136a8f2f7eb64960b5f8a3') }
}
```

Рисунок 1 – Результат выполнения запроса

Задание: вывести единорогов

Листинг

```
db.unicorns.find()
```

```
learn> db.unicorns.find()
[
  {
    _id: ObjectId('68136a8f2f7eb64960b5f899'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89a'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89b'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89c'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89d'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89e'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89f'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
```

Рисунок 2 – Результат выполнения запроса

Задание: добавить в коллекцию объект вторым способом

Листинг

```
document = ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})
db.unicorns.insert(document)
```

```
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.insert(document)
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('68136ae32f7eb64960b5f8a4') }
}
```

Рисунок 3 – Результат выполнения запроса

Рисунок 4 – Результат выполнения запроса

Практическое задание 2.2.1

Задание:

1. Сформировать запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.
2. Найти всех самок, которые любят carrot. Ограничить этот список первой особью с помощью функций findOne и limit.

Задание: сформировать запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.

Листинг

```
db.unicorns.find({gender: "f"}).sort({name: 1}).limit(3)
```

```
learn> db.unicorns.find({gender: "f"}).sort({name: 1}).limit(3)
[
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89a'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89e'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f8a1'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  }
]
```

Рисунок 5 – Результат выполнения запроса

Листинг 6

```
db.unicorns.find({gender: "m"}).sort({name: 1})
```

Рисунок 6 – Результат выполнения запроса

Задание: найти всех самок, которые любят carrot. Ограничить этот список первой особью с помощью функций findOne и limit.

Листинг

```
db.unicorns.find({“loves”: “carrot”}).limit(1)

db.unicorns.findOne({“loves”: “carrot”})
```

```
learn> db.unicorns.find({"loves": "carrot"}).limit(1)
[
  {
    _id: ObjectId('68136a8f2f7eb64960b5f899'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
]
learn> db.unicorns.findOne({"loves": "carrot"})
{
  _id: ObjectId('68136a8f2f7eb64960b5f899'),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
```

Рисунок 7 – Результат выполнения запроса

Практическое задание 2.2.2

Задание: модифицировать запрос для вывода списков самцов единорогов, исключив из результата информацию о предпочтениях и поле

Листинг

```
db.unicorns.find({gender: "m"}, {loves: false, _id: false}).sort({name: 1})
```

```
learn> db.unicorns.find({gender: "m"}, {loves: false, _id: false}).sort({name: 1})
[
  { name: 'Dunx', weight: 704, gender: 'm', vampires: 165 },
  { name: 'Horny', weight: 600, gender: 'm', vampires: 63 },
  { name: 'Kenny', weight: 690, gender: 'm', vampires: 39 },
  { name: 'Pilot', weight: 650, gender: 'm', vampires: 54 },
  { name: 'Raleigh', weight: 421, gender: 'm', vampires: 2 },
  { name: 'Roooooodles', weight: 575, gender: 'm', vampires: 99 },
  { name: 'Unicrom', weight: 984, gender: 'm', vampires: 182 }
]
learn>
```

Рисунок 8 – Результат выполнения запроса

Практическое задание 2.2.3

Задание: вывести список единорогов в обратном порядке добавления

Листинг

```
db.unicorns.find().sort({ $natural: -1})
```

Рисунок 9 – Результат выполнения запроса

Практическое задание 2.1.4

Задание: вывести список единорогов с названием первого любимого предпочтения, исключив идентификатор.

Листинг

```
db.unicorns.find({}, {_id: false, loves: {$slice: 1}})
```

Рисунок 10 – Результат выполнения запроса

Практическое задание 2.3.1

Задание: вывести список самок единорогов весом от полутонны до 700 кг, исключив вывод идентификатора

Листинг

```
db.unicorns.find({weight: {$gte: 500, $lt: 700}}, {_id: false})
```

```
learn> db.unicorns.find({weight: {$gte: 500, $lt: 700}}, {_id: false})
[
  {
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    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'
  }
]
```

Рисунок 11 – Результат выполнения запроса

Практическое задание 2.3.2

Задание: вывести список самцов единорогов весом от полутонны и предпочитающих grape и lemon, исключив вывод идентификатора

Листинг

```
db.unicorns.find({weight: {$gte: 500, $lt: 700}, gender: "m", loves: {$all: ["grape", "lemon"]}}, {_id: false})})
```

```
learn> db.unicorns.find({weight: {$gte: 500, $lt: 700}, gender: "m", loves: {$all: ["grape", "lemon"]}}, {_id: false})
[
  {
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
]
```

Рисунок 12 – Результат выполнения запроса

Практическое задание 2.3.3

Задание: найти всех единорогов, не имеющих ключ vampires

Листинг

```
db.unicorns.find({vampires: {$exists:false}})
```

```
learn> db.unicorns.find({vampires: {$exists:false}})
[
  {
    _id: ObjectId('68136a8f2f7eb64960b5f8a3'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
```

Рисунок 13 – Результат выполнения запроса

Практическое задание 2.3.4

Задание: вывести список упорядоченный список имен самцов единорогов с информацией об их первом предпочтении

Листинг

```
db.unicorns.find({gender: "m"}, {name: true, loves: {$slice: 1}, _id: false}).sort({name: 1})
```

```
learn> db.unicorns.find({gender: "m"}, {name: true, loves: {$slice: 1}, _id: false}).sort({name: 1})
[
  { name: 'Dunx', loves: [ 'grape' ] },
  { name: 'Horny', loves: [ 'carrot' ] },
  { name: 'Kenny', loves: [ 'grape' ] },
  { name: 'Pilot', loves: [ 'apple' ] },
  { name: 'Raleigh', loves: [ 'apple' ] },
  { name: 'Roooooodles', loves: [ 'apple' ] },
  { name: 'Unicrom', loves: [ 'energon' ] }
]
```

Рисунок 14 – Результат выполнения запроса

Запросы к базе данных Mongo DB. Выборка данных. Вложенные объекты. Использование курсоров. Агрегированные запросы. Изменение данных

Практическое задание 3.1.1

Задание:

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

Листинг

```
Db.towns.insertMany([
{name: “Punxsutawney “,
15opulation: 6200,
last_sensus: ISODate(“2008-01-31”),
famous_for: [“”],
mayor: {
  name: “Jim Wehrle”
}},
{name: “New York”,
15opulation: 22200000,
last_sensus: ISODate(“2009-07-31”),
famous_for: [“status 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”}}

])

```
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('6813839e2f7eb64960b5f8a5'),
    '1': ObjectId('6813839e2f7eb64960b5f8a6'),
    '2': ObjectId('6813839e2f7eb64960b5f8a7')
  }
}
```

Рисунок 15 – Результат выполнения запроса

Задание: сформировать запрос, который возвращает список городов с независимыми мэрами (party=»I»). Вывести только название города и информацию о мэре.

Листинг

```
db.towns.find({"mayor.party": "I"}, {name: true, mayor: true, _id: false})
```

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

Рисунок 16 – Вывод запроса

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

Листинг

```
db.towns.find({"mayor.party": {$exists:false}}, {name: true, mayor: true, _id: false})
```

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

Рисунок 17 – Вывод запроса

Практическое задание 3.1.2

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.

Задание: сформировать функцию для вывода списка самцов единорогов.

Листинг

```
fn = function() { return this.gender == 'm'; }
```

```
learn> fn = function() { return this.gender == 'm'; }
[Function: fn]
```

Рисунок 18 – Вывод запроса

Задание: создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке. Вывести результат, используя forEach

Листинг

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

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

Рисунок 19 – Вывод запроса

Практическое задание 3.2.1

Задание: вывести количество самок единорогов весом от полутонны до 600 кг

Листинг

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

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

Рисунок 20 – Вывод запроса

Практическое задание 3.2.2

Задание: вывести список предпочтений.

Листинг

```
db.unicorns.distinct("loves")
```

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

Рисунок 21 – Вывод запроса

Практическое задание 3.2.3

Задание: посчитать количество особей единорогов обоих полов

Листинг

```
db.unicorns.aggregate({"$group":{_id:"$gender",count:{$sum:1}}})
```

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

Рисунок 22 – Вывод запроса

Практическое задание 3.3.1

Задания:

1. Выполнить команду
2. Проверить содержимое коллекции unicorns

Листинг

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

```
learn> db.unicorns.save({name: 'Barny', loves: ['grape'],
... weight: 340, gender: 'm'})
...
TypeError: db.unicorns.save is not a function
```

Рисунок 23 – Вывод запроса

```
learn> db.unicorns.find({name: "Barny"})
```

Рисунок 24 – Вывод запроса

```
learn> db.unicorns.insertOne({
...   name: 'Barny',
...   loves: ['grape'],
...   weight: 340,
...   gender: 'm'
... })
{
  acknowledged: true,
  insertedId: ObjectId('681394762f7eb64960b5f8a8')
}
```

