

Movie and TV Show review and rating platform

Team Details:

Srirama V Swamy
Siddartha Kumar

PES1UG23CS598
PES1UG23CS573

Abstract:

In the digital entertainment era, viewers consume diverse content across movies and television shows but often lack a centralized, user-centric platform to discover, review, and rate content with granular control. Existing platforms provide limited flexibility for reviewing individual TV show episodes, offer inadequate search and filtering capabilities, and lack personalized user experiences with watchlists, custom lists, and social engagement features.

This project addresses these gaps by developing a comprehensive review and rating platform that enables users to submit ratings and reviews for both movies and individual TV show episodes, ensuring content quality through automated rating aggregation and constraint enforcement. The system provides advanced search functionality with multi-parameter filtering by genre, language, rating thresholds, and release years, allowing precise content discovery. Users can manage personalized watchlists, create custom content lists, and engage socially through review likes and follower systems.

Built with a Flask-based web application interfacing with a MySQL relational database, the platform leverages database views for efficient data retrieval, triggers for automatic rating updates, and stored procedures for complex operations such as review submission and trending content identification. The architecture ensures data integrity through foreign key constraints, check constraints, and normalized table structures encompassing users, movies, TV shows, episodes, reviews, genres, actors, directors, and social features.

This platform delivers an integrated solution for content exploration, community-driven reviews, and personalized content management, enhancing the user experience for entertainment enthusiasts.

User Requirement Specification:

Purpose of the Project

The purpose of the "Movie and Show Review and Rating Platform" is to develop a robust database management system that enables users to review and rate movies and TV shows, with the capability to provide granular feedback on individual TV show episodes. The system is designed as a DBMS project to demonstrate advanced database concepts including normalized schema design, referential integrity enforcement, automated triggers for real-time

data aggregation, stored procedures for complex transactions, and views for efficient data retrieval. The platform serves as a centralized repository for entertainment content metadata, user-generated reviews, and computed statistics, ensuring data consistency and reliability through well-designed database architecture.

Scope of the Project

The scope of this database management system project encompasses:

- **Core Database Schema Design:** Implementing normalized relational tables for users, movies, TV shows, episodes, reviews, genres, actors, and directors with appropriate primary keys, foreign keys, and constraints.
- **Many-to-Many Relationship Management:** Handling complex relationships through junction tables such as Movie_Genre, Show_Genre, Movie_Actor, and Movie_Director.
- **Review and Rating System:** Supporting user-submitted reviews for both movies and individual episodes with rating validation (0-5 scale) and ensuring one review per user per content item through unique constraints.
- **Automated Database Operations:** Implementing triggers to automatically update aggregate statistics (average ratings, review counts) in real-time when reviews are inserted, updated, or deleted.
- **Complex Query Support:** Creating database views for commonly accessed data patterns (popular movies, top-rated shows, user statistics) and stored procedures for advanced search functionality with multiple filter parameters.
- **Data Integrity and Validation:** Enforcing business rules through CHECK constraints, triggers for data validation, and referential integrity to prevent orphaned records.
- **Web Application Interface:** A Flask-based frontend that interacts with the MySQL database through parameterized queries to prevent SQL injection and ensure secure data access.

Detailed Description

The system is built on a **MySQL relational database** with a normalized schema following third normal form (3NF) principles to minimize data redundancy and ensure data integrity. The database architecture consists of the following core components:

1. Database Schema and Table Design

The system implements 12 core tables with well-defined relationships:

- **User Table:** Stores user credentials (username, password hash), profile information (name, DOB, email, phone, address) with unique constraints on username and email to prevent duplicates.
- **Movie Table:** Contains movie metadata including title, duration, description, box office revenue, release date, age rating, language, computed average rating, poster URL, and total review count.

- **TV Show (tvshow) Table:** Stores TV show information with fields for number of seasons/episodes, description, release date, age rating, language, average rating, poster URL, and show status (Ongoing/Completed/Cancelled).
- **Episode Table:** Links to the tvshow table via foreign key, storing season number, episode number, title, description, duration, and air date. Implements CASCADE delete to remove all episodes when a show is deleted.
- **Review Table:** Central to the rating system, containing user_id, movie_id or episode_id (mutually exclusive), date, rating (DECIMAL(2,1) constrained between 0-5), review text, and likes count. Enforces CHECK constraints to ensure exactly one of movie_id or episode_id is populated, and UNIQUE constraints to prevent duplicate reviews from the same user.
- **Genre, Actor, Director Tables:** Store entertainment industry metadata with auto-incrementing primary keys and descriptive fields like bio, profile images, nationality, and date of birth.
- **Junction Tables (Movie_Genre, Show_Genre, Movie_Actor, Movie_Director):** Implement many-to-many relationships with composite primary keys combining the related entity IDs.

2. Database Constraints and Integrity

- **Primary Keys:** All tables use auto-incrementing INT primary keys for efficient indexing and join operations.
- **Foreign Keys with Cascading:** ON DELETE CASCADE ensures referential integrity when parent records are deleted (e.g., deleting a user removes all their reviews).
- **CHECK Constraints:** Enforce business rules such as rating ranges (0-5), mutual exclusivity of movie_id and episode_id in reviews.
- **UNIQUE Constraints:** Prevent duplicate data entries such as duplicate usernames, emails, or multiple reviews from one user on the same content.

3. Triggers for Automated Data Maintenance

Eight triggers maintain data consistency automatically:

- **update_movie_stats_insert/update/delete:** Recalculate and update the Movie table's average rating and total review count whenever a movie review is added, modified, or removed.
- **update_show_rating_insert/update/delete:** Update TV show ratings based on aggregated episode review ratings when episode reviews change.
- **set_review_date:** Automatically sets the current date if a review is submitted without a date.
- **validate_review_type:** Prevents insertion of reviews that violate the movie XOR episode constraint before database-level CHECK constraints are evaluated.

These triggers eliminate the need for application-level rating recalculation and ensure database consistency.

4. Views for Efficient Data Retrieval

Eight materialized-like views provide pre-joined, aggregated data for common queries:

- **movie_reviews_view**: Joins Review, User, and Movie tables to display all movie reviews with reviewer details.
- **episode_reviews_view**: Similar to above but for TV show episode reviews, including show name and episode details.
- **popular_movies**: Lists movies ordered by rating and review count for homepage display.
- **top_rated_shows**: Aggregates show data with episode and review counts.
- **movie_details_view**: Comprehensive movie information with comma-separated lists of genres, directors, and actors using GROUP_CONCAT.
- **show_details_view**: Complete TV show information with genre lists.
- **user_stats_view**: User activity statistics including total reviews, average rating given, and likes received.
- **recent_activity_view**: Combined feed of all recent reviews across movies and episodes.

These views significantly improve query performance and simplify application code.

5. Stored Procedures for Complex Operations

Eleven stored procedures encapsulate complex database operations:

- **add_movie_review / add_episode_review**: Transaction-wrapped insert operations with error handling for adding reviews.
- **get_user_reviews**: Retrieves all reviews by a specific user with COALESCE for null handling.
- **search_movies_by_genre / get_movies_by_director / get_movies_by_actor**: Filtered search operations.
- **advanced_movie_search**: Multi-parameter search accepting optional filters for title, genre, minimum rating, language, and year range, returning up to 50 results ordered by rating.
- **get_show_details**: Returns complete show information and all episodes with review statistics in a single call.
- **get_user_statistics / get_top_grossing_movies / get_trending_movies**: Analytics and reporting procedures.

Stored procedures provide consistent data access patterns, improve security by preventing direct table access, and enable complex logic execution on the database server.

6. User-Defined Functions

Eight scalar functions provide reusable calculations:

- **get_user_avg_rating / count_user_reviews / get_user_total_likes:** User statistic calculations.
- **has_reviewed_movie / has_reviewed_episode:** Boolean functions to check review existence.
- **calculate_movie_popularity:** Algorithmic popularity score based on rating and review count logarithm.
- **count_movies_reviewed / count_episodes_reviewed:** Content-specific counting functions.

Functions enable embedding complex calculations in queries without repetitive code.

7. Flask Application Layer

The web interface uses Flask with MySQL Connector providing:

- **Database Connection Management:** Connection pooling with environment variable configuration for credentials.
- **Parameterized Queries:** fetch_all() and fetch_one() functions execute parameterized SQL to prevent SQL injection attacks.
- **Stored Procedure Calls:** callproc() function safely invokes stored procedures with parameter binding.
- **Route Handlers:** RESTful endpoints for home page, movie/show detail pages, search, review submission, and user authentication.
- **Session Management:** Secure user sessions with Flask's secret_key for CSRF protection.
- **Password Security:** werkzeug.security for password hashing and verification.

The application layer remains thin, delegating business logic to the database layer through views, procedures, and functions.

8. Key Technical Achievements

- **Normalized Schema Design:** Eliminates data redundancy through proper normalization with separate tables for entities and junction tables for relationships.
- **Referential Integrity:** Comprehensive foreign key relationships with appropriate CASCADE rules ensure data consistency.
- **Automated Aggregation:** Triggers automatically maintain denormalized aggregate data (ratings, counts) for query performance without manual updates.
- **Flexible Search:** The advanced_movie_search procedure demonstrates parameter handling with optional NULL values for flexible filtering.
- **Data Validation:** Multi-layered validation through CHECK constraints, UNIQUE constraints, and triggers prevents invalid data entry.
- **Transaction Safety:** Stored procedures wrap insert operations in transactions with error handlers for rollback on failure.

List of Software/Tools/Programming languages used:

Programming Languages

- **Python 3.x:** Primary backend programming language for application logic and database connectivity
- **SQL:** Database query language for table creation, stored procedures, triggers, views, and functions
- **HTML/CSS/JavaScript:** Frontend technologies for user interface templates (referenced via Flask's `render_template`)

Web Framework

- **Flask:** Lightweight Python web framework for building the web application with routing, session management, and template rendering

Database Management System

- **MySQL:** Relational database management system for storing and managing all application data with support for transactions, constraints, triggers, views, and stored procedures

Python Libraries and Modules

- **mysql-connector-python:** MySQL database driver for Python enabling database connectivity and query execution
- **flask:** Core Flask framework for web application development
- **werkzeug.security:** Provides password hashing utilities (`check_password_hash`) for secure authentication
- **python-dotenv:** Environment variable management for secure configuration of database credentials
- **os:** Standard Python module for environment variable access

Development Tools

- **MySQL Workbench** (implied): Database design, schema management, and SQL development tool for creating tables, triggers, procedures, and views
- **Text Editor/IDE:** Code editor for Python and SQL development (e.g., VS Code, PyCharm, or similar)
- **Web Browser:** For testing and accessing the web application interface

Server and Hosting

- **Flask Development Server:** Built-in Flask development server running on `localhost:5000` for testing
- **localhost:** Local MySQL server for database hosting during development

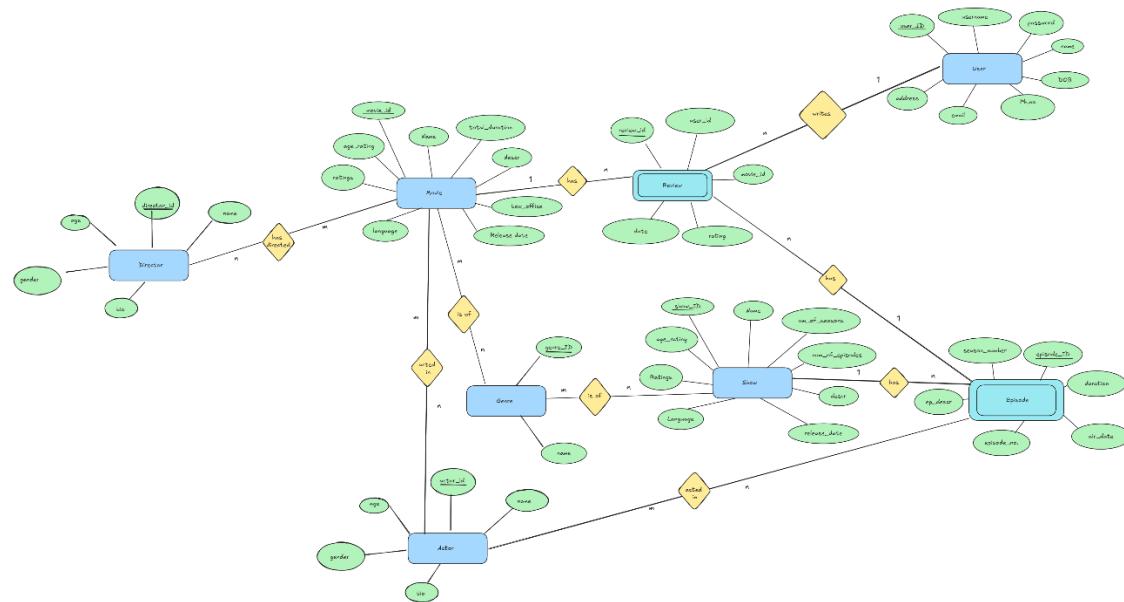
Configuration Management

- .env file:** Environment configuration file storing database credentials (DB_HOST, DB_USER, DB_PASS, DB_NAME)

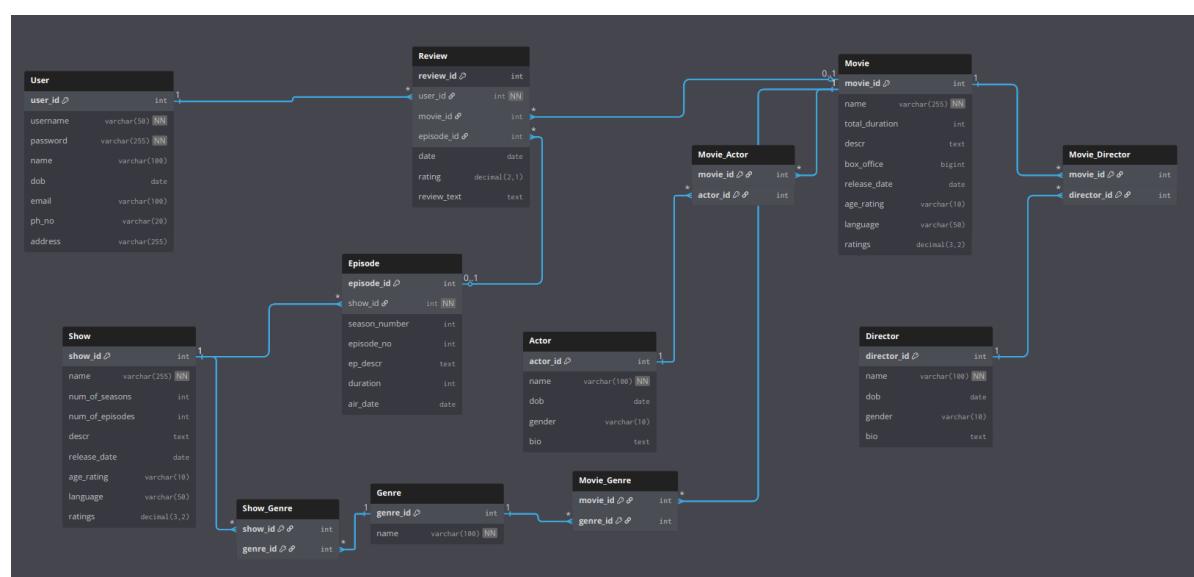
Security Features

- Parameterized Queries:** SQL injection prevention through parameter binding in all database queries
- Password Hashing:** werkzeug.security for secure password storage and verification
- Session Management:** Flask's built-in session handling with secret_key for CSRF protection

Entity Relationship Diagram



Relational Schema



Data Definition Language Commands

Here are the Data Definition Language (DDL) commands used to create the database schema for the Movie and Show Review and Rating Platform:

1. User Table

```
CREATE TABLE User (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    name VARCHAR(100),
    dob DATE,
    email VARCHAR(100) NOT NULL UNIQUE,
    ph_no VARCHAR(20),
    address VARCHAR(255)
);
```

2. Movie Table

```
CREATE TABLE Movie (
    movie_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    total_duration INT,
    descr TEXT,
    box_office BIGINT,
    release_date DATE,
    age_rating VARCHAR(10),
    language VARCHAR(50),
    ratings DECIMAL(3,2) DEFAULT 0.00,
    poster_url VARCHAR(500),
    total_reviews INT DEFAULT 0
);
```

