МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Самарский национальный исследовательский университет

имени академика С. П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Факультет информатики

Кафедра программных систем

ОТЧЁТ

по лабораторной работе № 3

«Создание таблиц и ограничений целостности, использование команд SQL для работы с данными»

по курсу «Распределенные базы данных»

Выполнил:

Лазарев М. Ю.

гр. 6132-020402D

Проверил:

Додонов М. В.

Самара 2025

Лабораторная работа № 3.

Ход выполнения лабораторной работы.

По теме магистерской работы «Разработка web-приложения по решению споров».

1. Строим модель данных в drawsql.app (Рис. 1)

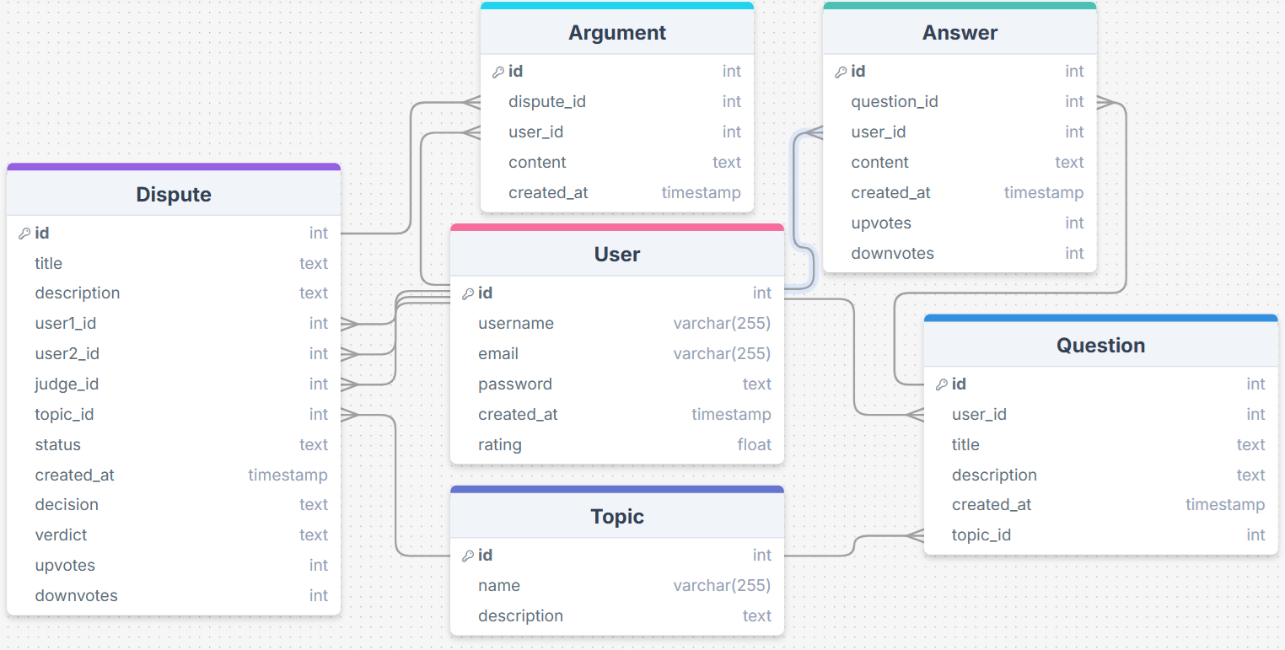


Рисунок 1 – ER-модель базы данных системы

1. Вводим SQL-запросы для создания таблиц с ключами и ограничениями (Рис. 2). Проверяем диаграмму БД (Рис. 3).

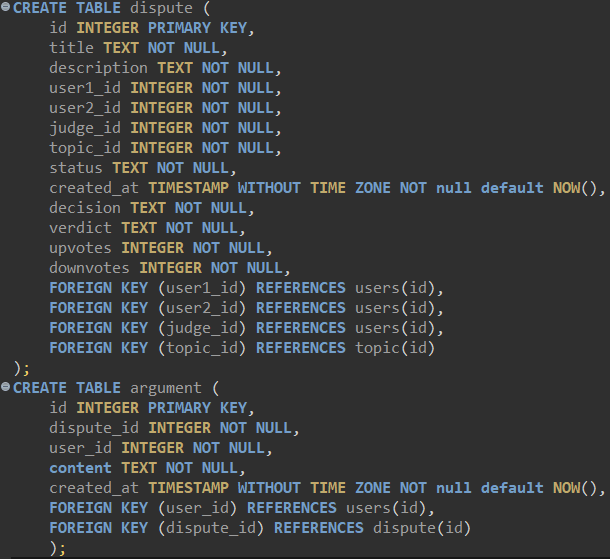
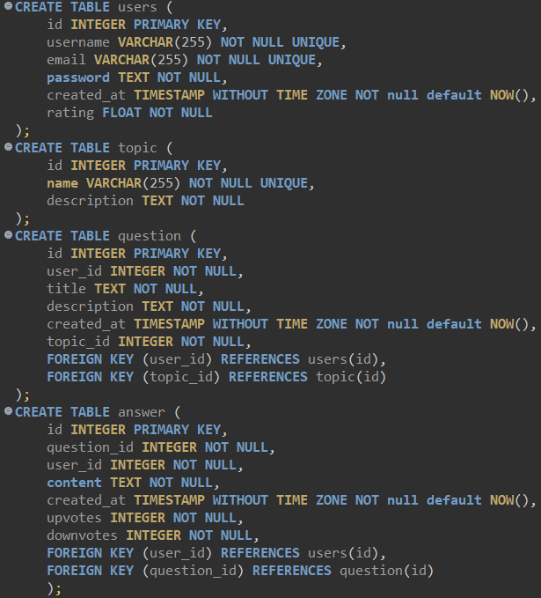


