**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика, искусственный интеллект и системы управления КАФЕДРА Системы обработки информации и управления

**Лабораторная работа №4**

**«MongoDb»**

ИСПОЛНИТЕЛИ: Гаврилов Л.Я.

ФИО

группа ИУ5-24М

подпись

"" март 2024 г.

ПРЕПОДАВАТЕЛЬ: Виноградова М.В.

ФИО

подпись

" " 2024 г.

Москва – 2024

1. **Цель работы**

* Изучить модель данных и способы работы с документными БД NoSql.
* Освоить методы создания документной БД и языки запросов к ним.
* Получить навыки работы с документной БД MongoDb.

1. **Задание**
2. Создать в среде MongoDb свою БД по теме «Подбор фильмов на основе глубокого обучения», содержащую не менее трех коллекций. Добавить в коллекции БД объекты сложной структуры, содержащие вложенные структуры и коллекции. Продемонстрировать (вывести на экран) содержимое коллекций.
3. Продемонстрировать изменение объектов БД:

 добавление элементов объекта,

 изменение элементов объекта,

 удаление элементов объекта,

 замена всего объекта,

 удаление объекта.

1. Выполнить запросы к базе данных:

 вывод всех элементов коллекции,

 вывод с фильтрацией (условия И, ИЛИ),

 проекция вывода (вывод части полей),

 сортировка,

 удаление дубликатов,

 условия на поля вложенных структур,

 поиск по вложенным коллекциям объекта

# Порядок выполнения

1. Создать в среде MongoDb свою БД по теме «Подбор фильмов на основе глубокого обучения», содержащую не менее трех коллекций. Добавить в коллекции БД объекты сложной структуры, содержащие вложенные структуры и коллекции. Продемонстрировать (вывести на экран) содержимое коллекций.

1. **Создание БД**

use MovieRecommendationDB

1. **Создание коллекций и добавление документов**

Создадим три коллекции: movies, users, reviews.

**movies**

db.movies.insertMany([

{

title: "Inception",

genre: ["Sci-Fi", "Thriller"],

year: 2010,

director: {

name: "Christopher Nolan",

birthYear: 1970,

awards: ["Oscar", "Golden Globe"]

},

ratings: [

{ source: "IMDb", score: 8.8 },

{ source: "Rotten Tomatoes", score: 87 }

]

},

{

title: "The Matrix",

genre: ["Sci-Fi", "Action"],

year: 1999,

director: {

name: "Wachowski Brothers",

birthYear: null,

awards: ["Oscar"]

},

ratings: [

{ source: "IMDb", score: 8.7 },

{ source: "Rotten Tomatoes", score: 87 }

]

}

])

**Users**

db.users.insertMany([

{

username: "john\_doe",

email: "john@example.com",

age: 25,

preferences: {

genres: ["Sci-Fi", "Action"],

directors: ["Christopher Nolan", "Wachowski Brothers"]

}

},

{

username: "jane\_smith",

email: "jane@example.com",

age: 30,

preferences: {

genres: ["Drama", "Thriller"],

directors: ["Steven Spielberg"]

}

}

])

**Reviews**

db.reviews.insertMany([

{

movieTitle: "Inception",

username: "john\_doe",

rating: 9,

comment: "Amazing movie with a mind-bending plot!"

