**Aldongar Yerkin and Mendiyarov Dias**

**SE 2318**

**Laboratory work 3. Data Exchange Formats in MongoDB. Data Modeling.**

1. **JSON** — текстовый формат обмена данными, используемый для передачи данных между клиентом и сервером или различными компонентами системы. Основные структуры: значение, массив.

Пример:  
{

"name": "John",

"age": 30,

"isStudent": false,

"courses": ["math", "history", "chemistry"],

"address": {

"street": "123 Main St",

"city": "New York",

"state": "NY"

}

}

**BSON** (Binary JSON) — бинарная версия JSON, оптимизированная для хранения и сериализации данных. Поддерживает дополнительные типы данных: бинарные данные, дату/время и другие. BSON используется в MongoDB для эффективного хранения и обработки данных.

1. **JSON** **основан на таких структурах основан как:**

Коллекция ключ-значение (объект).

Упорядоченная коллекция значений (массив).

1. **Форматы представления данных используемые в JSON:**

* Строки (string),
* числа (number),
* булевые значения (boolean),
* массивы (array),
* объекты (object)
* null.

1. Да, можно ссылаться на документы из одной коллекции MongoDB в другую. Для этого используются ссылки (References), которые реализуются через поле \_id и $lookup для выполнения операций агрегации.
2. Да, MongoDB поддерживает хранение массивов документов.
3. Да, вложенные документы активно используются для моделирования отношений "один ко многим" и "многие ко многим".

**Основы моделирования данных в MongoDB**

1. **Вложенные документы (Embedded Documents):**
   * Используются для представления отношений "1 to Many".
   * Удобны для быстрого доступа к связанным данным.
2. **Ссылки (References):**
   * Используются для связи между коллекциями.
   * Подходят для отношений "Many to Many" и работы с большими данными.

**Workload**

Нагрузка для системы "Форум об учебных заведениях" распределяется следующим образом:

* **Чтение постов (READ):**
  + Самая частая операция, выполняется примерно в 70% случаев.
  + Пользователи регулярно просматривают посты, связанные с определенным университетом, комментарии и количество лайков.
* **Создание новых данных (CREATE):**
  + Создание постов, комментариев, а также регистрация пользователей.
  + Около 20% от общего числа операций.
* **Обновление данных (UPDATE):**
  + Операции по добавлению лайков, редактированию постов или комментариев.
  + Составляет около 8% операций.
* **Удаление данных (DELETE):**
  + Удаление постов и связанных с ними комментариев.
  + Примерно 2% операций.

**Преимущества и недостатки выбранных моделей**

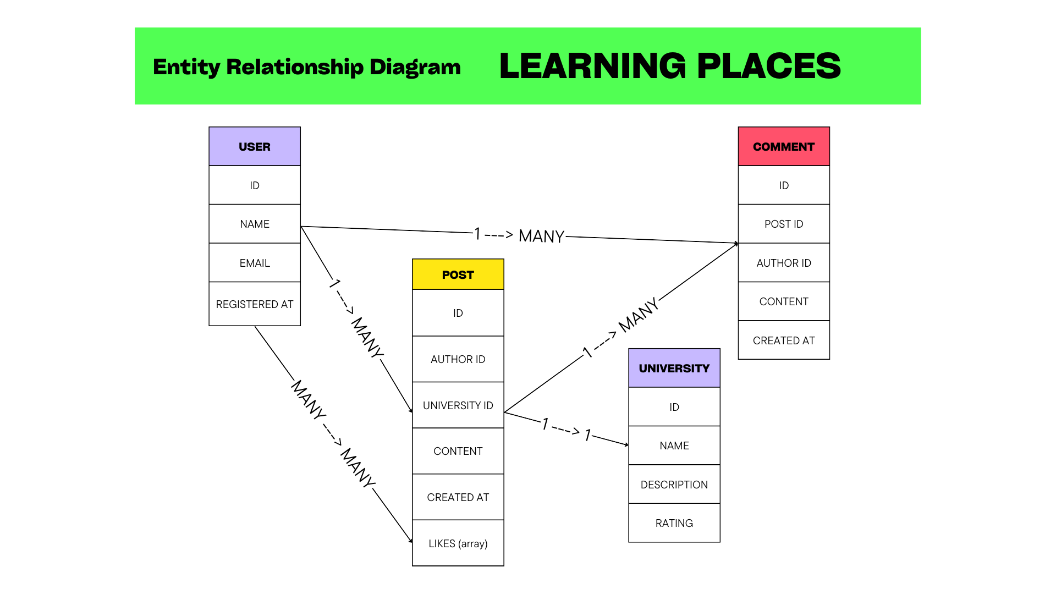
**Вложенные документы (Embedded Documents):**

