

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

**Институт №8 «Компьютерные науки и прикладная математика»
Кафедра №806 «Вычислительная математика и программирование»**

**Курсовой работа
по курсу «Базы данных»**

Разработка системы управления театром

**Выполнила: А.А.Серякова
Группа: 8О-309Б
Преподаватель: А.В Малахов**

Москва, 2024

Схема базы данных «театра»

Описание проекта:

Проект представляет собой систему для онлайн-платформы театра, которая предоставляет пользователям возможность просматривать спектакли, оставлять отзывы, ставить оценки и сохранять спектакли в избранное. В базе данных хранятся данные о спектаклях, жанрах, актерах, пользователях, а также их взаимодействиях с контентом (оценки, отзывы, добавление в избранное). Система поддерживает разделение ролей для пользователей, таких как "зритель" и "администратор", где администратор имеет права добавлять и удалять спектакли.

Проект использует PostgreSQL для управления данными и Streamlit для создания интерфейса. Система позволяет эффективно управлять данными о спектаклях.

Таблицы:

1. Performances

Хранит информацию о спектаклях, представленных на платформе. Включает данные, такие как название, год премьеры, описание, рейтинг, количество отзывов, жанр и возрастные ограничения. Эта таблица связана с жанрами, актерами и отзывами..

2. Users

Содержит информацию о зарегистрированных пользователях, включая их электронную почту, возраст, роль и хеш пароля. Также управляет правами доступа, что позволяет администраторам добавлять или удалять спектакли.

3. Genres

Перечень жанров, в которые классифицируются спектакли.

4. Reviews

Таблица, в которой пользователи могут оставлять текстовые отзывы о спектаклях. Каждый отзыв связан с пользователем и конкретным спектаклем, а также включает дату написания.

5. **Stars**

Хранит информацию о рейтинге спектаклей (оценки от 1 до 10). Включает ID спектакля, ID пользователя, количество выставленных оценок и среднюю оценку, что позволяет рассчитывать общую оценку спектакля.

6. **Actors**

Содержит данные об актерах, такие как имя и биография. Через связующую таблицу (actor_performances) пользователи могут узнавать, в каких спектаклях участвовал актёр.

7. **Liked_Performances**

Таблица, в которой пользователи могут сохранять спектакли в свой список избранного. Это позволяет платформе отображать пользователю его любимые спектакли.

8. **Actor_performances**

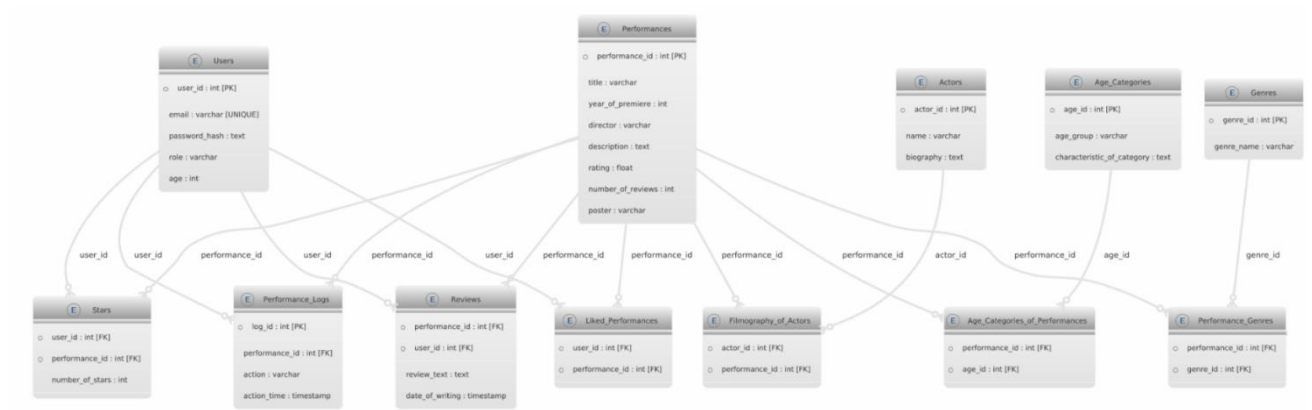
Связующая таблица для спектаклей и актёров, обеспечивающая возможность создавать список актёров для каждого спектакля.

9. **Performance_Genres**

Связующая таблица, которая связывает спектакли с жанрами. Поскольку у спектакля может быть несколько жанров, эта таблица помогает организовать классификацию спектаклей.

Эти таблицы поддерживают функциональность онлайн-театра, обеспечивая управление пользователями, классификацию спектаклей, отзывы и рейтинги.

Схема модели данных



Алгоритм работы приложения

Работа со спектаклями:

Пользователи могут просматривать список спектаклей, искать их по названию или фильтровать по жанрам, актерам и возрастным категориям. Для каждого спектакля показывается его описание, год премьеры, режиссер, рейтинг и количество отзывов.

