|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

Институт информационных технологий (ИИТ)

Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Распределенные системы управления базами данных»

**Практическое занятие № 6**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИКБО-11-22, Гришин Андрей Валерьевич* | (подпись) | |
| Преподаватель | *Красников Степан Альбертович* | (подпись) | |
| Отчет представлен | «\_\_\_»\_\_\_\_\_\_\_\_202\_\_г. | |  | |

Москва 2025 г.

**ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

В динамично растущих системах, объемы данных, как правило, быстро увеличиваются и рано или поздно можно столкнуться с проблемой, когда текущих ресурсов машины будет не хватать для нормальной работы.  
    Для решения этой проблемы применяют горизонтальное и вертикальное масштабирование. Шардирование является частным случаем горизонтального масштабирования. Суть его в разделении базы данных на отдельные части по определенному правилу так, чтобы каждую из них можно было вынести на отдельный сервер. Однако реплицирование данных является обязательным требованием для шардирования в Mongo. Сервер конфигурации и каждый шард представляют из себя так называемый Реплика-сет (группа экземпляров, хранящих одинаковые наборы данных).

Преимущества MongoDB перед Реляционными СУБД:

* Отсутствие четкой структуры − MongoDB — это база данных документов, в которой коллекции могут содержать разные документы. Количество полей, содержание и размер документа могут отличаться от одного документа к другому;
* Никаких сложных соединений;
* MongoDB поддерживает динамические запросы к документам с использованием языка запросов на основе документов, который почти такой же мощный, как SQL;
* Простота масштабирования;
* Преобразование/сопоставление объектов приложения с объектами базы данных не требуется;
* Использует внутреннюю память для хранения рабочего набора, обеспечивая более быстрый доступ к данным.

Зачем использовать MongoDB?

* Данные хранятся в виде документов в стиле JSON;
* Индексирование по любому атрибуту;
* Репликация и высокая доступность;
* Автоматическое разделение на фрагменты.

Где использовать MongoDB?

* Большие данные;
* Мобильная и социальная инфраструктура;
* Управление пользовательскими данными;
* Концентратор данных.

