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

|  |  |
| --- | --- |
| **Проверил**:  Говорова М.М. \_\_\_\_\_\_\_\_\_\_  **Дата**: «\_\_» \_\_\_\_ 2021 г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Выполнил**:  студент группы K3241  Каратецкая Мария |

**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.

**Практические задания**

**Практическое задание 8.1.1:**

1. *Создайте базу данных learn.*
2. *Заполните коллекцию единорогов unicorns:*

> use unicorns

switched to db unicorns

> 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', '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: [ 'grade', '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: [ 'grade', 'carrot'], weight: 540, gender: 'f'});

WriteResult({ "nInserted" : 1 })

*3) 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.insert(document)

WriteResult({ "nInserted" : 1 })

*4) Проверьте содержимое коллекции с помощью метода find.*

> db.unicorns.find()

{ "\_id" : ObjectId("60bd49d2dd3ed4fda3dd6056"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "\_id" : ObjectId("60bd4a02b4a6bf68c9e1a624"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60bd4a3fb4a6bf68c9e1a625"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60bd4aa0b4a6bf68c9e1a626"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60bd4ad4b4a6bf68c9e1a627"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }

{ "\_id" : ObjectId("60bd4af3b4a6bf68c9e1a628"), "name" : "Kenny", "loves" : [ "grade", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }

{ "\_id" : ObjectId("60bd4b19b4a6bf68c9e1a629"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60bd4b3eb4a6bf68c9e1a62a"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60bd4b5cb4a6bf68c9e1a62b"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60bd4b82b4a6bf68c9e1a62c"), "name" : "Nimue", "loves" : [ "grade", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60bd4c22b4a6bf68c9e1a62d"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

**Практическое задание 8.1.2:**

1. *Сформируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.*
2. *Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций findOne и limit.*

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

{ "\_id" : ObjectId("60bd49d2dd3ed4fda3dd6056"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "\_id" : ObjectId("60bd4ad4b4a6bf68c9e1a627"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }

{ "\_id" : ObjectId("60bd4b3eb4a6bf68c9e1a62a"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

> db.unicorns.findOne({loves: 'carrot', gender: 'f'})

{

"\_id" : ObjectId("60bd49d2dd3ed4fda3dd6056"),

"name" : "Aurora",

"loves" : [

"carrot",

"grape"

],

"weight" : 450,

"gender" : "f",

"vampires" : 43

}  
  
  
> db.unicorns.find({loves: 'carrot', gender: 'f'}).limit(1)

{ "\_id" : ObjectId("60bd49d2dd3ed4fda3dd6056"), "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 предпотениях и поле.*

> db.unicorns.find({gender:'m'}, {loves:0});

{ "\_id" : ObjectId("60bd499bea403ef6b2d14422"), "name" : "Horny", "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60be3133a0990b3ce62127d6"), "name" : "Unicrom", "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60be313ea0990b3ce62127d7"), "name" : "Roooooodles", "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60be315da0990b3ce62127da"), "name" : "Kenny", "weight" : 690, "gender" : "m", "vampires" : 39 }

{ "\_id" : ObjectId("60be3167a0990b3ce62127db"), "name" : "Raleigh", "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60be317ba0990b3ce62127dd"), "name" : "Pilot", "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60be3199a0990b3ce62127df"), "name" : "Dunx", "weight" : 704, "gender" : "m", "vampires" : 165 }

**Практическое задание 8.1.4:**

*Вывести список единорогов в обратном порядке добавления*

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

{ "\_id" : ObjectId("60be3199a0990b3ce62127df"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

{ "\_id" : ObjectId("60be3185a0990b3ce62127de"), "name" : "Nimue", "loves" : [ "grade", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60be317ba0990b3ce62127dd"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60be3172a0990b3ce62127dc"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60be3167a0990b3ce62127db"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60be315da0990b3ce62127da"), "name" : "Kenny", "loves" : [ "grade", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }

{ "\_id" : ObjectId("60be314fa0990b3ce62127d9"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }

{ "\_id" : ObjectId("60be3148a0990b3ce62127d8"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60be313ea0990b3ce62127d7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60be3133a0990b3ce62127d6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60bd499bea403ef6b2d14422"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

**Практическое задание 8.1.5:**

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

> db.unicorns.find({}, {loves:{$slice:-1}, \_id:false});

{ "name" : "Horny", "loves" : [ "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "name" : "Unicrom", "loves" : [ "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "name" : "Solnara", "loves" : [ "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "name" : "Ayna", "loves" : [ "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }

{ "name" : "Kenny", "loves" : [ "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }

{ "name" : "Raleigh", "loves" : [ "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "name" : "Leia", "loves" : [ "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "name" : "Pilot", "loves" : [ "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "name" : "Nimue", "loves" : [ "carrot" ], "weight" : 540, "gender" : "f" }

{ "name" : "Dunx", "loves" : [ "watermelon" ], "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"},{\_id:false}, {weight:{$gt:500, $lt:700}, "\_id" : false});

{ "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }

{ "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "name" : "Nimue", "loves" : [ "grade", "carrot" ], "weight" : 540, "gender" : "f" }

> db.unicorns.find({gender: "f"},{\_id:false}, {weight:{$gt:500, $lt:700}});

{ "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }

{ "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "name" : "Nimue", "loves" : [ "grade", "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( {loves:{$all:['lemon', 'grade']}}, {\_id:false}, {weight:{$gt:500}}, {gender:'m'});

{ "name" : "Kenny", "loves" : [ "grade", "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" : ObjectId("60be3185a0990b3ce62127de"), "name" : "Nimue", "loves" : [ "grade", "carrot" ], "weight" : 540, "gender" : "f" }

**Практическое 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("60be3199a0990b3ce62127df"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

{ "\_id" : ObjectId("60bd499bea403ef6b2d14422"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60be315da0990b3ce62127da"), "name" : "Kenny", "loves" : [ "grade" ], "weight" : 690, "gender" : "m", "vampires" : 39 }

{ "\_id" : ObjectId("60be317ba0990b3ce62127dd"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60be3167a0990b3ce62127db"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60be313ea0990b3ce62127d7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60be3133a0990b3ce62127d6"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

**Практическое задание 8.2.1:**

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

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 })  
  
db.towns.find({'mayor.party':'I'}, {'name':1, 'mayor':1} )

{ "\_id" : ObjectId("60be3f32a0990b3ce62127e1"), "name" : "New York", "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }  
  
db.towns.find({'mayor.party':{$exists:false}}, {'name':1, 'mayor':1} )

{ "\_id" : ObjectId("60be3f1ba0990b3ce62127e0"), "name" : "Punxsutawney ", "mayor" : { "name" : "Jim Wehrle" } }

**Практическое задание 8.2.2:**

1. *Сформировать функцию для вывода списка самцов единорогов.*
2. *Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.*
3. *Вывести результат, используя forEach.*

*Содержание коллекции единорогов unicorns:*> fn = function(){return this.gender == 'm'}

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

> db.unicorns.find(fn)

{ "\_id" : ObjectId("60bd499bea403ef6b2d14422"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60be3133a0990b3ce62127d6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60be313ea0990b3ce62127d7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60be315da0990b3ce62127da"), "name" : "Kenny", "loves" : [ "grade", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }

{ "\_id" : ObjectId("60be3167a0990b3ce62127db"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60be317ba0990b3ce62127dd"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60be3199a0990b3ce62127df"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

> var cursor = db.unicorns.find(fn)

> var cursor = db.unicorns.find(fn)

> cursor.sort({name:1}).limit(2)

{ "\_id" : ObjectId("60be3199a0990b3ce62127df"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

{ "\_id" : ObjectId("60bd499bea403ef6b2d14422"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

