

## Лабораторная работа № 15

Тема: Агрегация данных и поиск в MongoDB.

**Цель:** Научиться выполнять агрегирующие запросы, такие как фильтрация, группировка, сортировка и проекция данных. Освоить создание индексов для оптимизации поиска. Научиться выполнять текстовый поиск и использовать регулярные выражения в MongoDB.

### Задание:

1. Подготовка данных:

Создайте коллекцию products с документами следующего вида:

```
{
  "name": "Product Name",
  "category": "Category Name",
  "price": NumberDecimal("Цена"),
  "tags": ["tag1", "tag2"],
  "stock": NumberInt(Количество),
  "created_at": ISODate("Дата создания")
}
```

Добавьте не менее 10-15 документов с разными категориями и ценами.

2. Агрегационные запросы:

- Используйте \$match для выбора продуктов с ценой выше заданного значения. Например:

```
db.products.aggregate([
  {
    $match: {
      price: { $gt: 100 }
    }
  }
])
```

- Выполните \$group для подсчета количества продуктов в каждой категории. Например:

```
db.products.aggregate([
  {
    $group: {
      _id: "$category", // Группируем по полю "category"
      productCount: { $sum: 1 } // Считаем количество продуктов в каждой категории
    }
  }
])
```

- Примените \$sort для сортировки продуктов по цене в порядке убывания. Например:

```
db.products.aggregate([
  {
    $sort: {
      price: -1 // Сортировка по полю "price" в порядке убывания
    }
  }
])
```

- Используйте \$project для создания нового поля totalValue, равного произведению цены на количество на складе. Например:

```
db.products.aggregate([
  {
    $project: {
      name: 1, // Включаем поле "name" в результат
      category: 1, // Включаем поле "category"
      price: 1, // Включаем поле "price"
      stock: 1, // Включаем поле "stock"
      totalValue: { $multiply: ["$price", "$stock"] } // Создаем новое поле "totalValue"
    }
  }
])
```

- Осортируйте продукты по цене в порядке убывания и выведите 5 самых дорогих продуктов. Например:

```

db.products.aggregate([
  {
    $match: {
      category: "Electronics" // Фильтруем продукты категории "Electronics"
    }
  },
  {
    $sort: {
      price: -1 // Сортируем по цене в порядке убывания
    }
  },
  {
    $limit: 5 // Ограничиваем результат до 5 продуктов
  }
])

```

### 3. Индексация:

- Создайте индекс на поле category и выполните запрос с сортировкой по категориям. Например:

```
db.products.createIndex({ category: 1 })
```

```
db.products.find().sort({ category: 1 })
```

- Создайте составной индекс на полях price и stock и выполните запрос с сортировкой по цене и количеству товара на складе. Например:

```
db.products.createIndex({ price: 1, stock: -1 })
```

```
db.products.find().sort({ price: 1, stock: -1 })
```

### 4. Поиск:

- Создайте текстовый индекс на поле name. Выполните полнотекстовый поиск по ключевому слову. Например:

```
db.products.createIndex({ name: "text" })
```

```
db.products.find({ $text: { $search: "Laptop" } })
```

- Используйте регулярные выражения для поиска всех продуктов, имена которых начинаются с определенной буквы. Например:

```

db.products.find({
  name: { $regex: "^L", $options: "i" } // ^L означает "начинается с L",
})                                     опция "i" делает поиск нечувствительным к регистру

```

Выполните поиск продуктов с ценой в заданном диапазоне и отсортируйте их по дате создания. Например:

```
db.products.find({  
  price: { $gte: 100, $lte: 1000 } // Поиск по диапазону цен  
}).sort({ created_at: 1 }) // Сортировка по дате создания в порядке возрастания
```

**Отчет должен содержать (см. образец):**

- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- скриншоты подтверждающие выполнение заданий.

Отчеты в формате **pdf** отправлять на email: [colledge20education23@gmail.com](mailto:colledge20education23@gmail.com)