Movie Database Management System - Designed Application Features

1. Overview

The movie application is a backend system designed for managing, analyzing, and exploring movie-related data. It supports functionalities for various users, including movie enthusiasts, analysts, and producers.

2. Features

1. Movie Search

- **Full-Text Search:**
  - Search for movies using keywords or phrases in their titles or descriptions.
  - Example: A user can search for "action hero" to retrieve all relevant movies.
- **Efficiency:**
  - Uses full-text indexing for faster and optimized search results.

2. Genre Exploration

- **Categorized Movie Lists:**
  - View movies grouped by their respective genres (e.g., Action, Drama, Comedy).
  - Analyze the total number of movies available in each genre.
- **Popular Genres:**
  - Identify the most popular genres based on the number of movies.

3. Actor Insights

- **Biography Analysis:**
  - Perform full-text searches on actor biographies to find actors with specific attributes or accolades, such as "Oscar winner."
- **Collaboration Analysis:**
  - Discover actor pairs who have appeared together in multiple movies, helping users identify frequent collaborators.

4. Director Analytics

- **Genre Diversity:**
  - Identify directors who have worked across three or more genres, showcasing their versatility.
- **Movie Contributions:**
  - Analyze the number of movies directed by specific directors.

5. Advanced Query Execution

- **Predefined Queries:**

    - Perform predefined SQL queries for in-depth analysis, including:
        - Full-text searches for movies and actors.
        - Complex relational queries to analyze collaborations and genre diversity.
- **Customizability:**
    - Designed to support additional queries based on user needs.

6. Robust Data Ingestion

- **Automated Data Fetching:**
    - Automatically fetch movie-related data (genres, directors, actors, and movies) from external sources.
- **High Volume Support:**
    - Populate the database with thousands of records, ensuring scalability and efficiency.

7. Scalable Database Design

- **Normalization:**
    - The schema is designed with 3NF principles, reducing redundancy and ensuring consistency.
- **Indexes and Keys:**
    - Full-text indexes and foreign keys are implemented to improve query performance and ensure referential integrity.

8. Ease of Setup and Usage

- **Database Initialization:**
    - A Python script (create_db_script.py) creates the schema automatically.
- **Data Population:**
    - Data is fetched and populated programmatically using api_data_retrieve.py.
- **Query Demonstration:**
    - Queries can be executed and demonstrated through queries_db_script.py.

9. User-Focused Design

- **Use Cases:**
    - **Producers:** Analyze movie trends and characteristics of successful movies.
    - **Fans:** Explore their favorite actors, movies, and genres.
    - **Analysts:** Gain insights through complex relational queries.

10. Secure and Maintainable

- **SQL Injection Prevention:**
    - Parameterized queries are used throughout to prevent SQL injection attacks.
- **Error Handling:**
    - Logs and handles errors during database interactions to ensure smooth operations.