* **Преимущества:**
  + Все данные хранятся в одном документе.
* **Недостатки:**
  + Ограничение по размеру документа (16 МБ в MongoDB).
  + Сложность обновления данных внутри вложенных структур.

**Ссылки (References):**

* **Преимущества:**
  + Гибкость структуры данных, подход для больших объемов информации.
  + Упрощение работы с большими вложенными наборами, так как данные разделены между коллекциями.
* **Недостатки:**
  + Увеличение времени выполнения запросов из-за необходимости использовать $lookup.
  + Более сложная структура запросов для получения связанных данных.

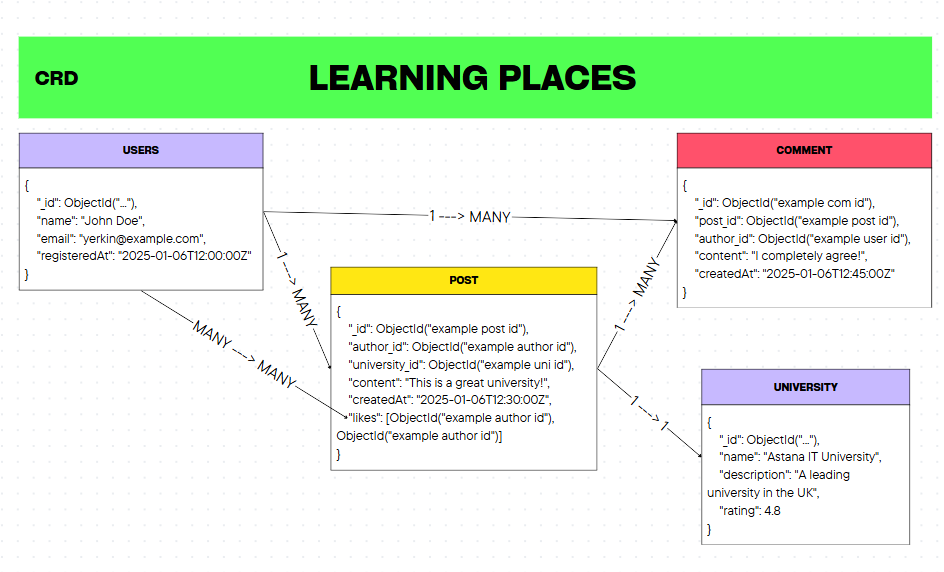
**ERD DIAGRAM**



**Связи между сущностями:**

* **User и Post:** Один пользователь может создавать много постов (1 → N).
* **Post и University:** Каждый пост связан с одним университетом (1 → 1).
* **Post и Comment:** Один пост может иметь множество комментариев (1 → N).
* **User и Comment:** Один пользователь может создавать множество комментариев (1 → N).
* **User и Like:** Много пользователей могут ставить лайки на много постов (M → N).

**CRD DIAGRAM**



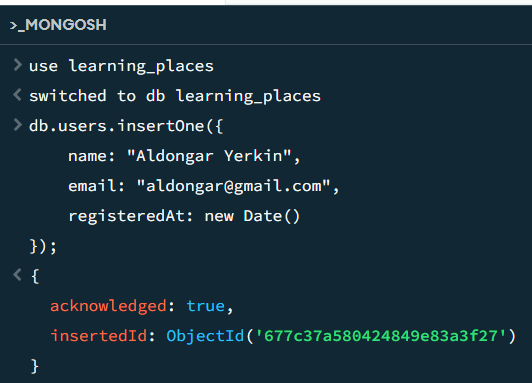
**Связи между коллекциями:**

* **users** ↔ **posts:** Связь через author\_id.
* **posts** ↔ **universities:** Связь через university\_id.
* **posts** ↔ **comments:** Связь через post\_id.
* **users** ↔ **comments:** Связь через author\_id.