**ПРАКТИЧЕСКАЯ ЧАСТЬ:**

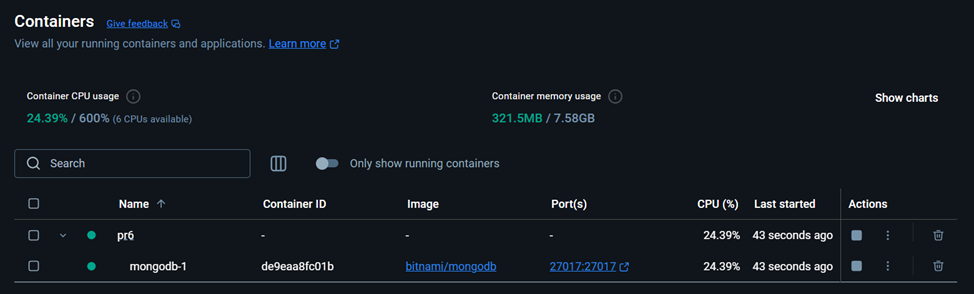


Рисунок 1 – Создание экземпляра MongoDB

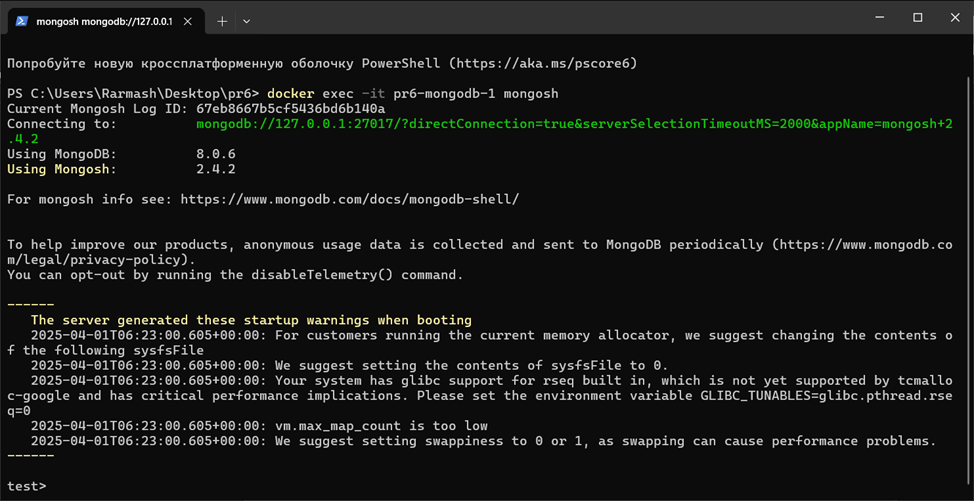


Рисунок 2 – Подключение к экземпляру MongoDB

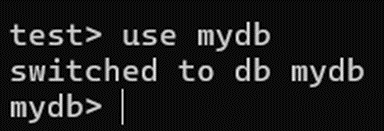


Рисунок 3 – Создание базы данных

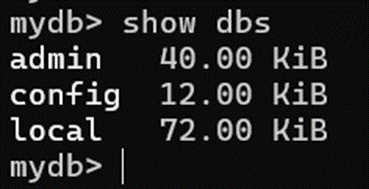


Рисунок 4 – Вывод непустых баз данных



Рисунок 5 – Заполнение базы данных

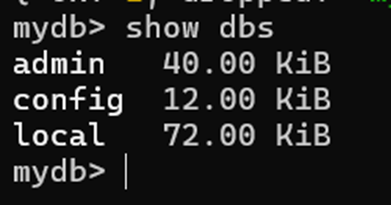


Рисунок 6 – Список доступных баз данных



Рисунок 7 – Удаление базы данных DATABASE\_STUDENT

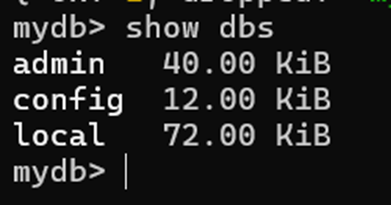


