

ЛАБОРАТОРНАЯ РАБОТА 5.2

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.

8.1 CRUD-ОПЕРАЦИИ В СУБД MONGODB. ВСТАВКА ДАННЫХ.

Практическое задание 8.1.1:

Заполните коллекцию единорогов unicorns:

```
> use learn
switched to db learn
> db.createCollection("unicorns")
{ "ok" : 1 }
> 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: 'Rooooooodles', 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 })
```

Используя второй способ, вставьте в коллекцию единорогов документ:

```
> 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.insertOne(document)
{
  "acknowledged" : true,
  "insertedId" : ObjectId("64767898e2d5f758ace3db93")
}
```

Проверьте содержимое коллекции с помощью метода find:

```
> db.unicorns.find()
{ "_id" : ObjectId("647676bd39ef05001120b95b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "v"
{ "_id" : ObjectId("647676bd39ef05001120b95c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "v"
{ "_id" : ObjectId("647676bd39ef05001120b95d"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m"
{ "_id" : ObjectId("647676bd39ef05001120b95e"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampir
{ "_id" : ObjectId("647676bd39ef05001120b95f"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gen
{ "_id" : ObjectId("647676bd39ef05001120b960"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f",
{ "_id" : ObjectId("647676bd39ef05001120b961"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vam
{ "_id" : ObjectId("647676bd39ef05001120b962"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "v
{ "_id" : ObjectId("647676bd39ef05001120b963"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f",
{ "_id" : ObjectId("647676bd39ef05001120b964"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m",
{ "_id" : ObjectId("647676c239ef05001120b965"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("64767898e2d5f758ace3db93"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m",
```

Практическое задание 8.1.2:

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("64767898e2d5f758ace3db93"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m",
{ "_id" : ObjectId("647676bd39ef05001120b95b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "v
{ "_id" : ObjectId("647676bd39ef05001120b961"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vam
{ "_id" : ObjectId("647676bd39ef05001120b964"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m",
{ "_id" : ObjectId("647676bd39ef05001120b962"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "v
{ "_id" : ObjectId("647676bd39ef05001120b95e"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampir
{ "_id" : ObjectId("647676bd39ef05001120b95d"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m"
> db.unicorns.find({ gender: "f" }).sort({ name: 1 }).limit(3)
{ "_id" : ObjectId("647676bd39ef05001120b95c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "v
{ "_id" : ObjectId("647676bd39ef05001120b960"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f",
{ "_id" : ObjectId("647676bd39ef05001120b963"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f",
```

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("647676bd39ef05001120b95c"),
  "name" : "Aurora",
  "loves" : [
    "carrot",
    "grape"
  ],
  "weight" : 450,
  "gender" : "f",
  "vampires" : 43
}
```

Практическое задание 8.1.3:

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

```
> db.unicorns.find({ gender: "m" }, { gender: false, loves: false })
{ "_id" : ObjectId("647676bd39ef05001120b95b"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("647676bd39ef05001120b95d"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
{ "_id" : ObjectId("647676bd39ef05001120b95e"), "name" : "Rooooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("647676bd39ef05001120b961"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("647676bd39ef05001120b962"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("647676bd39ef05001120b964"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("64767898e2d5f758ace3db93"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
```

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления:

```
> db.unicorns.find({}, {_id: false, name: true}).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:

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

```
> 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" : "Rooodoodles", "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:

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

```
> db.unicorns.find({ gender: "f", weight: { $gte: 500, $lt: 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" }
```

Практическое 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: { $gte: 500 }, loves: { $all: ["grape", "lemon"] } }, { _id: false })
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
```

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires:

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

Практическое задание 8.1.9:

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

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

8.2 ЗАПРОСЫ К БАЗЕ ДАННЫХ MONGODB. ВЫБОРКА ДАННЫХ. ВЛОЖЕННЫЕ ОБЪЕКТЫ. ИСПОЛЬЗОВАНИЕ КУРСОРОВ. АГРЕГИРОВАННЫЕ ЗАПРОСЫ. ИЗМЕНЕНИЕ ДАННЫХ

Практическое задание 8.2.1:

Создайте коллекцию towns, включающую следующие документы:

```
> db.createCollection("towns")
{ "ok" : 1 }
> db.towns.insertMany([
... {
...   name: "Punxsutawney ",
...   populatiuon: 6200,
...   last_sensus: ISODate("2008-01-31"),
...   famous_for: [""],
...   mayor: {
...     name: "Jim Wehrle",
...   },
... },
... ],
```

```

... {
...   name: "New York",
...   populatiuon: 22200000,
...   last_sensus: ISODate("2009-07-31"),
...   famous_for: ["status of liberty", "food"],
...   mayor: {
...     name: "Michael Bloomberg",
...     party: "I",
...   },
... },
... {
...   name: "Portland",
...   populatiuon: 528000,
...   last_sensus: ISODate("2009-07-20"),
...   famous_for: ["beer", "food"],
...   mayor: {
...     name: "Sam Adams",
...     party: "D",
...   },
... },
... ])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("6476838b877904b82a8a02ab"),
    ObjectId("6476838b877904b82a8a02ac"),
    ObjectId("6476838b877904b82a8a02ad")
  ]
}

```

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" }, { _id: false, mayor: true, name: true })
{ "name" : "New York", "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }

```

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

```

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

```

Практическое задание 8.2.2:

Сформировать функцию для вывода списка самцов единорогов:

```

> function findMaleUnicorns() { return this.gender === "m" }

```

Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке:

```

> var cursor = db.unicorns.find().sort({ name: 1 }).limit(2); null;
null

```

Вывести результат, используя forEach:

```

> cursor.forEach(function(doc){print(doc.name)})
Dunx
Horny

```

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг:

```

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

```

Практическое задание 8.2.4:

Вывести список предпочтений:

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

Практическое задание 8.2.5:

Посчитать количество особей единорогов обоих полов:

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

Практическое задание 8.2.6:

Выполнить команду:

```
> db.unicorns.save({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find({}, { _id: false })
{ "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "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" }
{ "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

Практическое задание 8.2.7:

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

```
> db.unicorns.updateOne({ name: "Ayna" }, { $set: { weight: 800, vampires: 51 } })
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.unicorns.find({}, { _id: false })
{ "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "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" }
{ "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

Практическое задание 8.2.8:

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

```
> db.unicorns.find({ name: "Raleigh" }, { _id: false })
{ "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
> db.unicorns.updateOne({ name: "Raleigh" }, { $push: { loves: "RedBull" } })
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.unicorns.find({ name: "Raleigh" }, { _id: false })
{ "name" : "Raleigh", "loves" : [ "apple", "sugar", "RedBull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
```

Практическое задание 8.2.9:

Всем самцам единорогов увеличить количество убитых вапмиров на 5:

```
> db.unicorns.find({ gender: "m" }, { _id: false })
{ "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "name" : "Rooodoodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Raleigh", "loves" : [ "apple", "sugar", "RedBull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
> db.unicorns.updateMany({ gender: "m" }, { $inc: { vampires: 5 } })
{ "acknowledged" : true, "matchedCount" : 8, "modifiedCount" : 8 }
> db.unicorns.find({ gender: "m" }, { _id: false })
{ "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "name" : "Rooodoodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "name" : "Raleigh", "loves" : [ "apple", "sugar", "RedBull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

Практическое задание 8.2.10:

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

```
> db.towns.find({}, { _id: false })
{ "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : {
{ "name" : "New York", "populatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty"
{ "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "ma
> db.towns.updateOne({ name: "Portland" }, { $unset: { "mayor.party": true } })
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.towns.find({}, { _id: false })
{ "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : {
{ "name" : "New York", "populatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty"
{ "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "ma
```

Практическое задание 8.2.11:

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

```
> db.unicorns.find({ name: "Pilot" }, { _id: false })
{ "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
> db.unicorns.updateOne({ name: "Pilot" }, { $push: { loves: "chocolate" } })
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.unicorns.find({ name: "Pilot" }, { _id: false })
{ "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
```

Практическое задание 8.2.12:

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

```
> db.unicorns.find({ name: "Aurora" }, { _id: false })
{ "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
> db.unicorns.updateOne({ name: "Aurora" }, { $push: { loves: { $each: ["sugar", "lemon"] } } })
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.unicorns.find({ name: "Aurora" }, { _id: false })
{ "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

Практическое задание 8.2.13:

Удалите документы с беспартийными мэрами:

```
> db.towns.find({}, { _id: false })
{ "name" : "Punxsutawney ", "popujatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "phil the groundho"
{ "name" : "New York", "popujatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty"
{ "name" : "Portland", "popujatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "ma
> db.towns.deleteMany({ "mayor.party": { $exists: false } })
{ "acknowledged" : true, "deletedCount" : 1 }
> db.towns.find({}, { _id: false })
{ "name" : "New York", "popujatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty"
{ "name" : "Portland", "popujatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "ma
> db.towns.drop()
true
> db.getCollectionNames()
[ "unicorns" ]
```

ССЫЛКИ И РАБОТА С ИНДЕКСАМИ В БАЗЕ ДАННЫХ MONGODB

Практическое задание 8.3.1:

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

```
> db.zones.find()
{ "_id" : "1", "name" : "zone_1" }
{ "_id" : "2", "name" : "zone_2" }
```

