

**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 «онлайн-кинотеатром»**

**Выполнила: Е.В. Варнак**

**Группа: 8О-306Б**

**Преподаватель: А.В. Малахов**

**Москва, 2024**

## **Схема базы данных «онлайн-кинотеатра»**

Таблицы:

### **1. Movies**

Хранит информацию о фильмах, доступных на платформе. Включает такие данные, как название, год выпуска, описание, рейтинг, количество оценок, категория фильма и возрастные ограничения. Таблица связана с жанрами, актёрами и отзывами.

### **2. Users**

Содержит данные о пользователях, включая электронную почту, возраст, роль и хеш пароля. Также отвечает за разделение прав доступа, позволяя, например, администраторам добавлять и удалять фильмы.

### **3. Genres**

Содержит список жанров фильмов.

### **4. Reviews**

Таблица отзывов. Позволяет пользователям оставлять текстовые комментарии к фильмам. Каждый отзыв связывается с пользователем и фильмом, к которому он относится, и имеет дату написания.

### **5. Stars**

Хранит оценки фильмов (число от 1 до 10). Состоит из ID фильма, ID пользователя, количества оценок, и среднюю оценку. Это позволяет системе рассчитывать средний рейтинг фильма.

### **6. Actors**

Содержит информацию об актёрах, таких как имя и биография. С помощью связующей таблицы (Filmography\_of\_actors) позволяет пользователям видеть, в каких фильмах снимался актёр.

## 7. Liked\_movies

Таблица избранных фильмов, в которой пользователи могут сохранять фильмы в свой список избранного. Это позволяет системе показывать пользователю его избранные фильмы.

## 8. Filmography\_of\_actors

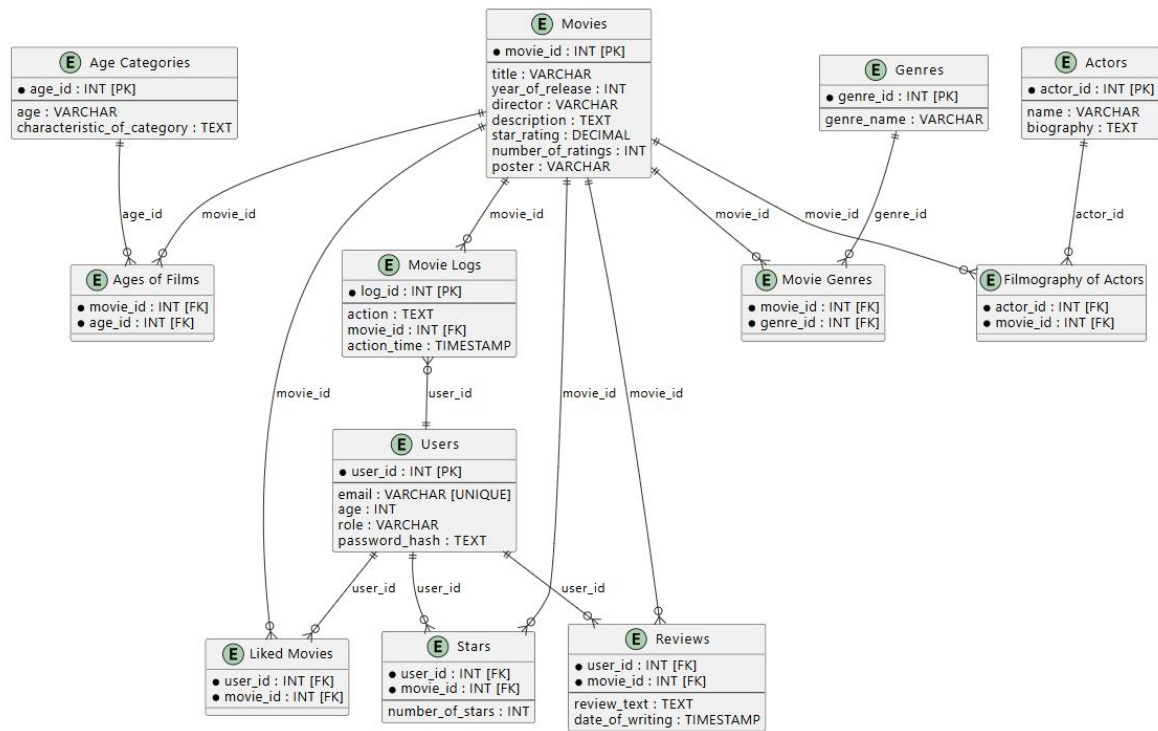
Связующая таблица для фильмов и актёров, обеспечивающая возможность создавать список актёров для каждого фильма.

## 9. Movie\_genres

Ещё одна связующая таблица, которая поддерживает связь между фильмами и жанрами. Так как у фильма может быть несколько жанров, таблица помогает управлять классификацией фильмов.

Эти таблицы обеспечивают функциональность онлайн-кинотеатра, включая управление пользователями, классификацию фильмов, оставление отзывов и оценок, а также формирование рекомендательного списка.

