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: «09» июня 2021г.   
Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:   
студент группы К3240

Костылев Иван

Санкт-Петербург 2021 г

**Цель работы**

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

**Практическое задание**

* + 1. ***Практическое задание***

1. ***Создать базу данных learn***
2. ***Добавить коллекцию единорогов в unicorn***

> use learn

switched to db learn

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

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 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: 'Rooooooooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

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

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 })

1. ***Добавить единорога в коллекцию вторым способом***

> doc = ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165});

> db.unicorns.insert(doc);

WriteResult({ "nInserted" : 1 })

1. ***Проверить содержание таблицы***

> db.unicorns.find()

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60c0be45e3a6baa5eed061ff"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60c0beaae3a6baa5eed06201"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60c0bf3de3a6baa5eed06203"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60c0c328e3a6baa5eed06206"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60c0c430e3a6baa5eed06207"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

* + 1. ***Выборка данных из БД***

*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("60c0c430e3a6baa5eed06207"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

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

{ "\_id" : ObjectId("60c0beaae3a6baa5eed06201"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "\_id" : ObjectId("60c0bf3de3a6baa5eed06203"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60c0c328e3a6baa5eed06206"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }

*2) Найдите всех самок, которые любят carrot. Ограничьте этот список первой*

*особью с помощью функций findOne и limit.*

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

{ "\_id" : ObjectId("60c0beaae3a6baa5eed06201"), "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"}, {loves: 0})

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60c0c430e3a6baa5eed06207"), "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("60c0c430e3a6baa5eed06207"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

{ "\_id" : ObjectId("60c0c328e3a6baa5eed06206"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60c0bf3de3a6baa5eed06203"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60c0beaae3a6baa5eed06201"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60c0be45e3a6baa5eed061ff"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

**Практическое задание 8.1.5**

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

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

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60c0be45e3a6baa5eed061ff"), "name" : "Solnara", "loves" : [ "apple" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60c0beaae3a6baa5eed06201"), "name" : "Aurora", "loves" : [ "carrot" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60c0bf3de3a6baa5eed06203"), "name" : "Leia", "loves" : [ "apple" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60c0c328e3a6baa5eed06206"), "name" : "Nimue", "loves" : [ "grape" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60c0c430e3a6baa5eed06207"), "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"}, {\_id: false}, {weight: {$lt: 700, $gt: 500}});

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

{ "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "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", loves: {$all: ["grape", "lemon"]}}, {\_id: false}, {weight: {$gt: 500}});

**Практическое задание 8.1.8:**

*Найти всех единорогов, не имеющих ключ vampires.*

> db.unicorns.find({vampires: {$exists: false}});

{ "\_id" : ObjectId("60c0c328e3a6baa5eed06206"), "name" : "Nimue", "loves" : [ "grape", "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({vampires: {$exists: false}});

{ "\_id" : ObjectId("60c0c328e3a6baa5eed06206"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }

> db.unicorns.find({gender: "m"}, {loves: {$slice: 1}}).sort({name: 1});

{ "\_id" : ObjectId("60c0c430e3a6baa5eed06207"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

**Практическое задание 8.2.1:**

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

{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"}}

1. *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, name:true, mayor: true});

{ "name" : "New York", "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }

1. *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, name:true});

{ "name" : "Punxsutawney " }

**Практическое задание 8.2.2:**

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

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

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

> db.unicorns.find(fn);

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60c0c430e3a6baa5eed06207"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

1. *Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.*
2. *Вывести результат, используя forEach.*

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

> cursor.forEach(function(obj) {

... print(obj.name);

... });

Dunx

Horny

**Практическое задание 8.2.3:**

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

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

4

**Практическое задание 8.2.4:**

*Вывести список предпочтений*

> db.unicorns.distinct("loves");

[

"apple",

"carrot",

"chocolate",

"energon",

"grape",

"papaya",

"redbull",

"sugar",

"watermelon"

]

**Практическое задание 8.2.5:**

*Посчитать количество особей единорогов обоих полов.*

> db.unicorns.aggregate({$group:{\_id:'$gender', total:{$sum:1}}})

{ "\_id" : "m", "total" : 6 }

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

**Практическое задание 8.2.6:**

1. *Выполнить команду:*

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

weight: 340, gender: 'm'})

1. *Проверить содержимое коллекции unicorns.*

> db.unicrons.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'});

WriteResult({ "nInserted" : 1 })

**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'}, { $set: {weight: 800, vampires: 51}});

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

> db.unicorns.find()

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60c0be45e3a6baa5eed061ff"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60c0beaae3a6baa5eed06201"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60c0bf3de3a6baa5eed06203"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60c0c328e3a6baa5eed06206"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60c0c430e3a6baa5eed06207"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

{ "\_id" : ObjectId("60c0e767e3a6baa5eed0620c"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }

**Практическое задание 8.2.8:**

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

> db.unicorns.update({name: "Raleigh", gender: 'm'}, {$set: {loves: ['redbull']}});

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

> db.unicorns.find()

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60c0be45e3a6baa5eed061ff"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

{ "\_id" : ObjectId("60c0beaae3a6baa5eed06201"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }

{ "\_id" : ObjectId("60c0bf3de3a6baa5eed06203"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60c0c328e3a6baa5eed06206"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60c0c430e3a6baa5eed06207"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

{ "\_id" : ObjectId("60c0e767e3a6baa5eed0620c"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }

**Практическое задание 8.2.9:**

1. *Всем самцам единорогов увеличить количество убитых вапмиров на 5.*
2. *Проверить содержимое коллекции unicorns.*

> db.unicorns.update({gender: "m"}, {$inc: {vampires: 5}}, {multi: true});

WriteResult({ "nMatched" : 6, "nUpserted" : 0, "nModified" : 6 })

> db.unicorns.find()

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 192 }

{ "\_id" : ObjectId("60c0be45e3a6baa5eed061ff"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }

{ "\_id" : ObjectId("60c0beaae3a6baa5eed06201"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }

{ "\_id" : ObjectId("60c0bf3de3a6baa5eed06203"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }

{ "\_id" : ObjectId("60c0c328e3a6baa5eed06206"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60c0c430e3a6baa5eed06207"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }

{ "\_id" : ObjectId("60c0e767e3a6baa5eed0620c"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }

**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()

{ "\_id" : ObjectId("60c0d528e3a6baa5eed06208"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last\_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous\_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }

{ "\_id" : ObjectId("60c0d538e3a6baa5eed06209"), "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("60c0d54be3a6baa5eed0620a"), "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.*
2. *Проверить содержимое коллекции unicorns.*

> db.unicorns.update({name: "Pilot", gender: "m"}, {$push: {loves: "chocolate"}});

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

> db.unicorns.find();

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 192 }

{ "\_id" : ObjectId("60c0be45e3a6baa5eed061ff"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }

{ "\_id" : ObjectId("60c0beaae3a6baa5eed06201"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }

{ "\_id" : ObjectId("60c0bf3de3a6baa5eed06203"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }

{ "\_id" : ObjectId("60c0c328e3a6baa5eed06206"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60c0c430e3a6baa5eed06207"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }

{ "\_id" : ObjectId("60c0e767e3a6baa5eed0620c"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }

**Практическое задание 8.2.12:**

1. *Изменить информацию о самке единорога* Aurora: *теперь она любит еще и сахар, и лимоны.*
2. *Проверить содержимое коллекции unicorns.*

> db.unicorns.update({name: "Aurora"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}});

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

> db.unicorns.find();

{ "\_id" : ObjectId("60c0bdcfe3a6baa5eed061fe"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 192 }

{ "\_id" : ObjectId("60c0be45e3a6baa5eed061ff"), "name" : "Solnara", "loves" : [ "apple", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60c0be7ee3a6baa5eed06200"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }

{ "\_id" : ObjectId("60c0beaae3a6baa5eed06201"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }

{ "\_id" : ObjectId("60c0beede3a6baa5eed06202"), "name" : "Rooooooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }

{ "\_id" : ObjectId("60c0bf3de3a6baa5eed06203"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60c0bf75e3a6baa5eed06204"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }

{ "\_id" : ObjectId("60c0c308e3a6baa5eed06205"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }

{ "\_id" : ObjectId("60c0c328e3a6baa5eed06206"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60c0c430e3a6baa5eed06207"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }

{ "\_id" : ObjectId("60c0e767e3a6baa5eed0620c"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }

**Практическое задание 8.2.13:**

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

{name: "Punxsutawney ",

popujatiuon: 6200,

last\_sensus: ISODate("2008-01-31"),

famous\_for: ["phil the groundhog"],

mayor: {

name: "Jim Wehrle"

}}

{name: "New York",

popujatiuon: 22200000,

last\_sensus: ISODate("2009-07-31"),

famous\_for: ["status of liberty", "food"],

mayor: {

name: "Michael Bloomberg",

party: "I"}}

{name: "Portland",

popujatiuon: 528000,

last\_sensus: ISODate("2009-07-20"),

famous\_for: ["beer", "food"],

mayor: {

name: "Sam Adams",

party: "D"}}

1. *Удалите документы с беспартийными мэрами.*
2. *Проверьте содержание коллекции.*

> db.towns.find();

{ "\_id" : ObjectId("60c0ed3de3a6baa5eed0620e"), "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("60c0ed4ce3a6baa5eed0620f"), "name" : "Portland", "popujatiuon" : 528000, "last\_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous\_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }

1. *Очистите коллекцию.*
2. *Просмотрите список доступных коллекций.*

> db.towns.remove({});

WriteResult({ "nRemoved" : 2 })

> show collections;

towns

**Практическое задание 8.3.1:**

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

> db.areas.insert({\_id: "rz", name: "real zone", description: "real life zone"});

WriteResult({ "nInserted" : 1 })

> db.areas.insert({\_id: "mz", name: "magic zone", description: "unreal life zone"});

WriteResult({ "nInserted" : 1 })

> db.unicorns.update({name: 'Horny'}, {$set: {area: {$ref: 'areas', $id: 'rz'}}});

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

> db.unicorns.update({name: 'Aurora'}, {$set: {area: {$ref: 'areas', $id: 'mz'}}});

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

> db.unicorns.update({name: 'Dunx'}, {$set: {area: {$ref: 'areas', $id: 'mz'}}});

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

> db.unicorns.find();

{ "\_id" : ObjectId("60c0eec3e3a6baa5eed06210"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63, "area" : DBRef("areas", "rz") }

{ "\_id" : ObjectId("60c0eec3e3a6baa5eed06211"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43, "area" : DBRef("areas", "mz") }

{ "\_id" : ObjectId("60c0eec3e3a6baa5eed06212"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }

{ "\_id" : ObjectId("60c0eec3e3a6baa5eed06213"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "\_id" : ObjectId("60c0eec3e3a6baa5eed06214"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }

{ "\_id" : ObjectId("60c0eec3e3a6baa5eed06215"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }

{ "\_id" : ObjectId("60c0eec3e3a6baa5eed06216"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }

{ "\_id" : ObjectId("60c0eec3e3a6baa5eed06217"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }

{ "\_id" : ObjectId("60c0eec3e3a6baa5eed06218"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }

{ "\_id" : ObjectId("60c0eec3e3a6baa5eed06219"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }

{ "\_id" : ObjectId("60c0eed6e3a6baa5eed0621a"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165, "area" : DBRef("areas", "mz") }