**Практическое задание 8.2.3:**

*Вывести количество самок единорогов весом от полутонны до 600 кг.*

> db.unicorns.find({gender: "f"},{\_id:false}, {weight:{$gt:500, $lt:700}, "\_id" : false}).count();

4

**Практическое задание 8.2.4:**

*Вывести список предпочтений.*

> db.unicorns.distinct('loves')

[

"apple",

"carrot",

"chocolate",

"energon",

"grade",

"grape",

"lemon",

"papaya",

"redbull",

"strawberry",

"sugar",

"watermelon"

]

**Практическое задание 8.2.5:**

*Посчитать количество особей единорогов обоих полов.*> db.unicorns.aggregate({$group:{\_id:'$gender', total:{$sum:1}}})

{ "\_id" : "f", "total" : 4 }

{ "\_id" : "m", "total" : 7 }

**Практическое задание 8.2.6:**

1. *Выполнить команду:*

> db.unicorns.save({name: 'Barny', loves: ['grape'],

weight: 340, gender: 'm'})

*Проверить содержимое коллекции unicorns.*db.unicorns.save({name: 'Barny', loves: ['grape'],

... weight: 340, gender: 'm'})

WriteResult({ "nInserted" : 1 })

> db.unicorns.find();

{ "\_id" : ObjectId("60bd499bea403ef6b2d14422"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60be3133a0990b3ce62127d6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60be313ea0990b3ce62127d7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60be3148a0990b3ce62127d8"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60be314fa0990b3ce62127d9"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }

{ "\_id" : ObjectId("60be315da0990b3ce62127da"), "name" : "Kenny", "loves" : [ "grade", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }

{ "\_id" : ObjectId("60be3167a0990b3ce62127db"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60be3172a0990b3ce62127dc"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60be317ba0990b3ce62127dd"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60be3185a0990b3ce62127de"), "name" : "Nimue", "loves" : [ "grade", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60be3199a0990b3ce62127df"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

{ "\_id" : ObjectId("60be4917a0990b3ce62127e3"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }

{ "\_id" : ObjectId("60be4918a0990b3ce62127e4"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }

**Практическое задание 8.2.7:**

1. *Для самки единорога* Ayna *внести изменения в БД: теперь ее вес 800, она убила 51 вапмира.*
2. *Проверить содержимое коллекции unicorns.*

db.unicorns.update({name:'Ayna'}, {name:'Ayna', weight:800, vampires:51})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.unicorns.find({name:'Ayna'});

{ "\_id" : ObjectId("60be314fa0990b3ce62127d9"), "name" : "Ayna", "weight" : 800, "vampires" : 51 }

**Практическое задание 8.2.8:**

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

*Проверить содержимое коллекции unicorns.*db.unicorns.update({name:'Raleigh', gender:'m'}, {name:'Raleigh', gender:'m', loves:'RedBull'})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.unicorns.find({name:'Raleigh'});

{ "\_id" : ObjectId("60be3167a0990b3ce62127db"), "name" : "Raleigh", "gender" : "m", "loves" : "RedBull" }

**Практическое задание 8.2.9:**

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

*Проверить содержимое коллекции unicorns.*> db.unicorns.update({ gender:'m'}, {$inc:{vampires:5}}, {multi:true})

WriteResult({ "nMatched" : 9, "nUpserted" : 0, "nModified" : 9 })

> db.unicorns.find({gender:'m'});

{ "\_id" : ObjectId("60bd499bea403ef6b2d14422"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 73 }

{ "\_id" : ObjectId("60be3133a0990b3ce62127d6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }

{ "\_id" : ObjectId("60be313ea0990b3ce62127d7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }

{ "\_id" : ObjectId("60be315da0990b3ce62127da"), "name" : "Kenny", "loves" : [ "grade", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }

{ "\_id" : ObjectId("60be3167a0990b3ce62127db"), "name" : "Raleigh", "gender" : "m", "loves" : "RedBull", "vampires" : 5 }

{ "\_id" : ObjectId("60be317ba0990b3ce62127dd"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }

{ "\_id" : ObjectId("60be3199a0990b3ce62127df"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }

{ "\_id" : ObjectId("60be4917a0990b3ce62127e3"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }

{ "\_id" : ObjectId("60be4918a0990b3ce62127e4"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }

**Практическое задание 8.2.10:**

1. *Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.*
2. *Проверить содержимое коллекции towns.*

> db.towns.update({name:'Portland'}, {$unset:{'mayor.party':1}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.towns.find({name:'Portland'})

{ "\_id" : ObjectId("60be3f45a0990b3ce62127e2"), "name" : "Portland", "populatiuon" : 528000, "last\_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous\_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams" } }

>

**Практическое задание 8.2.11:**

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

*Проверить содержимое коллекции unicorns.*> db.unicorns.update({ name:'Pilot'}, {$push:{loves:'chocolate'}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.unicorns.find({ name:'Pilot'})

{ "\_id" : ObjectId("60be317ba0990b3ce62127dd"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }

>

**Практическое задание 8.2.12:**

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

*Проверить содержимое коллекции unicorns.*db.unicorns.update({ name:'Aurora'}, {$addToSet: {loves: {$each:['sugar', 'lemon']}}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.unicorns.find({ name:'Aurora'})

{ "\_id" : ObjectId("60be4d6fa0990b3ce62127e5"), "name" : "Aurora", "dob" : ISODate("1991-01-24T10:00:00Z"), "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

>

**Практическое задание 8.2.13:**

1. *Создайте коллекцию towns, включающую следующие документы:*
2. *Удалите документы с беспартийными мэрами.*
3. *Проверьте содержание коллекции.*
4. *Очистите коллекцию.*
5. *Просмотрите список доступных коллекций.*

... 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 })

> db.towns.remove({'mayor.party':{$exists:false}})

WriteResult({ "nRemoved" : 3 })

> db.towns.find()

{ "\_id" : ObjectId("60be3f32a0990b3ce62127e1"), "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("60be4dfba0990b3ce62127e7"), "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("60be4e0ba0990b3ce62127e8"), "name" : "Portland", "popujatiuon" : 528000, "last\_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous\_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }

>

db.towns.remove({})

WriteResult({ "nRemoved" : 3 })

> db.towns.find()

>

**Практическое задание 8.3.1:**

1. *Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.*
2. *Включите для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания.*
3. *Проверьте содержание коллекции едиорогов.*

*Содержание коллекции единорогов unicorns:* > db.unicorns.update({ name:'Aurora'}, {$set:{zone:{$ref:'zones', $id:'fr'}}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.unicorns.update({ name:'Unicrom'}, {$set:{zone:{$ref:'zones', $id:'pr'}}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.unicorns.update({ name:'Ayna'}, {$set:{zone:{$ref:'zones', $id:'ds'}}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.unicorns.find()

{ "\_id" : ObjectId("60bd499bea403ef6b2d14422"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 73 }

{ "\_id" : ObjectId("60be3133a0990b3ce62127d6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187, "zone" : DBRef("zones", "pr") }

{ "\_id" : ObjectId("60be313ea0990b3ce62127d7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }

{ "\_id" : ObjectId("60be3148a0990b3ce62127d8"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60be314fa0990b3ce62127d9"), "name" : "Ayna", "weight" : 800, "vampires" : 51, "zone" : DBRef("zones", "ds") }

{ "\_id" : ObjectId("60be315da0990b3ce62127da"), "name" : "Kenny", "loves" : [ "grade", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }

{ "\_id" : ObjectId("60be3167a0990b3ce62127db"), "name" : "Raleigh", "gender" : "m", "loves" : "RedBull", "vampires" : 5 }