},

{

movieTitle: "The Matrix",

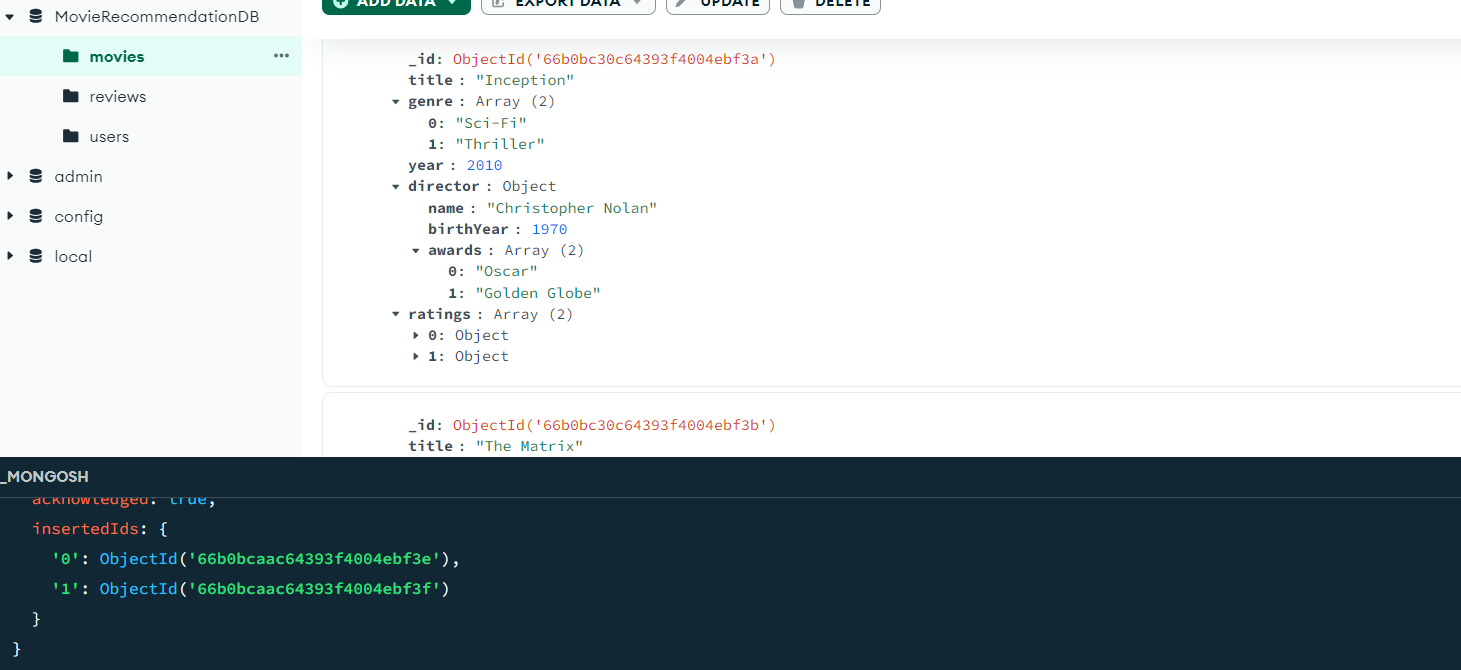
username: "jane\_smith",

rating: 8,

comment: "A groundbreaking film in the Sci-Fi genre."

}

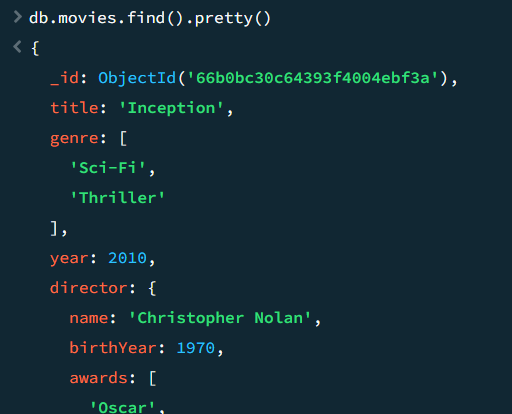
])



1. **Просмотр содержимого коллекций**

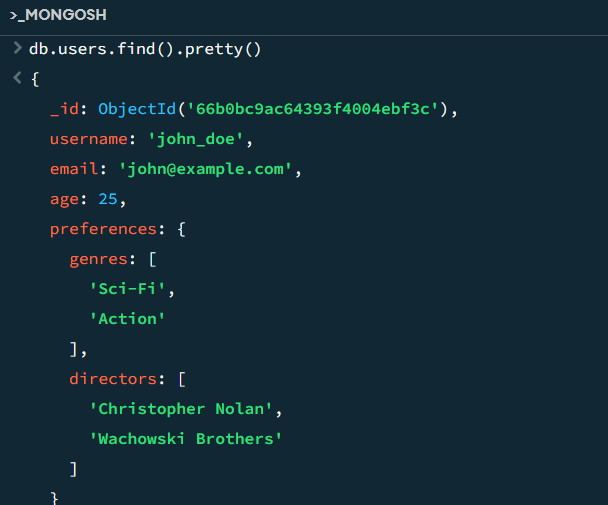
Коллекция **movies**:

db.movies.find().pretty()



