

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(УНИВЕРСИТЕТ ИТМО)

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6
по теме:
РАБОТА С БД В СУБД MONGODB

Руководитель НИР,
Преподаватель практики

подпись, дата

М.М. Говорова

Исполнитель НИР,
Студент

подпись, дата

А. А. Цырульников

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

1 Выполнение

1.1 Практическое задание 2.1.1

1.1.1 Создание базы данных learn

use learn

Листинг 1 — Создание базы данных learn

1.1.2 Заполнение коллекции 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'});
```

Листинг 2 — Вставка данных в базу данных 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('684bf2e3dc4dddfc31b5ffc') }
}
```

Рисунок 1 — Вывод консоли

1.1.3 Вывод единорогов

db.unicorns.find()

Листинг 3 — Команда вывода единорогов

```
learn> db.unicorns.find()
[
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff2'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff3'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff4'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff5'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  }
]
```

Рисунок 2 — Результат команды вывода единорогов ч.1

```

    },
    {
      _id: ObjectId('684bf2e3dc4dddfc31b5ff6'),
      name: 'Solnara',
      loves: [ 'apple', 'carrot', 'chocolate' ],
      weight: 550,
      gender: 'f',
      vampires: 80
    },
    {
      _id: ObjectId('684bf2e3dc4dddfc31b5ff7'),
      name: 'Ayna',
      loves: [ 'strawberry', 'lemon' ],
      weight: 733,
      gender: 'f',
      vampires: 40
    },
    {
      _id: ObjectId('684bf2e3dc4dddfc31b5ff8'),
      name: 'Kenny',
      loves: [ 'grape', 'lemon' ],
      weight: 690,
      gender: 'm',
    }
  ]
}

```

Рисунок 3 — Результат команды вывода единорогов ч.2

1.1.4 Добавление в коллекцию объекта вторым способом

```

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

db.unicorns.insert(document)

```

Листинг 4 — Команда добавления в коллекцию объекта вторым способом

```

learn> document = ({
...     name: 'Dunx',
...     loves: [ 'grape', 'watermelon' ],
...     weight: 704,
...     gender: 'm',
...     vampires: 165
...   })
...   db.unicorns.insert(document)
...   {
...     acknowledged: true,
...     insertedIds: { '0': ObjectId('684bf659dc4dddfc31b5ffd') }
...   }
learn>

```

Рисунок 4 — Результат выполнения запроса

```
learn> document = ({
...       name: 'Dunx',
...       loves: ['grape', 'watermelon'],
...       weight: 704,
...       gender: 'm',
...       vampires: 165
...     })
...     db.unicorns.insert(document)
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('684bf659dc4dddfc31b5ffd') }
}
learn>
```

Рисунок 5 — Результат выполнения запроса

1.2 Практическое задание 2.2.1

1.2.1 Вывод списков самок и самцов единорогов

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

Листинг 5 — Первые три самки, отсортированные по имени

```
learn> db.unicorns.find({gender: "f"}).sort({name: 1}).limit(3)
[
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff3'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff7'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffa'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  }
]
```

Рисунок 6 — Результат запроса самок

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

Листинг 6 — Список самцов, отсортированных по имени

```
learn> db.unicorns.find({gender: "m"}).sort({name: 1})
[
  {
    _id: ObjectId('684bf659dc4dddfc31b5ffdf'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 784,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff2'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 688,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff8'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffb'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 658,
```

Рисунок 7 — Результат запроса самцов ч.1

```
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff9'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff5'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff4'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  }
]
```

Рисунок 8 — Результат запроса самцов ч.2

1.2.2 Поиск самок, любящих carrot

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

Другой вариант

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

Листинг 7 — Поиск первой самки, любящей carrot

```
learn> db.unicorns.find({loves: "carrot"}).limit(1)
[
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff2'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
]
learn> db.unicorns.findOne({loves: "carrot"})
{
  _id: ObjectId('684bf2e3dc4dddfc31b5ff2'),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
learn>
```

Рисунок 9 — Результат выполнения запроса

1.3 Практическое задание 2.2.2

1.3.1 Исключение полей loves и _id у самцов

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

Листинг 8 — Вывод самцов без полей loves и _id

```
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 }
]
```

Рисунок 10 — Результат запроса

1.4 Практическое задание 2.2.3

1.4.1 Список единорогов в обратном порядке добавления

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

Листинг 9 — Вывод в обратном порядке

```
learn> db.unicorns.find().sort({$natural: -1})
[
  {
    _id: ObjectId('684bf659dc4dddfc31b5ffd'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffc'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffb'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffa'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
]
```