3. TV Show Table

```
CREATE TABLE tvshow (
```

```

show_id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(255) NOT NULL,
num_of_seasons INT,
num_of_episodes INT,
descr TEXT,
release_date DATE,
age_rating VARCHAR(10),
language VARCHAR(50),
ratings DECIMAL(3,2) DEFAULT 0.00,
poster_url VARCHAR(500),
status VARCHAR(20) DEFAULT 'Ongoing'
);

```

4. Episode Table

```

CREATE TABLE Episode (
episode_id INT PRIMARY KEY AUTO_INCREMENT,
show_id INT NOT NULL,
season_number INT,
episode_no INT,
ep_descr TEXT,
duration INT,
air_date DATE,
title VARCHAR(255),
FOREIGN KEY (show_id) REFERENCES tvshow(show_id) ON DELETE CASCADE
);

```

5. Review Table

```

CREATE TABLE Review (
review_id INT PRIMARY KEY AUTO_INCREMENT,
user_id INT NOT NULL,
movie_id INT,
episode_id INT,

```

```

date DATE,
rating DECIMAL(2,1) CHECK (rating >= 0 AND rating <= 5),
review_text TEXT,
likes_count INT DEFAULT 0,
FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
FOREIGN KEY (movie_id) REFERENCES Movie(movie_id) ON DELETE CASCADE,
FOREIGN KEY (episode_id) REFERENCES Episode(episode_id) ON DELETE
CASCADE,
UNIQUE KEY unique_user_episode (user_id, episode_id),
CHECK ((movie_id IS NOT NULL AND episode_id IS NOT NULL) OR
(movie_id IS NULL AND episode_id IS NOT NULL))
);

```

6.Genre Table

```

CREATE TABLE Genre (
genre_id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(100) NOT NULL UNIQUE,
description TEXT
);

```

7.Actor Table

```

CREATE TABLE Actor (
actor_id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(100) NOT NULL,
dob DATE,
gender VARCHAR(10),
bio TEXT,
profile_image_url VARCHAR(500)
);

```

8. Director Table

```

CREATE TABLE Director (

```

```

director_id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(100) NOT NULL,
dob DATE,
gender VARCHAR(10),
bio TEXT,
profile_image_url VARCHAR(500),
nationality VARCHAR(50)
);

```

JUNCTION TABLES (Many-to-Many Relationships)

9. Movie_Director Table

```

CREATE TABLE Movie_Director (
    movie_id INT NOT NULL,
    director_id INT NOT NULL,
    PRIMARY KEY (movie_id, director_id),
    FOREIGN KEY (movie_id) REFERENCES Movie(movie_id) ON DELETE CASCADE,
    FOREIGN KEY (director_id) REFERENCES Director(director_id) ON DELETE
    CASCADE
);

```

10. Movie_Actor Table

```

CREATE TABLE Movie_Actor (
    movie_id INT NOT NULL,
    actor_id INT NOT NULL,
    character_name VARCHAR(100),
    PRIMARY KEY (movie_id, actor_id),
    FOREIGN KEY (movie_id) REFERENCES Movie(movie_id) ON DELETE CASCADE,
    FOREIGN KEY (actor_id) REFERENCES Actor(actor_id) ON DELETE CASCADE
);

```

11. Movie_Genre Table

```
CREATE TABLE Movie_Genre (
    movie_id INT NOT NULL,
    genre_id INT NOT NULL,
    PRIMARY KEY (movie_id, genre_id),
    FOREIGN KEY (movie_id) REFERENCES Movie(movie_id) ON DELETE CASCADE,
    FOREIGN KEY (genre_id) REFERENCES Genre(genre_id) ON DELETE CASCADE
);
```

12. Show_Genre Table

```
CREATE TABLE Show_Genre (
    show_id INT NOT NULL,
    genre_id INT NOT NULL,
    PRIMARY KEY (show_id, genre_id),
    FOREIGN KEY (show_id) REFERENCES tvshow(show_id) ON DELETE CASCADE,
    FOREIGN KEY (genre_id) REFERENCES Genre(genre_id) ON DELETE CASCADE
);
```

13. Review_Likes Table

```
CREATE TABLE Review_Likes (
    like_id INT PRIMARY KEY AUTO_INCREMENT,
    review_id INT NOT NULL,
    user_id INT NOT NULL,
    liked_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (review_id) REFERENCES Review(review_id) ON DELETE CASCADE,
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    UNIQUE KEY unique_user_review_like (user_id, review_id)
);
```

14. User_Followers Table

```
CREATE TABLE User_Followers (
    follower_id INT NOT NULL,
```

```

following_id INT NOT NULL,
followed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (follower_id, following_id),
FOREIGN KEY (follower_id) REFERENCES User(user_id) ON DELETE CASCADE,
FOREIGN KEY (following_id) REFERENCES User(user_id) ON DELETE CASCADE,
CHECK (follower_id != following_id)
);

```

CRUD Operations

User table

```

mysql> INSERT INTO User (username, password, name, dob, email, ph_no, address)
-> VALUES ('newuser123', '$2b$12$hashed_password', 'John Smith', '1995-01-15', 'john.smith@email.com', '9876543220', '123 Test St, City');
Query OK, 1 row affected (0.12 sec)

mysql> SELECT * FROM User WHERE user_id = 1;
+-----+-----+-----+-----+-----+-----+-----+
| user_id | username | password | name | dob | email | address |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | john_doe | $2b$12$LQv3clyqBWHxkdBLHAKCOYz6TtxMQJqhNB/LewY5ztP8jYvZ8Ziu | John Doe | 1995-05-15 | john.doe@email.com | 9876543210 | 123 Main Street, New York, NY 10001 | 2025-10-31 14:33:56 | NULL
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> CALL get_user_statistics();
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| user_id | username | name | total_reviews | avg_rating_given | movies_reviewed | episodes_reviewed | total_likes_received | first_review_date | last_review_date |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | john_doe | John Doe | 7 | 4.85714 | 5 | 2 | 0 | 2024-01-15 | 2024-05-20
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.06 sec)

Query OK, 0 rows affected (0.08 sec)

mysql> UPDATE User SET email='newemail@test.com', ph_no='9876543299' WHERE user_id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT * FROM User WHERE user_id = 1;
+-----+-----+-----+-----+-----+-----+-----+
| user_id | username | password | name | dob | email | address |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | john_doe | $2b$12$LQv3clyqBWHxkdBLHAKCOYz6TtxMQJqhNB/LewY5ztP8jYvZ8Ziu | John Doe | 1995-05-15 | newemail@test.com | 9876543299 | 123 Main Street, New York, NY 10001 | 2025-10-31 14:33:56 | NULL
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

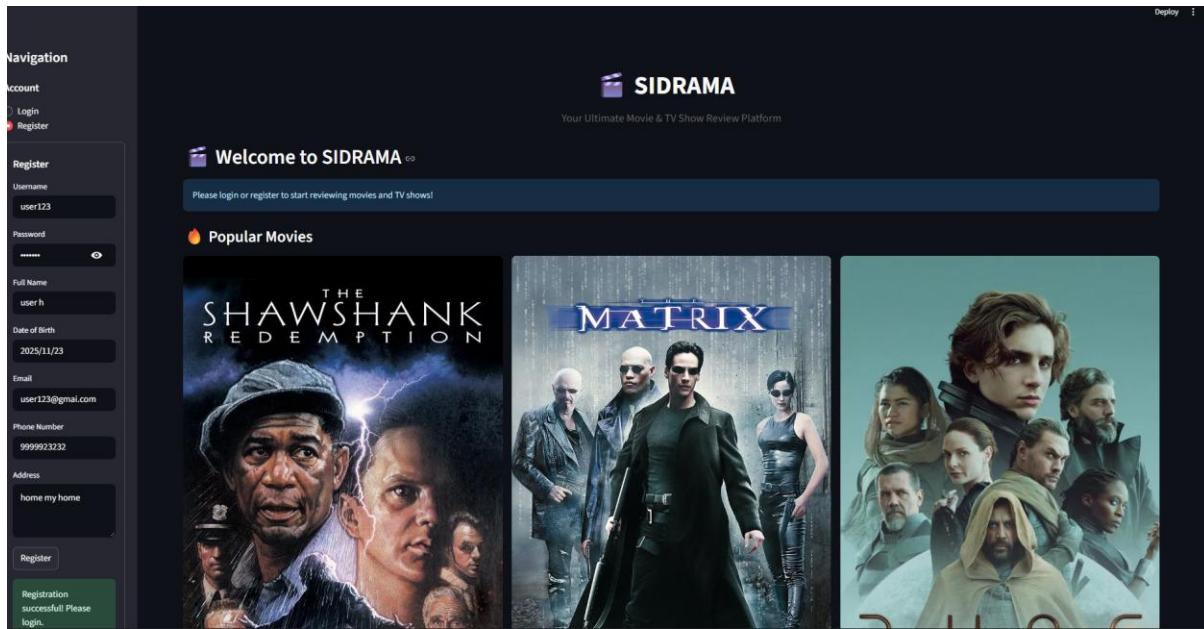
mysql> DELETE FROM User WHERE user_id = 10;
Query OK, 1 row affected (0.08 sec)

mysql> DELETE FROM User WHERE user_id = 10;
Query OK, 1 row affected (0.08 sec)

mysql> SELECT * FROM User WHERE user_id = 1;
+-----+-----+-----+-----+-----+-----+-----+
| user_id | username | password | name | dob | email | address |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | john_doe | $2b$12$LQv3clyqBWHxkdBLHAKCOYz6TtxMQJqhNB/LewY5ztP8jYvZ8Ziu | John Doe | 1995-05-15 | newemail@test.com | 9876543299 | 123 Main Street, New York, NY 10001 | 2025-10-31 14:33:56 | NULL
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM User;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| user_id | username | password | name | dob | email | ph_no | address | created_at | last_login |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | john_doe | $2b$12$LQv3clyqBWHxkdBLHAKCOYz6TtxMQJqhNB/LewY5ztP8jYvZ8Ziu | John Doe | 1995-05-15 | newemail@test.com | 9876543299 | 123 Main Street, New York, NY 10001 | 2025-10-31 14:33:56 | NULL
| 2 | jane_smith | $2b$12$92IXMpj00rQ05byM1.Yehlo/e3Re9lC/.sg/at2.uhW6/Igi | Jane Smith | 1998-08-22 | jane.smith@email.com | 9876543211 | 456 Oak Avenue, Los Angeles, CA 90001 | 2025-1
| 3 | mike_johnson | $2b$12$N3ZqmXrSa2zQvKt.PXKh5uGw/dz3y6JN8Gq5zXm3N7XGy8Zg9k | Mike Johnson | 1992-03-10 | mike.johnson@email.com | 9876543212 | 789 Pine Road, Chicago, IL 60601 | 2025-1
| 4 | sarah_williams | $2b$12$hash#HexExample | Sarah Williams | 1990-12-05 | sarah.w@email.com | 9876543213 | 321 Elm Street, Boston, MA 02101 | 2025-1
| 5 | alex_chen | $2b$12$hash#2024 | Alex Chen | 2000-07-18 | alex.chen@email.com | 9876543214 | 654 Maple Drive, Seattle, WA 98101 | 2025-1
| 6 | emily_davis | $2b$12$hash#example | Emily Davis | 1988-11-30 | emily.davis@email.com | 9876543215 | 987 Cedar Lane, Austin, TX 78701 | 2025-1
| 7 | david_martinez | $2b$12$hash#example | David Martinez | 1996-04-25 | david.m@email.com | 9876543216 | 246 Birch Street, Denver, CO 80201 | 2025-1
| 8 | rachel_green | $2b$12$hash#example | Rachel Green | 1994-09-14 | rachel.g@email.com | 9876543217 | 135 Willow Avenue, Miami, FL 33101 | 2025-1
| 9 | chris_wilson | $2b$12$hash#example | Chris Wilson | 1991-06-08 | chris.w@email.com | 9876543218 | 579 Spruce Road, Phoenix, AZ 85001 | 2025-1
| 10 | siddhartha | $2b$12$hash#example | Siddhartha | 2005-11-15 | sid@gmail.com | 9999999999 | 2025-1
| 11 | srirama | $2b$12$hash#example | Sri Rama | 2005-03-24 | sriramavishwa@gmail.com | 87338101114 | E303
| 12 | sid | $2b$12$hash#example | Sid | 2005-11-15 | sid@gmail.com | 9999999999 | 2025-1
| 13 | shruti | $2b$12$hash#example | Shruti | 2005-03-24 | shruti@gmail.com | 1234567890 | manhattan nyc | 2025-1
| 14 | john_smith | $2b$12$hash#example | John Smith | 1995-01-15 | john.smith@email.com | 9876543220 | 123 Test St, City | 2025-1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```



user registering and getting added to the table

Movie table

```
mysql> SELECT * FROM movie_details_view WHERE movie_id = 1;
+-----+-----+-----+-----+-----+-----+-----+-----+
| movie_id | movie_name | release_date | total_duration | ratings | total_reviews | poster_url
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Inception | 2010-07-16 | 148 | 4.88 | 4 | https://www.themoviedb.org/t/p/w780/edv5CZwJ09upOsy2Y6ImDhK8bt.jpg |
| directors | actors | box_office | age_rating | language | genres
+-----+-----+-----+-----+-----+-----+
| riller | Christopher Nolan | Leonardo DiCaprio | 829895144 | PG-13 | English | Action, Sci-Fi, Thriller
+-----+-----+-----+-----+-----+-----+
1 row in set (0.16 sec)

mysql> UPDATE Movie SET descr='Updated description', box_office=1000000000 WHERE movie_id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT * FROM movie_details_view WHERE movie_id = 1;
+-----+-----+-----+-----+-----+-----+-----+
| movie_id | movie_name | release_date | total_duration | ratings | total_reviews | poster_url
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Inception | 2010-07-16 | 148 | 4.88 | 4 | https://www.themoviedb.org/t/p/w780/edv5CZwJ09upOsy2Y6ImDhK8bt.jpg |
| directors | actors | box_office | age_rating | language | genres
+-----+-----+-----+-----+-----+-----+
| riller | Christopher Nolan | Leonardo DiCaprio | 1000000000 | PG-13 | English | Action, Sci-Fi, Thriller
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> DELETE FROM Movie WHERE movie_id = 15;
Query OK, 1 row affected (0.03 sec)
```

Tv show table

```
mysql> INSERT INTO tvshow (name, num_of_seasons, num_of_episodes, descr, release_date, age_rating, language, poster_url, status)
-> VALUES ('The Bear', 3, 28, 'A chef returns to Chicago to run his family sandwich shop', '2022-06-23', 'TV-MA', 'English', 'https://image.tmdb.org/t/p/w780/poster.jpg', 'Ongoing');
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM show_details_view WHERE show_id = 1;
+-----+-----+-----+-----+-----+-----+-----+-----+
| show_id | show_name | num_of_seasons | num_of_episodes | release_date | status | ratings | poster_url
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Breaking Bad | 5 | 62 | 2008-01-20 | Completed | 5.0 | https://www.themoviedb.org/t/p/w780/ggFHvNuYYISL9pCFoacjizRgt.jpg |
| age_rating | language | genres
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)

mysql> UPDATE tvshow SET num_of_episodes=30, status='Completed' WHERE show_id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM tvshow WHERE show_id = 10;
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO Review (user_id, movie_id, date, rating, review_text)
-> VALUES (1, 3, CURDATE()), 4.5, 'Amazing movie with great performances!';
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM movie_reviews_view WHERE movie_name = 'Inception';
+-----+-----+-----+-----+-----+-----+-----+
| review_id | username | user_name | movie_name | poster_url
| review_date | likes_count | rating | review_text
+-----+-----+-----+-----+-----+-----+
| 35 | john_doe | John Doe | Inception | https://www.themoviedb.org/t/p/w780/edv5CZwJ09upOsy2Y6ImDhK8bt.jpg | 5.0 | Absolutely mind-blowing! Inception is a masterpiece of storytelling. The layers of dreams within dreams keep you engaged throughout. Nolan at his absolute best!
| 38 | jane_smith | Jane Smith | Inception | https://www.themoviedb.org/t/p/w780/edv5CZwJ09upOsy2Y6ImDhK8bt.jpg | 4.5 | Complex and engaging narrative that needs multiple viewings to fully appreciate all the layers. Spectacular visuals and Hans Zimmer's score is incredible.
| 50 | film_critic_pro | Emily Davis | Inception | https://www.themoviedb.org/t/p/w780/edv5CZwJ09upOsy2Y6ImDhK8bt.jpg | 5.0 | A film that gets better with each viewing. The ending is still debated years later. Absolute genius filmmaking.
| 72 | sidd | siddhartha | Inception | https://www.themoviedb.org/t/p/w780/edv5CZwJ09upOsy2Y6ImDhK8bt.jpg | 5.0 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> UPDATE Review SET rating=5.0, review_text='Updated: Even better on second watch!' WHERE review_id = 10;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0 Changed: 0 Warnings: 0

mysql> DELETE FROM Review WHERE review_id = 15;
Query OK, 0 rows affected (0.00 sec)
```

```

mysql> INSERT INTO tvshow (name, num_of_seasons, num_of_episodes, descr, release_date, age_rating, language, poster_url, status)
-> VALUES ('The Bear', 3, 28, 'A chef returns to Chicago to run his family sandwich shop', '2022-06-23', 'TV-MA', 'English', 'https://image.tmdb.org/t/p/w780/poster.jpg', 'Ongoing');
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM show_details_view WHERE show_id = 1;
+-----+-----+-----+-----+-----+-----+-----+-----+
| show_id | show_name | num_of_seasons | num_of_episodes | release_date | status | ratings | poster_url
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Breaking Bad | 5 | 62 | 2008-01-20 | Completed | 5.00 | https://www.themoviedb.org/t/p/w780/ggFHVNu6YYI5L9pCf0acjizRGt.jpg | TV-MA | English | NULL | genres |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)

mysql> UPDATE tvshow SET num_of_episodes=30, status='Completed' WHERE show_id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM tvshow WHERE show_id = 10;
Query OK, 1 row affected (0.02 sec)

```

Review table

```

mysql> SELECT * FROM movie_reviews_view WHERE movie_name = 'Inception';
+-----+-----+-----+-----+-----+-----+-----+
| review_id | username | user_name | movie_name | poster_url | review_date | likes_count | rating | review_text
+-----+-----+-----+-----+-----+-----+-----+
| 35 | john_doe | John Doe | Inception | https://www.themoviedb.org/t/p/w780/cdVcZxWJ9upOsyzY6ImOhkBt.jpg | 2024-01-15 | 0 | 5.0 | Absolutely mind-blowing! Inception is a masterpiece of storytelling. The layers of dreams within dreams keep you engaged throughout. Nolan at his absolute best!
| 38 | jane_smith | Jane Smith | Inception | https://www.themoviedb.org/t/p/w780/edv5CzWJ9upOsyzY6ImOhkBt.jpg | 2024-02-10 | 0 | 4.5 | Complex and engaging narrative that needs multiple viewings to fully appreciate all the layers. Spectacular visuals and Hans Zimmer's score is incredible.
| 50 | film_critic_pro | Emily Davis | Inception | https://www.themoviedb.org/t/p/w780/edv5CzWJ9upOsyzY6ImOhkBt.jpg | 2024-04-29 | 0 | 5.0 | A film that gets better with each viewing. The ending is still debated years later. Absolute genius filmmaking.
| 72 | sid | siddhartha | Inception | https://www.themoviedb.org/t/p/w780/edv5CzWJ9upOsyzY6ImOhkBt.jpg | 2025-11-18 | 0 | 5.0 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT * FROM movie_reviews_view WHERE movie_name = 'The Dark Knight';
+-----+-----+-----+-----+-----+-----+-----+
| review_id | username | user_name | movie_name | poster_url | review_date | likes_count | rating | review_text
+-----+-----+-----+-----+-----+-----+-----+
| 37 | john_doe | John Doe | The Dark Knight | https://www.themoviedb.org/t/p/w780/qJ2tWGMUDux911r6m7haRef0WH.jpg | 2024-02-05 | 0 | 4.5 | Heath Ledger's Joker is unforgettable. This film transcends the superhero genre and becomes a true crime drama. Dark, gritty, and absolutely phenomenal.
| 41 | movie_buff_23 | Mike Johnson | The Dark Knight | https://www.themoviedb.org/t/p/w780/qJ2tWGMUDux911r6m7haRef0WH.jpg | 2024-03-05 | 0 | 5.0 | The best superhero movie ever made. Dark, intense, and gripping from start to finish. Heath Ledger's performance should have won more than just an Oscar.
| 90 | john_doe | John Doe | The Dark Knight | https://www.themoviedb.org/t/p/w780/qJ2tWGMUDux911r6m7haRef0WH.jpg | 2025-11-23 | 0 | 4.5 | Amazing movie with great performances!
| 91 | john_doe | John Doe | The Dark Knight | https://www.themoviedb.org/t/p/w780/qJ2tWGMUDux911r6m7haRef0WH.jpg | 2025-11-23 | 0 | 4.5 | Amazing movie with great performances!
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> DELETE FROM Review WHERE review_id = 90;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM movie_reviews_view WHERE movie_name = 'The Dark Knight';
+-----+-----+-----+-----+-----+-----+-----+
| review_id | username | user_name | movie_name | poster_url | review_date | likes_count | rating | review_text
+-----+-----+-----+-----+-----+-----+-----+
| 37 | john_doe | John Doe | The Dark Knight | https://www.themoviedb.org/t/p/w780/qJ2tWGMUDux911r6m7haRef0WH.jpg | 2024-02-05 | 0 | 4.5 | Heath Ledger's Joker is unforgettable. This film transcends the superhero genre and becomes a true crime drama. Dark, gritty, and absolutely phenomenal.
| 41 | movie_buff_23 | Mike Johnson | The Dark Knight | https://www.themoviedb.org/t/p/w780/qJ2tWGMUDux911r6m7haRef0WH.jpg | 2024-03-05 | 0 | 5.0 | The best superhero movie ever made. Dark, intense, and gripping from start to finish. Heath Ledger's performance should have won more than just an Oscar.
| 91 | john_doe | John Doe | The Dark Knight | https://www.themoviedb.org/t/p/w780/qJ2tWGMUDux911r6m7haRef0WH.jpg | 2025-11-23 | 0 | 4.5 | Amazing movie with great performances!
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

Episodes table

```

mysql> INSERT INTO Episode (show_id, season_number, episode_no, title, ep_descr, duration, air_date)
-> VALUES (1, 2, 1, 'Seven Thirty-Severen', 'Walt and Jesse deal with the aftermath', 47, '2009-03-08');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Episode WHERE show_id = 1 ORDER BY season_number, episode_no;
+-----+-----+-----+-----+-----+-----+-----+
| episode_id | show_id | season_number | episode_no | ep_descr | duration | air_date | title
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 1 | Walt White is diagnosed with lung cancer and turns to manufacturing meth to secure his family financial future | 58 | 2008-01-20 | Pilot
| 26 | 1 | 1 | 1 | Walt White is diagnosed with lung cancer and turns to manufacturing meth to secure his family financial future | 58 | 2008-01-20 | Pilot
| 2 | 1 | 1 | 2 | Walt and Jesse attempt to dispose of the two bodies in the RV which becomes increasingly complicated | 48 | 2008-01-27 | Cats in the Bag
| 27 | 1 | 1 | 2 | Walt and Jesse attempt to dispose of the two bodies in the RV which becomes increasingly complicated | 48 | 2008-01-27 | Cats in the Bag
| 3 | 1 | 1 | 3 | Walt is struggling with the fact that he must kill Krazy-8 | 48 | 2008-02-10 | And the Bags in the River
| 28 | 1 | 1 | 3 | Walt is struggling with the fact that he must kill Krazy-8 | 48 | 2008-02-10 | And the Bags in the River
| 60 | 1 | 2 | 1 | Walt and Jesse deal with the aftermath | 47 | 2009-03-08 | Seven Thirty-Severen
| 4 | 1 | 5 | 16 | Series Finale: Walt takes care of loose ends before facing his fate | 55 | 2013-09-29 | Felina
| 29 | 1 | 5 | 16 | Series Finale: Walt takes care of loose ends before facing his fate | 55 | 2013-09-29 | Felina
+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

```

```

mysql> UPDATE Episode SET title='Updated Title', duration=50 WHERE episode_id = 4;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Episode WHERE show_id = 1 ORDER BY season_number, episode_no;
+-----+-----+-----+-----+-----+
| episode_id | show_id | season_number | episode_no | ep_descr |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 1 | 1 | Walter White is diagnosed with lung cancer and turns to manufacturing meth to secure his family financial future |
| 26 | 1 | 1 | 1 | 1 | Walt and Jesse attempt to dispose of the two bodies in the RV which becomes increasingly complicated |
| 27 | 1 | 1 | 1 | 2 | Walt and Jesse attempt to dispose of the two bodies in the RV which becomes increasingly complicated |
| 3 | 1 | 1 | 1 | 3 | Walt is struggling with the fact that he must kill Krazy-8 |
| 28 | 1 | 1 | 1 | 3 | Walt is struggling with the fact that he must kill Krazy-8 |
| 60 | 1 | 1 | 2 | 1 | Walt and Jesse deal with the aftermath |
| 4 | 1 | 1 | 5 | 16 | Series finale: Walt takes care of loose ends before facing his fate |
| 29 | 1 | 1 | 5 | 16 | Series finale: Walt takes care of loose ends before facing his fate |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

+-----+-----+-----+-----+-----+
| episode_id | show_id | season_number | episode_no | ep_descr |
+-----+-----+-----+-----+-----+
| 20 | 1 | 4 | 1 | 3 | Walt is struggling with the fact that he must kill Krazy-8 |
| 60 | 1 | 1 | 2 | 1 | Walt and Jesse deal with the aftermath |
| 4 | 1 | 1 | 5 | 16 | Series finale: Walt takes care of loose ends before facing his fate |
| 29 | 1 | 1 | 5 | 16 | Series finale: Walt takes care of loose ends before facing his fate |
+-----+-----+-----+-----+-----+
rows in set (0.00 sec)

mysql> DELETE FROM Episode WHERE episode_id = 29;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Episode WHERE show_id = 1 ORDER BY season_number, episode_no;
+-----+-----+-----+-----+-----+
| episode_id | show_id | season_number | episode_no | ep_descr |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 1 | 1 | Walter White is diagnosed with lung cancer and turns to manufacturing meth to secure his family financial future |
| 26 | 1 | 1 | 1 | 1 | Walt and Jesse attempt to dispose of the two bodies in the RV which becomes increasingly complicated |
| 27 | 1 | 1 | 1 | 2 | Walt and Jesse attempt to dispose of the two bodies in the RV which becomes increasingly complicated |
| 3 | 1 | 1 | 1 | 3 | Walt is struggling with the fact that he must kill Krazy-8 |
| 28 | 1 | 1 | 1 | 3 | Walt is struggling with the fact that he must kill Krazy-8 |
| 60 | 1 | 1 | 2 | 1 | Walt and Jesse deal with the aftermath |
| 4 | 1 | 1 | 5 | 16 | Series finale: Walt takes care of loose ends before facing his fate |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

```

Genre Table

```

mysql> INSERT INTO Genre (name, description) VALUES ('Superhero', 'Movies featuring superheroes and comic book adaptations');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Genre;
+-----+-----+-----+
| genre_id | name | description |
+-----+-----+-----+
| 1 | Action | High-energy films with physical stunts and chase sequences |
| 2 | Drama | Character-driven stories with emotional depth |
| 3 | Comedy | Films designed to make audiences laugh |
| 4 | Sci-Fi | Science fiction with futuristic or space themes |
| 5 | Thriller | Suspenseful films that keep audiences on edge |
| 6 | Horror | Films designed to frighten and create suspense |
| 7 | Romance | Stories centered around love and relationships |
| 8 | Crime | Films about criminal activity and law enforcement |
| 9 | Fantasy | Films with magical or supernatural elements |
| 10 | Animation | Animated films and shows |
| 11 | Documentary | Non-fiction films documenting real events |
| 12 | Adventure | Action-oriented stories with exploration themes |
| 13 | Mystery | Films centered around solving puzzles or crimes |
| 14 | Western | Films set in the American Old West |
| 15 | Musical | Films featuring song and dance numbers |
| 16 | Superhero | Movies featuring superheroes and comic book adaptations |
+-----+-----+-----+
16 rows in set (0.00 sec)

mysql> UPDATE Genre SET description='Updated description for genre ' WHERE genre_id = 5;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT * FROM Genre;
+-----+-----+-----+
| genre_id | name | description |
+-----+-----+-----+
| 1 | Action | High-energy films with physical stunts and chase sequences |
| 2 | Drama | Character-driven stories with emotional depth |
| 3 | Comedy | Films designed to make audiences laugh |
| 4 | Sci-Fi | Science fiction with futuristic or space themes |
| 5 | Thriller | Updated description for genre |
| 6 | Horror | Films designed to frighten and create suspense |
| 7 | Romance | Stories centered around love and relationships |
| 8 | Crime | Films about criminal activity and law enforcement |
| 9 | Fantasy | Films with magical or supernatural elements |
| 10 | Animation | Animated films and shows |
| 11 | Documentary | Non-fiction films documenting real events |
| 12 | Adventure | Action-oriented stories with exploration themes |
| 13 | Mystery | Films centered around solving puzzles or crimes |
| 14 | Western | Films set in the American Old West |
| 15 | Musical | Films featuring song and dance numbers |
| 16 | Superhero | Movies featuring superheroes and comic book adaptations |
+-----+-----+-----+
16 rows in set (0.00 sec)

mysql> DELETE FROM Genre WHERE genre_id = 16;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Genre;
+-----+-----+-----+
| genre_id | name | description |
+-----+-----+-----+
| 1 | Action | High-energy films with physical stunts and chase sequences |
| 2 | Drama | Character-driven stories with emotional depth |
| 3 | Comedy | Films designed to make audiences laugh |
| 4 | Sci-Fi | Science fiction with futuristic or space themes |
| 5 | Thriller | Updated description for genre |
| 6 | Horror | Films designed to frighten and create suspense |
| 7 | Romance | Stories centered around love and relationships |
| 8 | Crime | Films about criminal activity and law enforcement |
| 9 | Fantasy | Films with magical or supernatural elements |
| 10 | Animation | Animated films and shows |
| 11 | Documentary | Non-fiction films documenting real events |
| 12 | Adventure | Action-oriented stories with exploration themes |
| 13 | Mystery | Films centered around solving puzzles or crimes |
| 14 | Western | Films set in the American Old West |
| 15 | Musical | Films featuring song and dance numbers |
+-----+-----+-----+
15 rows in set (0.00 sec)

```

Actors table

```
mysql> INSERT INTO Actor (name, dob, gender, bio, profile_image_url)
-> VALUES ('Margot Robbie', '1990-07-02', 'Female', 'Australian actress and producer', 'https://image.tmdb.org/t/p/w780/actor.jpg');
Query OK, 1 row affected (0.01 sec)

mysql> select * from actor;
+----+-----+-----+-----+-----+
| actor_id | name | dob | gender | bio |
+----+-----+-----+-----+-----+
| 1 | Leonardo DiCaprio | 1974-11-11 | Male | American actor and film producer, Oscar winner |
| 2 | Tom Hanks | 1956-07-09 | Male | American actor and filmmaker, two-time Oscar winner |
| 3 | Scarlett Johansson | 1984-11-22 | Female | American actress and singer |
| 4 | Robert Downey Jr. | 1965-04-04 | Male | American actor and producer, known for Iron Man |
| 5 | Morgan Freeman | 1937-06-01 | Male | American actor and narrator, Oscar winner |
| 6 | Meryl Streep | 1949-06-22 | Female | American actress with record Oscar nominations |
| 7 | Denzel Washington | 1954-12-28 | Male | American actor and director, two-time Oscar winner |
| 8 | Cate Blanchett | 1969-05-14 | Female | Australian actress, two-time Oscar winner |
| 9 | Christian Bale | 1974-01-30 | Male | British actor known for method acting |
| 10 | Natalie Portman | 1981-06-09 | Female | Israeli-American actress, Oscar winner |
| 11 | Brad Pitt | 1963-12-18 | Male | American actor and producer, Oscar winner |
| 12 | Jennifer Lawrence | 1990-08-15 | Female | American actress, Oscar winner |
| 13 | Tom Cruise | 1962-07-03 | Male | American actor and producer known for action films |
| 14 | Emma Stone | 1988-11-06 | Female | American actress, Oscar winner |
| 15 | Ryan Gosling | 1980-11-12 | Male | Canadian actor and musician |
| 16 | Christian Bale | 1974-01-30 | Male | British actor known for dramatic transformations |
| 17 | Emma Stone | 1988-11-06 | Female | Academy Award winner known for La La Land |
| 18 | Ryan Gosling | 1980-11-12 | Male | Canadian actor and musician |
| 19 | Jennifer Lawrence | 1990-08-15 | Female | Oscar winner for Silver Linings Playbook |
| 20 | Brad Pitt | 1963-12-18 | Male | American actor and film producer |
| 21 | Natalie Portman | 1981-06-09 | Female | Israeli-American actress and director |
| 22 | Matt Damon | 1970-10-08 | Male | Actor and screenwriter |
| 23 | Cate Blanchett | 1969-05-14 | Female | Australian actress with two Oscars |
| 24 | Hugh Jackman | 1968-10-12 | Male | Australian actor known for Wolverine |
| 25 | Anne Hathaway | 1982-11-12 | Female | Oscar winner for Les Misérables |
| 26 | Michael B. Jordan | 1987-02-09 | Male | Actor known for Creed and Black Panther |
| 27 | Timothée Chalamet | 1995-12-27 | Male | Young actor known for Call Me by Your Name |
| 28 | Zendaya | 1996-09-01 | Female | Actress and singer |
| 29 | Florence Pugh | 1996-01-03 | Female | British actress |
| 30 | Oscar Isaac | 1979-03-09 | Male | Guatemalan-American actor |
| 31 | Margot Robbie | 1990-07-02 | Female | Australian actress and producer |
| 32 | Chris Evans | 1981-06-13 | Male | Known for Captain America role |
| 33 | Samuel L. Jackson | 1948-12-21 | Male | Prolific American actor |
| 34 | Viola Davis | 1965-08-11 | Female | Triple Crown of Acting winner |
| 35 | Daniel Kaluuya | 1989-02-24 | Male | British actor, Oscar winner for Judas and the Black Messiah |
| 36 | Margot Robbie | 1990-07-02 | Female | Australian actress and producer |
+----+-----+-----+-----+-----+
36 rows in set (0.00 sec)
```

```
mysql> UPDATE Actor SET bio='Updated biography' WHERE actor_id = 36;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from actor;
+----+-----+-----+-----+-----+
| actor_id | name | dob | gender | bio |
+----+-----+-----+-----+-----+
| 1 | Leonardo DiCaprio | 1974-11-11 | Male | American actor and film producer, Oscar winner |
| 2 | Tom Hanks | 1956-07-09 | Male | American actor and filmmaker, two-time Oscar winner |
| 3 | Scarlett Johansson | 1984-11-22 | Female | American actress and singer |
| 4 | Robert Downey Jr. | 1965-04-04 | Male | American actor and producer, known for Iron Man |
| 5 | Morgan Freeman | 1937-06-01 | Male | American actor and narrator, Oscar winner |
| 6 | Meryl Streep | 1949-06-22 | Female | American actress with record Oscar nominations |
| 7 | Denzel Washington | 1954-12-28 | Male | American actor and director, two-time Oscar winner |
| 8 | Cate Blanchett | 1969-05-14 | Female | Australian actress, two-time Oscar winner |
| 9 | Christian Bale | 1974-01-30 | Male | British actor known for method acting |
| 10 | Natalie Portman | 1981-06-09 | Female | Israeli-American actress, Oscar winner |
| 11 | Brad Pitt | 1963-12-18 | Male | American actor and producer, Oscar winner |
| 12 | Jennifer Lawrence | 1990-08-15 | Female | American actress, Oscar winner |
| 13 | Tom Cruise | 1962-07-03 | Male | American actor and producer known for action films |
| 14 | Emma Stone | 1988-11-06 | Female | American actress, Oscar winner |
| 15 | Ryan Gosling | 1980-11-12 | Male | Canadian actor and musician |
| 16 | Christian Bale | 1974-01-30 | Male | British actor known for dramatic transformations |
| 17 | Emma Stone | 1988-11-06 | Female | Academy Award winner known for La La Land |
| 18 | Ryan Gosling | 1980-11-12 | Male | Canadian actor and musician |
| 19 | Jennifer Lawrence | 1990-08-15 | Female | Oscar winner for Silver Linings Playbook |
| 20 | Brad Pitt | 1963-12-18 | Male | American actor and film producer |
| 21 | Natalie Portman | 1981-06-09 | Female | Israeli-American actress and director |
| 22 | Matt Damon | 1970-10-08 | Male | Actor and screenwriter |
| 23 | Cate Blanchett | 1969-05-14 | Female | Australian actress with two Oscars |
| 24 | Hugh Jackman | 1968-10-12 | Male | Australian actor known for Wolverine |
| 25 | Anne Hathaway | 1982-11-12 | Female | Oscar winner for Les Misérables |
| 26 | Michael B. Jordan | 1987-02-09 | Male | Actor known for Creed and Black Panther |
| 27 | Timothée Chalamet | 1995-12-27 | Male | Young actor known for Call Me by Your Name |
| 28 | Zendaya | 1996-09-01 | Female | Actress and singer |
| 29 | Florence Pugh | 1996-01-03 | Female | British actress |
| 30 | Oscar Isaac | 1979-03-09 | Male | Guatemalan-American actor |
| 31 | Margot Robbie | 1990-07-02 | Female | Australian actress and producer |
| 32 | Chris Evans | 1981-06-13 | Male | Known for Captain America role |
| 33 | Samuel L. Jackson | 1948-12-21 | Male | Prolific American actor |
| 34 | Viola Davis | 1965-08-11 | Female | Triple Crown of Acting winner |
| 35 | Daniel Kaluuya | 1989-02-24 | Male | British actor, Oscar winner for Judas and the Black Messiah |
| 36 | Margot Robbie | 1990-07-02 | Female | Updated biography |
+----+-----+-----+-----+-----+
36 rows in set (0.00 sec)
```

actor_id	name	dob	gender	bio	profile_image_url
1	Leonardo DiCaprio	1974-11-11	Male	American actor and film producer, Oscar winner	https://example.com/dicaprio.jpg
2	Tom Hanks	1956-07-09	Male	American actor and filmmaker, two-time Oscar winner	https://example.com/hanks.jpg
3	Scarlett Johansson	1984-11-22	Female	American actress and singer	https://example.com/johansson.jpg
4	Robert Downey Jr.	1965-04-04	Male	American actor and producer, known for Iron Man	https://example.com/rdj.jpg
5	Morgan Freeman	1937-06-01	Male	American actor and narrator, Oscar winner	https://example.com/freeman.jpg
6	Meryl Streep	1949-06-22	Female	American actress with record Oscar nominations	https://example.com/streep.jpg
7	Denzel Washington	1954-12-28	Male	American actor and director, two-time Oscar winner	https://example.com/washington.jpg
8	Cate Blanchett	1969-05-14	Female	Australian actress, two-time Oscar winner	https://example.com/blanchett.jpg
9	Christian Bale	1974-01-30	Male	British actor known for method acting	https://example.com/bale.jpg
10	Natalie Portman	1981-06-09	Female	Israeli-American actress, Oscar winner	https://example.com/portman.jpg
11	Brad Pitt	1963-12-18	Male	American actor and producer, Oscar winner	https://example.com/pitt.jpg
12	Jennifer Lawrence	1990-08-15	Female	American actress, Oscar winner	https://example.com/lawrence.jpg
13	Tom Cruise	1962-07-03	Male	American actor and producer known for action films	https://example.com/cruise.jpg
14	Emma Stone	1988-11-08	Female	American actress, Oscar winner	https://example.com/stone.jpg
15	Ryan Gosling	1980-11-12	Male	Canadian actor and musician	https://example.com/gosling.jpg
16	Christian Bale	1974-01-30	Male	British actor known for dramatic transformations	NULL
17	Emma Stone	1988-11-06	Female	Academy Award winner known for La La Land	NULL
18	Ryan Gosling	1980-11-12	Male	Canadian actor and musician	NULL
19	Jennifer Lawrence	1990-08-15	Female	Oscar winner for Silver Linings Playbook	NULL
20	Brad Pitt	1963-12-18	Male	American actor and film producer	NULL
21	Natalie Portman	1981-06-09	Female	Israeli-American actress and director	NULL
22	Matt Damon	1970-10-08	Male	Actor and screenwriter	NULL
23	Cate Blanchett	1969-05-14	Female	Australian actress with two Oscars	NULL
24	Hugh Jackman	1968-10-12	Male	Australian actor known for Wolverine	NULL
25	Anna Hathaway	1982-11-12	Female	Oscar winner for Les Misérables	NULL
26	Michael B. Jordan	1987-02-09	Male	Actor known for Creed and Black Panther	NULL
27	Timothée Chalamet	1995-12-27	Male	Young actor known for Call Me by Your Name	NULL
28	Zendaya	1996-09-01	Female	Actress and singer	NULL
29	Florence Pugh	1996-01-03	Female	British actress	NULL
30	Oscar Isaac	1979-03-09	Male	Guatemalan-American actor	NULL
31	Margot Robbie	1990-07-02	Female	Australian actress and producer	NULL
32	Chris Evans	1981-06-13	Male	Known for Captain America role	NULL
33	Samuel L. Jackson	1948-12-21	Male	Prolific American actor	NULL
34	Viola Davis	1965-08-11	Female	Triple Crown of Acting winner	NULL
35	Daniel Kaluuya	1989-02-24	Male	British actor, Oscar winner for Judas and the Black Messiah	NULL

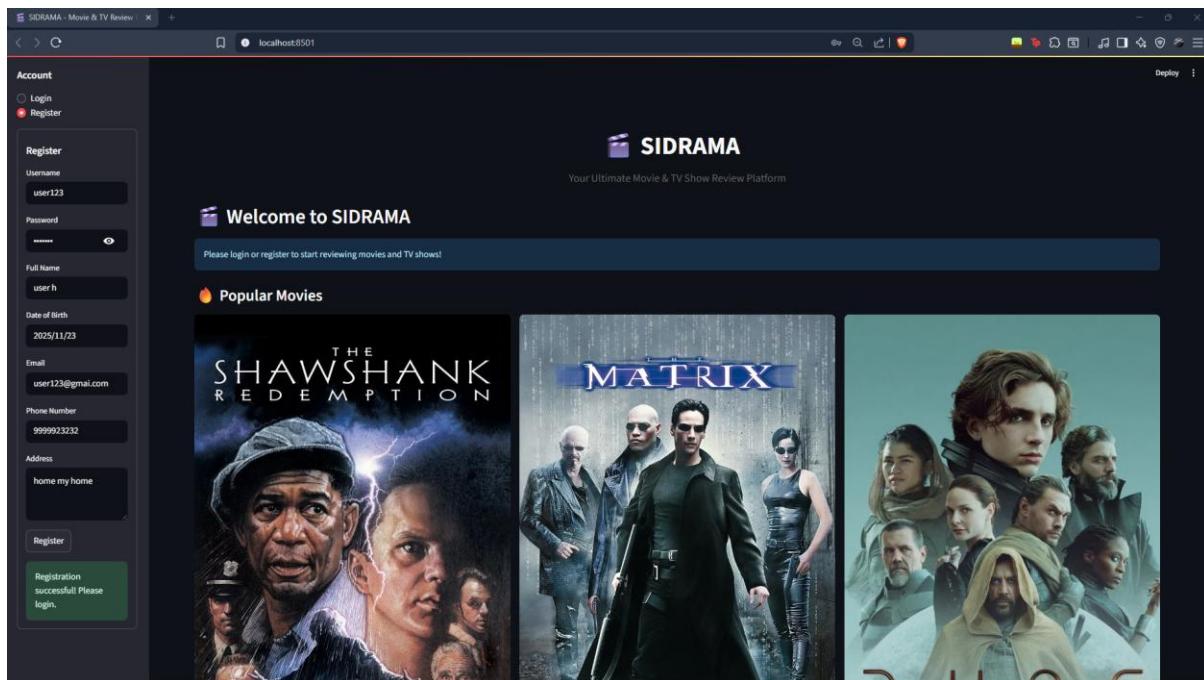
35 rows in set (0.00 sec)

Similarly can be done for Director table as well

List of functionalities/features of the application

1. User Management

- User registration and authentication
- User login/logout with session management
- User profile with statistics (total reviews, average rating, activity metrics)



Navigation

Account

- Login
- Register

Login

Username

user123

Password



[Login](#)

The image displays two screenshots of the SIDRAMA platform, a movie and TV show review platform.

Top Screenshot (Homepage):

- Navigation Bar:** Shows a dropdown menu for "user123" and links for Home, Movies, TV Shows, My Reviews, Search, Statistics, and Profile.
- Welcome Message:** "Hello, user123! Explore movies and TV shows below."
- Popular Movies Section:** Features three movie posters:
 - THE SHAWSHANK REDEMPTION**: Starring Morgan Freeman and Tim Robbins.
 - MATRIX**: Starring Keanu Reeves, Laurence Fishburne, and Carrie-Anne Moss.
 - DUNE**: Starring Timothée Chalamet, Zendaya, and Oscar Isaac.

Bottom Screenshot (User Profile):

- Navigation Bar:** Shows a dropdown menu for "rama" and links for Home, Movies, TV Shows, My Reviews, Search, Statistics, and Profile.
- User Profile Section:** "My Profile" section with a user icon.
- Profile Information:**
 - Username: rama
 - Name: srirama
 - Email: sriramavishwa@gmail.com
 - Phone: 07338101114
 - Date of Birth: 2005-03-24
 - Address: E303 Elegant Whispering Winds
- Account Statistics:**
 - Average Rating Given: **4.75/5.0**
 - Total Reviews: **6**
 - Movies Reviewed: **5**
 - Episodes Reviewed: **1**

SIDRAMA
Your Ultimate Movie & TV Show Review Platform

Statistics & Analytics

Your Activity

Total Reviews	Avg Rating Given	Movies Reviewed	Episodes Reviewed
6	4.75	5	1

Detailed Stats

Average Rating (Function): 4.75/5.0	Movies Reviewed (Function): 5
Total Review Count (Function): 6	

Platform Statistics

Top Rated Movies

- The Shawshank Redemption - 5.00 (3 reviews)
- The Matrix - 5.00 (1 reviews)
- Dune - 5.00 (2 reviews)
- La La Land - 5.00 (1 reviews)
- The Grand Budapest Hotel - 5.00 (1 reviews)

Top Rated Shows

- Breaking Bad - 5.00 (3 reviews)
- The Office - 5.00 (1 reviews)
- The Mandalorian - 5.00 (1 reviews)
- Succession - 5.00 (2 reviews)
- Stranger Things - 4.50 (2 reviews)

2. Movie Features

- Browse popular movies (sorted by ratings and review count)
- View detailed movie information (poster, description, genres, cast, directors, ratings)
- Search movies by title, genre, rating, language, and year
- Filter movies by director or actor
- View top-grossing movies by box office revenue
- View trending movies

Popular Movies

The Shawshank Redemption

Rating: 5.00/5.0
3 reviews
1994-09-23

The Matrix

Rating: 5.00/5.0
1 reviews
1999-03-31

Dune

Rating: 5.00/5.0
2 reviews
2021-10-22

The screenshot displays the SIDRAMA movie review platform interface. On the left, a dark sidebar navigation menu includes links for Home, Movies (which is the active tab), TV Shows, My Reviews, Search, Statistics, and Profile. The main content area features a header with the SIDRAMA logo and tagline "Your Ultimate Movie & TV Show Review Platform". A search bar at the top allows users to "Search movies" by entering a movie name. Below the search bar, there are filters for "Genre" (set to "All") and "Min Rating" (set to "0.00"). A list of movies is displayed, each with a title, rating, and genre information. The first few movies listed are Dune (5.00, Action, Drama, Sci-Fi), La La Land (5.00, Comedy, Drama, Romance), The Grand Budapest Hotel (5.00, Comedy, Drama), The Matrix (5.00, Action, Sci-Fi), The Shawshank Redemption (5.00, Drama), Parasite (4.88, Comedy, Drama, Thriller), Inception (4.88, Action, Sci-Fi, Thriller), Interstellar (4.75, Adventure, Drama, Sci-Fi), Pulp Fiction (4.75, Crime, Drama, Thriller), The Godfather (4.75, Crime, Drama), and The Dark Knight (4.67, Action, Crime, Thriller). The "Dune" movie page is currently selected, showing its full details. The page includes a large movie poster featuring the cast, a synopsis ("A noble family becomes embroiled in a war for control of the most valuable asset in the galaxy"), and technical specifications: Release (2021-10-22), Duration (155 min), Language (English), Box Office (\$400,671,789), and Age Rating (PG-13). A "Write a Review" button is also present. At the bottom of the movie page, there is a section for "Recent Reviews" with a single entry from user "shruj" dated 2025-11-13.

