Продвинутая работа с MongoDB и Python

Введение в индексы

Индексы — это специальные структуры данных, которые хранятся на диске, и которые значительно ускоряют выполнение запросов в базе данных. Без индексов, MongoDB должен сканировать всю коллекцию для выполнения запроса, что может быть очень медленным, особенно для больших коллекций. Индексы позволяют MongoDB быстро находить документы, соответствующие запросу, за счет хранения упорядоченной копии определенных полей.

Создание индекса

Создадим индекс на поле name в коллекции students. Индекс по этому полю позволит быстро искать студентов по их имени.

```
index_name = collection.create_index("name")
print(f"Created index: {index_name}")
```

Этот код создаст индекс, который автоматически будет сортировать данные по полю name. Индексы могут быть созданы как в порядке возрастания (по умолчанию, с помощью 1), так и в порядке убывания (с помощью -1).

Просмотр созданных индексов

После создания индексов, вы можете захотеть просмотреть, какие индексы уже созданы для коллекции. Это можно сделать с помощью метода index_information():

```
print("Indexes on collection:", collection.index_information())
```

Составные индексы

MongoDB также позволяет создавать составные индексы, которые включают в себя несколько полей. Например, мы можем создать индекс по name и age, который будет полезен, если нам часто нужно искать студентов по имени и возрасту одновременно:

composite_index_name = collection.create_index([("name", 1), ("age", -1)])

print(f"Created composite index: {composite_index_name}")

Этот индекс сначала будет сортировать данные по `name` в порядке возрастания, а затем по `age` в порядке убывания.

Удаление индексов

Если индекс больше не нужен, его можно удалить с помощью метода drop_index(). Допустим, мы хотим удалить созданный нами индекс на поле name:

collection.drop_index(index_name)

print(f"Dropped index: {index_name}")

Индексы нужны, когда:

- Есть большая коллекция данных.
- Часто нужно выполнять запросы по определенным полям (например, поиск студентов по имени или возрасту).
- Нужно ускорить поиск, сортировку или фильтрацию данных.

Агрегирование данных в MongoDB

Агрегирование в MongoDB используется для обработки и анализа данных. Это мощный инструмент, который позволяет выполнять такие задачи, как подсчет, фильтрация, сортировка, и группировка данных, аналогично SQL-запросам с GROUP BY, SUM, COUNT, AVG и другими операциями.

Основной компонент для агрегирования в MongoDB — это pipeline (конвейер), который представляет собой последовательность этапов (стадий), где каждый этап обрабатывает данные и передает их на следующий этап.

Основные операторы сравнения в MongoDB:

- $gt (greater than) — больше. Выбирает документы, где значение поля больше указанного.
- $gte (greater than or equal) — больше или равно. Выбирает документы, где значение поля больше или равно указанному.
- $lt (less than) — меньше. Выбирает документы, где значение поля меньше указанного.
- $lte (less than or equal) — меньше или равно. Выбирает документы, где значение поля меньше или равно указанному.
- $eq (equal) — равно. Выбирает документы, где значение поля равно указанному.
- $ne (not equal) — не равно. Выбирает документы, где значение поля не равно указанному.

Задача 1: Вставка и обновление данных в MongoDB

Создайте базу данных library и коллекцию books. Вставьте несколько документов с информацией о книгах, включая название, автора и год издания. Затем обновите год издания для всех книг, написанных определенным автором.

Задача 2: Агрегирование данных в MongoDB

Создайте базу данных company и коллекцию employees. Вставьте данные о сотрудниках, включая их имя, отдел и зарплату. Используйте агрегирование для подсчета общей суммы зарплаты по каждому отделу.

Задача 3: Использование индексов в MongoDB

Создайте базу данных fitness-club и коллекцию trainers. Вставьте данные о тренерах, включая их имя, стаж и тип тренировки. Создайте индекс на поле стаж. Затем выполните поиск тренеров по стажу.

Задача 4: Валидация данных при вставке в MongoDB

Создайте базу данных university и коллекцию courses. Вставьте несколько документов с информацией о курсах, включая название, преподавателя и количество кредитов (credits). Убедитесь, что при вставке документов количество кредитов всегда больше 0. Для этого примените соответствующую валидацию данных при вставке.