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

```
> db.unicorns.updateOne({ name: "Dunx" }, { $set: { zone: { $ref: "zones", $id: "1" } } })
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.unicorns.find({}, { _id: false })
{ "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "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" }
{ "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165, "zone" : DBRef("zones", "1")
{ "name" : "Rooodoodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
```

Практическое задание 8.3.2:

Проверьте, можно ли задать для коллекции unicorns индекс для ключа name с флагом unique:

```
> db.unicorns.ensureIndex({ name: 1 }, { unique: true })
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

Практическое задание 8.3.3:

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"
  }
]
```

```
{
  "v" : 2,
  "unique" : true,
  "key" : {
    "name" : 1
  },
  "name" : "name_1"
}
```

Удалите все индексы, кроме индекса для идентификатора:

```
> db.unicorns.dropIndex("name_1")
{ "nIndexesWas" : 2, "ok" : 1 }
```

Попытайтесь удалить индекс для идентификатора:

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

Практическое задание 8.3.4:

Создайте объемную коллекцию numbers, задействовав курсор:

```
> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
WriteResult({ "nInserted" : 1 })
```

Выберите последних четыре документа.

```
> db.numbers.find().sort({ $natural: -1 }).limit(4)
{ "_id" : ObjectId("6477c3d6cf6ae5a961fbd5f2"), "value" : 99999 }
{ "_id" : ObjectId("6477c3d6cf6ae5a961fbd5f1"), "value" : 99998 }
{ "_id" : ObjectId("6477c3d6cf6ae5a961fbd5f0"), "value" : 99997 }
{ "_id" : ObjectId("6477c3d6cf6ae5a961fbd5ef"), "value" : 99996 }
```

Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса:

```
> db.numbers.explain("executionStats").find({ value: 40000 })
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "learn.numbers",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "value" : {
        "$eq" : 40000
      }
    },
    "winningPlan" : {
      "stage" : "COLLSCAN",
      "filter" : {
        "value" : {
          "$eq" : 40000
        }
      },
      "direction" : "forward"
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 1,
    "executionTimeMillis" : 61,
    "totalKeysExamined" : 0,
    "totalDocsExamined" : 100000,
    "executionStages" : {
      "stage" : "COLLSCAN",
```



```

    "filter" : {
      "value" : {
        "$eq" : 40000
      }
    },
    "nReturned" : 1,
    "executionTimeMillisEstimate" : 6,
    "works" : 100002,
    "advanced" : 1,
    "needTime" : 100000,
    "needYield" : 0,
    "saveState" : 100,
    "restoreState" : 100,
    "isEOF" : 1,
    "direction" : "forward",
    "docsExamined" : 100000
  }
},
"serverInfo" : {
  "host" : "MiWiFi-R4AC-srv",
  "port" : 27017,
  "version" : "4.4.4",
  "gitVersion" : "8db30a63db1a9d84bdcad0c83369623f708e0397"
},
"ok" : 1
}

```

Создайте индекс для ключа value:

```

> 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"
  }
]

```

Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса:

```

> db.numbers.explain("executionStats").find({ value: 40000 })
{
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 1,
    "executionTimeMillis" : 7,
    "totalKeysExamined" : 1,
    "totalDocsExamined" : 1,
    "executionStages" : {
      "stage" : "FETCH",
      "nReturned" : 1,
      "executionTimeMillisEstimate" : 0,
      "works" : 2,
      "advanced" : 1,
      "needTime" : 0,
      "needYield" : 0,
      "saveState" : 0,
      "restoreState" : 0,
      "isEOF" : 1,
      "docsExamined" : 1,
      "alreadyHasObj" : 0,
      "inputStage" : {
        "stage" : "IXSCAN",
        "nReturned" : 1,

```

```

        "executionTimeMillisEstimate" : 0,
        "works" : 2,
        "advanced" : 1,
        "needTime" : 0,
        "needYield" : 0,
        "saveState" : 0,
        "restoreState" : 0,
        "isEOF" : 1,
        "keyPattern" : {
            "value" : 1
        },
        "indexName" : "value_1",
        "isMultiKey" : false,
        "multiKeyPaths" : {
            "value" : [ ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
            "value" : [
                "[40000.0, 40000.0]"
            ]
        },
        "keysExamined" : 1,
        "seeks" : 1,
        "dupsTested" : 0,
        "dupsDropped" : 0
    }
},
"serverInfo" : {
    "host" : "MiWiFi-R4AC-srv",
    "port" : 27017,
    "version" : "4.4.4",
    "gitVersion" : "8db30a63db1a9d84bdcad0c83369623f708e0397"
},
"ok" : 1
}

```

ЛАБОРАТОРНАЯ РАБОТА 5.2