Рисунок 25 – Вывод запроса

```
learn> db.unicorns.find({name:"Barny"})
[
  {
    _id: ObjectId('681394762f7eb64960b5f8a8'),
    name: 'Barny',
    loves: [ 'grape' ],
    weight: 340,
    gender: 'm'
  }
]
```

Рисунок 26 – Вывод запроса

Практическое задание 3.3.2

Задание: для самки единорога Ayna внести изменения в БД: теперь ее вес 800, она убила 51 вапмира

```
learn> db.unicorns.find({name: "Ayna"})
[
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89e'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  }
]
```

Рисунок 27 – Вывод запроса

```
learn> db.unicorns.update(
...   { name: "Ayna" },
...   {
...     name: "Ayna",
...     loves: ["strawberry", "lemon"],
...     weight: 800,
...     gender: "f",
...     vampires: 51
...   },
...   { upsert: true }
... )
...
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
MongoInvalidArgumentError: Update document requires atomic operators
```

Рисунок 28 – Вывод запроса

Рисунок 29 – Вывод запроса

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

Рисунок 30 – Вывод запроса

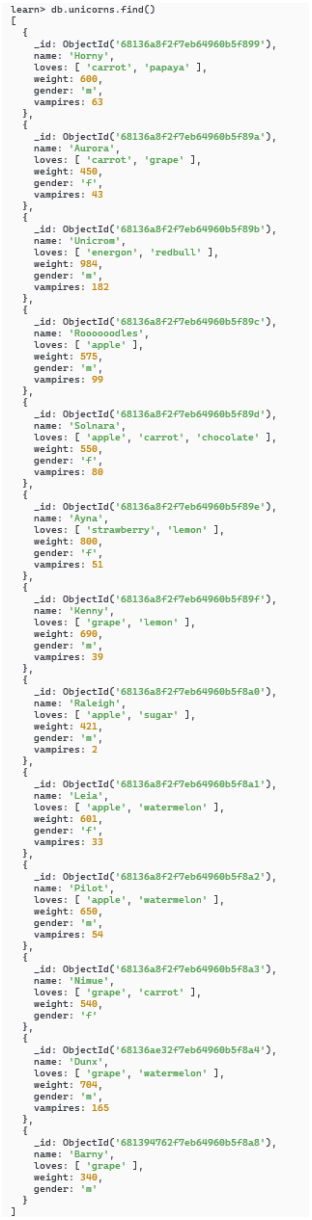

Рисунок 31 – Вывод запроса

Практическое задание 3.3.3

Задание:

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

Листинг

```
db.unicorns.updateOne(
  { name: "Raleigh" },
  { $set: { loves: ["redbull"] } }
)
```

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

Рисунок 32 – Вывод запроса

Рисунок 33 – Вывод запроса

Практическое задание 3.3.4

Задание:

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

```
learn> db.unicorns.find()
[
  {
    _id: ObjectId('68136a8f2f7eb64960b5f899'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89a'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89b'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89c'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89d'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89e'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 800,
    gender: 'f',
    vampires: 51
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89f'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f8a0'),
    name: 'Raleigh',
    loves: [ 'redbull' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f8a1'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f8a2'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f8a3'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('68136ae32f7eb64960b5f8a4'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('681394762f7eb64960b5f8a8'),
    name: 'Barny',
    loves: [ 'grape' ],
    weight: 340,
    gender: 'm'
  }
]
```

Рисунок 34 – Вывод запроса

Задание: Всем самцам единорогов увеличить количество убитых вапмиров на 5.

Листинг:

```
db.unicorns.updateMany(
  { gender: "m" },
  { $inc: { vampires: 5 } }
)
```

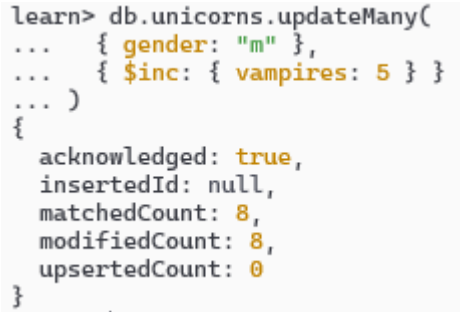

Рисунок 35 – Вывод запроса

Рисунок 36 – Вывод запроса

Практическое задание 3.3.5

Задание:

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

Листинг

```
db.towns.updateOne(
  { name: "Portland" },
  { $unset: { "mayor.party": "" } }
)
```