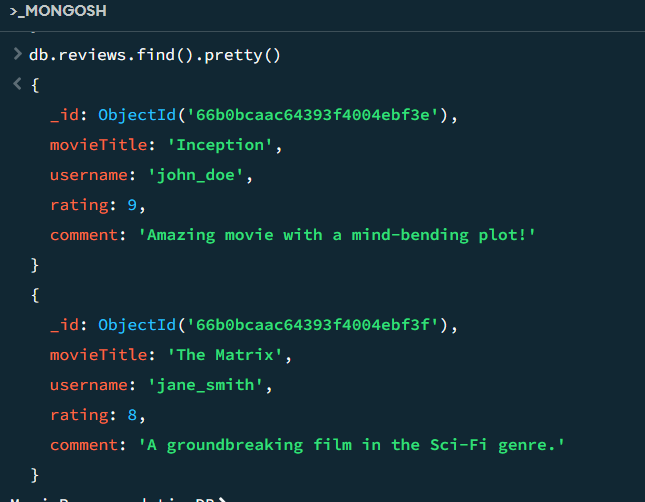
Коллекция **users**:

db.users.find().pretty()



Коллекция **reviews**:

db.reviews.find().pretty()



1. Продемонстрировать изменение объектов БД:

 добавление элементов объекта,

 изменение элементов объекта,

 удаление элементов объекта,

 замена всего объекта,

 удаление объекта.

1. **Добавление элемента в объект**

Добавим награду режиссеру фильма "The Matrix":

db.movies.updateOne(

{ title: "The Matrix" },

{ $push: { "director.awards": "BAFTA" } }

)



1. Изменение элемента объекта

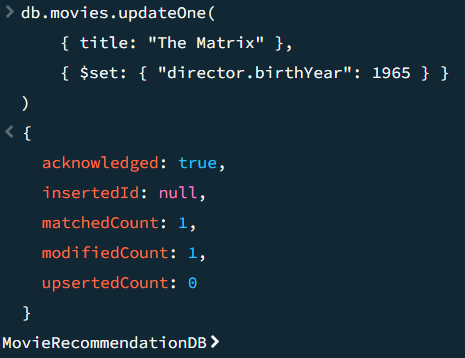
Изменим год рождения режиссера фильма "The Matrix":

db.movies.updateOne(

{ title: "The Matrix" },

{ $set: { "director.birthYear": 1965 } }

)



1. Удаление элемента из объекта

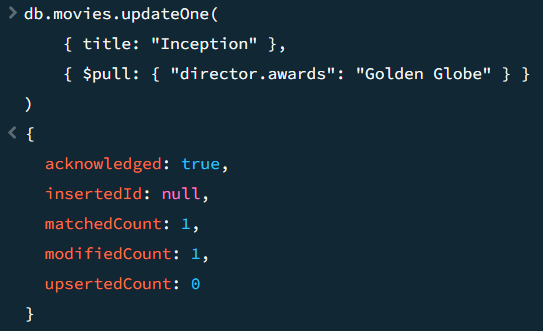
Удалим награду "Golden Globe" у режиссера фильма "Inception":

db.movies.updateOne(

{ title: "Inception" },

{ $pull: { "director.awards": "Golden Globe" } }

)



1. Замена всего объекта

Заменим документ пользователя "john\_doe":

db.users.replaceOne(

{ username: "john\_doe" },

{

username: "john\_doe",

email: "john\_new@example.com",

age: 26,

preferences: {

genres: ["Sci-Fi", "Adventure"],

directors: ["Christopher Nolan", "James Cameron"]

}

}

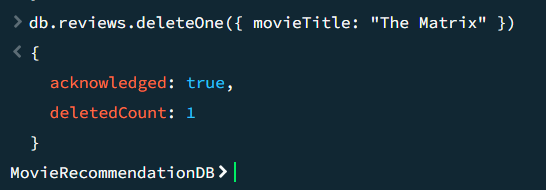
)



1. Удаление объекта

Удалим документ с рецензией на фильм "The Matrix":

db.reviews.deleteOne({ movieTitle: "The Matrix" })



1. Выполнить запросы к базе данных:

 вывод всех элементов коллекции,

 вывод с фильтрацией (условия И, ИЛИ),

 проекция вывода (вывод части полей),

 сортировка,

 удаление дубликатов,

 условия на поля вложенных структур,

 поиск по вложенным коллекциям объекта

1. Вывод всех элементов коллекции

Коллекция movies:

db.movies.find().pretty()



1. Вывод с фильтрацией (условия И, ИЛИ)