The figure consists of two vertically stacked screenshots of a movie review platform named SIDRAMA. Both screenshots show a dark-themed interface with a navigation bar on the left containing links for Home, Movies, TV Shows, My Reviews, Search, Statistics, and Profile. The top screenshot shows a search bar with 'Adventure' typed in, and a dropdown menu showing genre options: All, Action, Adventure, Animation, Comedy, Crime, and Documentary. Below the search bar, there is a list of movies: Interstellar (4.75), Dune (4.50), Avatar (4.50), and The Grand Budapest Hotel (0.00). The bottom screenshot shows a search bar with 'la la land' typed in, and a dropdown menu showing genre options: All, Action, Adventure, Animation, Comedy, Crime, and Documentary. Below the search bar, there is a list of movies: La La Land (5.00) with genres Comedy, Drama, Romance.

3. TV Show Features

- Browse top-rated TV shows
- View show details (poster, seasons, episodes, status, genres)
- View complete episode lists with individual ratings
- Episode-by-episode review tracking



The screenshots demonstrate the SIDRAMA movie review platform interface.

Top Screenshot: A list of TV shows with their names and ratings. The shows listed are:

- Breaking Bad ★ 5.00
- The Office ★ 5.00
- The Mandalorian ★ 5.00
- Succession ★ 5.00
- Stranger Things ★ 4.50
- The Crown ★ 4.50
- True Detective ★ 4.50
- The Last of Us ★ 4.00
- Game of Thrones ★ 3.83
- The Last of Us ★ 0.00
- The Crown ★ 0.00
- Succession ★ 0.00

Bottom Screenshot: A detailed view of the TV show "Breaking Bad". The page includes the show's poster, summary, and technical specifications.

Summary:

Seasons: 5 | **Episodes:** 30

Release Date: 2008-01-20

Language: English

Status: Completed

Age Rating: TV-MA

Description: A chemistry teacher diagnosed with cancer turns to cooking meth to provide for his family

View Episodes

The screenshot shows a dark-themed web application interface. On the left is a navigation sidebar with options like Home, Movies, TV Shows, My Reviews, Search, Statistics, and Profile. The main content area is titled "Episodes: Breaking Bad". It lists several episodes from Season 1 and Season 2, each with a thumbnail, title, synopsis, duration, release date, and a "Review" button.

Episode	Synopsis	Duration	Release Date	Action
S1E1 - Pilot	Walter White is diagnosed with lung cancer and turns to manufacturing meth to secure his family financial future	~58 min	2008-01-20	
S1E2 - Pilot	Walter White is diagnosed with lung cancer and turns to manufacturing meth to secure his family financial future	~58 min	2008-01-20	
S1E2 - Cats in the Bag	Walt and Jesse attempt to dispose of the two bodies in the RV which becomes increasingly complicated	~48 min	2008-01-27	
S1E2 - Cats in the Bag	Walt and Jesse attempt to dispose of the two bodies in the RV which becomes increasingly complicated	~48 min	2008-01-27	
S1E3 - And the Bags in the River	Walt is struggling with the fact that he must kill Krazy-8	~48 min	2008-02-10	
S1E3 - And the Bags in the River	Walt is struggling with the fact that he must kill Krazy-8	~48 min	2008-02-10	
S2E1 - Seven Thirty-Seven	Walt and Jesse deal with the aftermath	~47 min	2009-03-08	

4. Review System

- Add reviews for movies with rating (0-5) and text
- Add reviews for individual TV episodes
- View all reviews for any movie or episode
- View personal review history
- Edit and delete own reviews
- Automatic rating calculation via database triggers

The screenshot shows a movie detail page for "The Lord of the Rings: The Fellowship of the Ring". The page includes a large movie poster, the title, genres (Action, Drama, Fantasy), synopsis ("A meek Hobbit and companions set out to destroy a powerful ring"), release information (2001-12-19, 178 min), language (English), box office (\$871,368,364), age rating (PG-13), and a "Write a Review" button.

👉 Write a Review for: The Lord of the Rings: The Fellowship of the Ring

Your Rating ★

0.00 5.00

Your Review

Brilliant movie wonderful world building

Submit Review Cancel

The Lord of the Rings: The Fellowship of the Ring

Genres: Action Drama Fantasy

Synopsis:

A meek Hobbit and companions set out to destroy a powerful ring

📅 Release: 2001-12-19

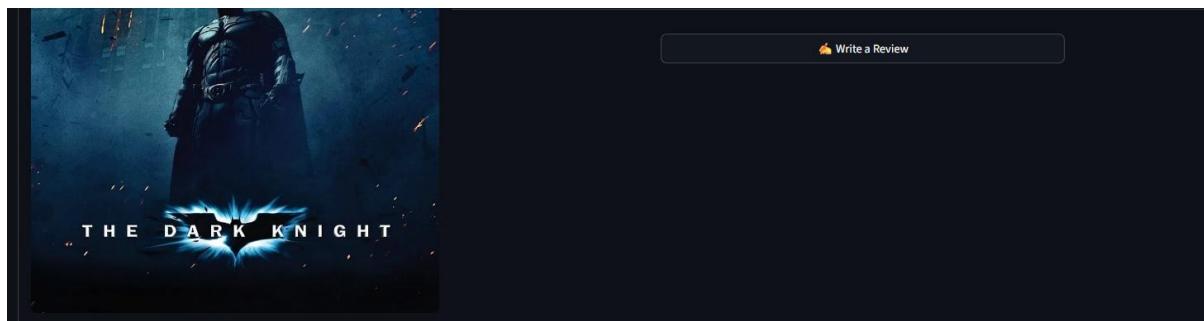
⌚ Duration: 178 min

🌐 Language: English

💰 Box Office: \$871,368,364

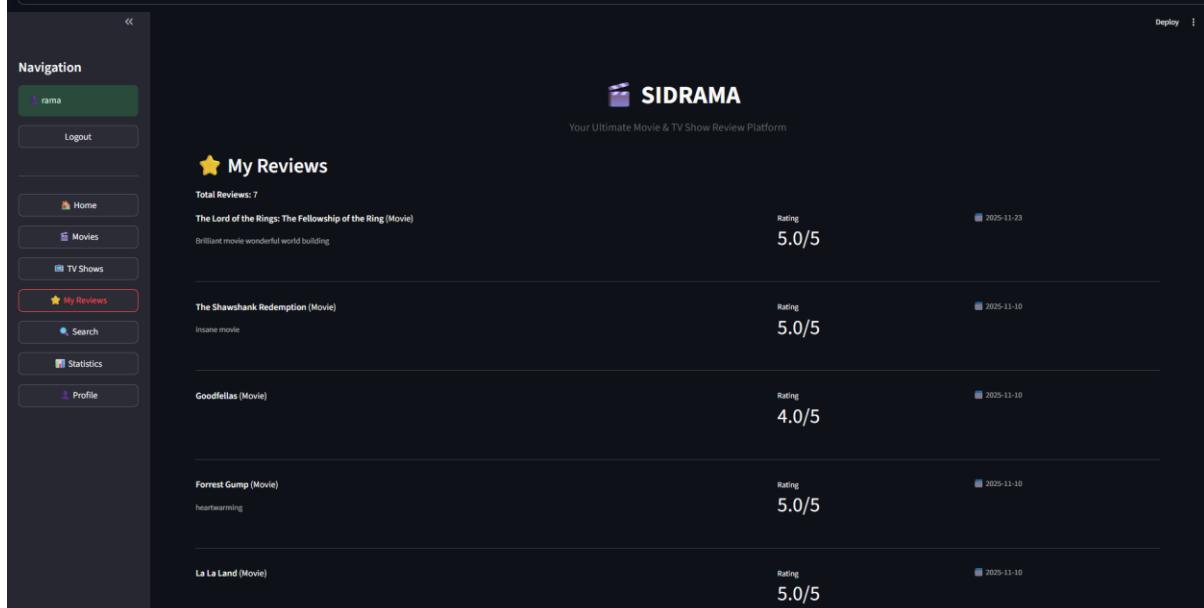
🔞 Age Rating: PG-13

⭐ Rating: 4.75/5.0



Recent Reviews

Reviewer	Date	Review Content	Rating
john_doe	2025-11-23	Amazing movie with great performances!	4.5/5
movie_buff_23	2024-03-05	The best superhero movie ever made. Dark, intense, and gripping from start to finish. Heath Ledgers performance should have won more than just an Oscar.	5.0/5
john_doe	2024-02-05	Heath Ledgers Joker is unforgettable. This film transcends the superhero genre and becomes a true crime drama. Dark, gritty, and absolutely phenomenal.	4.5/5



SIDRAMA
Your Ultimate Movie & TV Show Review Platform

My Reviews

Movie Title	Description	Rating	Date
The Lord of the Rings: The Fellowship of the Ring (Movie)	Brilliant movie wonderful world building	5.0/5	2025-11-23
The Shawshank Redemption (Movie)	Insane movie	5.0/5	2025-11-10
Goodfellas (Movie)		4.0/5	2025-11-10
Forrest Gump (Movie)	heartwarming	5.0/5	2025-11-10
La La Land (Movie)		5.0/5	2025-11-10

5. Search & Discovery

- Advanced search with multiple filters (title, genre, rating, language, year range)
- Quick search for movies and episodes by name
- Browse by genre, director, or actor
- View movies by specific criteria

The screenshot displays the SIDRAMA website interface, specifically the Advanced Search section. The top navigation bar includes links for Home, Logout, and various user reviews. The main header features the SIDRAMA logo and tagline: "Your Ultimate Movie & TV Show Review Platform". The Advanced Search section is titled "Advanced Search" and offers three search options: "By Genre", "By Director", and "By Actor". The "By Genre" tab is currently selected, showing a dropdown menu for "Select Genre" with "Comedy" chosen. Below this is a search bar with the placeholder "Search by Genre". A list of 7 movies found in the Comedy genre is displayed, each with a title, release date, and rating (e.g., La La Land (2016-12-09) ★ 5.00). The page then transitions to a "Search Movies by Director" section, where the user has entered "Christopher Nolan" into the search input field. The results show 3 movies directed by Christopher Nolan, with titles, release dates, and ratings (e.g., Interstellar (2014-11-07) ★ 4.75, Inception (2010-07-16) ★ 4.88, The Dark Knight (2008-07-18) ★ 4.67).

Advanced Search

By Genre By Director By Actor

Search Movies by Actor

Enter Actor Name

Found 3 movies:

- **The Shape of Water** (2017-12-22) ★ 4.50
Actor: Tom Hanks
- **The Shawshank Redemption** (1994-09-23) ★ 5.00
Actor: Tom Hanks
- **Forrest Gump** (1994-07-06) ★ 4.50
Actor: Tom Hanks

Message when no matching results

Advanced Search

By Genre By Director By Actor

Search Movies by Actor

Enter Actor Name

Search by Actor

No movies found for this actor.

6. Database Features (Backend)

- **8 SQL Views** for aggregated data (popular movies, user stats, movie details, show details, reviews, recent activity)
- **7 Triggers** for automatic rating updates and data validation
- **11 Stored Procedures** for search, statistics, and data operations
- **8 Functions** for calculations (user ratings, review counts, popularity scores)
- Complete CRUD operations for all entities

9. Automatic Features

- Auto-update movie/show ratings when reviews are added/edited/deleted
- Auto-set review dates
- Maintain review counts
- Cascade delete for related data

10. User Interface

- Responsive web design with Bootstrap
- Consistent navigation across all pages
- Flash messages for user feedback (success, error, info)
- Movie/show posters with fallback images
- Clean, modern layout

Triggers, Procedures/Functions, Nested query, Join, Aggregate queries

1. TRIGGERS

Trigger 1: Update Movie Rating After Insert

```
CREATE TRIGGER update_movie_stats_insert
AFTER INSERT ON Review
FOR EACH ROW
BEGIN
IF NEW.movie_id IS NOT NULL THEN
UPDATE Movie
SET ratings = (SELECT AVG(rating) FROM Review WHERE movie_id = NEW.movie_id),
total_reviews = (SELECT COUNT(*) FROM Review WHERE movie_id = NEW.movie_id)
WHERE movie_id = NEW.movie_id;
END IF;
END;
```

Purpose: Automatically recalculates movie average rating and review count when new review is added

Trigger 2: Update Movie Rating After Update

```
CREATE TRIGGER update_movie_stats_update
AFTER UPDATE ON Review
```

```

FOR EACH ROW
BEGIN
IF NEW.movie_id IS NOT NULL THEN
UPDATE Movie
SET ratings = (SELECT AVG(rating) FROM Review WHERE movie_id = NEW.movie_id),
total_reviews = (SELECT COUNT(*) FROM Review WHERE movie_id = NEW.movie_id)
WHERE movie_id = NEW.movie_id;
END IF;
END;

```

Purpose: Recalculates rating when review is modified

Trigger 3: Update Movie Rating After Delete

```

CREATE TRIGGER update_movie_stats_delete
AFTER DELETE ON Review
FOR EACH ROW
BEGIN
IF OLD.movie_id IS NOT NULL THEN
UPDATE Movie
SET ratings = COALESCE((SELECT AVG(rating) FROM Review WHERE movie_id =
OLD.movie_id), 0.00),
total_reviews = (SELECT COUNT(*) FROM Review WHERE movie_id = OLD.movie_id)
WHERE movie_id = OLD.movie_id;
END IF;
END;

```

Purpose: Updates rating when review is removed

Trigger 4: Update Show Rating After Insert

```

CREATE TRIGGER update_show_rating_insert
AFTER INSERT ON Review
FOR EACH ROW
BEGIN
IF NEW.episode_id IS NOT NULL THEN
UPDATE tvshow s
SET s.ratings = (SELECT AVG(r.rating) FROM Review r JOIN Episode e ON r.episode_id =
e.episode_id WHERE e.show_id = (SELECT show_id FROM Episode WHERE episode_id =
NEW.episode_id))
WHERE s.show_id = (SELECT show_id FROM Episode WHERE episode_id =
NEW.episode_id);
END IF;
END;

```

Purpose: Updates TV show rating when episode review is added

Trigger 5: Update Show Rating After Update

```

CREATE TRIGGER update_show_rating_update
AFTER UPDATE ON Review
FOR EACH ROW
BEGIN
IF NEW.episode_id IS NOT NULL THEN

```

```

UPDATE tvshow s
SET s.ratings = (SELECT AVG(r.rating) FROM Review r JOIN Episode e ON r.episode_id =
e.episode_id WHERE e.show_id = (SELECT show_id FROM Episode WHERE episode_id =
NEW.episode_id))
WHERE s.show_id = (SELECT show_id FROM Episode WHERE episode_id =
NEW.episode_id);
END IF;
END;

```

Purpose: Updates show rating when episode review is modified

Trigger 6: Update Show Rating After Delete

```

CREATE TRIGGER update_show_rating_delete
AFTER DELETE ON Review
FOR EACH ROW
BEGIN
IF OLD.episode_id IS NOT NULL THEN
UPDATE tvshow s
SET s.ratings = COALESCE((SELECT AVG(r.rating) FROM Review r JOIN Episode e ON
r.episode_id = e.episode_id WHERE e.show_id = (SELECT show_id FROM Episode
WHERE episode_id = OLD.episode_id)), 0.00)
WHERE s.show_id = (SELECT show_id FROM Episode WHERE episode_id =
OLD.episode_id);
END IF;
END;

```

Purpose: Updates show rating when episode review is deleted

Trigger 7: Auto-Set Review Date

```

CREATE TRIGGER set_review_date
BEFORE INSERT ON Review
FOR EACH ROW
BEGIN
IF NEW.date IS NULL THEN
SET NEW.date = CURDATE();
END IF;
END;

```

Purpose: Automatically sets current date if not provided

2. STORED PROCEDURES

Procedure 1: Add Movie Review

```

CREATE PROCEDURE add_movie_review(
IN p_user_id INT,
IN p_movie_id INT,
IN p_rating DECIMAL(2,1),
IN p_review_text TEXT
)
BEGIN
DECLARE EXIT HANDLER FOR SQLEXCEPTION

```

```

BEGIN
ROLLBACK;
SELECT 'Error: Review could not be added' AS error_message;
END;

text

START TRANSACTION;

INSERT INTO Review (user_id, movie_id, date, rating, review_text)
VALUES (p_user_id, p_movie_id, CURDATE(), p_rating, p_review_text);

COMMIT;

```

```

SELECT 'Review added successfully' AS message, LAST_INSERT_ID() AS review_id;
END;

