

# A SMART MOVIE RECOMMENDATION SYSTEM

## Project Mid Evaluation

Project Overview and Normalized Schema

Overview, objectives and features are defined in detail as part of the Project Overview. While a list of tables and their schemas are provided to justify the normalization up to 3-NF

Zain Baig – 22K4593

Asad Irfan – 22K4276

### Project Title: Smart Movie Recommendation System

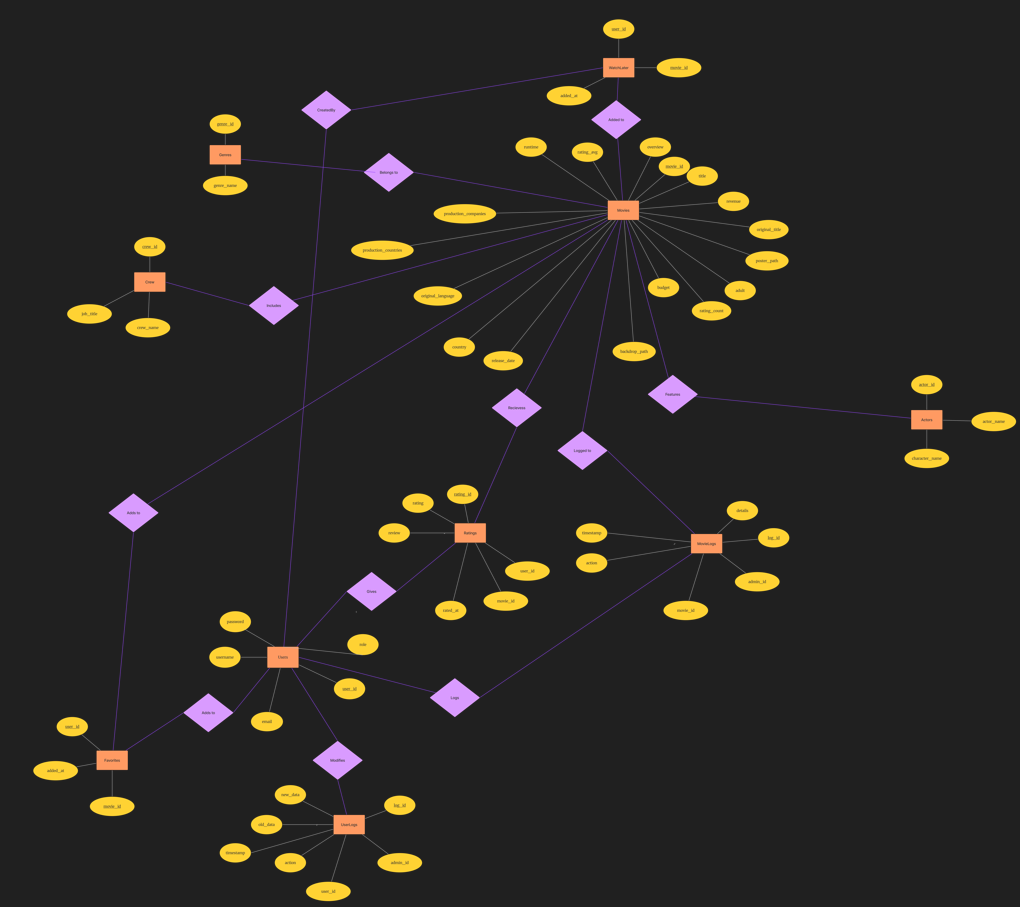
#### Overview

The **Smart Movie Recommendation System** is a dynamic web application crafted to elevate the movie-watching experience. It showcases trending movies, cast, and crew while empowering users with advanced features like search, sorting, user ratings, reviews, favorites, and watchlist management. Tailored for movie enthusiasts, this platform simplifies the discovery of popular content and provides seamless tools to organize preferences, creating a personalized and engaging cinematic journey.

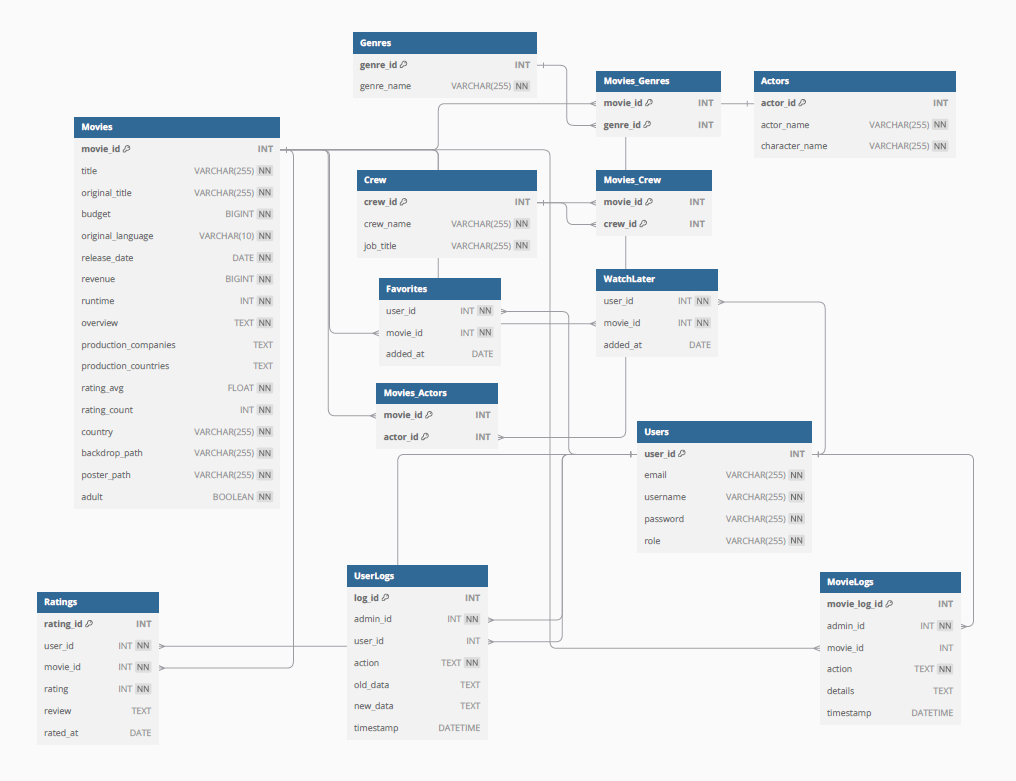
#### Project Objectives

1. **User Interaction:** Allow users to rate and review movies, add favorites, and create personalized watchlists to plan their movie-watching experience.
2. **Enhanced Browsing Experience:** Enable users to explore movies by genres, cast, crew, and country, providing a seamless and engaging discovery process.
3. **Trending Insights:** Identify and display trending movies, actors, directors, and producers, keeping users updated with the latest popular content.
4. **Sorting Options:** Offer sorting features based on top-rated movies and release dates for easy navigation and decision-making.
5. **Efficient Data Management:** Implement a structured database with well-defined relationships between entities like movies, users, genres, and crew members, ensuring efficient data querying and retrieval.

**Entity Relationship Diagram:**

****

**Schema Normalized upto 3nf:**

****

**Features:**

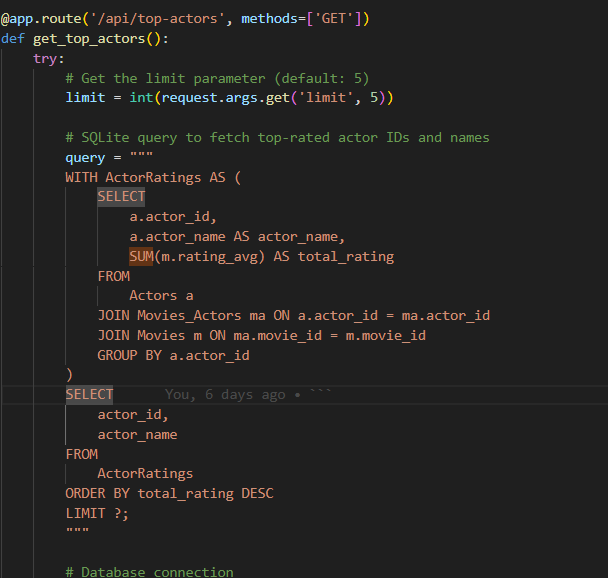
The system will perform several key functions:

1. **User Registration & Profile Management:** Allow users to create and manage profiles, including setting preferences for genres, actors, and other movie-related attributes to enhance the personalized experience.
2. **Search Functionality:** Provide an intuitive search feature, enabling users to find movies based on title, genre, actor, director, or country.
3. **Trending Movies & Insights:** Display lists of trending movies, actors, directors, and producers to keep users informed about popular and noteworthy content.
4. **Rating System:** Enable users to rate movies, influencing content rankings and enhancing community engagement.
5. **User Reviews & Comments:** Allow users to leave reviews and comments, fostering a community-driven platform with valuable user-generated feedback.
6. **Movie Details Page:** Present comprehensive details about each movie, including ratings, reviews, cast, crew, and release information.
7. **WatchLater & Favorites Management:** Provide users with the ability to add movies to their personalized watchlist or mark them as favorites for easy access.
8. **Genre-Based Recommendations:** Suggest movies based on genres of interest to the user, offering a tailored browsing experience.
9. **Actor & Director-Based Recommendations:** Recommend movies featuring actors, directors, or producers favored by the user, enhancing discovery.
10. **Country-Based Recommendations:** Filter and recommend movies based on their country of origin, allowing users to explore culturally specific content.
11. **Movie Database Management:** Allow admins to manage the movie database efficiently, including adding, updating, and deleting movies, either individually or in bulk.
12. **User Management Panel:** Provide an admin panel to manage user accounts, enabling actions like adding, editing, and deleting user information.
13. **Movie Logs & User Logs:** Maintain detailed logs of all changes made to movies and user data, ensuring a transparent history of updates.
14. **Admin Dashboard:** Offer a centralized admin panel for managing all platform features, including movies, users, and logs.
15. **Genre & Release Date Sorting:** Provide sorting options to organize movies by genre, top ratings, or release date for streamlined exploration.
16. **Batch Operations for Admins:** Facilitate bulk operations for adding, editing, or deleting movies and users to simplify database management tasks.

**\*** JOIN/Subqueries

****

\* Aggregate Functions



\* Stored Procedures



\* Triggers and Transactions



* For loop if condition works as trigger to update not only movies, but actors, genres, and crew for the movie being updated too.
* Conn.rollback works as transaction rollback if any of the query fails

**Real-World Applications:**

1. **User Personalization**:
   * Tailors movie recommendations based on user preferences (genres, actors, directors).
   * Real-world application: Streaming platforms like Netflix or Amazon Prime use similar algorithms to personalize content and increase user engagement.
2. **Social Interaction**:
   * Allows users to rate, review, and comment on movies.
   * Real-world application: Platforms like IMDb and Rotten Tomatoes enable users to share opinions and create a community-driven content hub.
3. **Content Discovery**:
   * Suggests trending movies, actor-based recommendations, or country-specific content.
   * Real-world application: Services like YouTube and Spotify help users discover content relevant to their interests using trends and recommendations.
4. **Admin Control and Data Management**:
   * Enables movie database and user account management with bulk operations.
   * Real-world application: Content platforms and online stores like Amazon allow administrators to manage large inventories, user accounts, and content efficiently.

**Challenges Faced:**

1. **Recommendation Accuracy**:
   * Ensuring that recommendations align with user preferences and accurately reflect diverse genres and tastes.
   * Challenge: Balancing content diversity with accuracy and handling cold-start problems (when there is insufficient data for a user or movie).
2. **Data Management**:
   * Handling large volumes of movie and user data efficiently.
   * Challenge: Maintaining database performance with high traffic, especially when managing large datasets or handling complex queries.
3. **Scalability**:
   * Scaling the system to handle increasing users, data, and interactions.
   * Challenge: Building a robust backend to handle concurrent users, real-time data updates, and batch processing for admin tasks.
4. **UI/UX Design**:
   * Providing an intuitive, smooth user experience while displaying complex data.
   * Challenge: Designing interfaces that are both aesthetically pleasing and functional, ensuring ease of navigation for users with varying tech proficiency.