**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по Лабораторной Работе № 5

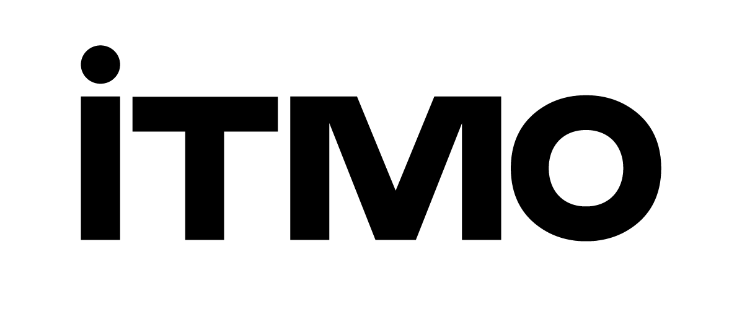
по дисциплине «**Проектирование и реализация баз данных**»

Автор: Барышников Владимир

Факультет: ФИКТ

Группа: K33391

Преподаватель: Говорова Марина Михайловна



Санкт-Петербург   
2023

# 8.1.1

*Создайте базу данных learn.*

use learn

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

*Заполните коллекцию единорогов:*

db.createCollection("unicorns")

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

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

db.unicorns.insert({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

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

db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

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

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

db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*Используя второй способ, вставьте в коллекцию единорогов документ:*

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

db.unicorns.insert(document)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

*Проверьте содержимое коллекции с помощью метода find.*

db.unicorns.find()

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

# 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})

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

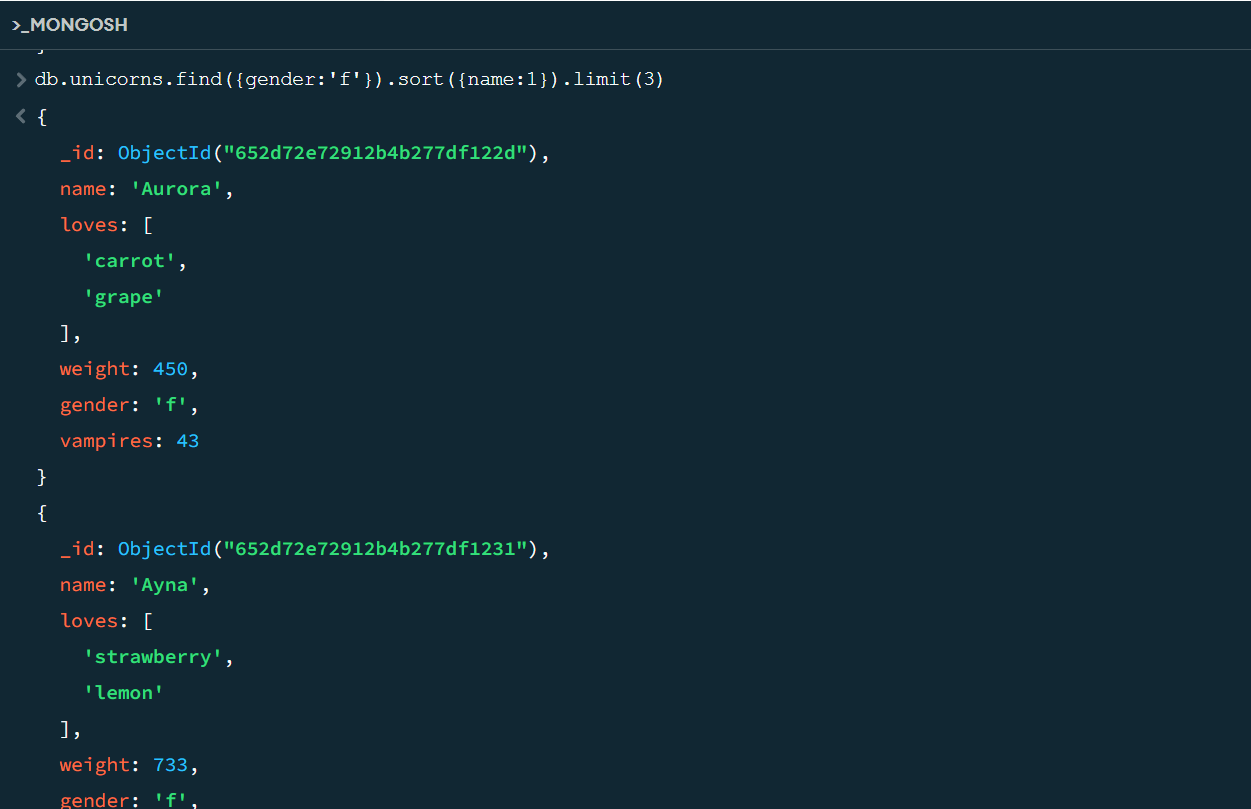
Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

 Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

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

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

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

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

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# 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, gender: 0})

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

# 8.1.4

*Вывести список единорогов в обратном порядке добавления.*

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

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

# 8.1.5

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

db.unicorns.find({}, {\_id: 0, loves: { $slice: 1}})

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

# 8.1.6

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

db.unicorns.find({gender: 'f', weight: { $gte: 500, $lte: 700}}, {\_id: 0})

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

# 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: 0})

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

# 8.1.8

*Найти всех единорогов, не имеющих ключ vampires.*

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

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# 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})

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

# 8.2.1

*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"}}

db.createCollection("towns")

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание

db.towns.insertMany([{ name: "Punxsutawney",

population: 6200,

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

famous\_for: [""],

mayor: {

name: "Jim Wehrle"}

},{

name: "New York",

population: 22200000,

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

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

mayor: {

name: "Michael Bloomberg",

party: "I"

}}, {

name: "Portland",

population: 528000,

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

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

mayor: {

name: "Sam Adams",

party: "D" }}

])

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*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: 0, name: 1, "mayor.name": 1, "mayor.party": 1})

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

*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: 0, name: 1, "mayor.name": 1, "mayor.party": 1})

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

# 8.2.2

*Сформировать функцию для вывода списка самцов единорогов.*

function ShowMaleUns() {

const maleUnics = db.unicorns.find({ gender: 'm' });

maleUnics.forEach(function (unicorn) {print(unicorn.name);});}

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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

const maleUnics\_Cursor = db.unicorns.find({ gender: 'm' }).sort({ name: 1 }).limit(2);

*Вывести результат, используя forEach.*

maleUnics\_Cursor.forEach(function(unicorn) {

print(unicorn.name);});

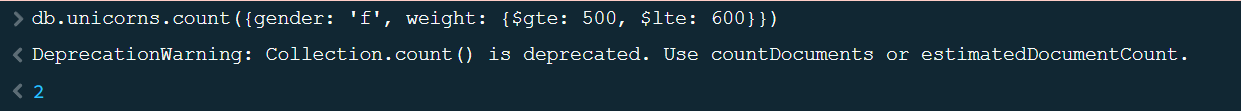
Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

# 8.2.3

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

db.unicorns.count({gender: 'f', weight: {$gte: 500, $lte: 600}})



# 8.2.4

*Вывести список предпочтений.*

db.unicorns.distinct("loves")

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# 8.2.5

*Посчитать количество особей единорогов обоих полов.*

db.unicorns.aggregate([{$group: {\_id: "$gender", count: {$sum: 1}}}])

db.unicorns.count() тоже сработает, но так как мы учимся, то выведем группировку по каждому из полов

db.unicorns.aggregate([{$group: {\_id: "$gender", count: {$sum: 1}}}])

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# 8.2.6

*Выполнить команду:*

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

weight: 340, gender: 'm'})

*Проверить содержимое коллекции unicorns.*

[*https://stackoverflow.com/questions/73118032/mongodb-db-save-type-error-db-collection-save-is-not-a-function*](https://stackoverflow.com/questions/73118032/mongodb-db-save-type-error-db-collection-save-is-not-a-function)

*Save не работает в новой версии, вместо этого используем insertOne*

db.unicorns.insertOne({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

db.unicorns.find() – ищем Барни

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# 8.2.7

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

db.unicorns.updateOne({name: 'Ayna', gender: 'f'},{$set: {weight: 800, vampires: 51}})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

db.unicorns.find()— ищем Айну

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# 8.2.8

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

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

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. *Проверить содержимое коллекции unicorns.*

db.unicorns.find()— ищем Рали

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

# 8.2.9

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

db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. *Проверить содержимое коллекции unicorns.*

*Сравним Рали из предыдущего задания:*

db.unicorns.find()

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# 8.2.10

db.towns.updateOne({name: "Portland"}, { $unset: {"mayor.party": ""}})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# 8.2.11

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

db.unicorns.updateOne({name: 'Pilot', gender: 'm' }, {$push: {loves: 'chocolate'}})

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

1. *Проверить содержимое коллекции unicorns.*

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# 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.updateOne({name: 'Aurora', gender: 'f'}, {$push: {loves: {$each: ['sugar', 'lemon']}}})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# 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"}}

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. *Удалите документы с беспартийными мэрами.*

db.towns.deleteMany({"mayor.party": {$exists: false}})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. *Проверьте содержание коллекции.*

db.towns.find()

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

1. *Очистите коллекцию.*

db.towns.deleteMany({})

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание

1. *Просмотрите список доступных коллекций.*

show collections

Изображение выглядит как текст, Шрифт, снимок экрана, дизайн

Автоматически созданное описание

# 8.3.1

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

db.zones.insertMany([

{

\_id: "zone1",

name: "Зона 1",

full\_name: "Зона спокойных",

description: "Место для спокойных единорогов"

},

{

\_id: "zone2",

name: "Зона 2",

full\_name: "Зона буйных",

description: "Место для буйных и энергичных единорогов"

},

{\_id: "zone3",

name: "Зона 3",

full\_name: "Вампирская деревня",

description: "Место для охоты на вампиров"}])

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

db.unicorns.updateOne({name: 'Aurora'}, {$set: {zone: { $ref: "zones", $id: "zone2" }}})

db.unicorns.updateOne({name: 'Ayna'}, {$set: {zone: { $ref: "zones", $id: "zone1" }}})

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*Проверьте содержание коллекции единорогов.*

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

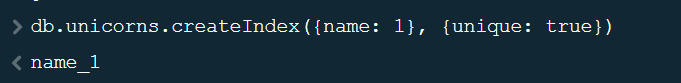
Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# 8.3.2

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

db.unicorns.createIndex({name: 1}, {unique: true})



# 8.3.3

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

db.unicorns.getIndexes()

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

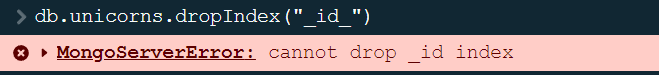
1. *Удалите все индексы, кроме индекса для идентификатора.*

db.unicorns.dropIndexes()

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. *Попытайтесь удалить индекс для идентификатора.*
2. db.unicorns.dropIndex("\_id\_")



# 8.3.4

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

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

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. *Выберите последних четыре документа.*

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

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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

db.numbers.find().sort({ \_id: -1 }).limit(4).explain("executionStats")

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

1. *Создайте индекс для ключа value.*

db.numbers.createIndex({value: 1})

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание

1. *Получите информацию обо всех индексах коллекции numbers.*

db.numbers.getIndexes()

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. *Выполните запрос 2.*

db.numbers.find().sort({ \_id: -1 }).limit(4).explain("executionStats")

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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

0 секунд, то есть компьютер почти не почувствовал нагрузки при работе с индексами.

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

Очевидно, более эффективен запрос с индексами, он облегчает работу в поиске и увеличивает скорость.

# Выводы

В данной лабораторной работе я ознакомился с большой частью функционала MongoDB, например, создание коллекций, агрегирование данных, изменение данных, а также научился работе со ссылками и индексами.