Рисунок 11 — Результат выполнения запроса ч.1

```

    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff8'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff7'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff6'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff5'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff4'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff3'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff2'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
]

```

Рисунок 12 — Результат выполнения запроса ч.2

1.5 Практическое задание 2.1.4

1.5.1 Первый любимый вкус каждого единорога

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

Листинг 10 — Вывод первого элемента массива loves

```
[learn> db.unicorns.find({}, {_id: false, loves: {$slice: 1}})
[
  {
    name: 'Horny',
    loves: [ 'carrot' ],
    weight: 690,
    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: 681,
    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: 784,
    gender: 'm',
    vampires: 165
  }
]
```

Рисунок 13 — Результат выполнения запроса

1.6 Практическое задание 2.3.1

1.6.1 Самки весом 500–700 кг

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

Листинг 11 — Самки в диапазоне веса

```
[learn> db.unicorns.find({weight: {$gte: 500, $lt: 700}, gender: "f"}, {_id: false})
[
  {
    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'
  }
]
```

Рисунок 14 — Результат выполнения запроса

1.7 Практическое задание 2.3.2

1.7.1 Самцы весом 500–700 кг, любящие grape и lemon

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

Листинг 12 — Самцы по сложному условию

```
[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 15 — Результат выполнения запроса

1.8 Практическое задание 2.3.3

1.8.1 Единороги без поля vampires

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

Листинг 13 — Поиск документов без ключа vampires

```
learn> db.unicorns.find({vampires: {$exists: false}})
[
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffc'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
```

Рисунок 16 — Результат выполнения запроса

1.9 Практическое задание 2.3.4

1.9.1 Имена самцов и их первый вкус

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

Листинг 14 — Имена самцов и первый элемент loves

```
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: 'Rooodles', loves: [ 'apple' ] },
  { name: 'Unicrom', loves: [ 'energon' ] }
]
```

Рисунок 17 — Результат выполнения запроса

1.10 Практическое задание 3.1.1

1.10.1 Создание коллекции towns

```
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: ["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"}
  }
])
```

Листинг 15 — Вставка документов в towns

```
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: ["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"}
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('684bfae1dc4dddfc31b5ffe'),
    '1': ObjectId('684bfae1dc4dddfc31b5fff'),
    '2': ObjectId('684bfae1dc4dddfc31b6000')
  }
}
```

Рисунок 18 — Результат выполнения запроса

1.10.2 Города с независимыми мэрами

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

Листинг 16 — Города с мэрами из партии «I»

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

Рисунок 19 — Результат выполнения запроса

1.10.3 Города без партийных мэров

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

Листинг 17 — Города без поля party в mayor

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

Рисунок 20 — Результат выполнения запроса

1.11 Практическое задание 3.1.2

1.11.1 Функция для фильтрации по полу

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

Листинг 18 — Определение функции фильтрации

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

Рисунок 21 — Результат выполнения определения функции

1.11.2 Курсор и вывод через forEach

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

Листинг 19 — Ограниченный курсор и перебор результатов

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

Рисунок 22 — Результат выполнения перебора

1.12 Практическое задание 3.2.1

1.12.1 Подсчёт самок 500–600 кг

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

Листинг 20 — Количество самок в заданном диапазоне веса

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

Рисунок 23 — Результат выполнения запроса

1.13 Практическое задание 3.2.2

1.13.1 Список всех предпочтений

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

Листинг 21 — Уникальные значения поля loves


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

Рисунок 24 — Результат выполнения запроса

1.14 Практическое задание 3.2.3

1.14.1 Агрегация по полу

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

Листинг 22 — Группировка и подсчёт по полу

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

Рисунок 25 — Результат выполнения агрегированного запроса

1.15 Практическое задание 3.3.1

1.15.1 Добавление единорога Barny через save

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

Листинг 23 — Использование метода save

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

learn> db.unicorns.insertOne({
...   name: 'Barny',
...   loves: ['grape'],
...   weight: 340,
...   gender: 'm'
... });
{
  acknowledged: true,
  insertedId: ObjectId('684bfd1bdc4dddfc31b6001')
}
learn> db.unicorns.find({name: 'Barny'})
[
  {
    _id: ObjectId('684bfd1bdc4dddfc31b6001'),
    name: 'Barny',
    loves: [ 'grape' ],
    weight: 340,
    gender: 'm'
  }
]
```

