Independent Film Distribution Platform - Database Requirements

Team Number - 11

Manasi Mundada Krish Jalan Ishaan Romil Inesh Shukla
2023101087 2023101074 2023114011 2023113008

1 System Overview

A comprehensive Python-based management system for independent film platforms, handling filmmakers, films, festivals, viewers, tickets, and analytics through a MySQL database.

2 Technical Requirements

- Python 3.x
- MySQL Server 8.0+
- PyMySQL package
- Windows OS

3 Installation & Setup

1. Install required package: `pip install pymysql`
2. Configure MySQL server
3. Import database schema
4. Set environment variables (optional)

4 Core Functions

4.1 Database Management

- **get_next_id(table, id_column)**
  Automatically generates sequential IDs for new entries in any table, ensuring unique identification.

- **clear_terminal()**
  Improves UI readability by clearing console output. Uses 'cls' command for Windows systems.

4.2 Filmmaker Management

- **insert_filmmaker()**
  Creates new filmmaker profiles with name, biography, contact details. Handles input validation and database constraints.

- **update_filmmaker()**
  Modifies existing filmmaker information with full error handling and constraint checking.

- **delete_filmmaker()**
  Removes filmmaker entries with cascade considerations for linked films and festivals.

4.3 Film Management

- **insert_film()**
  Registers new films with comprehensive metadata including title, genre, language, release year, and filmmaker association.

- **update_film()**
  Updates film details while maintaining referential integrity with filmmaker and viewer relationships.

- **delete_film()**
  Removes film entries with proper handling of dependent records (reviews, payments, etc.).

4.4 Festival Operations

- **insert_festival()**
  Creates festival entries with date validation and scheduling conflict checks.

- **update_festival()**
  Modifies festival information while preserving ticket and submission relationships.

- **delete_festival()**
  Handles festival removal with proper cleanup of associated tickets and submissions.

4.5 Viewer Management

- **insert_viewer()**
  Registers new viewers with complete profile information and viewing history tracking.
- **update_viewer()**
  Updates viewer profiles while maintaining payment and review relationships.
- **delete_viewer()**
  Manages viewer removal with proper handling of associated transactions and reviews.

4.6 Transaction Processing

- **insert_payment()**
  Records film purchase transactions with amount validation and viewer verification.
- **insert_ticket()**
  Manages festival ticket sales with access type validation and availability checks.
- **insert_review()**
  Processes film reviews with rating validation and duplicate prevention.

4.7 Analytics Functions

- **calculate_total_revenue()**
  Computes platform-wide revenue with detailed transaction analysis.
- **top_10_films_by_viewers()**
  Analyzes viewing patterns and ranks films by popularity.
- **top_10_grossing_films()**
  Tracks financial performance and identifies highest-earning content.
- **total_revenue_by_filmmaker()**
  Calculates filmmaker-specific earnings across all their films.
- **top_10_festivals_by_tickets()**
  Measures festival success through ticket sales analysis.

5 Menu System

- **main_menu()**
  Central navigation hub with access to all system functions through intuitive categorization.
- **execute_insert()**
  Handles all data creation operations with proper input validation and error handling.
- **execute_update()**
  Manages modification operations with consistency checks and constraint validation.
- **execute_delete()**
  Controls deletion operations with referential integrity preservation.
- **execute_retrieve()**
  Facilitates data retrieval with flexible search and filter options.
- **execute_analysis()**
  Provides access to analytical tools and reporting functions.

6 Error Handling

The system implements comprehensive error handling for:

- Database connection issues
- Input validation
- Constraint violations
- Transaction failures
- Data integrity issues