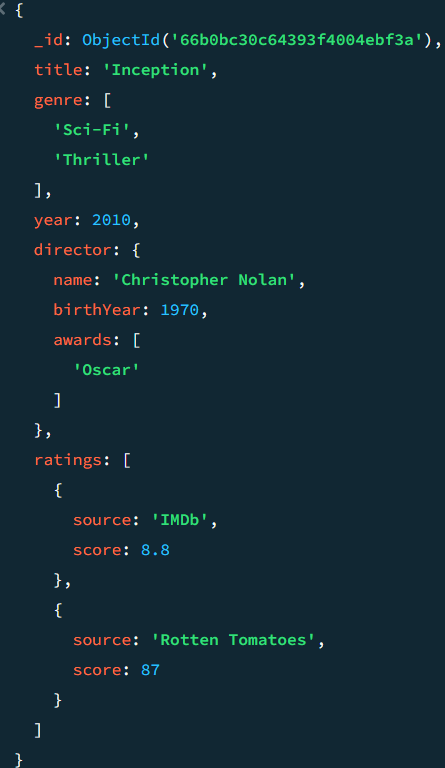
Фильтрация по году и жанру (И):

db.movies.find({ year: 2010, genre: "Sci-Fi" }).pretty()



Фильтрация по году или жанру (ИЛИ):

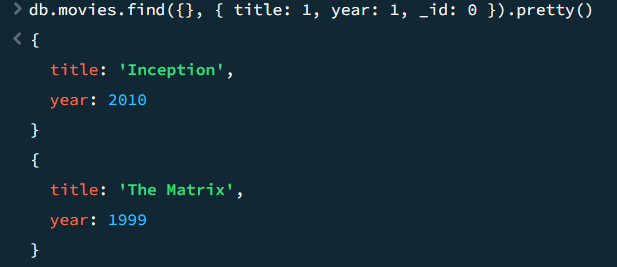
db.movies.find({ $or: [{ year: 2010 }, { genre: "Action" }] }).pretty()



1. Проекция вывода (вывод части полей)

Вывод только названий и года выхода фильмов:

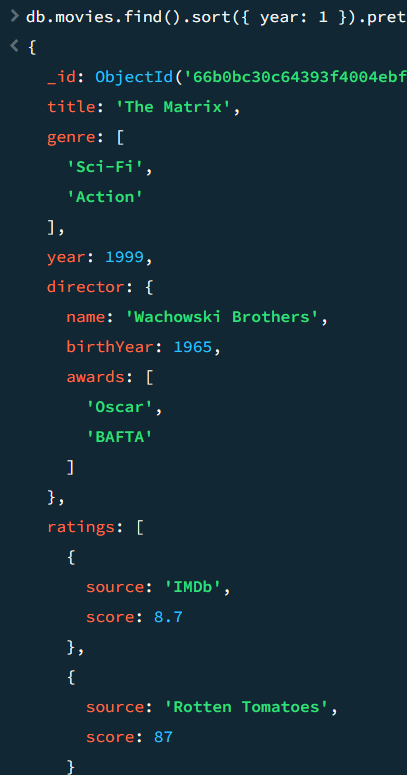
db.movies.find({}, { title: 1, year: 1, \_id: 0 }).pretty()



1. Сортировка

Сортировка фильмов по году выпуска (по возрастанию):

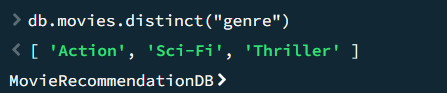
db.movies.find().sort({ year: 1 }).pretty()



1. Удаление дубликатов

Предположим, что у нас есть дублирующиеся документы, и мы хотим получить уникальные значения для жанров:

db.movies.distinct("genre")



1. Условия на поля вложенных структур

Найдем фильмы, где режиссер выиграл "Oscar":

db.movies.find({ "director.awards": "Oscar" }).pretty()



1. Поиск по вложенным коллекциям объекта

Найдем фильмы с оценкой выше 8.5 на IMDb:

db.movies.find({ "ratings": { $elemMatch: { source: "IMDb", score: { $gt: 8.5 } } } }).pretty()



**Вывод по выполненному заданию**

В процессе выполнения этого задания мы изучили и применили на практике различные операции с базой данных MongoDB. Рассмотрим ключевые аспекты, которые были изучены и продемонстрированы: