Контрольная работа по дисциплине

“Нерелеационные БД” (MongoDB)

Выполнил студент группы АБД19-1м

Полянчиков Владислав

Задание

Создать базу данных MongoDB новостной ленты для социального новостного сайта. Документ базы данных должен иметь следующую структуру:

* \_id,
* title - наименование статьи,
* theme – тематика,
* url-адрес новостного ресурса (например - http://primer.com/news\_1.txt),
* author - автор,
* vote\_count – подсчет голосов (количество «лайков»)
* tags - ключевые слова в виде массива строк
* image: { url: (например - http://primer.com/news\_1\_img.jpg),

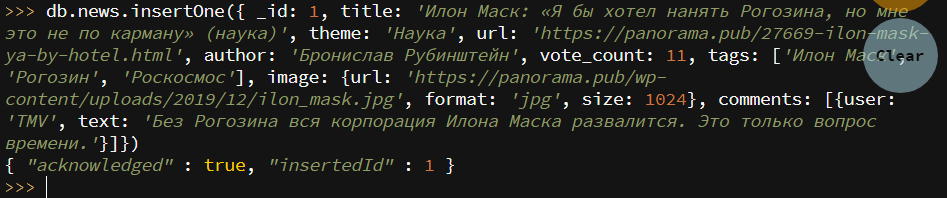
format: (jpg),

size: (1024)}

* comments – комментарии, хранящиеся в виде массива объектов, представляющие один комментарий (с указанием автора комментария (user) и самого комментария (text))

1. *Написать запрос на добавление одной новости в тематику «Наука»*

db.news.insertOne({ \_id: 1, title: 'Илон Маск: «Я бы хотел нанять Рогозина, но мне это не по карману» (наука)', theme: 'Наука', url: 'https://panorama.pub/27669-ilon-mask-ya-by-hotel.html', author: 'Бронислав Рубинштейн', vote\_count: 11, tags: ['Илон Маск', 'Рогозин', 'Роскосмос', 'финансовый'], image: {url: 'https://panorama.pub/wp-content/uploads/2019/12/ilon\_mask.jpg', format: 'jpg', size: 512}, comments: [{user: 'TMV', text: 'Без Рогозина вся корпорация Илона Маска развалится. Это только вопрос времени.'}]})



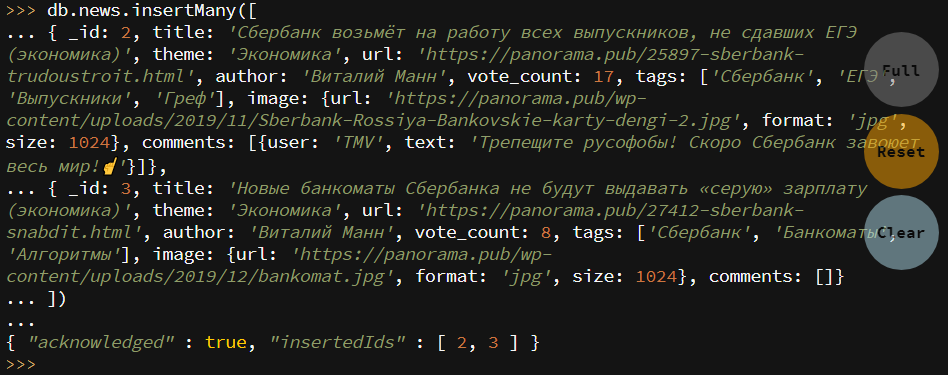
1. *Написать запрос для добавления 2-х статей в тематику «Экономика»*

db.news.insertMany([

{ \_id: 2, title: 'Сбербанк возьмёт на работу всех выпускников, не сдавших ЕГЭ (экономика)', theme: 'Экономика', url: 'https://panorama.pub/25897-sberbank-trudoustroit.html', author: 'Виталий Манн', vote\_count: 17, tags: ['Сбербанк', 'ЕГЭ', 'Выпускники', 'Греф', 'университет'], image: {url: 'https://panorama.pub/wp-content/uploads/2019/11/Sberbank-Rossiya-Bankovskie-karty-dengi-2.jpg', format: 'jpg', size: 1024}, comments: [{user: 'TMV', text: 'Трепещите русофобы! Скоро Сбербанк завоюет весь мир!☝'}]},

{ \_id: 3, title: 'Новые банкоматы Сбербанка не будут выдавать «серую» зарплату (экономика)', theme: 'Экономика', url: 'https://panorama.pub/27412-sberbank-snabdit.html', author: 'Виталий Манн', vote\_count: 8, tags: ['Сбербанк', 'Банкоматы', 'Алгоритмы', 'Москва'], image: {url: 'https://panorama.pub/wp-content/uploads/2019/12/bankomat.jpg', format: 'jpg', size: 512}, comments: []}

])

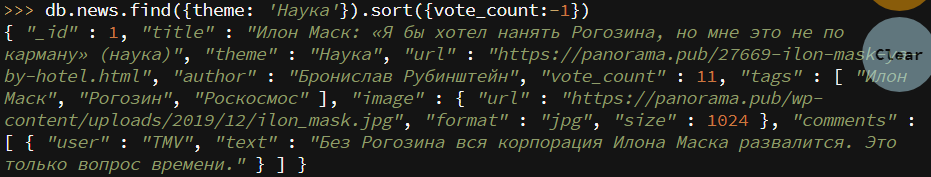


1. *Написать запрос для добавления статей из файла news.js*

load("C:/Program Files/news.js")

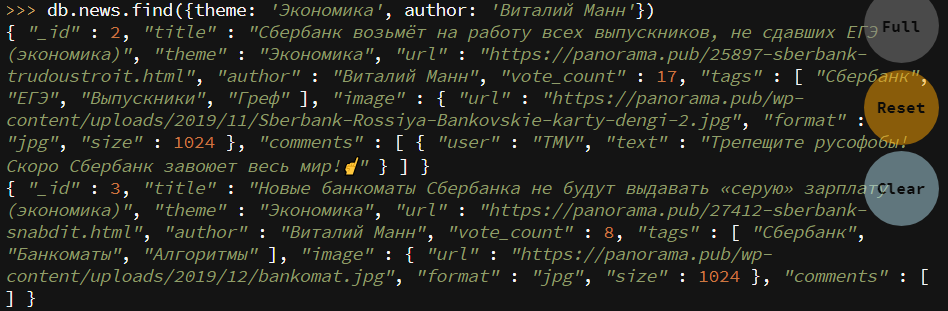
1. *Вывести все статьи тематики «Наука» отсортированные по количеству «лайков»*

db.news.find({theme: 'Наука'}).sort({vote\_count: -1})



1. *Вывести все статьи в тематике «Экономика и финансы» автора USER*

db.news.find({theme: 'Экономика', author: 'Виталий Манн'})



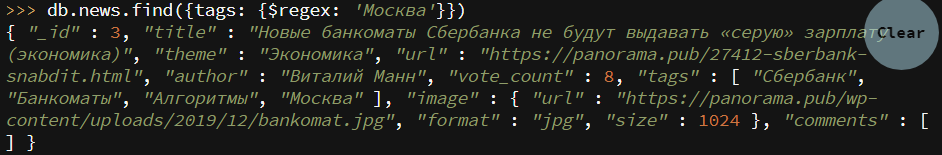
1. *Вывести список всех тематик*

db.news.distinct("theme")



1. *Вывести все статьи по ключевому слову «Москва»*

db.news.find({tags: {$regex: 'Москва'}})



1. *Вывести все статьи, имеющие оба ключевых слова: «финансовый», «университет»*

db.news.find({$and: [{tags: {$regex: 'финансовый'}}, {tags: {$regex: 'университет'}}]})



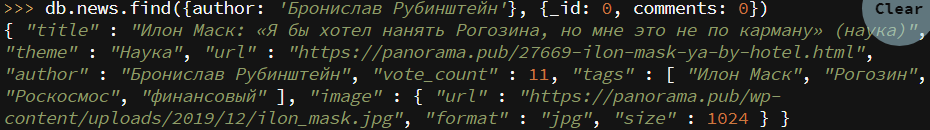
1. *Из тематики «Наука» вывести только наименования и авторов статей*

db.news.find({theme: 'Наука'}, {title: 1, author: 1, \_id: 0})



1. *Вывести статьи автора USER без вывода идентификатора статьи и комментариев*

db.news.find({author: 'Бронислав Рубинштейн'}, {\_id: 0, comments: 0})



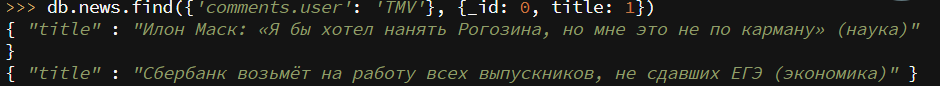
1. Вывести наименования статей с картинкой формата jpg больше 1000 Кб.

db.news.find({'image.size': {$gt: 1000}}, {\_id: 0, title: 1})



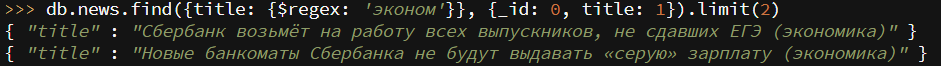
1. Вывести наименования статей, которые прокомментировал пользователь USER

db.news.find({'comments.user': 'TMV'}, {\_id: 0, title: 1})



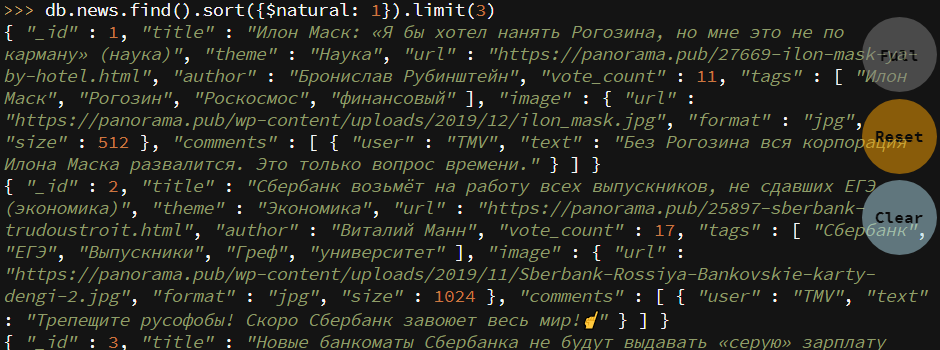
1. *Вывести первые две статьи, в названии которых есть часть слова «эконом»*

db.news.find({title: {$regex: 'эконом'}}, {\_id: 0, title: 1}).limit(2)



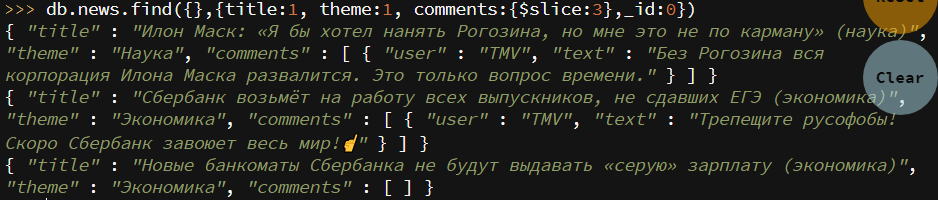
1. *Вывести первые 3 статьи, которые были добавлены в коллекцию*

db.news.find({\_id:0, title:1}).sort({$natural: 1}).limit(3)



1. *Вывести наименование статьи, тематику и 3 комментария к ней.*

db.news.find({},{title:1, theme:1, comments:{$slice:3},\_id:0})



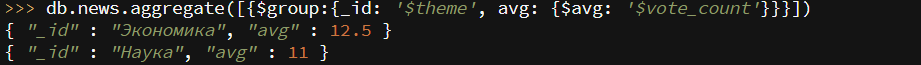
1. *Подсчитать количество комментариев пользователя USER*

db.news.find({'comments.user' : 'TMV' }).count()



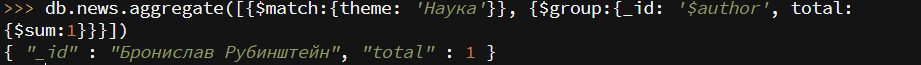
1. *Вывести среднее значение рекомендаций для каждой рубрики*

db.news.aggregate([{$group:{\_id: '$theme', avg: {$avg: '$vote\_count'}}}])



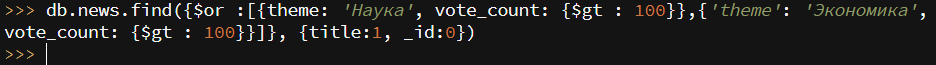
1. *Вывести количество статей, которые написал каждый пользователь в рубрику «Наука»*

db.news.aggregate([{$match:{theme: 'Наука'}}, {$group:{\_id: '$author', total: {$sum:1}}}])



1. *Вывести статьи тематик «Наука» и «Экономика» с количеством «лайков» больше 100*

db.news.find({$or :[{theme: 'Наука', vote\_count: {$gt : 100}},{'theme': 'Экономика', vote\_count: {$gt : 100}}]}, {title:1, \_id:0})



1. *Вывести максимальный размер изображения в новостях для каждого формата*

db.news.aggregate([{$group:{\_id: '$image.format', max: {$max: '$image.size'}}}])



1. *Вывести список статей, в которых количество комментариев больше 3*

db.news.find({$where: 'this.comments.length > 3'})



1. *Увеличить количество «лайков» у всех статей категории «Наука» на 3*

db.news.update({theme: 'Наука'}, {$inc: {vote\_count: 3}})



1. *Добавить комментарий к любой статье*

db.news.update({\_id: 3}, {$push: {comments: {$each:[{user: 'polyandtd', text: 'мда, опять к конвертам возвращаться'}]}}})



1. *Удалить все комментарии пользователя USER к статьям категории «Экономика»*

db.news.updateMany({theme : 'Экономика', 'comments.user' : 'polyandtd' }, {$pop: {comments:1 }})



1. *Добавить уникальный индекс на атрибуты: наименование статьи и тематика*

db.news.createIndex({ title: 1, theme: 1 } , {'unique' : true})