Рисунок 2 – SQL-скрипт для создания таблиц

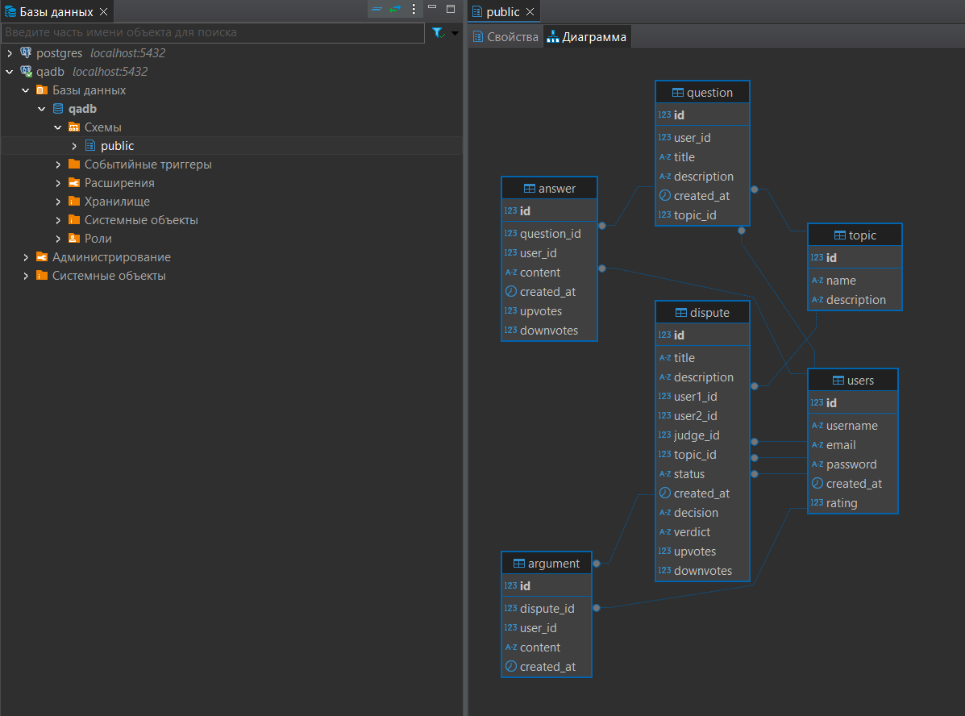


Рисунок 3 – Реализованные таблицы в БД

1. БД должна обеспечивать регистрацию и управление пользователями. Система должна поддерживать создание, редактирование и удаление вопросов, ответов и споров. Споры должны включать участников, судью и аргументы, с возможностью вынесения решения. Также необходимо управление темами, фильтрация и получение связанных данных.
2. Внесем тестовые значения в таблицы (Рис. 4). В данном случае произведены SQL-запросы с добавлением пользователей, темы, вопросов и ответов к ним.

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4 – Тестовые данные

1. Создаем SQL-запросы:

* выбор всех вопросов пользователя с ID = 1;

SELECT \* FROM question WHERE user\_id = 1

* выбор всех ответов на конкретный вопрос с ID = 1;

SELECT \* FROM answer WHERE question\_id = 1;

* выбрать всех пользователей с их рейтингом, отсортированным по убыванию;

SELECT users.id, users.username, users.rating FROM user ORDER BY users.rating DESC;

* получить все вопросы и количество ответов на них;

SELECT question.id, question.title, COUNT(answer.id) FROM question LEFT JOIN answer ON question.id = answer.question\_id GROUP BY question.id;

* получить пользователей и их вопросы;

SELECT users.id, users.username, question.title FROM users INNER JOIN question ON users.id = question.user\_id;

* получить все вопросы, у которых более 3-х ответов;

SELECT question.id, question.title, COUNT(answer.id) FROM question LEFT JOIN answer ON question.id = answer.question\_id GROUP BY question.id HAVING COUNT(answer.id) > 3;

* получить все вопросы и их авторов, даже если автора уже нет в базе;

SELECT question.id, question.title, user.username FROM question FULL OUTER JOIN user ON question.user\_id = users.id;

* найти пользователя с минимальным рейтингом;

SELECT users.id, users.username, users.rating FROM user ORDER BY users.rating ASC LIMIT 1;

* найти самый положительно оцененный ответ;

SELECT answer.id, answer.content, answer.upvotes FROM answer ORDER BY answer.upvotes DESC LIMIT 1;

* выбрать все вопросы, созданные в текущем месяце;

SELECT question.id, question.title, question.created\_at FROM question WHERE EXTRACT(MONTH FROM question.created\_at) = EXTRACT(MONTH FROM NOW());

* получить количество вопросов за последние 30 дней;

SELECT COUNT(question.id) FROM question WHERE question.created\_at >= NOW() - INTERVAL '30 days';

* вопросы, содержащие слово "сюжет";

SELECT \* FROM question WHERE title LIKE '%сюжет%';

* получить список всех ответов, упорядоченных по количеству голосов.

SELECT answer.content, answer.upvotes, answer.downvotes, users.usernameFROM answer JOIN users ON answer.user\_id = users.id ORDER BY answer.upvotes DESC, answer.downvotes ASC;