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

Рисунок 37 – Вывод запроса

```
learn> db.towns.find()
[
  {
    _id: ObjectId('6813839e2f7eb64960b5f8a5'),
    name: 'Punxsutawney ',
    populatiuon: 6200,
    last_sensus: ISODate('2008-01-31T00:00:00.000Z'),
    famous_for: [ '' ],
    mayor: { name: 'Jim Wehrle' }
  },
  {
    _id: ObjectId('6813839e2f7eb64960b5f8a6'),
    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('6813839e2f7eb64960b5f8a7'),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: ISODate('2009-07-20T00:00:00.000Z'),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams' }
  }
]
```

Рисунок 38 – Вывод запроса

Практическое задание 3.3.6

Задание: изменить информацию о самце единорога Pilot: теперь он любит и шоколад

Листинг

```
db.unicorns.updateOne(
  { name: "Pilot" },
  { $push: { loves: "chocolate" } }
)
```

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

Рисунок 39 – Выполнение запроса

```
learn> db.unicorns.find({name: "Pilot"})
[
  {
    _id: ObjectId('68136a8f2f7eb64960b5f8a2'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
```

Рисунок 40 – Вывод запроса

Практическое задание 3.3.7

Задание: изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и лимоны.

Листинг

```
db.unicorns.updateOne(
  { name: "Aurora" },
  {
   $addToSet: {
    loves: { $each: ["sugar", "lemon"] }
   }
  }
)
```

```
learn> db.unicorns.updateOne(
...    { name: "Aurora" },
...    {
...      $addToSet: {
...        loves: { $each: ["sugar", "lemon"] }
...      }
...    }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Рисунок 41 – Вывод запроса

```
learn> db.unicorns.find({name:"Aurora"})
[
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89a'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
```

Рисунок 42 – Вывод запроса

Практическое задание 3.4.1

Задание:

1. Создать коллекцию towns
2. Удалить документы с беспартийными мэрами.
3. Проверить содержание коллекции.
4. Очистить коллекцию.
5. Просмотреть список доступных коллекций.

Задание: создать коллекции towns

Листинг

```
db.towns.insertMany([
  {
    name: "Punxsutawney",
    popujatiion: 6200,
    last_sensus: ISODate("2008-01-31"),
    famous_for: ["phil the groundhog"],
    mayor: {
      name: "Jim Wehrle"
```

```
  }
},
{name: "New York",
 popujatiion: 22200000,
 last_sensus: ISODate("2009-07-31"),
 famous_for: ["status of liberty", "food"],
 mayor: {
  name: "Michael Bloomberg",
  party: "I"
 }},
{
 name: "Portland",
 popujatiion: 528000,
 last_sensus: ISODate("2009-07-20"),
 famous_for: ["beer", "food"],
 mayor: {
  name: "Sam Adams",
  party: "D"
 }}])
```

Рисунок 43 – Вывод запроса

Задание: удалить документы с беспартийными мэрами.

Листинг

```
db.towns.remove({ "mayor.party": { $exists: false } })
```

```
learn> db.towns.remove({ "mayor.party": { $exists: false } })
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 3 }
```

Рисунок 44 – Результат выполнения запроса

```
learn> db.towns.find()
[
  {
    _id: ObjectId('6813839e2f7eb64960b5f8a6'),
    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('68139e30dd65ef6cb4b5f89a'),
    name: 'New York',
    popujatiion: 22200000,
    last_sensus: ISODate('2009-07-31T00:00:00.000Z'),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  },
  {
    _id: ObjectId('68139e30dd65ef6cb4b5f89b'),
    name: 'Portland',
    popujatiion: 528000,
    last_sensus: ISODate('2009-07-20T00:00:00.000Z'),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
```

Рисунок 45 – Вывод запроса

Задание: очистить коллекцию.

Листинг

```
db.towns.deleteMany({})
```

```
learn> db.towns.deleteMany({})
{ acknowledged: true, deletedCount: 3 }
```

Рисунок 46 – Вывод запроса

Задание: посмотреть список доступных коллекций

Листинг

```
show collections
```

```
learn> show collections
towns
unicorns
```

Рисунок 47 – Вывод запроса

Практическое задание 4.1.1

Задание:

1. Создать коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.
2. Включить для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания.
3. Проверить содержание коллекции единорогов.

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

Листинг

```
db.habitats.insertMany([ { _id: "forest", full_name: "Green forest", description: "Very very green forest and beautiful forest" }, { _id: "mountains", full_name: "High mountains", description: "Very high mountains and very dangerous" }, { _id: "desert", full_name: "Infinity desert", description: "Only desert, no water" } ])
```