Рисунок 8 – Проверка списка доступных баз данных

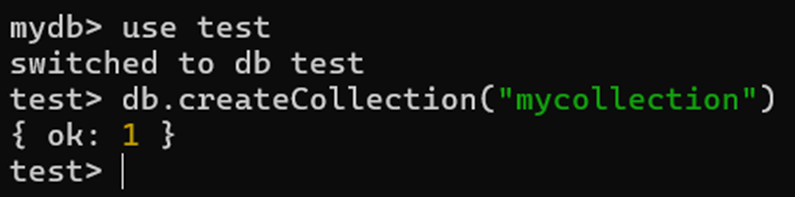


Рисунок 9 – Создание коллекции

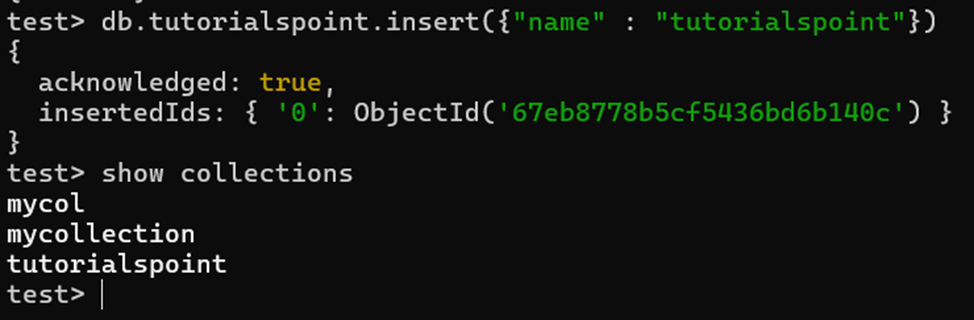


Рисунок 10 – Заполнение текущей коллекции

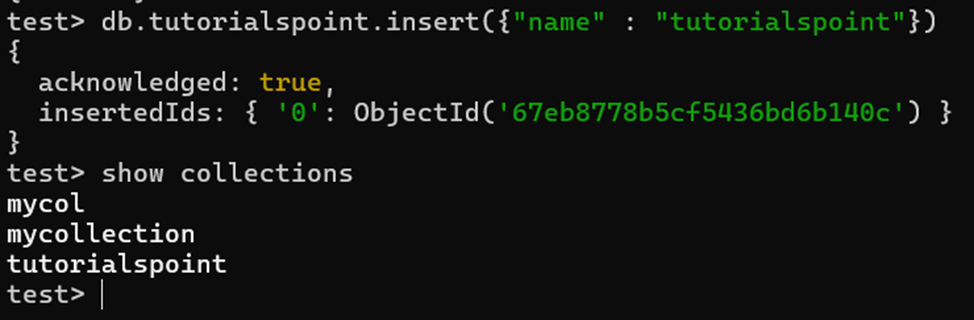
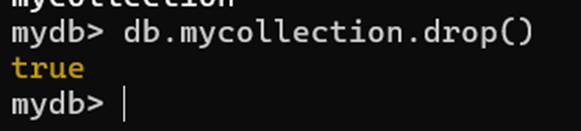


Рисунок 11 – Проверка списка доступных коллекций



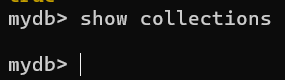


Рисунок 12 – Удаление и проверка коллекций в базе данных



Рисунок 13 – Вставка документа в коллекцию



Рисунок 14 - Передача массива документов в команду insert()

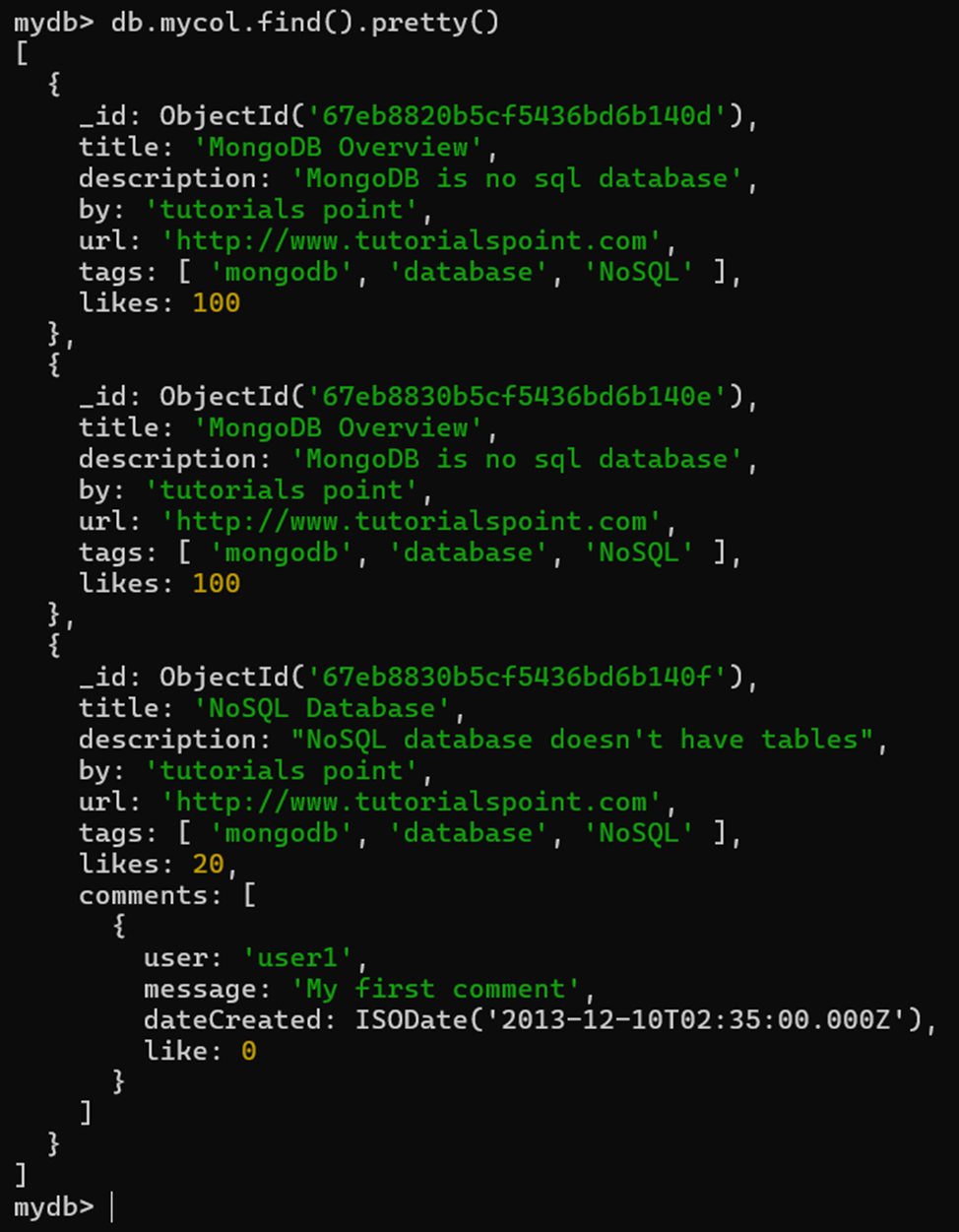


Рисунок 15 – Запрос данных из коллекции и отображение результата в отформатированном виде