{ "\_id" : ObjectId("60be3172a0990b3ce62127dc"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60be317ba0990b3ce62127dd"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }

{ "\_id" : ObjectId("60be3185a0990b3ce62127de"), "name" : "Nimue", "loves" : [ "grade", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60be3199a0990b3ce62127df"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }

{ "\_id" : ObjectId("60be4917a0990b3ce62127e3"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }

{ "\_id" : ObjectId("60be4918a0990b3ce62127e4"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }

{ "\_id" : ObjectId("60be4d6fa0990b3ce62127e5"), "name" : "Aurora", "dob" : ISODate("1991-01-24T10:00:00Z"), "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "zone" : DBRef("zones", "fr") }

**Практическое задание 8.3.2:**

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

> db.unicorns.ensureIndex({'name':1}, {'unique':true})

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 1,

"numIndexesAfter" : 2,

"ok" : 1

}

> db.unicorns.find()

{ "\_id" : ObjectId("60bd499bea403ef6b2d14422"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 73 }

{ "\_id" : ObjectId("60be3133a0990b3ce62127d6"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187, "zone" : DBRef("zones", "pr") }

{ "\_id" : ObjectId("60be313ea0990b3ce62127d7"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }

{ "\_id" : ObjectId("60be3148a0990b3ce62127d8"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60be314fa0990b3ce62127d9"), "name" : "Ayna", "weight" : 800, "vampires" : 51, "zone" : DBRef("zones", "ds") }

{ "\_id" : ObjectId("60be315da0990b3ce62127da"), "name" : "Kenny", "loves" : [ "grade", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }

{ "\_id" : ObjectId("60be3167a0990b3ce62127db"), "name" : "Raleigh", "gender" : "m", "loves" : "RedBull", "vampires" : 5 }

{ "\_id" : ObjectId("60be3172a0990b3ce62127dc"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60be317ba0990b3ce62127dd"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }

{ "\_id" : ObjectId("60be3185a0990b3ce62127de"), "name" : "Nimue", "loves" : [ "grade", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60be3199a0990b3ce62127df"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }

{ "\_id" : ObjectId("60be4d6fa0990b3ce62127e5"), "name" : "Aurora", "dob" : ISODate("1991-01-24T10:00:00Z"), "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "zone" : DBRef("zones", "fr") }

**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.*

> db.unicorns.getIndexes()

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_"

},

{

"v" : 2,

"unique" : true,

"key" : {

"name" : 1

},

"name" : "name\_1"

}

]

>

db.unicorns.dropIndexes('name\_1')

{ "nIndexesWas" : 2, "ok" : 1 }

> db.unicorns.dropIndexes('\_id\_') – попытка удаление индекса

uncaught exception: Error: error dropping indexes : {

"ok" : 0,

"errmsg" : "cannot drop \_id index",

"code" : 72,

"codeName" : "InvalidOptions"

**Практическое задание 8.3.4:**

1. *Создайте объемную коллекцию numbers, задействовав курсор:*

for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}

1. *Выберите последних четыре документа.*
2. *Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)*
3. *Создайте индекс для ключа value.*
4. *Получите информацию о всех индексах коллекции numbres.*
5. *Выполните запрос 2.*
6. *Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?*
7. *Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?*

> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}

WriteResult({ "nInserted" : 1 })

> db.numbers.getIndexes()

[ { "v" : 2, "key" : { "\_id" : 1 }, "name" : "\_id\_" } ]

> db.numbers.find({value:{$in:[99999, 99998, 99997, 99996]}})

{ "\_id" : ObjectId("60be557fa0990b3ce622ae85"), "value" : 99996 }

{ "\_id" : ObjectId("60be557fa0990b3ce622ae86"), "value" : 99997 }

{ "\_id" : ObjectId("60be557fa0990b3ce622ae87"), "value" : 99998 }

{ "\_id" : ObjectId("60be557fa0990b3ce622ae88"), "value" : 99999 }

> db.numbers.explain('executionStats').find({executionTimeMillis:1})

