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

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 8
по теме: Работа с БД в СУБД MongoDB
по дисциплине: Проектирование и реализация баз данных

Специальность:
09.03.03 Мобильные и сетевые технологии

Проверила: Выполнил:
Говорова М.М. _____   студент группы K3240
Дата: «___» _____ 2021 г.
Оценка _____   Поляков Андрей

Санкт-Петербург 2021 г.

ЦЕЛЬ РАБОТЫ

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

ВЫПОЛНЕНИЕ

Практическое задание 8.1.1:

```
> document = ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})
{
        "name" : "Dunx",
        "loves" : [
                "grape",
                "watermelon"
        ],
        "weight" : 704,
        "gender" : "m",
        "vampires" : 165
}
> db.unicorns.insert(document)
WriteResult({ "nInserted" : 1 })
> db.unicorns.find({"name": "Dunx"})
{ "_id" : ObjectId("60bcc92350056ae513106be9"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires"
: 165 }
{ "_id" : ObjectId("60bcc9c850056ae513106bea"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires"
: 165 }
>
```

```
> db.unicorns.find()
{ "_id" : ObjectId("60bcc83750056ae513106bdd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" :
63 }
{ "_id" : ObjectId("60bcc83850056ae513106bde"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" :
43 }
{ "_id" : ObjectId("60bcc83850056ae513106bdf"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampire
s" : 182 }
{ "_id" : ObjectId("60bcc83850056ae513106be0"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60bcc83850056ae513106be1"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f"
, "vampires" : 80 }
{ "_id" : ObjectId("60bcc83850056ae513106be2"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires"
: 40 }
{ "_id" : ObjectId("60bcc83850056ae513106be3"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 3
9 }
{ "_id" : ObjectId("60bcc83850056ae513106be4"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" :
2 }
{ "_id" : ObjectId("60bcc83850056ae513106be5"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires"
: 33 }
{ "_id" : ObjectId("60bcc83850056ae513106be6"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires
" : 54 }
{ "_id" : ObjectId("60bcc83850056ae513106be7"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60bcc92350056ae513106be9"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires"
: 165 }
{ "_id" : ObjectId("60bcc9c850056ae513106bea"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires"
: 165 }
>
```

Практическое задание 8.1.2:.

```
> db.unicorns.find({"gender": "m"}, {"name": 1, "_id": 0}).sort({"name": 1})
{ "name" : "Dunx" }
{ "name" : "Dunx" }
{ "name" : "Horny" }
{ "name" : "Kenny" }
{ "name" : "Pilot" }
{ "name" : "Raleigh" }
{ "name" : "Roooooodles" }
{ "name" : "Unicrom" }
>
```

```
> db.unicorns.find({"gender": "f"}, {"name": 1, "_id": 0}).sort({"name": 1}).limit(3)
{ "name" : "Aurora" }
{ "name" : "Ayna" }
{ "name" : "Leia" }
>
```

```
> db.unicorns.findOne({"gender": "f", "loves": "carrot"}, {"name": 1, "_id": 0})
{ "name" : "Aurora" }
>
```

```
> db.unicorns.find({"gender": "f", "loves": "carrot"}, {"name": 1, "_id": 0}).limit(1)
{ "name" : "Aurora" }
>
```

Практическое задание 8.1.3:

```
> db.unicorns.find({"gender": "m"}, {"loves": 0})
{ "_id" : ObjectId("60bcc83750056ae513106bdd"), "name" : "Horny", "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60bcc83850056ae513106bdf"), "name" : "Unicrom", "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60bcc83850056ae513106be0"), "name" : "Rooooooodles", "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60bcc83850056ae513106be3"), "name" : "Kenny", "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60bcc83850056ae513106be4"), "name" : "Raleigh", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60bcc83850056ae513106be6"), "name" : "Pilot", "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60bccfea50056ae513106beb"), "name" : "Dunx", "weight" : 704, "gender" : "m", "vampires" : 165 }
>
```

Практическое задание 8.1.4:

```
> db.unicorns.find({}, {"name": 1, "_id": 0}).sort({$natural: -1})
{ "name" : "Dunx" }
{ "name" : "Nimue" }
{ "name" : "Pilot" }
{ "name" : "Leia" }
{ "name" : "Raleigh" }
{ "name" : "Kenny" }
{ "name" : "Ayna" }
{ "name" : "Solnara" }
{ "name" : "Rooooooodles" }
{ "name" : "Unicrom" }
{ "name" : "Aurora" }
{ "name" : "Horny" }
>
```

Практическое задание 8.1.5:

```
> db.unicorns.find({}, {"loves": {$slice: 1}, "_id": 0})
{ "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Aurora", "loves" : [ "carrot" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Solnara", "loves" : [ "apple" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "name" : "Ayna", "loves" : [ "strawberry" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "name" : "Leia", "loves" : [ "apple" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "name" : "Nimue", "loves" : [ "grape" ], "weight" : 540, "gender" : "f" }
{ "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>
```

