

Домашнее задание 03 : Проектирование и оптимизация реляционной базы данных

Цель работы: Получить практические навыки работы с PostgreSQL, проектирования схемы БД, создания индексов и оптимизации запросов.

Задание

Для своего варианта выполните следующие задачи:

1. Проектирование схемы базы данных

- Изучите выбранный вариант задания
- Спроектируйте схему реляционной БД для вашей системы
- Определите таблицы для всех сущностей из вашего варианта
- Определите первичные ключи (PK) для каждой таблицы
- Определите внешние ключи (FK) для связей между таблицами
- Определите типы данных для всех колонок
- Определите ограничения (constraints): NOT NULL, UNIQUE, CHECK

2. Создание базы данных

- Создайте PostgreSQL базу данных (можно использовать Docker)
- Создайте все таблицы с помощью SQL CREATE TABLE
- Добавьте тестовые данные (минимум 10 записей в каждой таблице)

3. Создание индексов

- Проанализируйте, какие запросы будут выполняться часто
- Создайте индексы для:
 - Первичных ключей (автоматически создаются)
 - Внешних ключей
 - Колонок, используемых в WHERE условиях
 - Колонок, используемых в JOIN
- Объясните, для чего нужен каждый индекс

4. Оптимизация запросов

- Напишите SQL запросы для всех операций из вашего варианта задания
- Используйте EXPLAIN для анализа плана выполнения запросов
- Оптимизируйте минимум 2 запроса (добавьте индексы, перепишите запрос)
- Сравните планы выполнения до и после оптимизации

5. Партиционирование (опционально)

- Если в вашей системе есть таблицы с большим количеством данных, которые можно разделить по времени или другому критерию, спроектируйте партиционирование
- Опишите стратегию партиционирования

6. Подключение API к базе данных

- Подключите API, реализованные в прошлой домашней работе к созданной базе данных

Результат

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

- `schema.sql` – SQL скрипт для создания схемы БД (CREATE TABLE, индексы)
- `data.sql` – SQL скрипт для вставки тестовых данных
- `queries.sql` – SQL запросы для всех операций из варианта задания
- `optimization.md` – описание оптимизаций с планами выполнения (EXPLAIN)
- `README.md` – описание схемы БД, инструкции по созданию и запуску
- Dockerfile и docker-compose.yaml для запуска API и PostgreSQL

****Критерии оценки:****

- Корректность проектирования схемы БД
- Правильность использования типов данных и ограничений
- Обоснованность создания индексов
- Качество SQL запросов
- Эффективность оптимизаций
- Качество документации