

---

# Нереляционные базы данных и MongoDB

## *Урок 2. Введение в MongoDB*

---

### **Оглавление**

Задание 1.....	2
Задание 2.....	3
Задание 3.....	4
Задание 4.....	7

## Задание 1

Цель практической работы:

Научиться выполнять простые запросы в MongoDB.

### Что нужно сделать:

Из коллекции постов выберите документы, в которых среди топикиов встречается as, идентификатор автора содержит example.ru, а score больше 100.

---

```
skdb> db.posts.find({
...  topics: "as",
...  author: { $regex: "example\\.ru$" },
...  score: { $gt: 100 }
... })
```

```
[
```

```
{
  _id: ObjectId('6826704aa80bb5eef5c5a142'),
  author: 'lprudy@example.ru',
  creation_date: { '$date': '2020-06-27T00:00:00Z' },
  topics: [ 'as', 'very', 'a', 'but' ],
  score: 242,
  status: 'published',
```

message: 'us for over with rabbit and screamed to to life her sat and i to and be and and join don fell march said majesty found if ever and their stand to than natural doing dormouse alice know size and back go twinkle alice went askance commotion caused in the she all found taking croquet dormouse but hearing the i as evidence i put the your'

```
},
```

```
{
```

```
  _id: ObjectId('6826704aa80bb5eef5c5a146'),
  author: 'aalfred@example.ru',
  creation_date: { '$date': '2021-02-03T00:00:00Z' },
  topics: [ 'as', 'pleasure', 'hot' ],
  score: 4707,
  status: 'published',
```

message: 'all find the a that alice therefore off yet same and and it all rather and said the said out leaves tell this nor chorus just nine blasts made s plate if to nothing her round nose to except interesting alice say become is hatter grinned this and with close that over come and that all as mushroom interrupted then first time up the it wood what procession we moment pointing thought a round you'

```
},
```

```
{
```

```
  _id: ObjectId('6826704aa80bb5eef5c5a195'),
  author: 'sagnesse@example.ru',
  creation_date: { '$date': '2020-03-27T00:00:00Z' },
  topics: [ 'worth', 'as', 'pleasure', 'get' ],
  score: 369,
  status: 'published',
```

```
  message: 'sure for moment a it were it like could said mean fluttered'
```

```
}
```

```
]
```

## 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, topics должен быть списком из одного элемента mongodb;

creation\_date – 31 декабря 2021 года, автор – skbx@example.ru.

---

```
skdb> db.posts.insertMany([ {creation_date: new Date(), author: 'skbx@example.com', topics:
['mongodb']}, {creation_date: new Date('2021-12-31'), author: 'skbx@example.ru'} ])
```

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68267495a80bb5eef5c5a22a'),
    '1': ObjectId('68267495a80bb5eef5c5a22b')
  }
}
```

```
skdb> db.posts.find({author: 'skbx@example.com'})
```

```
[
  {
    _id: ObjectId('68267495a80bb5eef5c5a22a'),
    creation_date: ISODate('2025-05-15T23:11:17.177Z'),
    author: 'skbx@example.com',
    topics: [ 'mongodb' ]
  }
]
```

```
skdb> db.posts.find({author: 'skbx@example.ru'})
```

```
[
  {
    _id: ObjectId('68267495a80bb5eef5c5a22b'),
    creation_date: ISODate('2021-12-31T00:00:00.000Z'),
    author: 'skbx@example.ru'
  }
]
```

### 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.

---

```
skdb> db.users.createIndexes([{'first_name': 1}, {'last_name': 2}])
[ 'first_name_1', 'last_name_2' ]
```

Проверка:

```
skdb> db.users.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { first_name: 1 }, name: 'first_name_1' },
  { v: 2, key: { last_name: 2 }, name: 'last_name_2' }
]
```

Проверка:

```
skdb> db.users.find({ first_name: 'Rebekkah' }).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'skdb.users',
    parsedQuery: { first_name: { '$eq': 'Rebekkah' } },
    indexFilterSet: false,
    queryHash: '7792D0FE',
    planCacheShapeHash: '7792D0FE',
    planCacheKey: '8F53FA6D',
    optimizationTimeMillis: 0,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { first_name: 1 },
        indexName: 'first_name_1',
        isMultiKey: false,
        multiKeyPaths: { first_name: [ ] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { first_name: [ ["Rebekkah", "Rebekkah"] ] }
      }
    }
  },
```

```

    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 1,
    executionTimeMillis: 0,
    totalKeysExamined: 1,
    totalDocsExamined: 1,
    executionStages: {
      isCached: false,
      stage: 'FETCH',
      nReturned: 1,
      executionTimeMillisEstimate: 0,
      works: 2,
      advanced: 1,
      needTime: 0,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      docsExamined: 1,
      alreadyHasObj: 0,
      inputStage: {
        stage: 'IXSCAN',
        nReturned: 1,
        executionTimeMillisEstimate: 0,
        works: 2,
        advanced: 1,
        needTime: 0,
        needYield: 0,
        saveState: 0,
        restoreState: 0,
        isEOF: 1,
        keyPattern: { first_name: 1 },
        indexName: 'first_name_1',
        isMultiKey: false,
        multiKeyPaths: { first_name: [] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { first_name: [ ["Rebekkah", "Rebekkah"] ] },
        keysExamined: 1,
        seeks: 1,
        dupsTested: 0,
        dupsDropped: 0
      }
    }
  },
  queryShapeHash:
'3B108858BA7D3A30CF57C297B38173023A61E2EA7E9B5CFE482CBFE6C49AF8C2',
  command: { find: 'users', filter: { first_name: 'Rebekkah' }, '$db': 'skdb' },

```

```
serverInfo: {
  host: 'helga-test',
  port: 27017,
  version: '8.0.9',
  gitVersion: 'f882ef816d531ecfbb593843e4c554fda90ca416'
},
serverParameters: {
  internalQueryFacetBufferSizeBytes: 104857600,
  internalQueryFacetMaxOutputDocSizeBytes: 104857600,
  internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
  internalDocumentSourceGroupMaxMemoryBytes: 104857600,
  internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
  internalQueryProhibitBlockingMergeOnMongoS: 0,
  internalQueryMaxAddToSetBytes: 104857600,
  internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
  internalQueryFrameworkControl: 'trySbeRestricted',
  internalQueryPlannerIgnoreIndexWithCollationForRegex: 1
},
ok: 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.

---

```
skdb> db.users.aggregate([
... { $match: { visits: { $gt: 300 } } },
... {
...   $group: {
...     _id: { $substr: ['$first_name', 0, 1] },
...     total_karma: { $sum: "$karma" }
...   }
... }
... ])
```

```
[
  { _id: 'M', total_karma: 516 },
  { _id: 'A', total_karma: -28 },
  { _id: 'H', total_karma: 79 },
  { _id: 'C', total_karma: 176 },
  { _id: 'L', total_karma: 243 },
  { _id: 'V', total_karma: -43 },
  { _id: 'K', total_karma: 153 },
  { _id: 'Z', total_karma: -82 },
  { _id: 'T', total_karma: -68 },
  { _id: 'P', total_karma: 94 },
  { _id: 'G', total_karma: 199 },
  { _id: 'E', total_karma: 120 },
  { _id: 'R', total_karma: 53 },
  { _id: 'S', total_karma: 296 },
  { _id: 'J', total_karma: 419 },
  { _id: 'O', total_karma: 71 },
  { _id: 'D', total_karma: -64 },
  { _id: 'B', total_karma: 323 }
]
```

---

Ссылка на репозиторий:

<https://github.com/olgashenkel/GeekBrains-specialization-ELECTIVES/tree/main/11.%20Non-relational%20databases%20and%20MongoDB>