Практическое задание 8.1.6:

```
> db.unicorns.find({"weight": {$gte: 500, $lte: 700}}, {"_id": 0})
{ "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Rooooooodles", "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" }
>
```

Практическое задание 8.1.7:

```
> db.unicorns.find({"gender": "m", "weight": {$gte: 500}, $or: [{"loves": "grape"}, {"loves": "lemon"}]}, {"_id": 0})
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>
```

Практическое задание 8.1.8:

```
> db.unicorns.find({"vampires": {$exists: false}}, {"_id": 0})
{ "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
>
```

Практическое задание 8.1.9:

```
> db.unicorns.find({"gender": "m"}, {"_id": 0, "name": 1, "loves": 1}).sort({"name": 1})
{ "name" : "Dunx", "loves" : [ "grape", "watermelon" ] }
{ "name" : "Horny", "loves" : [ "carrot", "papaya" ] }
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ] }
{ "name" : "Pilot", "loves" : [ "apple", "watermelon" ] }
{ "name" : "Raleigh", "loves" : [ "apple", "sugar" ] }
{ "name" : "Rooooooodles", "loves" : [ "apple" ] }
{ "name" : "Unicrom", "loves" : [ "energon", "redbull" ] }
>
```

Практическое задание 8.2.1:

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

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

Практическое задание 8.2.2:

```
> fn = function() {return this.gender=="m"; }
function() {return this.gender=="m"; }
> db.unicorns.find(fn)
{ "_id" : ObjectId("60bcc83750056ae513106bdd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" :
 63 }
{ "_id" : ObjectId("60bcc83850056ae513106bdf"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampire
s" : 182 }
{ "_id" : ObjectId("60bcc83850056ae513106be0"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60bcc83850056ae513106be3"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 3
9 }
{ "_id" : ObjectId("60bcc83850056ae513106be4"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" :
 2 }
{ "_id" : ObjectId("60bcc83850056ae513106be6"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires
" : 54 }
{ "_id" : ObjectId("60bccfea50056ae513106beb"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires"
 : 165 }
>
```

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

Практическое задание 8.2.3:

```
> db.unicorns.find({"gender": "f", "weight": {$lte: 600}}).count()
3
>
```

Практическое задание 8.2.4:

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

Практическое задание 8.2.5:

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

Практическое задание 8.2.6:

```
> db.unicorns.find({name: "Barny"})
> db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find({name: "Barny"})
{ "_id" : ObjectId("60bcef20d2c12a9f8359fe39"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

Практическое задание 8.2.7:

```
> db.unicorns.find({name: "Ayna"})
{ "_id" : ObjectId("60bcf090d2c12a9f8359fe3a"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires"
: 40 }
> db.unicorns.update({name: "Ayna"}, {name:'Ayna', loves: ['strawberry', 'lemon'], weight: 800, gender: 'f', vampires: 51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Ayna"})
{ "_id" : ObjectId("60bcf090d2c12a9f8359fe3a"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires"
: 51 }
>
```

Практическое задание 8.2.8

```
> db.unicorns.find({name: "Raleigh", gender: "m"})
{ "_id" : ObjectId("60bcc83850056ae513106be4"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" :
2 }
> db.unicorns.update({name: "Raleigh", gender: "m"}, {$set: {loves: ["redbull"]}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Raleigh", gender: "m"})
{ "_id" : ObjectId("60bcc83850056ae513106be4"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
>
```

Практическое задание 8.2.9:

```
> db.unicorns.find({gender: "m"}, {vampires: 1})
{ "_id" : ObjectId("60bcc83750056ae513106bdd"), "vampires" : 63 }
{ "_id" : ObjectId("60bcc83850056ae513106bdf"), "vampires" : 182 }
{ "_id" : ObjectId("60bcc83850056ae513106be0"), "vampires" : 99 }
{ "_id" : ObjectId("60bcc83850056ae513106be3"), "vampires" : 39 }
{ "_id" : ObjectId("60bcc83850056ae513106be4"), "vampires" : 2 }
{ "_id" : ObjectId("60bcc83850056ae513106be6"), "vampires" : 54 }
{ "_id" : ObjectId("60bccfea50056ae513106beb"), "vampires" : 165 }
{ "_id" : ObjectId("60bcef20d2c12a9f8359fe39") }
> db.unicorns.update({gender: "m"}, {$inc: {vampires: 5}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({gender: "m"}, {vampires: 1})
{ "_id" : ObjectId("60bcc83750056ae513106bdd"), "vampires" : 68 }
{ "_id" : ObjectId("60bcc83850056ae513106bdf"), "vampires" : 182 }
{ "_id" : ObjectId("60bcc83850056ae513106be0"), "vampires" : 99 }
{ "_id" : ObjectId("60bcc83850056ae513106be3"), "vampires" : 39 }
{ "_id" : ObjectId("60bcc83850056ae513106be4"), "vampires" : 2 }
{ "_id" : ObjectId("60bcc83850056ae513106be6"), "vampires" : 54 }
{ "_id" : ObjectId("60bccfea50056ae513106beb"), "vampires" : 165 }
{ "_id" : ObjectId("60bcef20d2c12a9f8359fe39") }
>
```

Практическое задание 8.2.10:

```
> db.towns.find({name: "Portland"})
{ "_id" : ObjectId("60bcdb88db81d6243c653a4c"), "name" : "Portland", "population" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "fa
mous_for" : [ "beer", "food" ], "mayor" : ( "name" : "Sam Adams", "party" : "D" ) }
> db.towns.update({name: "Portland"}, {$unset: {"mayor.party": 1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.towns.find({name: "Portland"})
{ "_id" : ObjectId("60bcdb88db81d6243c653a4c"), "name" : "Portland", "population" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "fa
mous_for" : [ "beer", "food" ], "mayor" : ( "name" : "Sam Adams" ) }
>
```

Практическое задание 8.2.11:

```
> db.unicorns.find({"name": "Pilot"})
{ "_id" : ObjectId("60bcc83850056ae513106be6"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires
" : 54 }
> db.unicorns.update({"name": "Pilot"}, {$push: {loves: "chocolate"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({"name": "Pilot"})
{ "_id" : ObjectId("60bcc83850056ae513106be6"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "
m", "vampires" : 54 }
>
```

Практическое задание 8.2.12:

```
> db.unicorns.find({name: "Aurora"})
{ "_id" : ObjectId("60bcc83850056ae513106bde"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" :
43 }
> db.unicorns.update({name: "Aurora"}, {$addToSet: {"loves": {$each: ["sugar", "lemon"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Aurora"})
{ "_id" : ObjectId("60bcc83850056ae513106bde"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" :
"f", "vampires" : 43 }
>
```

Практическое задание 8.2.13:

```
> db.towns.find()
{ "_id" : ObjectId("60bcfa43fce444937f9b28fe"), "name" : "Punxsutawney", "population" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "
famous_for" : [ "phil the groundhog" ], "mayor" : ( "name" : "Jim Wehrle" ) }
{ "_id" : ObjectId("60bcfa43fce444937f9b28ff"), "name" : "New York", "population" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "
famous_for" : [ "statue of liberty", "food" ], "mayor" : ( "name" : "Michael Bloomberg", "party" : "I" ) }
{ "_id" : ObjectId("60bcfa43fce444937f9b2900"), "name" : "Portland", "population" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "fa
mous_for" : [ "beer", "food" ], "mayor" : ( "name" : "Sam Adams", "party" : "D" ) }
> db.towns.remove({"mayor.party": {$exists: false}})
WriteResult({ "nRemoved" : 1 })
> db.towns.find()
{ "_id" : ObjectId("60bcfa43fce444937f9b28ff"), "name" : "New York", "population" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "
famous_for" : [ "statue of liberty", "food" ], "mayor" : ( "name" : "Michael Bloomberg", "party" : "I" ) }
{ "_id" : ObjectId("60bcfa43fce444937f9b2900"), "name" : "Portland", "population" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "fa
mous_for" : [ "beer", "food" ], "mayor" : ( "name" : "Sam Adams", "party" : "D" ) }
>
```

```
> db.towns.drop()
true
> show collections
unicorns
>
```

Практическое задание 8.3.1:

```
> db.zones.insertMany([
...    {id: "forest",
...     name: "forest",
...     description: "A forest with lots of trees"),
...
...    {id: "b_forest",
...     name: "birch forest",
...     description: "A forest with lots of birch trees"),
...
...    {id: "j_forest",
...     name: "jungle",
...     description: "A forest with lots of jungle trees"),
...
...    {id: "t_forest",
...     name: "taiga",
...     description: "A forest with lots of spruce trees"}
... ])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("60bd2289fce444937f9b2905"),
                ObjectId("60bd2289fce444937f9b2906"),
                ObjectId("60bd2289fce444937f9b2907"),
                ObjectId("60bd2289fce444937f9b2908")
        ]
}
> db.zones.find()
{ "_id" : ObjectId("60bd2289fce444937f9b2905"), "id" : "forest", "name" : "forest", "description" : "A forest with lots of trees" }
{ "_id" : ObjectId("60bd2289fce444937f9b2906"), "id" : "b_forest", "name" : "birch forest", "description" : "A forest with lots of birch trees"
}
{ "_id" : ObjectId("60bd2289fce444937f9b2907"), "id" : "j_forest", "name" : "jungle", "description" : "A forest with lots of jungle trees" }
{ "_id" : ObjectId("60bd2289fce444937f9b2908"), "id" : "t_forest", "name" : "taiga", "description" : "A forest with lots of spruce trees" }
>
```

```
> db.unicorns.update({"name": "Leia"}, {$set: {zone: {$ref: "zones", $id: "j_forest"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Leia"})
{ "_id" : ObjectId("60bcc83850056ae513106be5"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires"
: 33, "zone" : DBRef("zones", "j_forest") }
>
```

```
> db.unicorns.find()
{ "_id" : ObjectId("60bcc83750056ae513106bdd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" :
68, "zone" : DBRef("zones", "forest") }
{ "_id" : ObjectId("60bcc83850056ae513106bde"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" :
"f", "vampires" : 43, "zone" : DBRef("zones", "j_forest") }
{ "_id" : ObjectId("60bcc83850056ae513106bdf"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampire
s" : 182 }
{ "_id" : ObjectId("60bcc83850056ae513106be0"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60bcc83850056ae513106be1"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f"
, "vampires" : 80 }
{ "_id" : ObjectId("60bcc83850056ae513106be3"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 3
9 }
{ "_id" : ObjectId("60bcc83850056ae513106be4"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60bcc83850056ae513106be5"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires"
: 33 }
{ "_id" : ObjectId("60bcc83850056ae513106be6"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "
m", "vampires" : 54, "zone" : DBRef("zones", "j_forest") }
{ "_id" : ObjectId("60bcc83850056ae513106be7"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60bccfea50056ae513106beb"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires"
: 165, "zone" : DBRef("zones", "b_forest") }
{ "_id" : ObjectId("60bcef20d2c12a9f8359fe39"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
{ "_id" : ObjectId("60bcf090d2c12a9f8359fe3a"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires"
: 51, "zone" : DBRef("zones", "b_forest") }
>
```

Практическое задание 8.3.2:

```
> db.unicorns.ensureIndex({"name": 1}, {"unique" : true})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.unicorns.insert({name: "Pilot"})
WriteResult({
        "nInserted" : 0,
        "writeError" : {
                "code" : 11000,
                "errmsg" : "E11000 duplicate key error collection: learn.unicorns index: name_1 dup key: { name: \"Pilot\" }"
        }
})
>
```

Практическое задание 8.3.3:

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

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

```
> db.unicorns.dropIndex("_id_")
{
        "ok" : 0,
        "errmsg" : "cannot drop _id index",
        "code" : 72,
        "codeName" : "InvalidOptions"
}
>
```

Практическое задание 8.3.4:

```
> db.numbers.explain("executionStats").find().sort({value: -1}).limit(4)
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "learn.numbers",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "winningPlan" : {
                        "stage" : "SORT",
                        "sortPattern" : {
                                "value" : -1
                        },
                        "memLimit" : 104857600,
                        "limitAmount" : 4,
                        "type" : "simple",
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "direction" : "forward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 4,
                "executionTimeMillis" : 76,
                "totalKeysExamined" : 0,
                "totalDocsExamined" : 100000,
                "executionStages" : {
                        "stage" : "SORT",
                        "nReturned" : 4,
                        "executionTimeMillisEstimate" : 3,
```

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

9

```
> db.numbers.explain("executionStats").find().sort({value: -1}).limit(4)
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "learn.numbers",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "winningPlan" : {
                        "stage" : "LIMIT",
                        "limitAmount" : 4,
                        "inputStage" : {
                                "stage" : "FETCH",
                                "inputStage" : {
                                        "stage" : "IXSCAN",
                                        "keyPattern" : {
                                                "value" : 1
                                        },
                                        "indexName" : "value_1",
                                        "isMultiKey" : false,
                                        "multiKeyPaths" : {
                                                "value" : [ ]
                                        },
                                        "isUnique" : false,
                                        "isSparse" : false,
                                        "isPartial" : false,
                                        "indexVersion" : 2,
                                        "direction" : "backward",
                                        "indexBounds" : {
                                                "value" : [
                                                        "[MaxKey, MinKey]"
                                                ]
                                        }
                                }
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 4,
                "executionTimeMillis" : 0,
                "totalKeysExamined" : 4,
                "totalDocsExamined" : 4,
                "executionStages" : {
                        "stage" : "LIMIT",
                        "nReturned" : 4,
                        "executionTimeMillisEstimate" : 0,
```

ВЫВОДЫ

В ходе работы были рассмотрены стандартные механизмы работы с MongoDB, а также был сделан вывод, что MongoDB предоставляет собой сильный инструмент для выполнения CRUD операций в CLI.