```
learn> db.habitats.insertMany([
...   {
...     _id: "forest",
...     full_name: "Green forest",
...     description: "Very very green forest and beautiful forest"
...   },
...   {
...     _id: "mountains",
...     full_name: "High mountains",
...     description: "Very high mountains and very dangerous"
...   },
...  {
...     _id: "desert",
...     full_name: "Infinity desert",
...     description: "Only desert, no water"
...   }
... ])
{
  acknowledged: true,
  insertedIds: { '0': 'forest', '1': 'mountains', '2': 'desert' }
}
```

Рисунок 48 – Вывод запроса

Задание: включить для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания

Листинг

```
db.unicorns.updateOne(
  { name: "Horny" },
  { $set: {
     habitat: {
       $ref: "habitats",
       $id: "forest"}}})
db.unicorns.updateOne(
  { name: "Pilot" },
  {set: { habitat: {
       $ref: "habitats",
       $id: "mountains"}}})
```

```
db.unicorns.updateOne(
  { name: "Nimue" },
  { $set: { habitat: {$ref: "habitats",
      $id: "desert"}}})
```

```
learn> db.unicorns.updateOne(
...   { name: "Horny" },
...   {
...     $set: {
...       habitat: {
...         $ref: "habitats",
...         $id: "forest"
...       }
...     }
...   }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.updateOne(
...   { name: "Pilot" },
...   {
...     $set: {
...       habitat: {
...         $ref: "habitats",
...         $id: "mountains"
...       }
...     }
...   }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.updateOne(
...   { name: "Nimue" },
...   {
...     $set: {
...       habitat: {
...         $ref: "habitats",
...         $id: "desert"
...       }
...     }
...   }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Рисунок 49 – Вывод запроса

Задание: проверить содержание коллекции единорогов

```
learn> db.unicorns.find()
[
  {
    _id: ObjectId('68136a8f2f7eb64960b5f899'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 68,
    habitat: DBRef('habitats', 'forest')
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89a'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89b'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 187
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89c'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 104
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89d'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89e'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 800,
    gender: 'f',
    vampires: 51
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f89f'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 44
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f8a0'),
    name: 'Raleigh',
    loves: [ 'redbull' ],
    weight: 421,
    gender: 'm',
    vampires: 7
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f8a1'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f8a2'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 650,
    gender: 'm',
    vampires: 59,
    habitat: DBRef('habitats', 'mountains')
  },
  {
    _id: ObjectId('68136a8f2f7eb64960b5f8a3'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f',
    habitat: DBRef('habitats', 'desert')
  },
  {
    _id: ObjectId('68136ae32f7eb64960b5f8a4'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 170
  },
  {
    _id: ObjectId('681394762f7eb64960b5f8a8'),
    name: 'Barny',
    loves: [ 'grape' ],
    weight: 340,
    gender: 'm',
    vampires: 5
  }
]
```

Рисунок 50 – Вывод запроса

Практическое задание 4.2.1

Задание: проверить, можно ли задать для коллекции unicorns индекс для ключа name с флагом unique.

Листинг

```
db.unicorns.createIndex(
  { name: 1 },
  { unique: true }
)
```

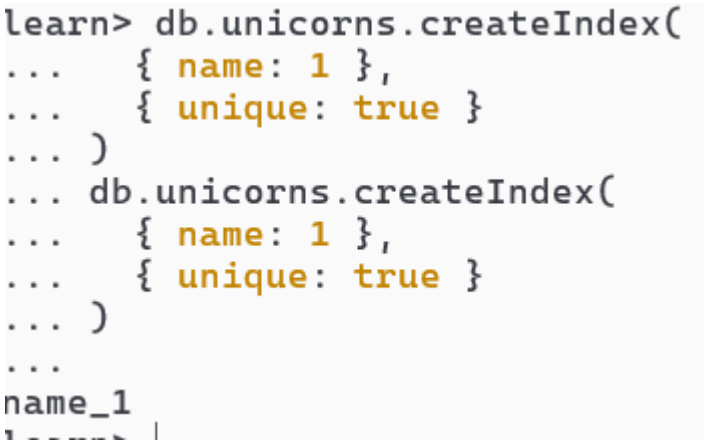

Рисунок 51 – Вывод запроса

Практическое задание 4.3.1

Задание:

1. Получить информацию обо всех индексах коллекции unicorns.
2. Удалить все индексы, кроме индекса для идентификатора.
3. Попытаться удалить индекс для идентификатора.

