

Домашнее задание 04: Проектирование и работа с MongoDB

Цель работы: Получить практические навыки работы с MongoDB, проектирования документной модели данных, выполнения CRUD операций и валидации схем.

Задание

Для своего варианта задания выполните следующие задачи:

1. Проектирование документной модели

- Изучите выбранный вариант задания
- Спроектируйте структуру коллекций MongoDB для вашей системы
- Определите, какие сущности будут храниться в отдельных коллекциях
- Определите структуру документов для каждой коллекции
- Решите, где использовать embedded documents, а где references
- Обоснуйте выбор между embedded и references для каждой связи

2. Создание базы данных и коллекций

- Создайте MongoDB базу данных (использовать Docker)
- Создайте все необходимые коллекции
- Добавьте тестовые данные (минимум 10 документов в каждой коллекции)
- Используйте различные типы данных MongoDB (String, Number, Date, Array, Object и т.д.)

3. Реализация CRUD операций

- Напишите MongoDB запросы (или код на выбранном языке) для всех операций из вашего варианта задания:
 - Create (вставка документов)
 - Read (поиск документов с различными условиями)
 - Update (обновление документов)
 - Delete (удаление документов)
- Используйте различные операторы: \$eq, \$ne, \$gt, \$lt, \$in, \$and, \$or и т.д.
- Используйте операции с массивами: \$push, \$pull, \$addToSet

4. Валидация схем

- Создайте валидацию схем для минимум одной коллекции используя ``$jsonSchema``
- Определите обязательные поля (required)
- Определите типы данных для полей
- Определите дополнительные ограничения (например, min/max для чисел, pattern для строк)
- Протестируйте валидацию, попытавшись вставить невалидные данные

5. Агрегации (опционально)

- Создайте минимум один aggregation pipeline для сложного запроса
- Используйте стадии: \$match, \$group, \$project, \$sort и т.д.

6. Подключение базы данных MongoDB к API

- Реализуйте API для одной или нескольких сущностей из второго домашнего задания для работы с MongoDB

Результат

Результат должен быть оформлен в виде следующих файлов, размещенных в вашем GitHub репозитории:

- ``schema_design.md`` – описание проектирования документной модели с обоснованием выбора embedded/references
- ``data.js`` или ``data.json`` – скрипт или файл с тестовыми данными
- ``queries.js`` или ``queries.md`` – MongoDB запросы для всех операций
- ``validation.js`` – скрипт с валидацией схем
- ``README.md`` – описание проекта, инструкции по запуску MongoDB и выполнению запросов
- Dockerfile и docker-compose.yml для запуска API и MongoDB (расширьте предыдущую работу)

Критерии оценки:

- Корректность проектирования документной модели
- Обоснованность выбора между embedded и references
- Качество MongoDB запросов
- Правильность валидации схем
- Качество документации