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

**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.insert({name: 'Horny', dob: new Date(1992,2,13,7,47), loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', dob: new Date(1991, 0, 24, 13, 0), loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', dob: new Date(1973, 1, 9, 22, 10), loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', dob: new Date(1979, 7, 18, 18, 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', dob: new Date(1985, 6, 4, 2, 1), loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', dob: new Date(1998, 2, 7, 8, 30), loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', dob: new Date(1997, 6, 1, 10, 42), loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

db.unicorns.insert({name: 'Raleigh', dob: new Date(2005, 4, 3, 0, 57), loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', dob: new Date(2001, 9, 8, 14, 53), loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', dob: new Date(1997, 2, 1, 5, 3), loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insert ({name: 'Nimue', dob: new Date(1999, 11, 20, 16, 15), loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

db.unicorns.insert {name: 'Dunx', dob: new Date(1976, 6, 18, 18, 18), loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165

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

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 1,

"numIndexesAfter" : 2,

"ok" : 1

}

**Практическое задание 8.3.3:**

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

> db.unicorns.getIndexes();

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_"

},

{

"v" : 2,

"unique" : true,

"key" : {

"name" : 1

},

"name" : "name\_1"

}

]

1. *Удалите все индексы, кроме индекса для идентификатора.*
2. *Попытайтесь удалить индекс для идентификатора.*

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

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

> db.unicorns.dropIndexes('\_id\_');

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})}

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

WriteResult({ "nInserted" : 1 })

1. *Выберите последних четыре документа.*

> db.numbers.find().skip(99996);

{ "\_id" : ObjectId("60c0f517e3a6baa5eed1e8c3"), "value" : 99996 }

{ "\_id" : ObjectId("60c0f517e3a6baa5eed1e8c4"), "value" : 99997 }

{ "\_id" : ObjectId("60c0f517e3a6baa5eed1e8c5"), "value" : 99998 }

{ "\_id" : ObjectId("60c0f517e3a6baa5eed1e8c6"), "value" : 99999 }

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

> db.numbers.find().skip(99996).explain("executionStats");

{

"queryPlanner" : {

"plannerVersion" : 1,

"namespace" : "learn.numbers",

"indexFilterSet" : false,

"parsedQuery" : {

},

"winningPlan" : {

"stage" : "SKIP",

"skipAmount" : 0,

"inputStage" : {

"stage" : "COLLSCAN",

"direction" : "forward"

}

},

"rejectedPlans" : [ ]

},

"executionStats" : {

"executionSuccess" : true,

"nReturned" : 4,

"executionTimeMillis" : 23,

"totalKeysExamined" : 0,

"totalDocsExamined" : 100000,

"executionStages" : {

"stage" : "SKIP",

"nReturned" : 4,

"executionTimeMillisEstimate" : 0,

"works" : 100002,

"advanced" : 4,

"needTime" : 99997,

"needYield" : 0,

"saveState" : 100,

"restoreState" : 100,

"isEOF" : 1,

"skipAmount" : 0,

"inputStage" : {

"stage" : "COLLSCAN",

"nReturned" : 100000,

"executionTimeMillisEstimate" : 0,

"works" : 100002,

"advanced" : 100000,

"needTime" : 1,

"needYield" : 0,

"saveState" : 100,

"restoreState" : 100,

"isEOF" : 1,

"direction" : "forward",

"docsExamined" : 100000

}

}

},

"serverInfo" : {

"host" : "LAPTOP-RFUC65E1",

"port" : 27017,

"version" : "4.4.6",

"gitVersion" : "72e66213c2c3eab37d9358d5e78ad7f5c1d0d0d7"

},

"ok" : 1

}

1. *Создайте индекс для ключа value.*
2. *Получите информацию о всех индексах коллекции numbres.*
3. *Выполните запрос 2.*

> 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"

}

]

> db.numbers.find().skip(99996);

{ "\_id" : ObjectId("60c0f517e3a6baa5eed1e8c3"), "value" : 99996 }

{ "\_id" : ObjectId("60c0f517e3a6baa5eed1e8c4"), "value" : 99997 }

{ "\_id" : ObjectId("60c0f517e3a6baa5eed1e8c5"), "value" : 99998 }

{ "\_id" : ObjectId("60c0f517e3a6baa5eed1e8c6"), "value" : 99999 }

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

*На выполнение данного запроса тоже потребовалось 23 секунды.*

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