Задание: получить информацию обо всех индексах коллекции unicorns.

Листинг

```
db.unicorns.getIndexes()
```

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

Рисунок 52 – Результат выполнения запроса

Задание: удалить все индексы, кроме индекса для идентификатора.

Листинг

```
db.unicorns.dropIndex("name_1")
```

```
learn> db.unicorns.dropIndex("name_1")
{ nIndexesWas: 2, ok: 1 }
learn> db.unicorns.getIndexes()
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
```

Рисунок 53 – Результат выполнения запроса

Задание: попытаться удалить индекс для идентификатора

Листинг

```
db.unicorns.dropIndex("_id_")
```

```
learn> db.unicorns.dropIndex("_id_")
MongoServerError[InvalidOptions]: cannot drop _id index
learn> db.unicorns.getIndexes()
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
```

Рисунок 54 – Результат выполнения запроса

Практическое задание 4.4.1

Задание:

1. Создать объемную коллекцию numbers, задействовав курсор
2. Выбрать последних четыре документа.
3. Проанализировать план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)
4. Создать индекс для ключа value.
5. Получить информацию обо всех индексах коллекции numbres.
6. Выполнить запрос 2.
7. Проанализировать план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?
8. Сравнить время выполнения запросов с индексом и без. Дать ответ на вопрос: какой запрос более эффективен?

Задание: создать объемную коллекцию numbers, задействовав курсор

Листинг

```
for (i = 0; i < 100000; i++) {
  db.numbers.insert({ value: i })
}
```

```
learn> for (i = 0; i < 100000; i++) {
...   db.numbers.insert({ value: i })
... }
...
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.


{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6813a2c0dd65ef6cb4b77f3b') }
}
```

Рисунок 55 – Результат выполнения запроса

Задание: выбрать последних четыре документа

Листинг

```
db.numbers.find().sort({ $natural: -1 }).limit(4)
```

```
learn> db.numbers.find().sort({ $natural: -1 }).limit(4)
[
  { _id: ObjectId('6813a2c0dd65ef6cb4b77f3b'), value: 99999 },
  { _id: ObjectId('6813a2c0dd65ef6cb4b77f3a'), value: 99998 },
  { _id: ObjectId('6813a2c0dd65ef6cb4b77f39'), value: 99997 },
  { _id: ObjectId('6813a2c0dd65ef6cb4b77f38'), value: 99996 }
]
```

Рисунок 56 – Результат выполнения запроса

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

Листинг

```
db.numbers.explain("executionStats")
  .find()
  .sort({ $natural: -1 })
  .limit(4)
```

Рисунок 57 – Результат выполнения запроса

MongoDB выполнил полный поиск коллекции, просмотрев 4 документа, и вернул 4 результата.

Значение executionStats.executionTimeMillis равно 1 мс.

Задание: создать индекс для ключа value.

Листинг

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

```
learn> db.numbers.createIndex({ value: 1 })
value_1
```

Рисунок 58 – Результат выполнения запроса

Задание: получить информацию обо всех индексах коллекции numbers.

Листинг

```
db.numbers.getIndexes()
```

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

Рисунок 59 – Результат выполнения запроса

Задание: выполнить запрос 2.

Листинг

```
db.numbers.find().sort({ value: -1 }).limit(4)
```

```
learn> db.numbers.find().sort({ value: -1 }).limit(4)
[
  { _id: ObjectId('6813a2c0dd65ef6cb4b77f3b'), value: 99999 },
  { _id: ObjectId('6813a2c0dd65ef6cb4b77f3a'), value: 99998 },
  { _id: ObjectId('6813a2c0dd65ef6cb4b77f39'), value: 99997 },
  { _id: ObjectId('6813a2c0dd65ef6cb4b77f38'), value: 99996 }
]
```

Рисунок 60 – Результат выполнения запроса

Задание: проанализировать план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?

Листинг

```
db.numbers.explain("executionStats")
  .find()
  .sort({ value: -1 })
  .limit(4)
```

Рисунок 61 – Результат выполнения запроса

С установленным индексом MongoDB вместо полного сканирования коллекции выполняет индексное сканирование по индексу value_1. Время выполнения executionTimeMillis оказалось 0 мс.

При полном сканировании коллекции (без индекса) запрос занял 1 мс, а с индексом – 0 мс, поэтому использование индекса оказалось более эффективным.

ВЫВОД

Я овладел 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.