```

Purpose: Inserts new movie review with error handling and transaction management

Procedure 2: Add Episode Review

```

CREATE PROCEDURE add_episode_review(
IN p_user_id INT,
IN p_episode_id INT,
IN p_rating DECIMAL(2,1),
IN p_review_text TEXT
)
BEGIN
DECLARE EXIT HANDLER FOR SQLEXCEPTION
BEGIN
ROLLBACK;
SELECT 'Error: Review could not be added' AS error_message;
END;

text

START TRANSACTION;

INSERT INTO Review (user_id, episode_id, date, rating, review_text)
VALUES (p_user_id, p_episode_id, CURDATE(), p_rating, p_review_text);

COMMIT;

```

```

SELECT 'Episode review added successfully' AS message, LAST_INSERT_ID() AS
review_id;
END;

```

Purpose: Inserts new episode review with validation

Procedure 3: Get User Reviews

```

CREATE PROCEDURE get_user_reviews(IN p_user_id INT)
BEGIN
SELECT r.review_id, COALESCE(m.name, CONCAT(s.name, ' - S', e.season_number, 'E',
e.episode_no)) AS content_name, CASE WHEN r.movie_id IS NOT NULL THEN 'Movie'
ELSE 'TV Show Episode' END AS content_type, r.rating, r.review_text, r.date, r.likes_count
FROM Review r
LEFT JOIN Movie m ON r.movie_id = m.movie_id
LEFT JOIN Episode e ON r.episode_id = e.episode_id
LEFT JOIN tvshow s ON e.show_id = s.show_id
WHERE r.user_id = p_user_id
ORDER BY r.date DESC;
END;

```

Purpose: Returns all reviews written by specific user with content details

Procedure 4: Search Movies by Genre

```

CREATE PROCEDURE search_movies_by_genre(IN p_genre_name VARCHAR(100))
BEGIN
SELECT m.movie_id, m.name, m.release_date, m.ratings, m.total_reviews, m.language,
m.poster_url
FROM Movie m
JOIN Movie_Genre mg ON m.movie_id = mg.movie_id
JOIN Genre g ON mg.genre_id = g.genre_id
WHERE g.name = p_genre_name
ORDER BY m.ratings DESC, m.total_reviews DESC;
END;

```

Purpose: Filters and returns movies of specified genre sorted by rating

Procedure 5: Get Movies by Director

```

CREATE PROCEDURE get_movies_by_director(IN p_director_name VARCHAR(100))
BEGIN
SELECT m.movie_id, m.name, m.release_date, m.ratings, m.total_reviews, m.poster_url,
d.name AS director_name, d.profile_image_url AS director_image
FROM Movie m
JOIN Movie_Director md ON m.movie_id = md.movie_id
JOIN Director d ON md.director_id = d.director_id
WHERE d.name LIKE CONCAT('%', p_director_name, '%')
ORDER BY m.release_date DESC;
END;

```

Purpose: Returns all movies directed by specified director with ratings

Procedure 6: Get Movies by Actor

```

CREATE PROCEDURE get_movies_by_actor(IN p_actor_name VARCHAR(100))
BEGIN
SELECT m.movie_id, m.name, m.release_date, m.ratings, m.total_reviews, m.poster_url,
a.name AS actor_name, a.profile_image_url AS actor_image, ma.character_name
FROM Movie m
JOIN Movie_Actor ma ON m.movie_id = ma.movie_id
JOIN Actor a ON ma.actor_id = a.actor_id

```

```

WHERE a.name LIKE CONCAT('%', p_actor_name, '%')
ORDER BY m.release_date DESC;
END;

```

Purpose: Returns all movies featuring specified actor with character names

Procedure 7: Get Show Details with Episodes

```

CREATE PROCEDURE get_show_details(IN p_show_id INT)
BEGIN
SELECT s.*, GROUP_CONCAT(DISTINCT g.name ORDER BY g.name SEPARATOR ',')
AS genres
FROM tvshow s
LEFT JOIN Show_Genre sg ON s.show_id = sg.show_id
LEFT JOIN Genre g ON sg.genre_id = g.genre_id
WHERE s.show_id = p_show_id
GROUP BY s.show_id;

SELECT e.episode_id, e.season_number, e.episode_no, e.title, e.ep_descr, e.duration,
e.air_date, COUNT(r.review_id) AS review_count, AVG(r.rating) AS avg_rating
FROM Episode e
LEFT JOIN Review r ON e.episode_id = r.episode_id
WHERE e.show_id = p_show_id
GROUP BY e.episode_id, e.season_number, e.episode_no, e.title, e.ep_descr, e.duration,
e.air_date
ORDER BY e.season_number, e.episode_no;
END;

```

Purpose: Returns complete show information including all episodes with review statistics

Procedure 8: Advanced Movie Search

```

CREATE PROCEDURE advanced_movie_search(
IN p_title VARCHAR(255),
IN p_genre VARCHAR(100),
IN p_min_rating DECIMAL(3,2),
IN p_language VARCHAR(50),
IN p_min_year INT,
IN p_max_year INT
)
BEGIN
SELECT DISTINCT m.movie_id, m.name, m.release_date, m.ratings, m.total_reviews,
m.language, m.poster_url, GROUP_CONCAT(DISTINCT g.name ORDER BY g.name
SEPARATOR ',') AS genres
FROM Movie m
LEFT JOIN Movie_Genre mg ON m.movie_id = mg.movie_id
LEFT JOIN Genre g ON mg.genre_id = g.genre_id
WHERE (p_title IS NULL OR m.name LIKE CONCAT('%', p_title, '%'))
AND (p_genre IS NULL OR g.name = p_genre)
AND (p_min_rating IS NULL OR m.ratings >= p_min_rating)

```

```

AND (p_language IS NULL OR m.language = p_language)
AND (p_min_year IS NULL OR YEAR(m.release_date) >= p_min_year)
AND (p_max_year IS NULL OR YEAR(m.release_date) <= p_max_year)
GROUP BY m.movie_id, m.name, m.release_date, m.ratings, m.total_reviews, m.language,
m.poster_url
ORDER BY m.ratings DESC, m.total_reviews DESC
LIMIT 50;
END;

```

Purpose: Multi-criteria search for movies with flexible filtering options

Procedure 9: Get User Statistics

```

CREATE PROCEDURE get_user_statistics(IN p_user_id INT)
BEGIN
SELECT u.user_id, u.username, u.name, u.email, COUNT(DISTINCT r.review_id) AS
total_reviews, AVG(r.rating) AS avg_rating_given, COUNT(DISTINCT CASE WHEN
r.movie_id IS NOT NULL THEN r.movie_id END) AS movies_reviewed,
COUNT(DISTINCT CASE WHEN r.episode_id IS NOT NULL THEN r.episode_id END)
AS episodes_reviewed, SUM(r.likes_count) AS total_likes_received, MIN(r.date) AS
first_review_date, MAX(r.date) AS last_review_date
FROM User u
LEFT JOIN Review r ON u.user_id = r.user_id
WHERE u.user_id = p_user_id
GROUP BY u.user_id, u.username, u.name, u.email;
END;

```

Purpose: Calculates comprehensive user activity metrics including total reviews, average ratings, and engagement data

Procedure 10: Get Top Grossing Movies

```

CREATE PROCEDURE get_top_grossing_movies(IN p_limit INT)
BEGIN
SELECT m.movie_id, m.name, m.box_office, m.release_date, m.ratings, m.total_reviews,
m.poster_url, GROUP_CONCAT(DISTINCT d.name ORDER BY d.name SEPARATOR ',')
AS directors
FROM Movie m
LEFT JOIN Movie_Director md ON m.movie_id = md.movie_id
LEFT JOIN Director d ON md.director_id = d.director_id
WHERE m.box_office IS NOT NULL
GROUP BY m.movie_id, m.name, m.box_office, m.release_date, m.ratings, m.total_reviews,
m.poster_url
ORDER BY m.box_office DESC
LIMIT p_limit;
END;

```

Purpose: Returns highest earning movies by box office revenue with director information

Procedure 11: Get Trending Movies

```

CREATE PROCEDURE get_trending_movies(IN p_days INT)
BEGIN
SELECT m.movie_id, m.name, m.ratings, m.poster_url, COUNT(r.review_id) AS

```

```

recent_review_count, AVG(r.rating) AS recent_avg_rating
FROM Movie m
JOIN Review r ON m.movie_id = r.movie_id
WHERE r.date >= DATE_SUB(CURDATE(), INTERVAL p_days DAY)
GROUP BY m.movie_id, m.name, m.ratings, m.poster_url
HAVING recent_review_count >= 2
ORDER BY recent_avg_rating DESC, recent_review_count DESC
LIMIT 20;
END;

```

Purpose: Shows recently reviewed movies with high ratings within specified time period

3. FUNCTIONS

Function 1: Get User Average Rating

```

CREATE FUNCTION get_user_avg_rating(p_user_id INT)
RETURNS DECIMAL(3,2)
DETERMINISTIC
BEGIN
DECLARE avg_rating DECIMAL(3,2);
SELECT COALESCE(AVG(rating), 0.00) INTO avg_rating FROM Review WHERE user_id
= p_user_id;
RETURN avg_rating;
END;

```

Purpose: Returns user's average rating given across all reviews

Function 2: Count User Reviews

```

CREATE FUNCTION count_user_reviews(p_user_id INT)
RETURNS INT
DETERMINISTIC
BEGIN
DECLARE review_count INT;
SELECT COUNT(*) INTO review_count FROM Review WHERE user_id = p_user_id;
RETURN review_count;
END;

```

Purpose: Returns total number of reviews by user

Function 3: Check if User Reviewed Movie

```

CREATE FUNCTION has_reviewed_movie(p_user_id INT, p_movie_id INT)
RETURNS BOOLEAN
DETERMINISTIC
BEGIN
DECLARE reviewed BOOLEAN;
SELECT EXISTS(SELECT 1 FROM Review WHERE user_id = p_user_id AND movie_id =
p_movie_id) INTO reviewed;
RETURN reviewed;
END;

```

Purpose: Returns TRUE if user has reviewed movie, FALSE otherwise

Function 4: Get User Total Likes

```
CREATE FUNCTION get_user_total_likes(p_user_id INT)
RETURNS INT
DETERMINISTIC
BEGIN
DECLARE total_likes INT;
SELECT COALESCE(SUM(likes_count), 0) INTO total_likes FROM Review WHERE
user_id = p_user_id;
RETURN total_likes;
END;
```

Purpose: Returns sum of all likes received on user's reviews

Function 5: Calculate Movie Popularity

```
CREATE FUNCTION calculate_movie_popularity(p_movie_id INT)
RETURNS DECIMAL(5,2)
DETERMINISTIC
BEGIN
DECLARE popularity DECIMAL(5,2);
DECLARE avg_rating DECIMAL(3,2);
DECLARE review_count INT;
SELECT ratings, total_reviews INTO avg_rating, review_count FROM Movie WHERE
movie_id = p_movie_id;
SET popularity = (COALESCE(avg_rating, 0) * 0.7) + (LOG10(COALESCE(review_count,
0) + 1) * 3);
RETURN COALESCE(popularity, 0.00);
END;
```

Purpose: Returns popularity score calculated using rating and logarithmic review count

Function 6: Count Movies Reviewed

```
CREATE FUNCTION count_movies_reviewed(p_user_id INT)
RETURNS INT
DETERMINISTIC
BEGIN
DECLARE movie_count INT;
SELECT COUNT(DISTINCT movie_id) INTO movie_count FROM Review WHERE
user_id = p_user_id AND movie_id IS NOT NULL;
RETURN COALESCE(movie_count, 0);
END;
```

Purpose: Returns number of distinct movies reviewed by user

Function 7: Count Episodes Reviewed

```
CREATE FUNCTION count_episodes_reviewed(p_user_id INT)
RETURNS INT
DETERMINISTIC
BEGIN
DECLARE episode_count INT;
SELECT COUNT(DISTINCT episode_id) INTO episode_count FROM Review WHERE
```

```

user_id = p_user_id AND episode_id IS NOT NULL;
RETURN COALESCE(episode_count, 0);
END;

```

Purpose: Returns number of distinct episodes reviewed by user

Function 8: Check if User Reviewed Episode

```

CREATE FUNCTION has_reviewed_episode(p_user_id INT, p_episode_id INT)
RETURNS BOOLEAN
DETERMINISTIC
BEGIN
DECLARE reviewed BOOLEAN;
SELECT EXISTS(SELECT 1 FROM Review WHERE user_id = p_user_id AND episode_id
= p_episode_id) INTO reviewed;
RETURN reviewed;
END;

```

Purpose: Returns TRUE if user has reviewed specific episode, FALSE otherwise

4. NESTED QUERIES

Example 1: Movies Not Reviewed by User

```
SELECT * FROM Movie WHERE movie_id NOT IN (SELECT movie_id FROM Review
WHERE user_id = 1);
```

Purpose: Finds all movies that a specific user has not yet reviewed by using a subquery with the NOT IN operator to exclude movies present in the user's review history.

Example 2: Users with Above Average Review Count

```

SELECT user_id, username, COUNT() as review_count
FROM User u JOIN Review r ON u.user_id = r.user_id
GROUP BY u.user_id, username
HAVING COUNT() > (SELECT AVG(review_count) FROM (SELECT COUNT(*) as
review_count FROM Review GROUP BY user_id) as avg_reviews);

```

Purpose: Identifies active users by comparing each user's review count against the average review count across all users using a nested aggregate query in the HAVING clause.

Example 3: Movies with Rating Higher Than Average

```
SELECT name, ratings FROM Movie WHERE ratings > (SELECT AVG(ratings) FROM
Movie);
```

Purpose: Selects movies whose ratings exceed the average rating of all movies in the database using a scalar subquery in the WHERE clause.

Example 4: Find Similar Users for Recommendations

```

SELECT DISTINCT r2.user_id
FROM Review r1
JOIN Review r2 ON r1.movie_id = r2.movie_id
WHERE r1.user_id = 1 AND r2.user_id != 1 AND ABS(r1.rating - r2.rating) <= 1;

```

Purpose: Discovers users with similar taste preferences by finding users who have reviewed the same movies with similar ratings using a correlated subquery with join conditions.

5. JOIN QUERIES

Example 1: INNER JOIN - Movies with Directors

```
SELECT m.name, d.name AS director_name FROM Movie m INNER JOIN Movie_Director md ON m.movie_id = md.movie_id INNER JOIN Director d ON md.director_id = d.director_id;
```

Purpose: Uses inner join to combine Movie, Movie_Director, and Director tables to display movies alongside their directors, returning only records where all relationships exist.

Example 2: LEFT JOIN - All Movies with Review Count

```
SELECT m.name, COUNT(r.review_id) AS total_reviews FROM Movie m LEFT JOIN Review r ON m.movie_id = r.movie_id GROUP BY m.movie_id, m.name;
```

Purpose: Uses left outer join to include all movies regardless of whether they have reviews, showing zero for movies without reviews using aggregate COUNT function.

Example 3: RIGHT JOIN - All Reviews with Movie Info

```
SELECT u.username, m.name, r.rating FROM Movie m RIGHT JOIN Review r ON m.movie_id = r.movie_id JOIN User u ON r.user_id = u.user_id;
```

Purpose: Uses right outer join to ensure all reviews are included even if movie information is missing, combining user and movie details with review data.

Example 4: SELF JOIN - Users with Similar Taste

```
SELECT u1.username, u2.username, m.name FROM Review r1 JOIN Review r2 ON r1.movie_id = r2.movie_id AND r1.user_id < r2.user_id JOIN User u1 ON r1.user_id = u1.user_id JOIN User u2 ON r2.user_id = u2.user_id JOIN Movie m ON r1.movie_id = m.movie_id;
```

Purpose: Performs self-join on the Review table to find pairs of users who have reviewed the same movies, useful for recommendation systems and identifying user communities.

Example 5: MULTIPLE JOINS - Complete Movie Details

```
SELECT m.name, GROUP_CONCAT(DISTINCT g.name) AS genres,
GROUP_CONCAT(DISTINCT d.name) AS directors, GROUP_CONCAT(DISTINCT a.name) AS actors
FROM Movie m
LEFT JOIN Movie_Genre mg ON m.movie_id = mg.movie_id
LEFT JOIN Genre g ON mg.genre_id = g.genre_id
LEFT JOIN Movie_Director md ON m.movie_id = md.movie_id
LEFT JOIN Director d ON md.director_id = d.director_id
LEFT JOIN Movie_Actor ma ON m.movie_id = ma.movie_id
LEFT JOIN Actor a ON ma.actor_id = a.actor_id
GROUP BY m.movie_id, m.name;
```

Purpose: Combines six tables using multiple left joins to aggregate comprehensive movie information including genres, directors, and actors using GROUP_CONCAT function for many-to-many relationships.

6. AGGREGATE QUERIES

Average Rating by Genre

```
SELECT g.name AS genre, AVG(m.ratings) AS avg_rating FROM Genre g JOIN
Movie_Genre mg ON g.genre_id = mg.genre_id JOIN Movie m ON mg.movie_id =
m.movie_id GROUP BY g.genre_id, g.name;
```

Purpose: Uses AVG aggregate function with GROUP BY to calculate average movie ratings for each genre, helping identify which genres tend to receive higher ratings.

Count Reviews per User

```
SELECT u.username, COUNT(r.review_id) AS total_reviews FROM User u LEFT JOIN
Review r ON u.user_id = r.user_id GROUP BY u.user_id, u.username ORDER BY
total_reviews DESC;
```

Purpose: Uses COUNT aggregate function with LEFT JOIN to determine user activity levels by counting total reviews submitted by each user ordered by review count.

Sum Total Box Office by Director

```
SELECT d.name, SUM(m.box_office) AS total_revenue FROM Director d JOIN
Movie_Director md ON d.director_id = md.director_id JOIN Movie m ON md.movie_id =
m.movie_id GROUP BY d.director_id, d.name ORDER BY total_revenue DESC;
```

Purpose: Uses SUM aggregate function to calculate total box office revenue earned by each director across all their films, demonstrating financial impact.

Maximum and Minimum Rated Movies

```
SELECT name, ratings FROM Movie WHERE ratings = (SELECT MAX(ratings) FROM
Movie);
SELECT name, ratings FROM Movie WHERE ratings = (SELECT MIN(ratings) FROM
Movie);
```

Purpose: Uses MAX and MIN aggregate functions in subqueries to identify the highest and lowest rated movies in the database for quality analysis.

Active Users with HAVING Clause

```
SELECT u.username, COUNT(r.review_id) AS review_count FROM User u JOIN Review r
ON u.user_id = r.user_id GROUP BY u.user_id, u.username HAVING COUNT(r.review_id)
>= 5;
```

Purpose: Uses COUNT with GROUP BY and HAVING clause to filter users who have written at least five reviews, identifying engaged community members.

Count Distinct Genres per User

```
SELECT u.username, COUNT(DISTINCT g.genre_id) AS genres_reviewed
FROM User u
JOIN Review r ON u.user_id = r.user_id
JOIN Movie m ON r.movie_id = m.movie_id
JOIN Movie_Genre mg ON m.movie_id = mg.movie_id
JOIN Genre g ON mg.genre_id = g.genre_id
GROUP BY u.user_id, u.username;
```

Purpose: Uses COUNT DISTINCT to calculate the variety of genres each user has reviewed, indicating user diversity in content consumption and breadth of interest.

Code snippets for invoking the Procedures/Functions/Trigger

```
mysql> CALL add_movie_review(1, 3, 4.5, 'Amazing movie with great cinematography!');
+-----+
| message |
+-----+
| Review added successfully |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.02 sec)

mysql> CALL add_episode_review(2, 6, 5.0, 'Best episode of the season!');
+-----+-----+
| message | review_id |
+-----+-----+
| Episode review added successfully | 95 |
+-----+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL get_user_reviews(1);
+-----+-----+-----+-----+-----+
| review_id | content_name | date | content_type | likes_count | rating | review_text |
+-----+-----+-----+-----+-----+
| 91 | The Dark Knight | Movie | 2025-11-23 | 0 | 4.5 | Amazing movie with great performances! |
| 93 | The Dark Knight | Movie | 2025-11-23 | 0 | 4.5 | Amazing movie with great cinematography! |
| 56 | Breaking Bad - S1E16 | TV Show Episode | 2024-05-15 | 0 | 5.0 | The finale we deserved. Felina wraps up every storyline perfectly. One of the best series finales in television history. |
| 55 | Breaking Bad - S1E1 | TV Show Episode | 2024-05-15 | 0 | 5.0 | Perfect pilot episode. Immediately hooks you into Walter Whites transformation. Bryan Cranston is phenomenal from the very first scene. |
| 80 | Gravity | Movie | 2024-04-06 | 0 | 4.5 | Everything Everywhere All at Once is pure creativity unleashed. Chaotic, emotional, and absolutely brilliant. |
| 75 | Parasite | Movie | 2024-04-01 | 0 | 4.5 | Dune is a visual masterpiece. Denis Villeneuve crafted an epic sci-fi experience that stays true to the book while being accessible to newcomers. |
| 37 | The Dark Knight | Movie | 2024-02-05 | 0 | 4.5 | Heath Ledgers Joker is unforgettable. This film transcends the superhero genre and becomes a true crime drama. Dark, gritty, and absolutely phenomenal. |
| 36 | The Shawshank Redemption | Movie | 2024-01-28 | 0 | 5.0 | One of the greatest films ever made. A timeless story about hope, friendship, and redemption. Morgan Freeman and the cast deliver perfect performances. |
| 35 | Inception | Movie | 2024-01-15 | 0 | 5.0 | Absolutely mind-blowing! Inception is a masterpiece of storytelling. The layers of dreams within dreams keep you engaged throughout. Nolan at his absolute best! |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

Query OK, 0 rows affected (0.16 sec)

mysql> CALL search_movies_by_genre('Action');
+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | release_date | ratings | total_reviews | language | poster_url |
+-----+-----+-----+-----+-----+-----+
| 16 | Dune | 2021-10-22 | 5.00 | 2 | English | https://image.tmdb.org/t/p/w500/dSNxSkLXo8qyIVkgV9IXAghTcIC.jpg |
| 6 | The Matrix | 1999-03-31 | 5.00 | 1 | English | https://www.themoviedb.org/t/p/w788/f893AD1e181s9GkdDpExUxKh5h.jpg |
| 1 | Inception | 2010-07-16 | 4.88 | 4 | English | https://www.themoviedb.org/t/p/w788/edv5CzVaj9puDoy2Y67wkhkBbt.jpg |
| 28 | The Lord of the Rings: The Fellowship of the Ring | 2001-12-19 | 4.75 | 2 | English | https://image.tmdb.org/t/p/w500/6com5QYQ3yTMJbnnvdk89Ch06.jpg |
| 3 | The Dark Knight | 2008-07-18 | 4.63 | 4 | English | https://www.themoviedb.org/t/p/w788/aj3jZdWfOOG91Ldrd7mHef9m.jpg |
| 19 | Avatar | 2009-12-18 | 4.60 | 0 | English | https://image.tmdb.org/t/p/w500/0u0688zLYvIwPQnqfzJy3y.jpg |
| 39 | Mad Max: Fury Road | 2015-05-15 | 4.00 | 0 | English | https://image.tmdb.org/t/p/w500/hA2plq9qkqpxw3hKWhriipsrJr.jpg |
| 31 | Avengers: Endgame | 2019-04-26 | 0.00 | 0 | English | https://image.tmdb.org/t/p/w500/or66FN3QkaStduK1e9l16pB31y.jpg |
| 33 | Spider-Man: Into the Spider-Verse | 2018-12-14 | 0.00 | 0 | English | https://image.tmdb.org/t/p/w500/iiZZddoQBEYbv6id8su7ImLboCb0.jpg |
| 34 | 1917 | 2019-12-25 | 0.00 | 0 | English | https://image.tmdb.org/t/p/w500/lzf80kyE25z1sage4SYFLCCrM19.jpg |
| 37 | Top Gun: Maverick | 2022-05-27 | 0.00 | 0 | English | https://image.tmdb.org/t/p/w500/62HCnlf7ziyWcpab0211Dx171H.jpg |
| 38 | The Batman | 2022-03-04 | 0.00 | 0 | English | https://image.tmdb.org/t/p/w500/7NxTEgt7H36fpoo58r9T25onhq.jpg |
+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

```

mysql> CALL search_movies_by_genre('Drama');
+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | release_date | ratings | total_reviews | language | poster_url |
+-----+-----+-----+-----+-----+-----+-----+
| 2 | The Shawshank Redemption | 1994-09-23 | 5.00 | 3 | English | https://www.themoviedb.org/t/p/w780/q6yGoIstsGEsmrFryDOJo3dEmqu.jpg |
| 16 | Dune | 2021-10-22 | 5.00 | 2 | English | https://image.tmbd.org/t/p/w580/d5NSXkLx0EqyIVkgV9uXAgMlCkC.jpg |
| 23 | La La Land | 2016-12-09 | 5.00 | 1 | English | https://image.tmbd.org/t/p/w580/UD0z2DhfwwoFdKsifxkUJt0RF0.jpg |
| 27 | The Grand Budapest Hotel | 2014-03-28 | 5.00 | 1 | English | https://image.tmbd.org/t/p/w580/wldWqTqeja6z6C6cqzHwpHrrPo.jpg |
| 8 | Parasite | 2019-05-30 | 4.88 | 4 | Korean | https://www.themoviedb.org/t/p/w780/71itTgloJzvG11TAywmCf6f13vT.jpg |
| 4 | Pulp Fiction | 1994-10-14 | 4.75 | 2 | English | https://www.themoviedb.org/t/p/w780/ds11Fn5o0Imszv2Bp8JPfbxD.jpg |
| 7 | Interstellar | 2014-07-09 | 4.75 | 2 | English | https://image.tmbd.org/t/p/w780/3Bhkrj58Vtu7enYsRoLD1Fzdjai.jpg |
| 9 | The Godfather | 1974-03-29 | 4.75 | 2 | English | https://www.themoviedb.org/t/p/w780/1Ai1Y5tEqICRqo64sV9z1ZvQ.jpg |
| 20 | The Lord of the Rings: The Fellowship of the Ring | 2001-12-19 | 4.75 | 2 | English | https://image.tmbd.org/t/p/w780/1Ai1Y5tEqICRqo64sV9z1ZvQ.jpg |
| 5 | Forrest Gump | 1994-07-06 | 4.50 | 2 | English | https://image.tmbd.org/t/p/w780/2Rm6NvV1SfPwDltv0c1xG8k6cnNc.jpg |
| 18 | Goodfellas | 1990-09-19 | 4.50 | 2 | English | https://www.themoviedb.org/t/p/w780/71iTglJzvG11TAywmCf6f13vT.jpg |
| 11 | Dune | 2021-10-22 | 4.50 | 1 | English | https://www.themoviedb.org/t/p/w780/ds11Fn5o0Imszv2Bp8JPfbxD.jpg |
| 18 | Fight Club | 1999-10-15 | 4.50 | 1 | English | https://image.tmbd.org/t/p/w580/888P7pSp6e6t7vZQ1bQ3PmJK.jpg |
| 28 | Gravity | 2013-10-04 | 4.50 | 1 | English | https://image.tmbd.org/t/p/w580/7fnp62ij5l13xTme25g1Lceuedo.jpg |
| 29 | The Shape of Water | 2017-12-22 | 4.50 | 1 | English | https://image.tmbd.org/t/p/w580/k4FwhlMuR5B1SYZGe20R9nZwZ.jpg |
| 13 | Titanic | 1997-12-19 | 4.25 | 2 | English | https://www.themoviedb.org/t/p/w780/9xjzS2R5Vxe8SFx8pc3aIGCOV.jpg |
| 14 | The Grand Budapest Hotel | 2014-03-28 | 0.00 | 0 | English | https://www.themoviedb.org/t/p/w780/nX5otM9ypcKanRH4fzQo1nMJ.jpg |
| 17 | The Godfather | 1972-03-24 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/3bhkrj58Vtu7enYsRoLD1Fzdjai.jpg |
| 21 | Parasite | 2019-05-30 | 0.00 | 0 | Korean | https://image.tmbd.org/t/p/w580/edv5CZvWl09upOs2Y61wkh8bT.jpg |
| 22 | Whiplash | 2014-10-10 | 0.00 | 0 | English | https://www.themoviedb.org/t/p/w780/71iTglJzvG11TAywmCf6f13vT.jpg |
| 24 | Get Out | 2017-02-24 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/7fnp62ij5l13xTme25g1Lceuedo.jpg |
| 26 | Blade Runner 2049 | 2017-10-06 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/gva2L0pYkEWjzgFLXCAVBE5.jpg |
| 32 | Joker | 2019-10-04 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/udcLJohjfjb8Ekgsd4FDftedOkU.jpg |
| 34 | 1917 | 2019-12-25 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/1zf0kyrE25zlsageHSFLCCR1M9.jpg |
| 35 | Oppenheimer | 2023-07-21 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/Gvx8vgSCU0GDyktEgv7R1n2ua.jpg |
| 37 | Top Gun: Maverick | 2022-05-27 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/62HCnUTziyWcpDab0211DX17Ljh.jpg |
+-----+-----+-----+-----+-----+-----+-----+
26 rows in set (0.00 sec)

Query OK, 0 rows affected (0.23 sec)

mysql> CALL search_movies_by_genre('Sci-Fi');
+-----+-----+-----+-----+-----+-----+
| movie_id | name | release_date | ratings | total_reviews | language | poster_url |
+-----+-----+-----+-----+-----+-----+
| 16 | Dune | 2021-10-22 | 5.00 | 2 | English | https://image.tmbd.org/t/p/w580/d5NSXkLx0EqyIVkgV9uXAgMlC.jpg |
| 6 | The Matrix | 1999-03-31 | 5.00 | 1 | English | https://www.themoviedb.org/t/p/w780/f89u3AdrioiB1s9GkdP0EpXu5kH.jpg |
| 1 | Inception | 2010-07-16 | 4.88 | 4 | English | https://www.themoviedb.org/t/p/w780/edv5CZvWl09upOs2Y61wkh8bT.jpg |
| 7 | Interstellar | 2014-11-07 | 4.75 | 2 | English | https://image.tmbd.org/t/p/w780/raiyTfGQgDCR1Iqo664sY9XZIVQ.jpg |
| 12 | Avatar | 2009-12-18 | 4.50 | 2 | English | https://www.themoviedb.org/t/p/w780/kmcqLZGaSh2oZpbtuoF0Cdnn7dT.jpg |
| 11 | Dune | 2021-10-22 | 4.50 | 1 | English | https://www.themoviedb.org/t/p/w780/ds11Fn5o0Imszv2Bp8JPfbxD.jpg |
| 28 | Gravity | 2013-10-04 | 4.50 | 1 | English | https://image.tmbd.org/t/p/w580/uWwz2BjHaixVXqgZfPwMSVkZ.jpg |
| 19 | Avatar | 2009-12-18 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/3bhkrj58Vtu7enYsRoLD1Fzdjai.jpg |
| 26 | Blade Runner 2049 | 2017-10-06 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/gjaV2L6pWzgFLXCAVBE5.jpg |
| 30 | Mad Max: Fury Road | 2015-05-15 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/hA2ple9qLqnmwpxpnhVnhrpisip.jpg |
| 31 | Avengers: Endgame | 2019-04-26 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/or96Fn3Dp5tuk1e9s1l6pB3i.jpg |
| 33 | Spider-Man: Into the Spider-Verse | 2018-12-14 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/11zzd6QBEYBvId8s71m0eChD.jpg |
| 36 | Everything Everywhere All at Once | 2022-03-25 | 0.00 | 0 | English | https://image.tmbd.org/t/p/w580/w3LxiVwMmEVn5RYq6jkb1.jpg |
+-----+-----+-----+-----+-----+-----+
36 rows in set (0.00 sec)

Query OK, 0 rows affected (0.05 sec)

mysql> CALL get_movies_by_director('Nolan');
+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | release_date | ratings | total_reviews | poster_url | director_name | director_image |
+-----+-----+-----+-----+-----+-----+-----+
| 7 | Interstellar | 2014-11-07 | 4.75 | 2 | https://image.tmbd.org/t/p/w780/rAiYfWqGDCR1Iqo664sY9XZIVQ.jpg | Christopher Nolan | https://example.com/nolan.jpg |
| 1 | Inception | 2010-07-16 | 4.88 | 4 | https://www.themoviedb.org/t/p/w780/edv5CZvWl09upOs2Y61wkh8bT.jpg | Christopher Nolan | https://example.com/nolan.jpg |
| 3 | The Dark Knight | 2008-07-18 | 4.63 | 4 | https://www.themoviedb.org/t/p/w780/qJ2tWGMUDux91lr6m7haRef0Wh.jpg | Christopher Nolan | https://example.com/nolan.jpg |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.05 sec)

mysql> CALL get_movies_by_director('Tarantino');
+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | release_date | ratings | total_reviews | poster_url | director_name | director_image |
+-----+-----+-----+-----+-----+-----+-----+
| 4 | Pulp Fiction | 1994-10-14 | 4.75 | 2 | https://www.themoviedb.org/t/p/w780/d511Fn5o0Imszv2Bp8JPfbxD.jpg | Quentin Tarantino | https://example.com/tarantino.jpg |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL get_movies_by_director('Scorsese');
+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | release_date | ratings | total_reviews | poster_url | director_name | director_image |
+-----+-----+-----+-----+-----+-----+-----+
| 2 | The Shawshank Redemption | 1994-09-23 | 5.00 | 3 | https://www.themoviedb.org/t/p/w780/q6yGoIstsGEsmrFryDOJo3dEmqu.jpg | Martin Scorsese | https://example.com/scorsese.jpg |
| 10 | Goodfellas | 1990-09-19 | 4.50 | 2 | https://image.tmbd.org/t/p/w780/3Rm6NvV1SfPwDltv0c1xG8k6cnNc.jpg | Martin Scorsese | https://example.com/scorsese.jpg |
| 9 | The Godfather | 1974-03-24 | 4.75 | 2 | https://www.themoviedb.org/t/p/w780/3Bhkrj58Vtu7enYsRoLD1Fzdjai.jpg | Martin Scorsese | https://example.com/scorsese.jpg |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.03 sec)

mysql> CALL get_movies_by_actor('Tom Hanks');
+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | release_date | ratings | total_reviews | poster_url | actor_name | actor_image | character_name |
+-----+-----+-----+-----+-----+-----+-----+
| 29 | The Shape of Water | 2017-12-22 | 4.50 | 1 | https://image.tmbd.org/t/p/w580/k4FwhlMuR5B1SYZGe20R9nZwZ.jpg | Tom Hanks | https://example.com/hanks.jpg | Pete Mitchell |
| 2 | The Shawshank Redemption | 1994-09-23 | 5.00 | 3 | https://www.themoviedb.org/t/p/w780/q6yGoIstsGEsmrFryDOJo3dEmqu.jpg | Tom Hanks | https://example.com/hanks.jpg | Andy Dufresne |
| 5 | Forrest Gump | 1994-07-06 | 4.50 | 2 | https://www.themoviedb.org/t/p/w780/saHP97zTPSSeLurLQEcaNklrsFl.jpg | Tom Hanks | https://example.com/hanks.jpg | Forrest Gump |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.05 sec)

mysql> CALL get_movies_by_actor('Morgan Freeman');
+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | release_date | ratings | total_reviews | poster_url | actor_name | actor_image | character_name |
+-----+-----+-----+-----+-----+-----+-----+
| 12 | Avatar | 2009-12-18 | 4.50 | 2 | https://www.themoviedb.org/t/p/w780/kmcqLZGaSh2oZpbtuoF0Cdnn7dT.jpg | Morgan Freeman | https://example.com/freeman.jpg | Frodo Baggins |
| 2 | The Shawshank Redemption | 1994-09-23 | 5.00 | 3 | https://www.themoviedb.org/t/p/w780/q6yGoIstsGEsmrFryDOJo3dEmqu.jpg | Morgan Freeman | https://example.com/freeman.jpg | Ellis Boyd Redding |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.03 sec)

```

```

mysql> CALL get_show_details(2);
+-----+-----+-----+-----+-----+-----+-----+-----+
| show_id | name | num_of_seasons | num_of_episodes | descr | status | genres |
+-----+-----+-----+-----+-----+-----+-----+
| 2 | Game of Thrones | 8 | 73 | Nine noble families fight for control over the lands of Westeros while an ancient enemy returns | Completed | NULL |
| https://www.themoviedb.org/t/p/w780/1X5iql89opfnbl8MnZY1O1uJx.jpg |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

+-----+-----+-----+-----+-----+-----+-----+
| episode_id | season_number | episode_no | title | ep_descr | duration | air_date | review_c |
+-----+-----+-----+-----+-----+-----+-----+
| 5 | 1 | 1 | Updated Title | Eddard Stark is torn between his family and old friend when asked to serve at the side of King Robert Baratheon | 50 | 2011-04-17 | |
| 4750000 | 30 | 1 | Winter Is Coming | Eddard Stark is torn between his family and old friend when asked to serve at the side of King Robert Baratheon | 62 | 2011-04-17 |
| 6 | NULL | 1 | 2 | The Kingsroad | While Bran recovers from his fall the Lannisters plot to ensure his silence | 56 | 2011-04-24 |
| 5000000 | 31 | 1 | 2 | The Kingsroad | While Bran recovers from his fall the Lannisters plot to ensure his silence | 56 | 2011-04-24 |
| 7 | NULL | 8 | 6 | The Iron Throne | Series finale: Destinies are fulfilled as the Great War comes to an end | 80 | 2019-05-19 |
| 2000000 | 32 | 8 | 6 | The Iron Throne | Series finale: Destinies are fulfilled as the Great War comes to an end | 80 | 2019-05-19 |
| 8 | NULL |  |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)

Query OK, 0 rows affected (0.09 sec)

mysql> CALL advanced_movie_search('Dark', NULL, 4.0, NULL, 2000, 2020);
+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | release_date | ratings | total_reviews | language | poster_url | genres |
+-----+-----+-----+-----+-----+-----+-----+
| 3 | The Dark Knight | 2008-07-18 | 4.63 | 4 | English | https://www.themoviedb.org/t/p/w780/qJ2tW6wMUDux91lr6m7haRef0WH.jpg | Action, Crime, Thriller |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.02 sec)

mysql> CALL advanced_movie_search(NULL, 'Action', 4.5, 'English', NULL, NULL);
+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | release_date | ratings | total_reviews | language | poster_url | genres |
+-----+-----+-----+-----+-----+-----+-----+
| 16 | Dune | 2021-10-22 | 5.00 | 2 | English | https://image.tmdb.org/t/p/w500/d5NSkXlyo8qyIVkgV9UXAgMickC.jpg | Action |
| 6 | The Matrix | 1999-03-31 | 5.00 | 1 | English | https://www.themoviedb.org/t/p/w780/f89U3ADrl0i1s9GkdPOExUkSH.jpg | Action |
| 1 | Inception | 2010-07-16 | 4.88 | 4 | English | https://www.themoviedb.org/t/p/w780/edv5CZwJ99upPsy2YGiDhKh8bt.jpg | Action |
| 20 | The Lord of the Rings: The Fellowship of the Ring | 2001-12-19 | 4.75 | 2 | English | https://image.tmdb.org/t/p/w500/soom5Q02yQTMJ1bnvbkBL9ch06.jpg | Action |
| 3 | The Dark Knight | 2008-07-18 | 4.63 | 4 | English | https://www.themoviedb.org/t/p/w780/qJ2tW6wMUDux91lr6m7haRef0WH.jpg | Action |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

Query OK, 0 rows affected (0.05 sec)

mysql> CALL advanced_movie_search('The', NULL, NULL, NULL, 1990, 2000);
+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | release_date | ratings | total_reviews | language | poster_url | genres |
+-----+-----+-----+-----+-----+-----+-----+
| 2 | The Shawshank Redemption | 1994-09-23 | 5.00 | 3 | English | https://www.themoviedb.org/t/p/w780/q6y8GoitsGEsmfFryDOjo3dEmqu.jpg | Drama |
| 6 | The Matrix | 1999-03-31 | 5.00 | 1 | English | https://www.themoviedb.org/t/p/w780/f89U3ADrl0i1s9GkdPOExUkSH.jpg | Action, Sci-Fi |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)

mysql> CALL get_user_statistics();
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| user_id | username | name | email | total_reviews | avg_rating_given | movies_reviewed | episodes_reviewed | total_likes_received | first_review_date | last_review_date |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | john_doe | John Doe | newemail@test.com | 9 | 4.77778 | 5 | 2 | 0 | 2024-01-15 | 2025-11-23 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)

mysql> CALL get_user_statistics();
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| user_id | username | name | email | total_reviews | avg_rating_given | movies_reviewed | episodes_reviewed | total_likes_received | first_review_date | last_review_date |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | jane_smith | Jane Smith | jane.smith@email.com | 8 | 4.50000 | 5 | 3 | 0 | 2024-02-10 | 2025-11-23 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL get_user_statistics();
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| user_id | username | name | email | total_reviews | avg_rating_given | movies_reviewed | episodes_reviewed | total_likes_received | first_review_date | last_review_date |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | movie_buff_23 | Mike Johnson | mike.johnson@email.com | 6 | 4.58333 | 4 | 2 | 0 | 2024-03-05 | 2024-06-10 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL get_top_grossing_movies(5);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | box_office | release_date | ratings | total_reviews | poster_url | directors |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 19 | Avatar | 2923706026 | 2009-12-18 | 0.00 | 0 | https://image.tmdb.org/t/p/w500/kycqIdyUX66B5iVhRqosgbkJyK.jpg | NULL |
| 12 | Avatar | 2923706026 | 2009-12-18 | 4.50 | 2 | https://www.themoviedb.org/t/p/w780/kmcq1ZGSh2RpBwF9Cdri7dt.jpg | James Cameron |
| 31 | Avengers: Endgame | 2923706026 | 2019-04-26 | 0.00 | 0 | https://image.tmdb.org/t/p/w500/1LcPjZCzHgF0eB007df.jpg | NULL |
| 33 | Titanic | 2264703305 | 1997-12-19 | 4.25 | 2 | https://www.themoviedb.org/t/p/w780/9/xjZS2rVxn8Fx8dC3aiGCCQV.jpg | James Cameron |
| 37 | Top Gun: Maverick | 1488732821 | 2022-05-27 | 0.00 | 0 | https://image.tmdb.org/t/p/w500/62ChnUtziyWcpdab0211DX17Ljh.jpg | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

Query OK, 0 rows affected (0.09 sec)

mysql> CALL get_top_grossing_movies(10);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| movie_id | name | box_office | release_date | ratings | total_reviews | poster_url | directors |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 19 | Avatar | 2923706026 | 2009-12-18 | 0.00 | 0 | https://image.tmdb.org/t/p/w500/kycqIdyUX66B5iVhRqosgbkJyK.jpg | NULL |
| 12 | Avatar | 2923706026 | 2009-12-18 | 4.50 | 0 | https://www.themoviedb.org/t/p/w780/kmcq1ZGSh2RpBwF9Cdri7dt.jpg | James Cameron |
| 31 | Avengers: Endgame | 2923706026 | 2019-04-26 | 0.00 | 0 | https://image.tmdb.org/t/p/w500/1LcPjZCzHgF0eB007df.jpg | NULL |
| 33 | Titanic | 2264703305 | 1997-12-19 | 4.25 | 1 | https://www.themoviedb.org/t/p/w780/9/xjZS2rVxn8Fx8dC3aiGCCQV.jpg | James Cameron |
| 37 | Top Gun: Maverick | 1488732821 | 2022-05-27 | 0.00 | 0 | https://image.tmdb.org/t/p/w500/62ChnUtziyWcpdab0211DX17Ljh.jpg | NULL |
| 3 | The Dark Knight | 10004558444 | 2008-07-18 | 4.63 | 4 | https://www.themoviedb.org/t/p/w500/3J2tW6wMUDux91lr6m7haRef0WH.jpg | Christopher Nolan |
| 1 | Inception | 10000000000 | 2010-07-16 | 4.88 | 4 | https://www.themoviedb.org/t/p/w780/edv5CZwJ99upPsy2YGiDhKh8bt.jpg | Christopher Nolan |
| 35 | Oppenheimer | 9520000000 | 2023-07-21 | 0.00 | 0 | https://image.tmdb.org/t/p/w500/86xvg5FCU0GDyIgv72rIn2ua.jpg | Christopher Nolan |
| 28 | The Lord of the Rings: The Fellowship of the Ring | 871368364 | 2001-12-19 | 4.75 | 2 | https://image.tmdb.org/t/p/w500/soom5Q02yQTMJ1bnvbkBL9ch06.jpg | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

Query OK, 0 rows affected (0.12 sec)

```

These are the Procedure calls that are used in the code as well to bring the info from the database and present it to the user in the frontend.

Here are the functions that were used :

```
mysql> SELECT get_user_avg_rating(1) AS user_avg_rating;
+-----+
| user_avg_rating |
+-----+
|      4.78 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT get_user_avg_rating(2) AS user_avg_rating;
+-----+
| user_avg_rating |
+-----+
|      4.50 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT count_user_reviews(1) AS total_reviews;
+-----+
| total_reviews |
+-----+
|         9 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT count_user_reviews(2) AS total_reviews;
+-----+
| total_reviews |
+-----+
|         8 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT has_reviewed_movie(1, 3) AS has_reviewed;
+-----+
| has_reviewed |
+-----+
|         1 |
+-----+
1 row in set (0.01 sec)

mysql> SELECT has_reviewed_movie(2, 5) AS has_reviewed;
+-----+
| has_reviewed |
+-----+
|         0 |
+-----+
1 row in set (0.00 sec)
```

```

mysql> SELECT calculate_movie_popularity(1) AS popularity_score;
+-----+
| popularity_score |
+-----+
|      5.51 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT calculate_movie_popularity(3) AS popularity_score;
+-----+
| popularity_score |
+-----+
|      5.34 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT count_movies_reviewed(1) AS movies_count;
+-----+
| movies_count |
+-----+
|      5 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT count_movies_reviewed(2) AS movies_count;
+-----+
| movies_count |
+-----+
|      5 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT count_episodes_reviewed(1) AS episodes_count;
+-----+
| episodes_count |
+-----+
|      2 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT count_episodes_reviewed(2) AS episodes_count;
+-----+
| episodes_count |
+-----+
|      3 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT user_id, username, get_user_avg_rating(user_id) AS avg_rating, count_user_reviews(user_id) AS review_count FROM User LIMIT 5;
+-----+-----+-----+-----+
| user_id | username | avg_rating | review_count |
+-----+-----+-----+-----+
|      9 | action_fan_42 |      4.50 |          1 |
|      4 | cinephile_99 |      4.83 |          6 |
|      8 | drama_queen |      4.25 |          2 |
|      6 | film_critic_pro |      5.00 |          4 |
|      2 | jane_smith |      4.50 |          8 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

Triggers were as follows:

```

mysql> INSERT INTO Review (user_id, movie_id, date, rating, review_text) VALUES (1, 1, '2024-11-23', 4.5, 'Great movie!');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT movie_id, name, ratings, total_reviews FROM Movie WHERE movie_id = 1;
+-----+-----+-----+-----+
| movie_id | name      | ratings | total_reviews |
+-----+-----+-----+-----+
|       1  | Inception |    4.80 |          5  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> UPDATE Review SET rating = 5.0 WHERE review_id = 1;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0  Changed: 0  Warnings: 0

mysql> SELECT movie_id, name, ratings, total_reviews FROM Movie WHERE movie_id = 1;
+-----+-----+-----+-----+
| movie_id | name      | ratings | total_reviews |
+-----+-----+-----+-----+
|       1  | Inception |    4.80 |          5  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> DELETE FROM Review WHERE review_id = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT movie_id, name, ratings, total_reviews FROM Movie WHERE movie_id = 1;
+-----+-----+-----+-----+
| movie_id | name      | ratings | total_reviews |
+-----+-----+-----+-----+
|       1  | Inception |    4.80 |          5  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> UPDATE Review SET rating = 4.5 WHERE episode_id = 1 AND user_id = 2;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0  Changed: 0  Warnings: 0

mysql> SELECT show_id, name, ratings FROM tvshow WHERE show_id = (SELECT show_id FROM Episode WHERE episode_id = 1);
+-----+-----+-----+
| show_id | name      | ratings |
+-----+-----+-----+
|       1  | Breaking Bad |    5.00 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> DELETE FROM Review WHERE episode_id = 1 AND user_id = 2;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT show_id, name, ratings FROM tvshow WHERE show_id = (SELECT show_id FROM Episode WHERE episode_id = 1);
+-----+-----+-----+
| show_id | name      | ratings |
+-----+-----+-----+
|       1  | Breaking Bad |    5.00 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO Review (user_id, movie_id, rating, review_text) VALUES (3, 2, 4.0, 'Good movie without specifying date');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT review_id, user_id, movie_id, date, rating FROM Review ORDER BY review_id DESC LIMIT 1;
+-----+-----+-----+-----+-----+
| review_id | user_id | movie_id | date      | rating |
+-----+-----+-----+-----+-----+
|       97  |       3  |       2  | 2025-11-23 |    4.0 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```