Рисунок 16 – Запрос документа на основе условия И



Рисунок 17 – Запрос документа на основе условия ИЛИ

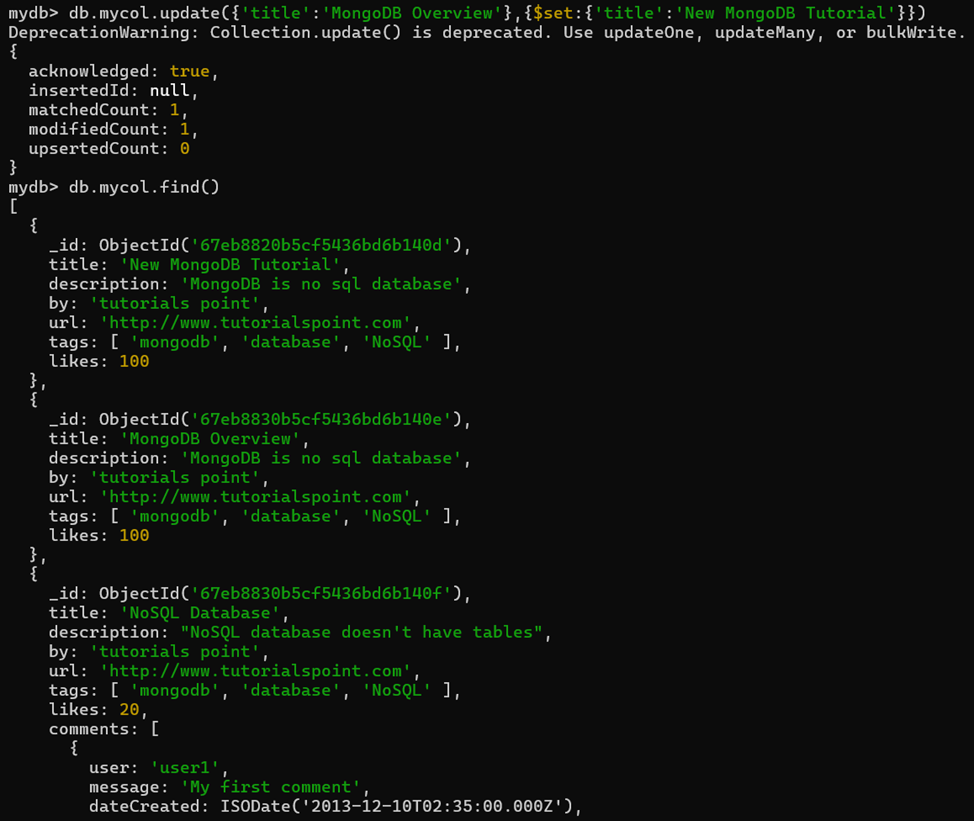


Рисунок 18 – Обновление данных в одном документе

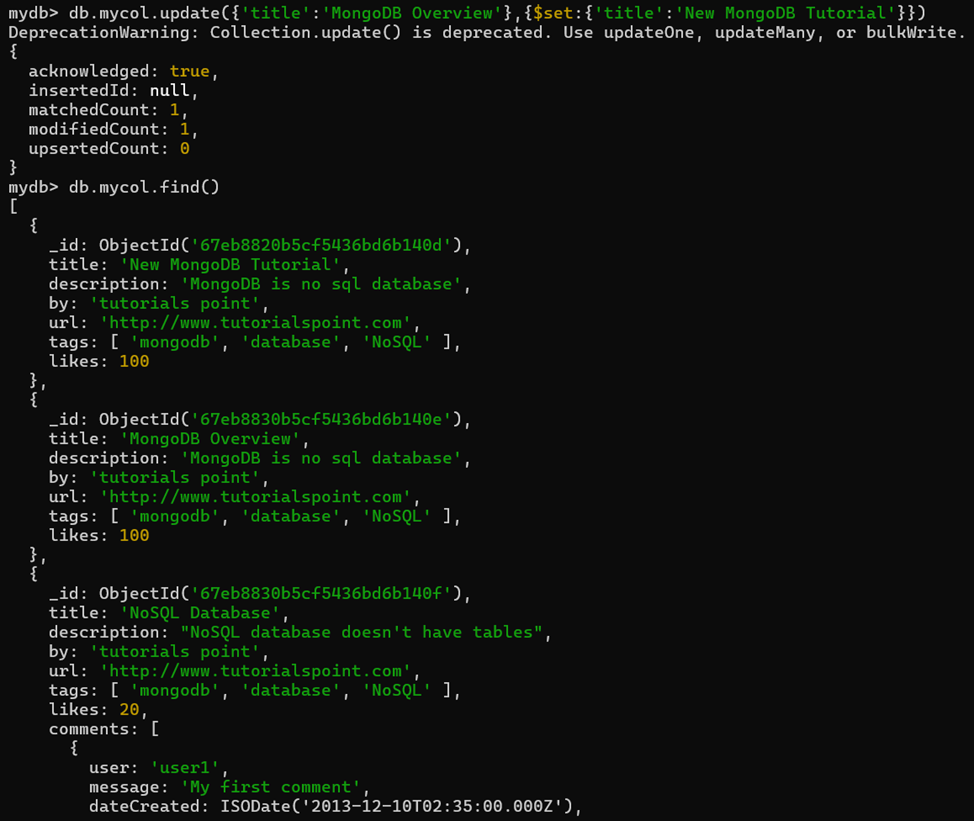


Рисунок 19 – Результат выполнения команды



Рисунок 20 – Обновление данных в нескольких документах

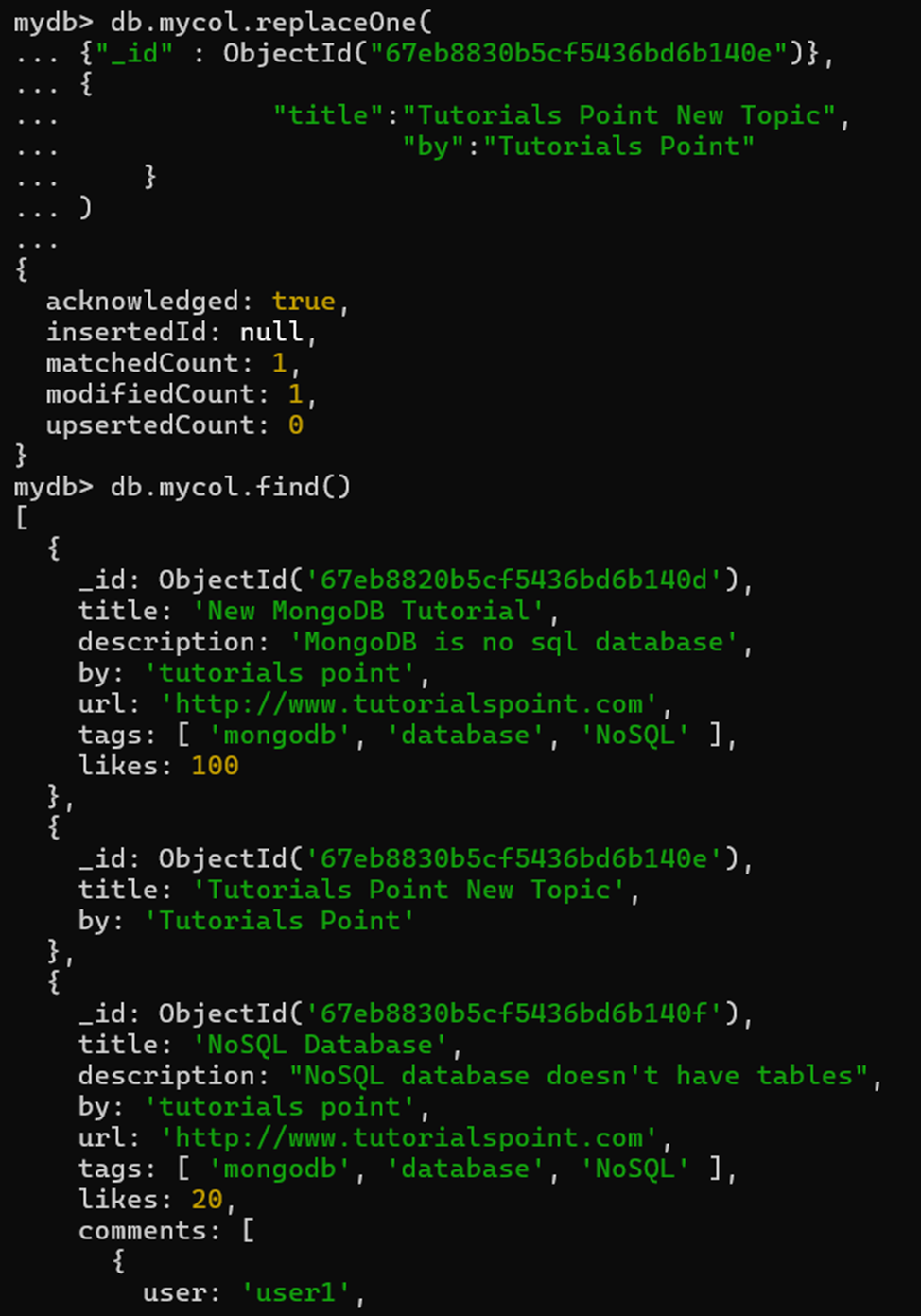


Рисунок 21 – Замена существующего документа новым

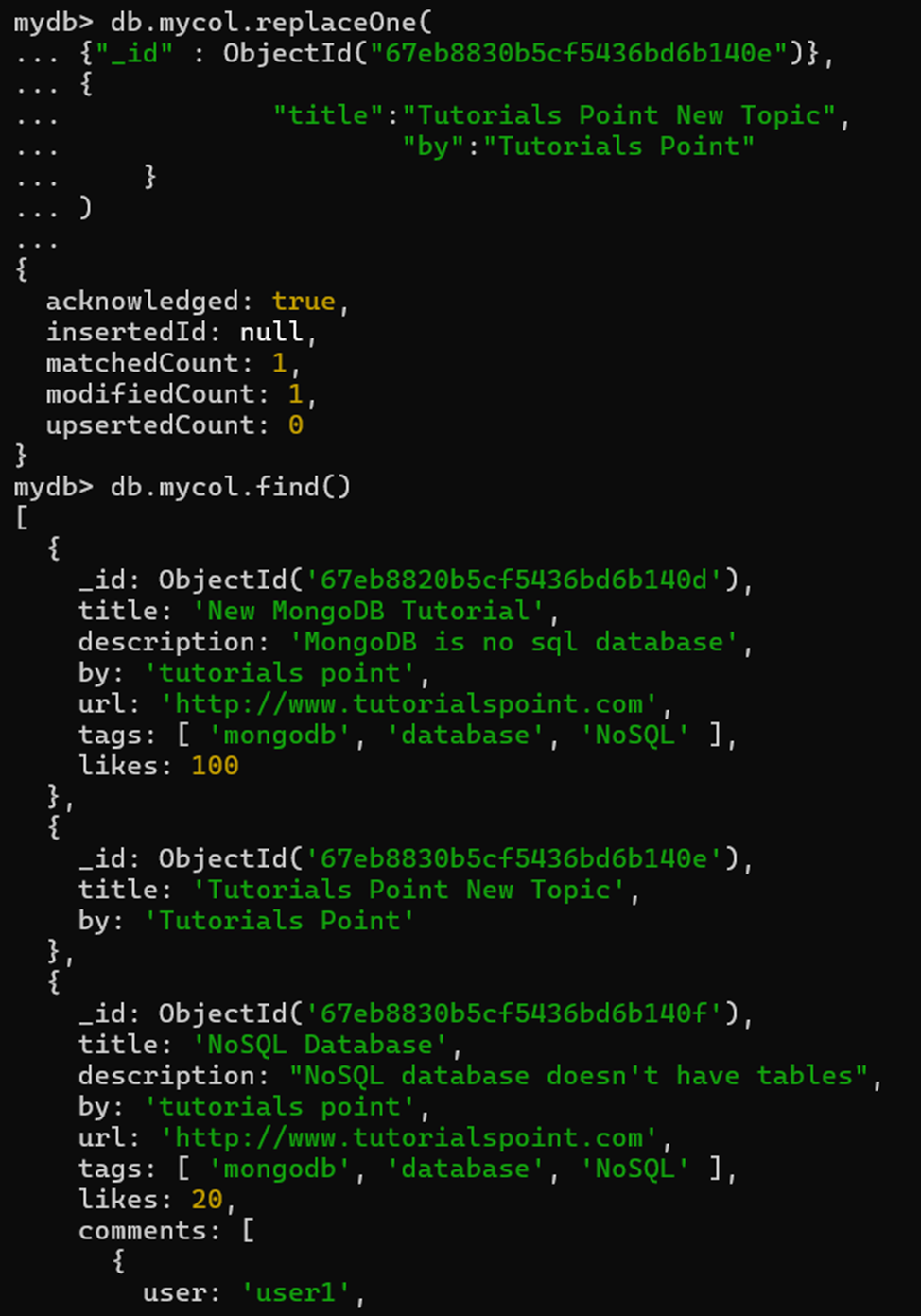


Рисунок 22 – Результат выполнения команды



Рисунок 23 – Удаление одного документа

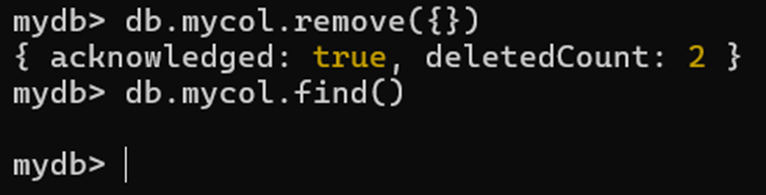


Рисунок 24 – Удаление всех данных из коллекции

