



Прочитай и проанализируй файл. Это ТЗ к курсовой работе по Базам Данных. Придумай тему для данной курсовой, при этом не банальную, чтобы я мог спокойно написать данный курсач не на сильно огромный код. Плюс напиши краткое описание курсовой

Проанализировав ТЗ курсовой работы по Базам Данных, предлагаю тему, которая будет интересной, не банальной и при этом не потребует огромного кодирования.^[1]

Предлагаемая тема

"Система управления соревнованиями по спортивному программированию (Competitive Programming Contest Management System)"

Описание курсовой

Информационная система для организации и проведения олимпиад по программированию, которая автоматизирует процесс регистрации участников, управления задачами, приёма решений и формирования турнирных таблиц.^[1]

Основные возможности

Система включает управление пользователями (участники, жюри, администраторы), контестами и их статусами, задачами с тестовыми данными, посылками решений с вердиктами (Accepted, Wrong Answer, Time Limit), и автоматический расчёт рейтингов и статистики.^[1]

Структура базы данных (9-10 таблиц)

- **Users** — пользователи с ролями и рейтингами
- **Contests** — соревнования с типами (ACM ICPC, Codeforces)
- **Problems** — задачи с лимитами времени/памяти
- **Testcases** — тестовые данные для задач
- **Submissions** — посылки решений от участников
- **Verdicts** — результаты проверки посылок
- **Standings** — турнирные таблицы

- **Tags** — теги для задач (DP, графы, строки)
- **ProblemTags** — связь многие-ко-многим
- **AuditLog** — журнал всех изменений

Функциональные преимущества

Триггеры автоматически пересчитывают рейтинг пользователя при добавлении вердикта и обновляют турнирную таблицу. Скалярные функции вычисляют штрафное время по правилам ACM, а табличные генерируют статистику по задачам и пользователям. Представления показывают топ решателей, сложность задач и активность контестов.^[1]

Почему это удобно

Тема близка твоему опыту в алгоритмическом программировании, структура БД логична и естественна, объём данных генерируется просто (5000+ посылок для разных контестов), а API получается компактным но функциональным. Backend можно реализовать на Go или Python с PostgreSQL, что позволит выполнить все требования ТЗ без избыточной сложности.^[1]

**

1. Kursovaia_rabota_BD_2025.pdf