Оценки и отзывы:

Зарегистрированные пользователи могут выставять оценки спектаклям, создавая записи в таблице Stars. Также пользователи могут оставлять текстовые отзывы, которые сохраняются в таблице Reviews.

Добавление спектаклей в избранное:

Пользователи могут добавлять спектакли в раздел "Избранное", что сохраняется в таблице Liked_Performances.

Рекомендации:

Система формирует рекомендации, основываясь на жанрах, оценках и возрастных категориях пользователей.

Роли пользователей:

Система поддерживает разные роли, такие как "зритель" (viewer) и "администратор" (admin), которые определяются в поле role таблицы Users. Администратор может добавлять или удалять спектакли.

Логирование действий:

Все действия по добавлению или удалению спектаклей в таблице Performances логируются в таблице Performance_Logs с использованием триггеров.

Классификация по возрастным категориям:

Спектакли распределяются по возрастным категориям через таблицу Age_Categories_of_Performances, что позволяет фильтровать контент для пользователей разного возраста.

Связь с актерами и жанрами:

Пользователи могут искать спектакли с любимыми актерами или в интересных жанрах через таблицы Filmography_of_Actors и Performance_Genres.

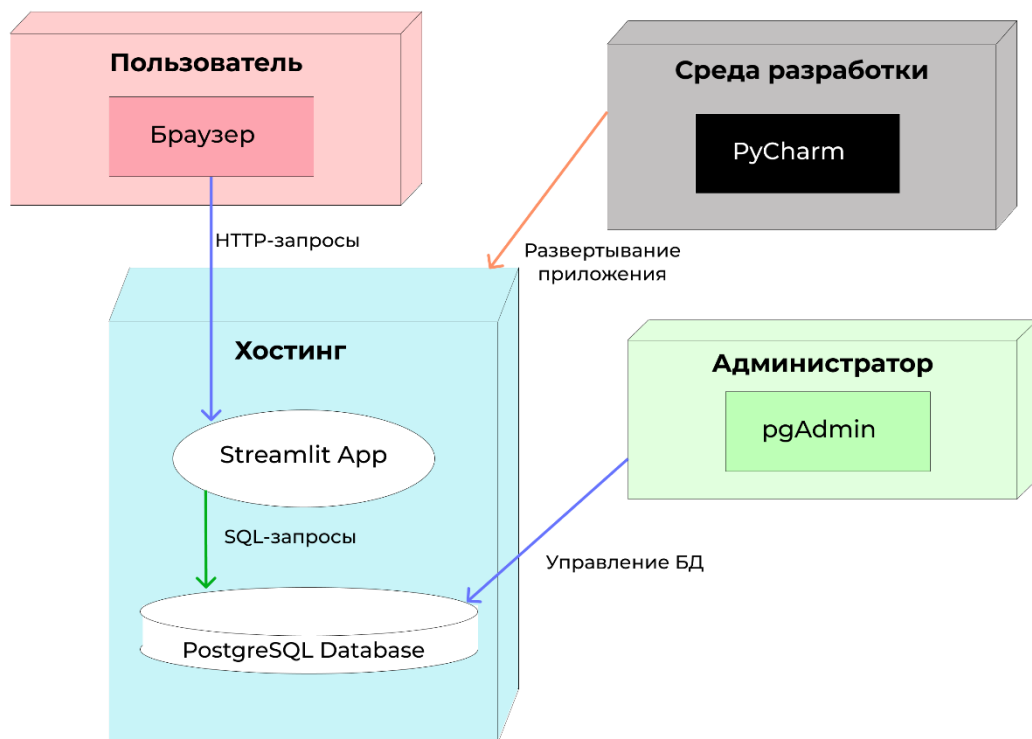
Технологии, использованные в проекте:

- **PostgreSQL** для управления базой данных.
- **Streamlit** для разработки пользовательского интерфейса.
- **PyCharm** как среда разработки.
- Дополнительные инструменты: **pgAdmin** для управления базой данных и **Python** для реализации функционала.

Пример работы системы:

1. Пользователь проходит регистрацию или использует уже существующие учетные данные, открывает интерфейс и просматривает список доступных спектаклей.
2. Он выбирает понравившийся спектакль, читает описание, добавляет его в избранное или оценивает, оставляя отзыв.
3. После добавления новой оценки обновляется общий рейтинг спектакля в базе данных.
4. Администратор может добавить новый спектакль через интерфейс или напрямую в базу данных. Все действия логируются.

Проект демонстрирует эффективное управление данными в реляционной базе данных, автоматизацию задач через триггеры и функции, а также взаимодействие различных компонентов системы. Пользователи могут удобно взаимодействовать с контентом, а администраторы — управлять базой данных.



QR-код на репозиторий GitHub:

