Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение

высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики управления и технологий

Кузьмина Дарья Юрьевна БД-241м

Инструменты хранения и анализа больших данных

**Лабораторная работа 2.1 Часть 1.**

**Изучение методов хранения данных на основе NoSQL**

**Вариант 11**

Направление подготовки/специальность

38.04.05 - Бизнес-информатика

Бизнес-аналитика и большие данные

(очная форма обучения)

Руководитель дисциплины:

Босенко Т.М., доцент департамента

информатики, управления и технологий,

доктор экономических наук

Москва

2025

Содержание

[Введение 2](#_Toc192845116)

[Основная часть 2](#_Toc192845117)

[Заключение 13](#_Toc192845118)

## Введение

**Цель**

получить практические навыки работы с документо-ориентированной базой данных MongoDB путем выполнения основных операций по управлению данными.

**Задачи:**

1. Изучить основы работы с MongoDB:

- Подключение к базе данных.

- Создание коллекций.

- Добавление документов.

2. Освоить основные операции с документами:

- Поиск документов.

- Обновление документов.

- Удаление документов.

3. Научиться работать с индексами:

- Создание индексов.

- Использование индексов для оптимизации.

4. Освоить базовые принципы агрегации данных:

- Группировка данных.

- Подсчет статистики.

- Сортировка результатов.

5. Научиться выполнять текстовый поиск:

- Создание текстовых индексов.

- Поиск по текстовым полям.

**Необходимое ПО:**

* **Операционная система**: Ubuntu 22.
* **СУБД**: GraphDB.
* **Docker**: для запуска контейнера с MongoDB.
* **Среда разработки**: MongoDB Compass - редактор для выполнения запросов.

## Основная часть

Задача 1.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Запускаем докер композер

Изображение выглядит как текст, Шрифт, веб-страница, снимок экрана

Автоматически созданное описание

Проверяем, запущена ли служба в контейнере.

Сервер доступен, версия верная, стабильная, команды будут работать, мы находимся в тестовой базе данных.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Подключились монго дБ компас

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Удаляем ссылку, вставляем свой документ для импорта данных

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Проверяем видимость коллекции через терминал

Изображение выглядит как текст, программное обеспечение, снимок экрана, Мультимедийное программное обеспечение

Автоматически созданное описание

Выполнили подсчет документов в базе

Изображение выглядит как программное обеспечение, снимок экрана, текст, Мультимедийное программное обеспечение

Автоматически созданное описание

Добавим еще один фильм

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, веб-страница

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Добавим персон

Изображение выглядит как текст, программное обеспечение, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, веб-страница, программное обеспечение

Автоматически созданное описание

Задаем новый рейтинг.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Оптимизация производительности создаем тайтл.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Индивидуальное задание **Вариант 11**

Пояснения к заданиям варианта 11:

1. Создайте коллекцию "locations" с информацией о местах съёмок для 5 фильмов

2. Найдите все фильмы с exactly 2 режиссерами

3. Добавьте поле "remakes" в виде массива для фильмов до 1980 года

4. Создайте индекс по полю runtime

5. Посчитайте количество фильмов каждого жанра для каждого десятилетия

1. Создание коллекции "locations" с информацией о местах съёмок для 5 фильмов

Цель задания — создать новую коллекцию для хранения данных о локациях съемок фильмов. Для этого используется команда «insertMany()», которая добавляет несколько документов за один запрос. Каждый документ содержит идентификатор фильма («filmId»), его название («title») и массив мест съемок («locations»). Примеры данных включают города и страны, где снимались фильмы, такие как "Los Angeles, USA" для "Pulp Fiction".

db.locations.insertMany([

{

filmId: "tt0110912",

title: "Pulp Fiction",

locations: ["Los Angeles, USA", "Paris, France"]

},

{

filmId: "tt0133093",

title: "The Matrix",

locations: ["Sydney, Australia", "San Francisco, USA"]

},

{

filmId: "tt1375666",

title: "Inception",

locations: ["Tokyo, Japan", "London, UK", "Morocco"]

},

{

filmId: "tt0068646",

title: "The Godfather",

locations: ["New York, USA", "Sicily, Italy"]

},

{

filmId: "tt0088763",

title: "Back to the Future",

locations: ["Los Angeles, USA", "Hill Valley (фиктивное место)"]

}

]);

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

2. Поиск фильмов с точно 2 режиссерами

Задача — найти фильмы, у которых в списке режиссеров («directors») ровно два человека. Для этого применяется оператор «$size», который проверяет длину массива. Запрос «db.movies.find({ "directors": { $size: 2 } })» вернет все подходящие документы. Важно учитывать, что если поле «directors» отсутствует или содержит не массив, такие документы не будут найдены.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

3. Добавление поля "remakes" для фильмов до 1980 года

Цель — добавить новое поле «remakes» (ремейки) в виде пустого массива к фильмам, выпущенным до 1980 года. Используется команда «updateMany()» с условием «{ year: { $lt: 1980 } }» для выбора старых фильмов и оператором «$set» для добавления поля. Если таких фильмов нет в коллекции, запрос не изменит документы.

Изображение выглядит как текст, Шрифт, снимок экрана, дизайн

Автоматически созданное описание

4. Создание индекса по полю runtime

Индекс по полю «runtime» (продолжительность фильма) ускоряет поиск и сортировку по этому параметру. Команда «db.movies.createIndex({ runtime: 1 })» создает индекс с сортировкой по возрастанию. Индексы улучшают производительность чтения, но могут замедлить операции записи.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

5. Подсчет количества фильмов каждого жанра для каждого десятилетия

Задача — проанализировать распределение жанров по десятилетиям. Для этого используется агрегация:

- «$unwind» разбивает массив жанров на отдельные документы.

- «$project» вычисляет десятилетие как «(год // 10) 10» (например, 1994 → 1990).

- «$group» группирует данные по жанру и десятилетию, подсчитывая количество фильмов.

- «$sort» упорядочивает результат по десятилетию и жанру.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Выгружаем csv.

# Заключение

**Вывод**:

В ходе лабораторной работы были освоены основные операции работы с MongoDB, включая создание и управление базами данных, коллекциями и документами. Изучены методы поиска, обновления и удаления данных, а также работа с индексами и агрегацией. Практические задания позволили закрепить навыки использования командной строки и графических интерфейсов для взаимодействия с MongoDB. Работа продемонстрировала гибкость и эффективность документо-ориентированной СУБД для обработки структурированных и неструктурированных данных.