**Задание 1**

Цель практической работы: Научиться выполнять простые запросы в MongoDB.

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({"author": /.\*example.ru.\*, "topics": /.\*as.\*/ , "score": {$gt: 100}})

**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": ["mongodb"]}, {"creation\_date": new Date("2021-12-31"), "author": "skbx@example.ru", "topics": ["mongodb"]}])

**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.find({})

**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([{$group: {\_id: substr("$first\_name", 0, 1), sum: {$sum: "$karma"}}}, {$match: {sum: {$gt: 300}}}])

**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.

db.system.js.insert({"\_id": "shuffle", "value": function(x) { return x.split("").sort(function() { return Math.random() - 0.5 }).join(""); }}) db.users.aggregate([{$project: {\_id: 0, name: shuffle("abcdefg")}}])