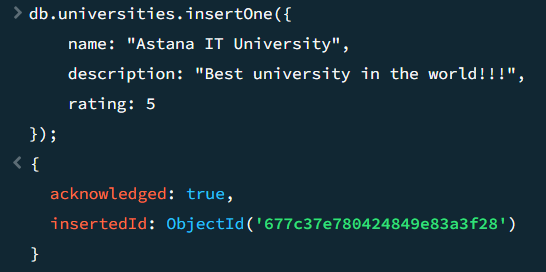
**Реализация запросов в MongoDB**

**1. CREATE (Создание данных)**

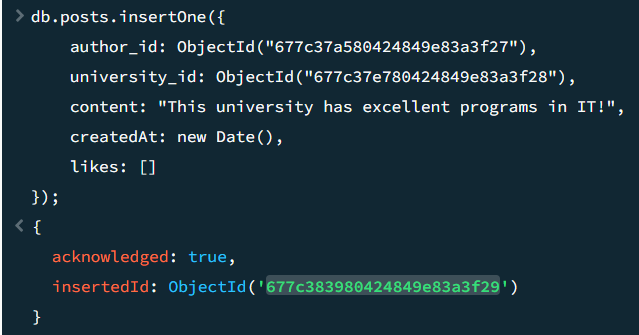
Добавить пользователя:



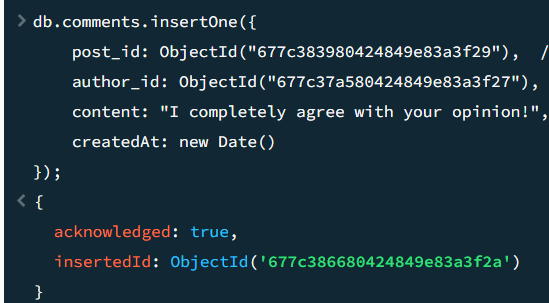
Добавить университет:



Добавить пост:

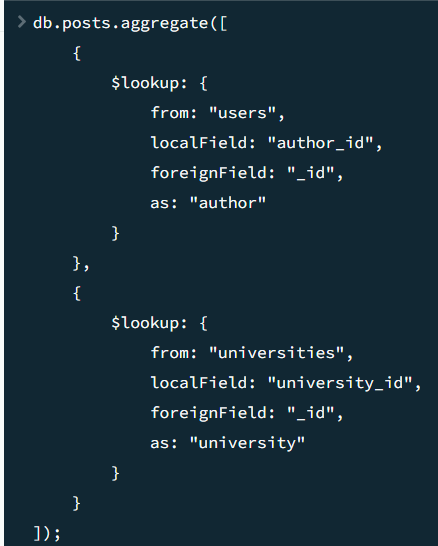


Добавить комментарий:

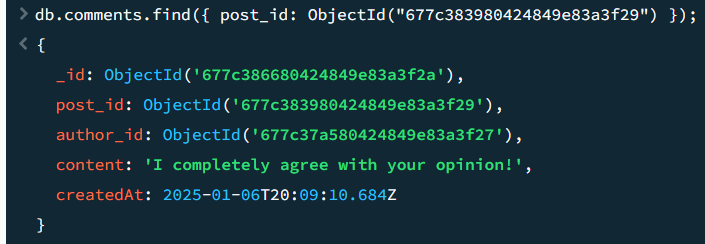


**2. READ (Чтение данных)**

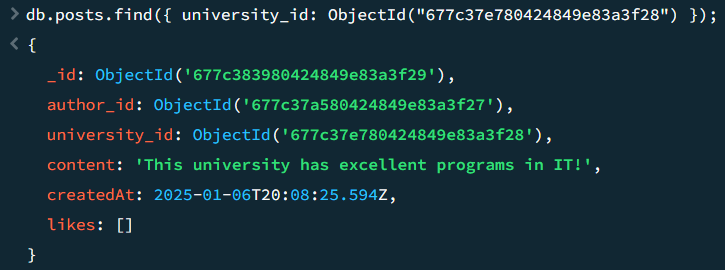
Получить все посты с их авторами и университетами:



Получить все комментарии к конкретному посту:

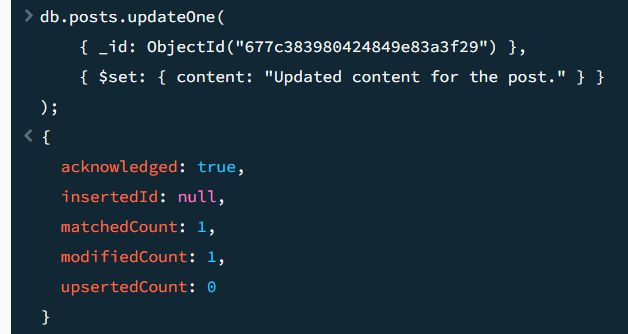


Получить все посты по конкретному университету:

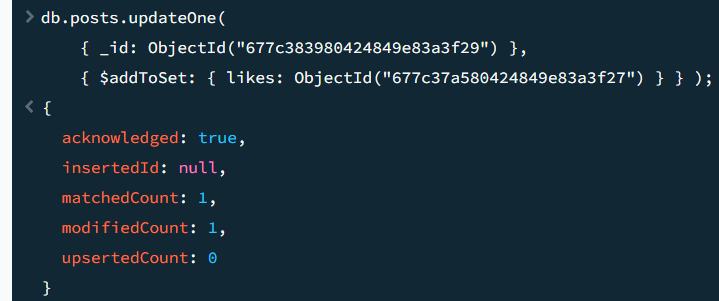


**3. UPDATE (Обновление данных)**

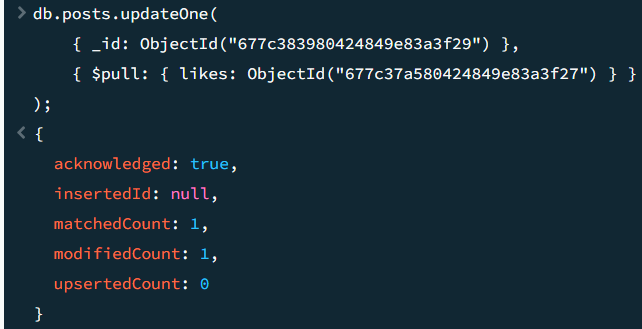
Обновить текст поста:



Добавить лайк к посту:

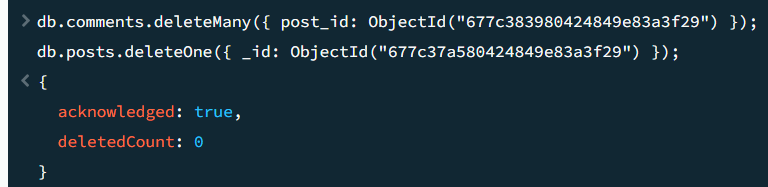


Убрать лайк с поста:



**4. DELETE (Удаление данных)**

**Удалить пост и все связанные с ним комментарии:**



Удалить пользователя



**5. Агрегация и связи ($lookup)**

**Получить посты с комментариями и информацией об авторе:**



Получить информацию о пользователях, которые поставили лайк на пост:



**Заключение**

В ходе выполнения лабораторной работы были изучены форматы данных JSON и BSON, а также особенности моделирования данных в MongoDB. Были построены диаграммы ERD и CRD, описаны отношения между данными, их преимущества и недостатки. Также выполнены запросы CRUD и агрегации. Работа показала, что MongoDB является мощным инструментом для работы с данными, предоставляющим гибкость и высокую производительность.