

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

Проверила:
Говорова М.М.

Выполнил:
студент группы К32402
Балакирев Даниил Андреевич

Санкт-Петербург 2023

Практическое задание 8.1.1:

Создайте базу данных *learn*.

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

Заполните коллекцию единорогов *unicorns*:

```
db.unicorns.insert({name: 'Horny', loves: ['carrot', 'papaya'], weight: 600, gender:
'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender:
'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984,
gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm',
vampires: 99});

db.unicorns.insert({name: 'Solnara', loves: ['apple', 'carrot', 'chocolate'],
weight: 550, gender: 'f', vampires: 80});

db.unicorns.insert({name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733,
gender: 'f', vampires: 40});

db.unicorns.insert({name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender:
'm', vampires: 39});

db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender:
'm', vampires: 2});

db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601,
gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650,
gender: 'm', vampires: 54});

db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender:
'f'});
```

Используя второй способ, вставьте в коллекцию единорогов документ:

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

```
learn> db.unicorns.insertOne(document)
{
  acknowledged: true,
  insertedId: ObjectId("651cd49a9dffa31ebfa43421")
}
```

Проверьте содержимое коллекции с помощью метода *find*.

```
learn> db.unicorns.find({name: {$eq: 'Dunx'}})
[
  {
    _id: ObjectId("651cd49a9dffa31ebfa43421"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }
]
```

Практическое задание 8.1.2:

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

```
learn> db.unicorns.find({gender: 'm'}).sort({name:1}).limit(3)
[
  {
    _id: ObjectId("651cd49a9dffa31ebfa43421"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId("651cd2779dffa31ebfa43414"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("651cd29e9dffa31ebfa4341a"),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
]
```

```
learn> db.unicorns.find({gender: 'f'}).sort({name:1}).limit(3)
[
  {
    _id: ObjectId("651cd27e9dffa31ebfa43415"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("651cd2989dffa31ebfa43419"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId("651cd2ab9dffa31ebfa4341c"),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  }
]
learn>
```

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

```
learn> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
{
  _id: ObjectId("651cd27e9dffa31ebfa43415"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
learn>
```

Практическое задание 8.1.3:

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

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

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

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

Практическое задание 8.1.5:

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

```
learn> db.unicorns.find({}, {_id: false, 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
  },
]
```

Практическое задание 8.1.6:

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

```
learn> db.unicorns.find({ gender: 'f', weight: { $gte: 500, $lte: 700 } }, { _id: 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'
  }
]
learn>
```

Практическое задание 8.1.7:

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

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

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.


```
learn> db.unicorns.find({ vampires: { $exists: false } })
[
  {
    _id: ObjectId("651cd2b79dffa31ebfa4341e"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
learn>
```

Практическое задание 8.1.9:

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

```
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' ] }
]
learn> |
```

Практическое задание 8.2.1:

1. Создайте коллекцию towns, включающую следующие документы:

```
learn> db.towns.insert({name: "Punxsutawney ", populatiuon: 6200, last_sensus: ISODate("2008-01-31"), famous_for: [""], mayor: { name: "Jim Wehrle" }})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("651cdc9a9dffa31ebfa43425") }
}
learn> db.towns.insert({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,
  insertedIds: { '0': ObjectId("651cdcc29dffa31ebfa43426") }
}
learn> db.towns.insert({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("651cdcee9dffa31ebfa43427") }
}
learn>
```

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

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

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

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

Практическое задание 8.2.2:

1. Сформировать функцию для вывода списка самцов единорогов.

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
learn> fn = function() { return this.gender === "m"; }
NaN
```

```
learn> db.unicorns.find({ $where: fn })
[
  {
    _id: ObjectId("651ce11a9dffa31ebfa43428"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("651ce1249dffa31ebfa4342a"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId("651ce1299dffa31ebfa4342b"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
  }
]
```

Вывести результат, используя `forEach`.

```
learn> var cursor = db.unicorns.find({ $where: fn }).limit(2).sort({ name:1 });null;null
null
learn> cursor.forEach(function(obj) {print(obj.name)})
Dunx
Horny
```

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
learn> db.unicorns.find({ gender: "f" }).count()
5
learn> |
```

Практическое задание 8.2.4:

Вывести список предпочтений.

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

Практическое задание 8.2.5:

Посчитать количество особей единорогов обоих полов.

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

Практическое задание 8.2.6:

1. Выполнить команду:

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

```
learn> db.unicorns.insertOne({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})
{
  acknowledged: true,
  insertedId: ObjectId("651ce3a59dffa31ebfa43434")
}
learn> |
```

```
{
  _id: ObjectId("651ce14a9dffa31ebfa43432"),
  name: 'Nimue',
  loves: [ 'grape', 'carrot' ],
  weight: 540,
  gender: 'f'
},
{
  _id: ObjectId("651ce1779dffa31ebfa43433"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
},
{
  _id: ObjectId("651ce3a59dffa31ebfa43434"),
  name: 'Barney',
  loves: [ 'grape' ],
  weight: 340,
  gender: 'm'
}
learn>
```

Практическое задание 8.2.7:

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

Проверить содержимое коллекции unicorns.

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

```
learn> db.unicorns.find({name: "Ayna"})
[
  {
    _id: ObjectId("651ce1339dffa31ebfa4342d"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 800,
    gender: 'f',
    vampires: 51
  }
]
learn> |
```

Практическое задание 8.2.8:

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

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

```
learn> db.unicorns.find({name: "Raleigh"})
[
  {
    _id: ObjectId("651ce13c9dffa31ebfa4342f"),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar', 'redbull' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  }
]
learn> |
```

Практическое задание 8.2.9:

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

Проверить содержимое коллекции `unicorns`.

```
learn> db.unicorns.find({gender: "m"}).limit(3)
[
  {
    _id: ObjectId("651ce11a9dffa31ebfa43428"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 68
  },
  {
    _id: ObjectId("651ce1249dffa31ebfa4342a"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 187
  },
  {
    _id: ObjectId("651ce1299dffa31ebfa4342b"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 104
  }
]
```

Практическое задание 8.2.10:

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

Проверить содержимое коллекции `towns`.

```
learn> db.towns.updateOne({ name: "Portland" }, {$unset: { "mayor.party": 1 }})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.towns.find({name: "Portland"})
[
  {
    _id: ObjectId("651cdcee9dffa31ebfa43427"),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams' }
  }
]
learn>
```

Практическое задание 8.2.11:

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

Проверить содержимое коллекции `unicorns`.

```

learn> db.unicorns.find({name: "Pilot"})
[
  {
    _id: ObjectId("651ce1459dffa31ebfa43431"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
learn> db.unicorns.updateOne({name: "Pilot"}, { $push: { loves: "chocolate" } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn>

```

```

learn> db.unicorns.find({name: "Pilot"})
[
  {
    _id: ObjectId("651ce1459dffa31ebfa43431"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
learn> |

```

Практическое задание 8.2.12:

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

Проверить содержимое коллекции `unicorns`.

```
]
learn> db.unicorns.find({name: "Aurora"})
[
  {
    _id: ObjectId("651ce11f9dffa31ebfa43429"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
```

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

```
learn> db.unicorns.find({name: "Aurora"})
[
  {
    _id: ObjectId("651ce11f9dffa31ebfa43429"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemons' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
learn>
```

Практическое задание 8.2.13:

1. Создайте коллекцию *towns*, включающую следующие документы:

```

learn> db.towns.insert({name: "Punxsutawney ",
... popujatiuon: 6200,
... last_sensus: ISODate("2008-01-31"),
... famous_for: ["phil the groundhog"],
... mayor: {
...   name: "Jim Wehrle"
... }}
... )
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("651cea369dffa31ebfa43438") }
}
learn> db.towns.insert({name: "New York",
... popujatiuon: 22200000,
... last_sensus: ISODate("2009-07-31"),
... famous_for: ["status of liberty", "food"],
... mayor: {
...   name: "Michael Bloomberg",
...   party: "I"}}
... )
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("651cea429dffa31ebfa43439") }
}
learn> db.towns.insert({name: "Portland",
... popujatiuon: 528000,
... last_sensus: ISODate("2009-07-20"),
... famous_for: ["beer", "food"],
... mayor: {
...   name: "Sam Adams",
...   party: "D"}}
... )
{