Рисунок 26 — Результат выполнения запроса

1.16 Практическое задание 3.3.2

1.16.1 Обновление веса и vampires у Ayna

```
db.unicorns.updateOne(  
  {name: "Ayna"},  
  {$set: {weight: 800, vampires: 51}}  
)
```

Листинг 24 — Обновление документа Ayna

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

Рисунок 27 — Результат выполнения запроса

1.17 Практическое задание 3.3.3

1.17.1 Обновление предпочтений у Raleigh

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

Листинг 25 — Обновление массива loves

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

Рисунок 28 — Результат обновления

1.18 Практическое задание 3.3.4

1.18.1 Увеличение vampires у всех самцов на 5

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

Листинг 26 — Инкремент поля vampires

```
learn> db.unicorns.updateMany(  
...   {gender: "m"},  
...   {$inc: {vampires: 5}}  
[...  ]  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 8,  
  modifiedCount: 8,  
  upsertedCount: 0  
}  
learn> db.unicorns.find()  
[  
  {  
    _id: ObjectId('684bf2e3dc4dddfc31b5ff2'),  
    name: 'Horny',  
    loves: [ 'carrot', 'papaya' ],  
    weight: 600,  
    gender: 'm',  
    vampires: 68  
  },  
  {  
    _id: ObjectId('684bf2e3dc4dddfc31b5ff3'),  
    name: 'Aurora',  
    loves: [ 'carrot', 'grape' ],  
    weight: 450,  
    gender: 'f',  
    vampires: 43  
  },  
  {  
    _id: ObjectId('684bf2e3dc4dddfc31b5ff4'),  
    name: 'Unicrom',  
    loves: [ 'energon', 'redbull' ],  
    weight: 984,  
    gender: 'm',  
    vampires: 187  
  },  
  {  
    _id: ObjectId('684bf2e3dc4dddfc31b5ff5'),  
    name: 'RoooooodLes',  
    loves: [ 'apple' ],  
    weight: 575,  
  }  
]
```

Рисунок 29 — Результат выполнения запроса

1.19 Практическое задание 3.3.5

1.19.1 Удаление партии мэра в Portland

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

Листинг 27 — Удаление поля party

```
[learn> db.towns.find({name: 'Portland'})  
[  
  {  
    _id: ObjectId('684bfae1dc4dddfc31b6000'),  
    name: 'Portland',  
    population: 528000,  
    last_sensus: ISODate('2009-07-20T00:00:00.000Z'),  
    famous_for: [ 'beer', 'food' ],  
    mayor: { name: 'Sam Adams', party: 'D' }  
  }  
]  
[learn> db.towns.updateOne(  
...   {name: "Portland"},  
...   {$unset: {"mayor.party": ""}}  
[...  )  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
[learn> db.towns.find({name: 'Portland'})  
[  
  {  
    _id: ObjectId('684bfae1dc4dddfc31b6000'),  
    name: 'Portland',  
    population: 528000,  
    last_sensus: ISODate('2009-07-20T00:00:00.000Z'),  
    famous_for: [ 'beer', 'food' ],  
    mayor: { name: 'Sam Adams' }  
  }  
]  
]
```

Рисунок 30 — Результат выполнения запроса

1.20 Практическое задание 3.3.6

1.20.1 Добавление chocolate в loves у Pilot

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

Листинг 28 — Добавление элемента в массив

```
learn> db.unicorns.find({name: "Pilot"})
[
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffb'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 658,
    gender: 'm',
    vampires: 59
  }
]
learn> db.unicorns.updateOne(
...   {name: "Pilot"},
...   {$push: {loves: "chocolate"}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.find({name: "Pilot"})
[
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffb'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 658,
    gender: 'm',
    vampires: 59
  }
]
1
```

Рисунок 31 — Результат выполнения запроса

1.21 Практическое задание 3.3.7

1.21.1 Добавление sugar и lemon в loves у Aurora

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

Листинг 29 — Добавление нескольких элементов уникально

```
learn> db.unicorns.find({name: "Aurora"})
[
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffb3'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
learn> db.unicorns.updateOne(
...   {name: "Aurora"},
...   {$addToSet: {loves: {$each: ["sugar", "lemon"]}}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.find({name: "Aurora"})
[
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffb3'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
1
```

