Введение в MongoDB

**Задание 1**

Цель практической работы:  
Научиться выполнять простые запросы в MongoDB.

mongoimport -d skdb -c posts --legacy posts.json

mongoimport -d skdb -c users --legacy users.json

**Что нужно сделать:**  
Из коллекции постов выберите документы, в которых среди топиков встречается as, идентификатор автора содержит example.ru, а score больше 100.

db.posts.find({"score": {$gt:100}, "author": /.\*example.com\*/, "topics": /.\*as.\*/})  
[

{

\_id: ObjectId('663cd6075486ec41c74fe478'),

author: 'orosanne@example.com',

creation\_date: ISODate('2020-12-15T00:00:00.000Z'),

topics: [ 'that', 'as', 'had', 'tired' ],

score: 183,

status: 'published',

message: 'on well she in your wouldn will the cats would the alice the who all explain getting i alice it and not said a i better sleepy longed that sure leaves i sing lesson classics the the a and it as i tears m rumbling well about didn clear of examine i week alice like explain the said them had alice but said the back that in hare forget it all small at dormouse so with it it your a sharply with it ve mock tale front them from oh you temper or wish to last roared these suppose a he a goes what a her i'

},

{

\_id: ObjectId('663cd6075486ec41c74fe485'),

author: 'orosanne@example.com',

creation\_date: ISODate('2021-08-23T00:00:00.000Z'),

topics: [ 'pleasure', 'is' ],

score: 4247,

status: 'published',

message: 'seemed up chorus into some not do you the coast because only and you the best are be no think me of some helpless one locked an his door face m way thought than t always the do so as miss well a i the sister birds i feet gather as so and it i however running the curls never call and though and opened what swam on said nor as she stairs box the if laid of and guessed said rabbit then never time all the outside and but then see was but i the she mock find pool and she sitting that the all jurors and the then i eyes the we the and here good to taller alice ridge will right alas was i door that candle no cakes in rattling don civil was once of he advisable of an boy why long an youth from and going that king constant himself to larger be i funny t looked just to the a a as a yawning said again before be said side mine currants for it you about began you nothing muttered but so all leave the who her the till use you not now reason up hatter no a pleasant a the in once mock if'

},

{

\_id: ObjectId('663cd6075486ec41c74fe4f6'),

author: 'rdaphine@example.com',

creation\_date: ISODate('2021-05-09T00:00:00.000Z'),

topics: [ 'pleasure', 'hot' ],

score: 2497,

status: 'published',

message: 'like to they of once very that and so and no m as pocket and know yet not to she yes of looking you and and long alice salmon shilling and large minute repeated the spectacles we parchment after of squeaking to s voice i to he and you till at above said in taking that told stairs to it round timidly what to as like said was the had nothing head silence with her indeed i youth cake upon sister tipped i un while nothing two remarked time up and in loud presents as last to another on learnt way now queen give nearly is quite'

}

]

**Задание 2**

Цель практической работы:  
Научиться писать запросы с использованием различных структур данных в MongoDB.

**Что нужно сделать:**  
Одним запросом добавьте два документа к коллекции posts:  
creation\_date — текущее время, автор — [skbx@example.com](mailto:skbx@example.com), topics должен быть списком из одного элемента mongodb;  
creation\_date — 31 декабря 2021 года, автор — [skbx@example.ru](mailto:skbx@example.ru).

db.posts.insertMany([{"creation\_date": new Date(), "author": "skbx@example.com", "topics": ["oh"]},{"creation\_date": "2021-12-31", "author": "skbx@example.com"}])  
  
**Задание 3**

Цель практической работы:  
Научиться анализировать запросы и создавать индексы в MongoDB.  
  
**Что нужно сделать:**  
Создайте композитный индекс для коллекции users, в него войдут поля first\_name и last\_name. Приведите запросы: на создание индекса и на проверку, что индекс используется.

db.users.createIndex({first\_name: 1, last\_name: 1})

db.users.getIndexes()

[

{ v: 2, key: { \_id: 1 }, name: '\_id\_' },

{

v: 2,

key: { firstName: 1, last\_name: 1 },

name: 'firstName\_1\_last\_name\_1'

},

{

v: 2,

key: { first\_name: 1, last\_name: 1 },

name: 'first\_name\_1\_last\_name\_1'

}

]

**Задание 4**

Цель практической работы:  
Научиться писать аналитические запросы в MongoDB.  
  
**Что нужно сделать:**  
Посчитайте сумму кармы по первым буквам имён пользователей для тех пользователей, у которых больше 300 визитов.  
Советы и указания  
Для выбора первой буквы имени используйте ключевое слово substr.

db.users.aggregate([{$match: {"visits": {$gt:300}}}, {$project:{karma:"$karma", word:{ "$substr":["$first\_name",0,1]}}},{$group:{\_id: "$word", sum\_karma:{$sum:"$karma"}}}])

[

{ \_id: 'A', sum\_karma: -28 },

{ \_id: 'L', sum\_karma: 243 },

{ \_id: 'D', sum\_karma: -39 },

{ \_id: 'T', sum\_karma: -68 },

{ \_id: 'B', sum\_karma: 323 },

{ \_id: 'P', sum\_karma: 94 },

{ \_id: 'V', sum\_karma: -43 },

{ \_id: 'M', sum\_karma: 516 },

{ \_id: 'E', sum\_karma: 120 },

{ \_id: 'R', sum\_karma: 53 },

{ \_id: 'S', sum\_karma: 296 },

{ \_id: 'J', sum\_karma: 477 },

{ \_id: 'H', sum\_karma: 79 },

{ \_id: 'C', sum\_karma: 176 },

{ \_id: 'Z', sum\_karma: -82 },

{ \_id: 'K', sum\_karma: 153 },

{ \_id: 'G', sum\_karma: 199 },

{ \_id: 'O', sum\_karma: 71 }

]

**Задание 5**

Цель практической работы:  
Научиться писать хранимые процедуры в MongoDB.  
  
**Что нужно сделать:**  
Создайте хранимую функцию shuffle, которая принимает один параметр — строку и возвращает строку со случайно переставленными символами.  
  
**Советы и указания:**  
Используйте встроенный в JavaScript метод Math.random() для сортировки символов в строке.

Скорее всего так:

db.system.js.save({\_id: "shuffle", value: function (str) {

var arr = str.split('');

for (var i = arr.length - 1; i > 0; i--) {

var j = Math.floor(Math.random() \* (i + 1));

var temp = arr [i];

arr [i] = arr [j];

arr [j] = temp;

}

return arr.join(''); }});

db.eval("return shuffle("hello"))

но save и eval устарели, вместо save так:

db.system.js.insert({\_id: "shuffle", value: function shuffle (str){

var arr = str.split('');

for (var i = arr.length - 1; i > 0; i--) {

var j = Math.floor(Math.random() \* (i + 1));

var temp = arr [i];

arr [i] = arr [j];

arr [j] = temp;

}

return arr.join(''); }});

но как выводить не понимаю.