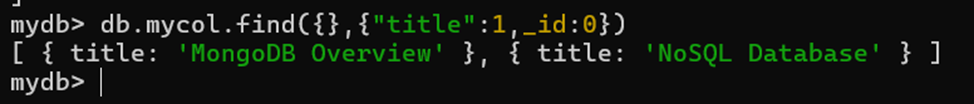


Рисунок 25 – Проекция данных из коллекции

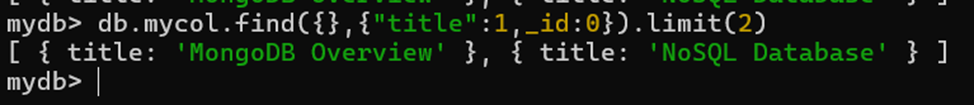


Рисунок 26 – Ограничение количества записей для вывода

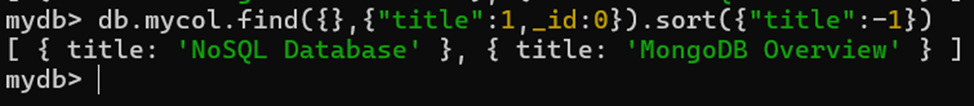


Рисунок 27 – Cортировка записей коллекции

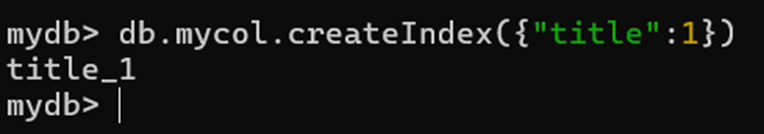


Рисунок 28 – Создание индекса по возрастанию

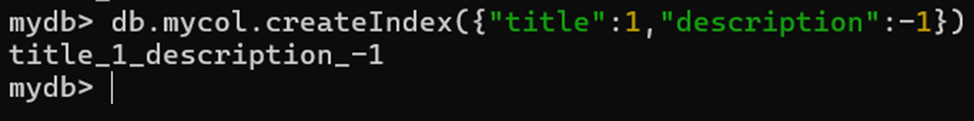


Рисунок 29 – Создание индекса для нескольких полей

**Заключение:**

В ходе выполнения практических заданий была изучена работа с одной из наиболее популярных NoSQL-баз данных – MongoDB. Практика охватывала основные аспекты работы с этой СУБД, включая установку, базовые операции с данными и их оптимизацию.

Были выполнены следующие шаги:

1. **Установка и настройка** – развертывание MongoDB и подготовка окружения.
2. **Запуск и подключение** – запуск сервера и работа с базой данных через консоль или клиент.
3. **Создание и удаление баз данных** – изучение команд use <dbName> и db.dropDatabase().
4. **Работа с коллекциями** – создание (db.createCollection()) и удаление (db.collection.drop()).
5. **Операции с документами**:
   * Добавление (insertOne, insertMany),
   * Чтение (find()),
   * Фильтрация с использованием операторов ($eq, $gt, $lt, $in и др.),
   * Обновление (updateOne, updateMany),
   * Удаление (deleteOne, deleteMany).
6. **Оптимизация выборок**:
   * Использование проекции (find({}, {fieldName: 1})) для выборки только необходимых полей,
   * Ограничение (limit()) и сортировка (sort()).
7. **Индексация** – создание индексов (createIndex()) для ускорения запросов.

Выполненная работа позволила глубже разобраться в принципах работы MongoDB, её возможностях по обработке больших объемов данных и преимуществах перед реляционными базами. Освоенные инструменты создают фундамент для изучения более сложных тем, таких как кластеризация, репликация и шардирование.