Рисунок 32 — Результат выполнения запроса

1.22 Практическое задание 3.4.1

1.22.1 Создать коллекцию towns

```
db.towns.insertMany([
  {
    name: "Punxsutawney",
    popujation: 6200,
    last_sensus: ISODate("2008-01-31"),
    famous_for: ["phil the groundhog"],
    mayor: {
      name: "Jim Wehrle"
    }
  },
  {name: "New York",
    popujation: 22200000,
    last_sensus: ISODate("2009-07-31"),
    famous_for: ["status of liberty", "food"],
    mayor: {
      name: "Michael Bloomberg",
      party: "I"
    }
  },
  {
    name: "Portland",
    popujation: 528000,
    last_sensus: ISODate("2009-07-20"),
    famous_for: ["beer", "food"],
    mayor: {
      name: "Sam Adams",
      party: "D"
    }
  }
])

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('684c045adc4dddfc31b6002'),
    '1': ObjectId('684c045adc4dddfc31b6003'),
    '2': ObjectId('684c045adc4dddfc31b6004')
  }
}
```

Рисунок 33 — Результат выполнения запроса

1.22.2 Удаление беспартийных мэров

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

Листинг 31 — Удаление документов по условию

```
learn> db.towns.find({"mayor.party": {$exists: false}})
[
  {
    _id: ObjectId('684bfae1dc4dddfc31b5ffe'),
    name: 'Punxsutawney',
    population: 6200,
    last_sensus: ISODate('2008-01-31T00:00:00.000Z'),
    famous_for: [ 'phil the groundhog' ],
    mayor: { name: 'Jim Wehrle' }
  },
  {
    _id: ObjectId('684bfae1dc4dddfc31b6000'),
    name: 'Portland',
    population: 520000,
    last_sensus: ISODate('2009-07-20T00:00:00.000Z'),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams' }
  },
  {
    _id: ObjectId('684c045adc4dddfc31b6002'),
    name: 'Punxsutawney',
    population: 6200,
    last_sensus: ISODate('2008-01-31T00:00:00.000Z'),
    famous_for: [ 'phil the groundhog' ],
    mayor: { name: 'Jim Wehrle' }
  }
]
learn> db.towns.remove({"mayor.party": {$exists: false}})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 3 }
learn> db.towns.find({"mayor.party": {$exists: false}})
```

Рисунок 34 — Результат выполнения запроса

1.22.3 Очистка коллекции towns

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

Листинг 32 — Очистка коллекции

```
learn> db.towns.countDocuments()
3
learn> db.towns.deleteMany({})
{ acknowledged: true, deletedCount: 3 }
learn> db.towns.countDocuments()
0
```

Рисунок 35 — Результат выполнения запроса

1.22.4 Список доступных коллекций

```
show collections
```

Листинг 33 — Просмотр коллекций в базе

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

Рисунок 36 — Результат выполнения команды

1.23 Практическое задание 4.1.1

1.23.1 Создание коллекции habitats

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

Листинг 34 — Вставка документов в habitats

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

Рисунок 37 — Результат выполнения запроса

1.23.2 Ссылка на зону обитания для единорогов

```
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"}}})
```

Листинг 35 — Установка ссылок через DBRef


```

learn> db.unicorns.updateOne({name: "Horny"}, {$set: {habitat: {$ref: "habitats", $id: "forest"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({name: "Pilot"}, {$set: {habitat: {$ref: "habitats", $id: "mountains"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({name: "Nimue"}, {$set: {habitat: {$ref: "habitats", $id: "desert"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
learn> db.unicorns.find({ habitat: { $exists: true } })
[
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ff2'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 68,
    habitat: DBRef('habitats', 'forest')
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffb'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 650,
    gender: 'm',
    vampires: 59,
    habitat: DBRef('habitats', 'mountains')
  },
  {
    _id: ObjectId('684bf2e3dc4dddfc31b5ffc'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f',
    habitat: DBRef('habitats', 'desert')
  }
]

```

Рисунок 38 — Результат выполнения запросов

1.24 Практическое задание 4.2.1

1.24.1 Создание уникального индекса по name

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

Листинг 36 — Создание уникального индекса

```

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

```

Рисунок 39 — Результат выполнения команды

1.25 Практическое задание 4.3.1

1.25.1 Работа с индексами коллекции unicorns

```
db.unicorns.getIndexes()  
# Удалить индекс name_1  
db.unicorns.dropIndex("name_1")  
# Удалить индекс _id_  
db.unicorns.dropIndex("_id_")
```

Листинг 37 — Просмотр и удаление индексов

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

Рисунок 40 — Изначальные индексы

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

Рисунок 41 — После удаления индекса name_1

```
learn> db.unicorns.dropIndex("_id_")  
MongoServerError[InvalidOptions]: cannot drop _id index
```

Рисунок 42 — Результат попытки удалить _id_

1.26 Практическое задание 4.4.1

1.26.1 Генерация коллекции numbers

```
for (i = 0; i < 1000000; i++) {  
  db.numbers.insert({value: i})  
}
```

Листинг 38 — Заполнение коллекции большого объёма

```
learn> for (i = 0; i < 1000000; i++) {
...     db.numbers.insert({value: i})
... }
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('684c0d0ddc4dddfc32c28e4') }
}
```

Рисунок 43 — Результат выполнения цикла

1.26.2 Выбор последних четырёх документов без индекса

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

Листинг 39 — Поиск без индекса

```
learn> db.numbers.find().sort({$natural: -1}).limit(4)
[
  { _id: ObjectId('684c0d0ddc4dddfc32c28e4'), value: 999999 },
  { _id: ObjectId('684c0d0ddc4dddfc32c28e3'), value: 999998 },
  { _id: ObjectId('684c0d0ddc4dddfc32c28e2'), value: 999997 },
  { _id: ObjectId('684c0d0ddc4dddfc32c28e1'), value: 999996 }
]
learn> |
```

Рисунок 44 — Результат выполнения запроса

1.26.3 План выполнения без индекса

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

Листинг 40 — Explain без индекса

```

learn> db.numbers.explain("executionStats")
...   .find()
...   .sort({$natural: -1})
...   .limit(4)
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'learn.numbers',
    parsedQuery: {},
    indexFilterSet: false,
    queryHash: '8F2383EE',
    planCacheShapeHash: '8F2383EE',
    planCacheKey: '7DF350EE',
    optimizationTimeMillis: 0,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'LIMIT',
      limitAmount: 4,
      inputStage: { stage: 'COLLSCAN', direction: 'backward' }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 4,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 4,
    executionStages: {

```

Рисунок 45 — Результат explain: полный скан, до 1 мс

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

Значение `executionStats.executionTimeMillis` равно 0, что означает, что операция была выполнена очень быстро (менее 1 мс, округлено до 0).

1.26.4 Создание индекса по value

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

Листинг 41 — Создание индекса

```

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

```

Рисунок 46 — Результат выполнения команды

1.26.5 План выполнения с индексом

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

Листинг 42 — Explain с индексом

```
learn> db.numbers.explain("executionStats")
...   .find()
...   .sort({$natural: -1})
...   .limit(4)
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'learn.numbers',
    parsedQuery: {},
    indexFilterSet: false,
    queryHash: '8F2383EE',
    planCacheShapeHash: '8F2383EE',
    planCacheKey: '7DF350EE',
    optimizationTimeMillis: 0,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'LIMIT',
      limitAmount: 4,
      inputStage: { stage: 'COLLSCAN', direction: 'backward' }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 4,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 4,
    executionStages: {
```

Рисунок 47 — Результат explain: индексный скан, 0 мс

2 Вывод

В ходе выполнения практических заданий было:

1. Освоено создание, заполнение и удаление коллекций в MongoDB.
2. Отработано выполнение CRUD-операций: вставка (`insert`, `insertMany`), чтение (`find`, `findOne`, `distinct`), обновление (`updateOne`, `updateMany`, `save`) и удаление (`remove`, `deleteMany`).
3. Научено фильтровать, сортировать и ограничивать результаты с помощью `sort()`, `limit()` и `findOne()`.
4. Получен опыт работы со вложенными объектами и ссылками (`$ref`, `$id`) для организации связанных данных.
5. Изучено построение агрегированных запросов (`aggregate`, `$group`) и подсчёт статистик (`count()`, `distinct()`).
6. Овладение созданием и управлением индексами (`createIndex`, `getIndexes`, `dropIndex`), а также анализом планов выполнения запросов через `explain("executionStats")` для оценки эффективности.
7. Понято использование курсоров и JavaScript-функций в Mongo Shell (определение функций, перебор результатов через `forEach`).
8. Закреплены приёмы выборки последних добавленных документов, выборки по диапазону значений и управления полями с помощью операторов `$set`, `$unset`, `$push`, `$addToSet`, `$inc`.

Таким образом, приобретён полный набор инструментов для эффективной работы с данными в MongoDB и готовность применять их в реальных проектах.