```

Удалите документы с беспартийными мэрами.


```

learn> db.towns.deleteMany({"mayor.party": { $exists: false }})
{ acknowledged: true, deletedCount: 1 }
learn> db.towns.fins({})
TypeError: db.towns.fins is not a function
learn> db.towns.find({})
[
  {
    _id: ObjectId("651cea429dffa31ebfa43439"),
    name: 'New York',
    popujatiuon: 22200000,
    last_sensus: ISODate("2009-07-31T00:00:00.000Z"),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  },
  {
    _id: ObjectId("651cea4c9dffa31ebfa4343a"),
    name: 'Portland',
    popujatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]

```

```

learn> db.towns.deleteMany({})
{ acknowledged: true, deletedCount: 2 }
learn>

```

```

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

```

Практическое задание 8.3.1:

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

```

learn> db.regions.insert({ _id: "us-west-1", name: "California", description: "Here influence
r unicorns leave" })
{ acknowledged: true, insertedIds: { '0': 'us-west-1' } }
learn> |

```

```

learn> db.regions.find()
[
  {
    _id: 'us-west-1',
    name: 'California',
    description: 'Here influencer unicorns leave'
  }
]
learn> |

```

```

]
learn> db.regions.insert({ _id: "us-east-1", name: "Miami", description: "Here is the place o
f hurricane unicorns" })
{ acknowledged: true, insertedIds: { '0': 'us-east-1' } }
learn> db.regions.insert({ _id: "eu-central-1", name: "Frankfurt" , decription: " Beer unicor
s live here"})
Uncaught:
SyntaxError: Unexpected token , expected " " (1:47)
learn> db.regions.insert({ _id: "eu-central-1", name: "Frankfurt" , decription: " Beer unicor
ns live here"})
{ acknowledged: true, insertedIds: { '0': 'eu-central-1' } }
learn> |

```

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

```

learn> db.unicorns.updateOne({name: "Dunx"}, {$set: {region: {$ref: "regions", $id: "us-east-1"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({name: "Barney"}, {$set: {region: {$ref: "regions", $id: "us-west-1"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.updateOne({name: "Nimue"}, {$set: {region: {$ref: "regions", $id: "eu-central-1"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```

gender: 'm',
vampires: 59
},
{
  _id: ObjectId("651ce14a9dffa31ebfa43432"),
  name: 'Nimue',
  loves: [ 'grape', 'carrot' ],
  weight: 540,
  gender: 'f',
  region: DBRef("regions", 'eu-cantral-1')
},
{
  _id: ObjectId("651ce1779dffa31ebfa43433"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 170,
  region: DBRef("regions", 'us-east-1')
},
{
  _id: ObjectId("651ce3a59dffa31ebfa43434"),
  name: 'Barney',
  loves: [ 'grape' ],
  weight: 340,
  gender: 'm',
  vampires: 5,
  region: DBRef("regions", 'us-west-1')
}

```

Практическое

задание 8.3.2:

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

Содержание	коллекции	единорогов	<code>unicorns:</code>
<pre> learn> db.unicorns.ensureIndex({"name" : 1}, {"unique" : true}) ['name_1'] learn> </pre>			

Практическое задание 8.3.3:

1. Получите информацию о всех индексах коллекции `unicorns`.
2. Удалите все индексы, кроме индекса для идентификатора.
3. Попытайтесь удалить индекс для идентификатора.

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

```
learn> db.unicorns.dropIndex("name_1")
{ nIndexesWas: 2, ok: 1 }
learn>
learn> db.unicorns.dropIndex("_id_")
MongoServerError: cannot drop _id index
learn>
```