{

"queryPlanner" : {

"plannerVersion" : 1,

"namespace" : "learn.numbers",

"indexFilterSet" : false,

"parsedQuery" : {

"executionTimeMillis" : {

"$eq" : 1

}

},

"winningPlan" : {

"stage" : "COLLSCAN",

"filter" : {

"executionTimeMillis" : {

"$eq" : 1

}

},

"direction" : "forward"

},

"rejectedPlans" : [ ]

},

"executionStats" : {

"executionSuccess" : true,

"nReturned" : 0,

"executionTimeMillis" : 58,

"totalKeysExamined" : 0,

"totalDocsExamined" : 100000,

"executionStages" : {

"stage" : "COLLSCAN",

"filter" : {

"executionTimeMillis" : {

"$eq" : 1

}

},

"nReturned" : 0,

"executionTimeMillisEstimate" : 0,

"works" : 100002,

"advanced" : 0,

"needTime" : 100001,

"needYield" : 0,

"saveState" : 100,

"restoreState" : 100,

"isEOF" : 1,

"direction" : "forward",

"docsExamined" : 100000

}

},

"serverInfo" : {

"host" : "User-PC",

"port" : 27017,

"version" : "4.4.6",

"gitVersion" : "72e66213c2c3eab37d9358d5e78ad7f5c1d0d0d7"

},

"ok" : 1

}

> db.numbers.ensureIndex({'value':1}, {'unique':true})

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 1,

"numIndexesAfter" : 2,

"ok" : 1

}

> db.numbers.getIndexes()

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_"

},

{

"v" : 2,

"unique" : true,

"key" : {

"value" : 1

},

"name" : "value\_1"

}

]

> db.numbers.find({value:{$in:[99999, 99998, 99997, 99996]}})

{ "\_id" : ObjectId("60be557fa0990b3ce622ae85"), "value" : 99996 }

{ "\_id" : ObjectId("60be557fa0990b3ce622ae86"), "value" : 99997 }

{ "\_id" : ObjectId("60be557fa0990b3ce622ae87"), "value" : 99998 }

{ "\_id" : ObjectId("60be557fa0990b3ce622ae88"), "value" : 99999 }

> db.numbers.explain('executionStats').find({executionTimeMillis:1})

{

"queryPlanner" : {

"plannerVersion" : 1,

"namespace" : "learn.numbers",

"indexFilterSet" : false,

"parsedQuery" : {

"executionTimeMillis" : {

"$eq" : 1

}

},

"winningPlan" : {

"stage" : "COLLSCAN",

"filter" : {

"executionTimeMillis" : {

"$eq" : 1

}

},

"direction" : "forward"

},

"rejectedPlans" : [ ]

},

"executionStats" : {

"executionSuccess" : true,

"nReturned" : 0,

"executionTimeMillis" : 57,

"totalKeysExamined" : 0,

"totalDocsExamined" : 100000,

"executionStages" : {

"stage" : "COLLSCAN",

"filter" : {

"executionTimeMillis" : {

"$eq" : 1

}

},

"nReturned" : 0,

"executionTimeMillisEstimate" : 3,

"works" : 100002,

"advanced" : 0,

"needTime" : 100001,

"needYield" : 0,

"saveState" : 100,

"restoreState" : 100,

"isEOF" : 1,

"direction" : "forward",

"docsExamined" : 100000

}

},

"serverInfo" : {

"host" : "User-PC",

"port" : 27017,

"version" : "4.4.6",

"gitVersion" : "72e66213c2c3eab37d9358d5e78ad7f5c1d0d0d7"

},

"ok" : 1

}

C индексированием запрос был чуть быстрее

**ВЫВОДЫ**

MongoDB предоставляет мощный CLI интерфейс для выполнения CRUD операций, отличительной особенностью является интеграция полноценного языка программирования: Javascript.