## Схема модели данных



## Схема работы приложения

**Работа с фильмами:** Пользователи могут видеть список фильмов, искать их по названию или фильтровать по жанрам, актерам и возрастным категориям. Для каждого фильма отображается его описание, год выпуска, режиссер, текущий рейтинг и количество оценок.

**Оценки и отзывы:** Зарегистрированные пользователи могут оставлять оценки фильмам. Для этого создается запись в таблице Stars. Пользователи могут оставлять текстовые отзывы, которые сохраняются в таблице Reviews.

**Добавление фильмов в избранное:** Пользователи могут добавлять фильмы в раздел "Избранное". Эти данные записываются в таблицу Liked Movies.

**Рекомендации:** На основе жанров, оценок и возрастной категории пользователя формируется список фильмов, которые могут быть ему интересны.

**Роли пользователей:** В системе предусмотрены роли пользователей, такие как "зритель" (viewer) и "администратор" (admin). Роль определяется через поле role в таблице Users. Администратор имеет возможность добавлять или удалять фильмы из базы данных.

**Логирование действий:** При добавлении или удалении фильмов в таблице Movies автоматически срабатывает триггер, который записывает действие в таблицу Movie Logs.

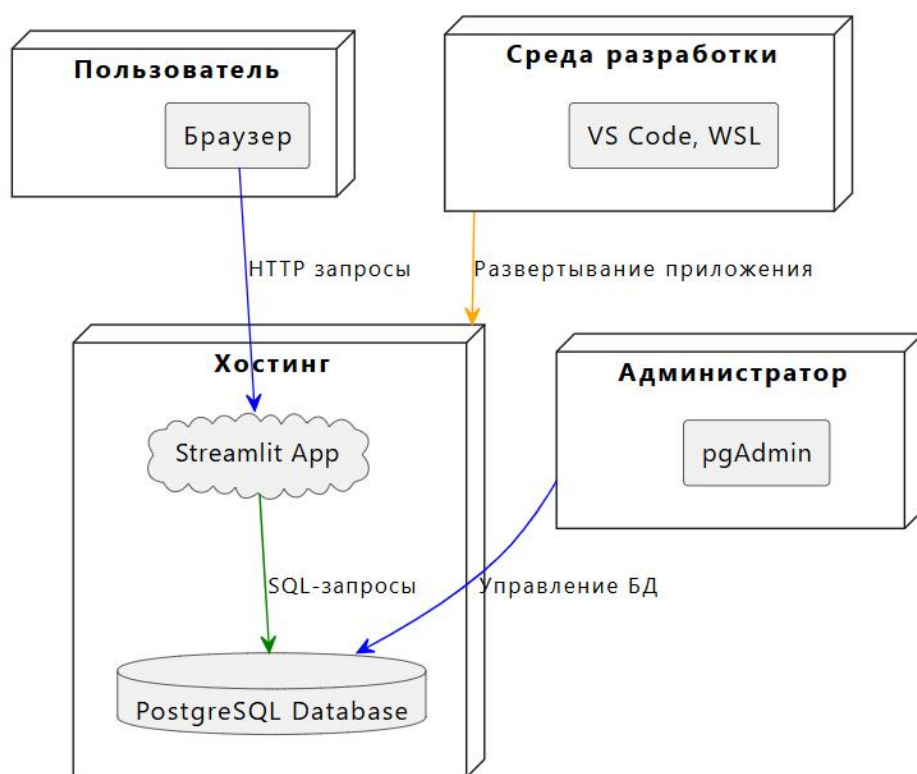
**Классификация по возрастным категориям:** Фильмы распределены по возрастным категориям через таблицу Ages of Films, что позволяет легко фильтровать фильмы для пользователей разных возрастов.

**Связь с актерами и жанрами:** Пользователь может найти фильмы с любимыми актерами или в определенных жанрах через таблицы Filmography of Actors и Movie Genres.

Проект реализован с использованием следующих технологий: PostgreSQL для управления базой данных. Streamlit для построения пользовательского интерфейса. Среда разработки - VS Code. Дополнительные инструменты: pgAdmin для управления базой данных, Python для реализации функционала.

Пример работы:

1. Пользователь регистрируется или вводит уже известные ему логин и пароль, открывает интерфейс и видит список фильмов.
2. Выбирает понравившийся фильм, читает описание, добавляет его в избранное или ставит оценку и пишет отзыв.
3. После оценки обновляется общий рейтинг фильма в базе данных.
4. Администратор добавляет новый фильм через интерфейс или напрямую в базу данных. Это действие логируется.



Проект демонстрирует возможности управления данными в реляционной базе данных, автоматизации задач (через триггеры и функции) и взаимодействия различных компонент системы. Пользователи получают доступ к удобной платформе для выбора фильмов и взаимодействия с контентом, а администратор — к инструментам для управления базой данных.

## Qr-код на репозиторий GitHub

