Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Интеллектуальные системы в гуманитарной сфере**

Направление подготовки (специальность) **45.03.04 Интеллектуальные системы в гуманитарной сфере**

Лабораторная работа №5

по дисциплине «Проектирование и реализация баз данных»

тема: «Разработка интерфейса для выполнения CRUD-операций над базой данных средствами PHP»

Обучающийся: Шикалова Софья Сергеевна, K3242

Работа выполнена с оценкой _____

Преподаватель (и): Говорова М.М.

_____
(подпись)

Дата 30.05.2022

Санкт-Петербург 2022

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

Ход выполнения работы:

Задание 8.1.1:

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

```
> db.unicorns.insertOne({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}
... )
{
        "acknowledged" : true,
        "insertedId" : ObjectId("629c60f5583fb65f23db0ba7")
}
```

```
> db.unicorns.find()
{ "_id" : ObjectId("629c5e60101a6996580228dc"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629c5e60101a6996580228dd"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629c5e60101a6996580228de"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629c5e60101a6996580228df"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629c5e60101a6996580228e0"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629c5e60101a6996580228e1"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("629c5e60101a6996580228e2"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629c5e60101a6996580228e3"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629c5e60101a6996580228e4"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629c5e60101a6996580228e5"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629c5e60101a6996580228e6"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629c60f5583fb65f23db0ba7"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

Задание 8.1.2:

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

```
> db.unicorns.find({gender: 'm'}).sort({name: 1})
{ "_id" : ObjectId("629c60f5583fb65f23db0ba7"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629c5e60101a6996580228dc"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629c5e60101a6996580228e2"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629c5e60101a6996580228e5"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629c5e60101a6996580228e3"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629c5e60101a6996580228df"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629c5e60101a6996580228de"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
> db.unicorns.find({gender: 'f'}).sort({name: 1}).limit(3)
{ "_id" : ObjectId("629c5e60101a6996580228dd"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629c5e60101a6996580228e1"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("629c5e60101a6996580228e4"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
```

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

```
> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
{
        "_id" : ObjectId("629c5e60101a6996580228dd"),
        "name" : "Aurora",
        "loves" : [
                "carrot",
                "grape"
        ],
        "weight" : 450,
        "gender" : "f",
        "vampires" : 43
}
> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
{ "_id" : ObjectId("629c5e60101a6996580228dd"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

Задание 8.1.3:

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

```
> db.unicorns.find({gender: 'm'}, {gender:0, loves: 0}).sort({name: 1})
{ "_id" : ObjectId("629c60f5583fb65f23db0ba7"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("629c5e60101a6996580228dc"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("629c5e60101a6996580228e2"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("629c5e60101a6996580228e5"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("629c5e60101a6996580228e3"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("629c5e60101a6996580228df"), "name" : "Roooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("629c5e60101a6996580228de"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
```

Задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({ $natural: -1 })
{ "_id" : ObjectId("629c60f5583fb65f23db0ba7"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629c5e60101a6996580228e6"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629c5e60101a6996580228e5"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629c5e60101a6996580228e4"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629c5e60101a6996580228e3"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629c5e60101a6996580228e2"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629c5e60101a6996580228e1"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("629c5e60101a6996580228e0"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629c5e60101a6996580228df"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629c5e60101a6996580228de"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629c5e60101a6996580228dd"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629c5e60101a6996580228dc"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

Задание 8.1.5:

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

```
> 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" : "Roooooodles", "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 }
```

Задание 8.1.6:

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

```
> db.unicorns.find({gender: 'f', weight: {$gt: 500, $lt: 700}}, {_id: 0})
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

Задание 8.1.7:

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

```
> db.unicorns.find({gender: 'm', weight: {$gt: 500}, loves: {$all: ['grape', 'lemon']}}, {_id: 0})
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
```

Задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.

```
> db.unicorns.find({vampires: {$exists: false}})
{ "_id" : ObjectId("629c5e60101a6996580228e6"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

Задание 8.1.9:

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

```
> db.unicorns.find({gender: 'm'}, {loves: {$slice: 1}}).sort({name: 1})
{ "_id" : ObjectId("629c60f5583fb65f23db0ba7"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629c5e60101a6996580228dc"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629c5e60101a6996580228e2"), "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629c5e60101a6996580228e5"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629c5e60101a6996580228e3"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629c5e60101a6996580228df"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629c5e60101a6996580228de"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
```

Задание 8.2.1:

1. *Создайте коллекцию towns.*

```
> db.towns.insert({name: "Punxsutawney ", populatiuon: 6200,  last_sensus: ISODate("2008-01-31"),  famous_for: [""],  mayor: {    name: "Jim Wehrle"  }})
WriteResult({ "nInserted" : 1 })
> 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"}}
... )
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "Portland",
... populatiuon: 528000,
... last_sensus: ISODate("2009-07-20"),
... famous_for: ["beer", "food"],
... mayor: {
...     name: "Sam Adams",
... party: "D"}}
... )
WriteResult({ "nInserted" : 1 })
```

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

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

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

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

Задание 8.2.2:

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

```
> func = function() {return this.gender == 'm'}
function() {return this.gender == 'm'}
> db.unicorns.find(func)
{ "_id" : ObjectId("629c5e60101a6996580228dc"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629c5e60101a6996580228de"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629c5e60101a6996580228df"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629c5e60101a6996580228e2"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629c5e60101a6996580228e3"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629c5e60101a6996580228e5"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629c60f5583fb65f23db0ba7"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

2. *Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.*

```
> var curs = db.unicorns.find({gender: 'm'}); null;
null
> curs.sort({name: 1}).limit(2); null;
null
```

3. *Вывести результат, используя forEach.*

```
> curs.forEach(function(func) {print (func.name);})
Dunx
Horny
```

Задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

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

Задание 8.2.4:

Вывести список предпочтений.

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

Задание 8.2.5:

Посчитать количество особей единорогов обоих полов.

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

Задание 8.2.6:

1. *Выполнить команду.*

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

2. *Проверить содержимое коллекции unicorns.*

```
> db.unicorns.find()
{ "_id" : ObjectId("629c5e60101a6996580228dc"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629c5e60101a6996580228dd"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629c5e60101a6996580228de"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629c5e60101a6996580228df"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629c5e60101a6996580228e0"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629c5e60101a6996580228e1"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("629c5e60101a6996580228e2"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629c5e60101a6996580228e3"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629c5e60101a6996580228e4"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629c5e60101a6996580228e5"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629c5e60101a6996580228e6"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629c60f5583fb65f23db0ba7"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629c825d583fb65f23db0bab"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

Задание 8.2.7:

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

```
> db.unicorns.update({name: 'Ayna', gender: 'f'}, {name: 'Ayna', gender: 'f', weight: 800, vampires: 51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

2. *Проверить содержимое коллекции* `unicorns.`

```
> db.unicorns.find({name: 'Ayna'})
{ "_id" : ObjectId("629c5e60101a6996580228e1"), "name" : "Ayna", "gender" : "f", "weight" : 800, "vampires" : 51 }
```

Задание 8.2.8:

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

```
> db.unicorns.update({name: 'Raleigh', gender: 'm'}, {$set: {loves: 'redbull'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

2. *Проверить содержимое коллекции unicorns.*

```
> db.unicorns.find({name: 'Raleigh'})
{ "_id" : ObjectId("629c5e60101a6996580228e3"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421, "gender" : "m", "vampires" : 2 }
```

Задание 8.2.9:

1. *Всем самцам единорогов увеличить количество убитых вапмиров на 5.*

```
> db.unicorns.update({gender: 'm'}, {$inc: {vampires: 5}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

2. *Проверить содержимое коллекции unicorns.*

```
> db.unicorns.find({gender: 'm'})
{ "_id" : ObjectId("629c5e60101a6996580228dc"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("629c5e60101a6996580228de"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629c5e60101a6996580228df"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629c5e60101a6996580228e2"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629c5e60101a6996580228e3"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629c5e60101a6996580228e5"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629c60f5583fb65f23db0ba7"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629c825d583fb65f23db0bab"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

Задание 8.2.10:

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

```
> db.towns.update({name: 'Portland'}, {$unset: {'mayor.party': 1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

2. *Проверить содержимое коллекции* `towns`.

```
> db.towns.find()
{ "_id" : ObjectId("629c6ab4583fb65f23db0ba8"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for"
 : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("629c6ad5583fb65f23db0ba9"), "name" : "New York", "populatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for"
: [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("629c6afc583fb65f23db0baa"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" :
[ "beer", "food" ], "mayor" : { "name" : "Sam Adams" } }
```

Задание 8.2.11:

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

```
> db.unicorns.update({name: 'Pilot', gender: 'm'}, {$push: {loves: 'chocolate'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

2. *Проверить содержимое коллекции unicorns.*

```
> db.unicorns.find({name: 'Pilot'})
{ "_id" : ObjectId("629c5e60101a6996580228e5"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m",
 "vampires" : 54 }
```

Задание 8.2.12:

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

```
> db.unicorns.update({name: 'Aurora'}, {$addToSet: {loves: {$each: ['sugar', 'lemon']}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

2. *Проверить содержимое коллекции* `unicorns`.

```
{ "_id" : ObjectId("629c5e60101a6996580228dd"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" :
 "f", "vampires" : 43 }
```

Задание 8.2.13:

1. *Создайте коллекцию towns, включающую указанные документы.*

```
> db.towns.insert({name: "Punxsutawney ",
... popujatiuon: 6200,
... last_sensus: ISODate("2008-01-31"),
... famous_for: ["phil the groundhog"],
... mayor: {
...     name: "Jim Wehrle"
... }}
... )
WriteResult({ "nInserted" : 1 })
> 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"}}
... )
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "Portland",
... popujatiuon: 528000,
... last_sensus: ISODate("2009-07-20"),
... famous_for: ["beer", "food"],
... mayor: {
...     name: "Sam Adams",
... party: "D"}}
... )
WriteResult({ "nInserted" : 1 })
```

2. *Удалите документы с беспартийными мэрами.*

```
> db.towns.remove({'mayor.party': {$exists: false}})
WriteResult({ "nRemoved" : 3 })
```

3. *Проверьте содержание коллекции.*

```
> db.towns.find()
{ "_id" : ObjectId("629c6ad5583fb65f23db0ba9"), "name" : "New York", "populatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"),
"famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("629c8e70583fb65f23db0bad"), "name" : "New York", "popujatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"),
"famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("629c8e8f583fb65f23db0bae"), "name" : "Portland", "popujatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "f
amous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
```

4. *Очистите коллекцию.*

```
> db.towns.remove({})
WriteResult({ "nRemoved" : 3 })
```

5. *Просмотрите список доступных коллекций.*

```
> show collections
towns
unicorns
```

Задание 8.3.1:

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

```
> db.habitats.insert({_id: 'Fst', short: 'Fst', full: 'Forest', desc: 'Wild forest with normal temperature and low light level'})
WriteResult({ "nInserted" : 1 })
> db.habitats.insert({_id: 'Dst', short: 'Dst', full: 'Desert', desc: 'Dry desert with extremely high temperature and high light level'})
WriteResult({ "nInserted" : 1 })
> db.habitats.insert({_id: 'Mnt', short: 'Mnt', full: 'Mountains', desc: 'Very high mountains with low temperature and high light level'})
WriteResult({ "nInserted" : 1 })
> db.habitats.insert({_id: 'Spc', short: 'Spc', full: 'Space', desc: 'Infinite space with extremely low temperatures and low light level'})
WriteResult({ "nInserted" : 1 })
> db.habitats.insert({_id: 'Cld', short: 'Cld', full: 'Clouds', desc: 'Soft clouds with low temperature and high light level'})
WriteResult({ "nInserted" : 1 })
> db.habitats.find()
{ "_id" : "Fst", "short" : "Fst", "full" : "Forest", "desc" : "Wild forest with normal temperature and low light level" }
{ "_id" : "Dst", "short" : "Dst", "full" : "Desert", "desc" : "Dry desert with extremely high temperature and high light level" }
{ "_id" : "Mnt", "short" : "Mnt", "full" : "Mountains", "desc" : "Very high mountains with low temperature and high light level" }
{ "_id" : "Spc", "short" : "Spc", "full" : "Space", "desc" : "Infinite space with extremely low temperatures and low light level" }
{ "_id" : "Cld", "short" : "Cld", "full" : "Clouds", "desc" : "Soft clouds with low temperature and high light level" }
```

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

```
> db.unicorns.update({name: 'Horny'}, {$set: {'habitat': {$ref: 'habitats', $id: 'Mnt'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Aurora'}, {$set: {'habitat': {$ref: 'habitats', $id: 'Cld'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Unicrom'}, {$set: {'habitat': {$ref: 'habitats', $id: 'Spc'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Kenny'}, {$set: {'habitat': {$ref: 'habitats', $id: 'Fst'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Leia'}, {$set: {'habitat': {$ref: 'habitats', $id: 'Dst'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

3. *Проверьте содержание коллекции едиорогов.*

```
> db.unicorns.find()
{ "_id" : ObjectId("629c5e60101a6996580228dc"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m
", "vampires" : 68, "habitat" : DBRef("habitats", "Mnt") }
{ "_id" : ObjectId("629c5e60101a6996580228dd"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" :
450, "gender" : "f", "vampires" : 43, "habitat" : DBRef("habitats", "Cld") }
{ "_id" : ObjectId("629c5e60101a6996580228de"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender"
: "m", "vampires" : 182, "habitat" : DBRef("habitats", "Spc") }
{ "_id" : ObjectId("629c5e60101a6996580228df"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "v
ampires" : 99 }
{ "_id" : ObjectId("629c5e60101a6996580228e0"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550,
 "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629c5e60101a6996580228e1"), "name" : "Ayna", "gender" : "f", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("629c5e60101a6996580228e2"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m",
 "vampires" : 39, "habitat" : DBRef("habitats", "Fst") }
{ "_id" : ObjectId("629c5e60101a6996580228e3"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421, "gender" : "m", "vampire
s" : 2 }
{ "_id" : ObjectId("629c5e60101a6996580228e4"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" :
"f", "vampires" : 33, "habitat" : DBRef("habitats", "Dst") }
{ "_id" : ObjectId("629c5e60101a6996580228e5"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 65
0, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629c5e60101a6996580228e6"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f"
 }
{ "_id" : ObjectId("629c60f5583fb65f23db0ba7"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" :
"m", "vampires" : 165 }
{ "_id" : ObjectId("629c825d583fb65f23db0bab"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

Задание 8.3.2:

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

```
> db.unicorns.ensureIndex({'name': 1}, {'unique': true})
uncaught exception: TypeError: db.unicorns.ensureIndex is not a function :
@(shell):1:1
> db.unicorns.createIndex({'name': 1}, {'unique': true})
{
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "createdCollectionAutomatically" : false,
        "ok" : 1
}
```

Задание 8.3.3:

1. *Получите информацию о всех индексах коллекции unicorns.*

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

2. *Удалите все индексы, кроме индекса для идентификатора.*

3. *Попытайтесь удалить индекс для идентификатора.*

```
> db.unicorns.dropIndex('name_1')
{ "nIndexesWas" : 2, "ok" : 1 }
> db.unicorns.dropIndex('_id_')
{
        "ok" : 0,
        "errmsg" : "cannot drop _id index",
        "code" : 72,
        "codeName" : "InvalidOptions"
}
```

Задание 8.3.4:

1. *Создайте объемную коллекцию numbers, задействовав курсор:*
for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}

2. *Выберите последних четыре документа.*

```
> db.numbers.find().sort({$natural: -1}).limit(4)
{ "_id" : ObjectId("629e97ddc64eaf726b185037"), "value" : 99999 }
{ "_id" : ObjectId("629e97ddc64eaf726b185036"), "value" : 99998 }
{ "_id" : ObjectId("629e97ddc64eaf726b185035"), "value" : 99997 }
{ "_id" : ObjectId("629e97ddc64eaf726b185034"), "value" : 99996 }
```

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

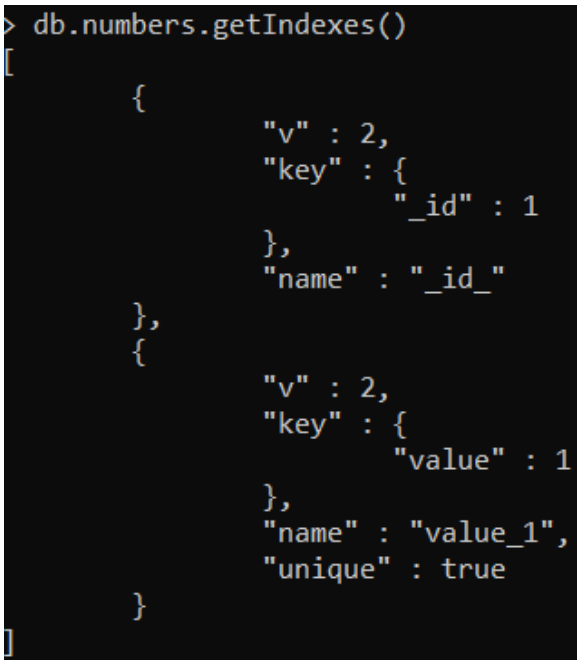
```
> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
WriteResult({ "nInserted" : 1 })
> db.numbers.explain('executionStats').find().sort({$natural: -1}).limit(4)
{
        "explainVersion" : "1",
        "queryPlanner" : {
                "namespace" : "learn.numbers",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "maxIndexedOrSolutionsReached" : false,
                "maxIndexedAndSolutionsReached" : false,
                "maxScansToExplodeReached" : false,
                "winningPlan" : {
                        "stage" : "LIMIT",
                        "limitAmount" : 4,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "direction" : "backward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 4,
                "executionTimeMillis" : 14,
                "totalKeysExamined" : 0,
                "totalDocsExamined" : 4,
                "executionStages" : {
                        "stage" : "LIMIT",
                        "nReturned" : 4,
                        "executionTimeMillisEstimate" : 0,
                        "works" : 6,
                        "advanced" : 4,
```

(14 миллисекунд)

4. *Создайте индекс для ключа value.*

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

5. *Получите информацию о всех индексах коллекции numbres.*



6. *Выполните запрос 2.*

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

(1 миллисекунда)

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

Очевидно создание индексов может значительно ускорить обработку данных, в моём примере время обработки проиндексированной коллекции меньше в 14 раз.

Вывод:

Овладела практическими навыками работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB.