PopularMovies Inc.

The "Top 50 Movies" Information Database System

Created by Ayowade Owojori & Noam Yaffe

Overview

PopularMovies Inc. is a trusted resource that helps movie fanatics seek information on the top 50 highest-rated films from the 2000s. Our database system streamlines this process by providing an individual with all the information they may need about a film: the title, release year, genre, a brief description of the film (spoiler-free, of course!), director, actors, and personal anecdotes from verified audience members. Any personal input, rating, or review in our database is encouraged from users, as it can impact the specific movie's placement in our top 50 list.

Database Outline

Movies: umbrella entity that tracks the details of each Movie entry included in our Top 50 list

- idMovie: int, auto_increment, unique, not NULL, PK
- title: VARCHAR(145), not NULL
- releaseYear: DATE, not NULL
- genres_idGenre: INT, not NULL, FK
- description: VARCHAR(300), unique, not NULL
- averageRating: decimal(19,1), not NULL
- Relationships:
    - A M:N relationship is formed between Movies and Directors, where both are mandatory for the connection to exist. This relationship is implemented with an intersection table, Movies_has_Directors, where the intersection table has Foreign Keys for both Movies (movies_idMovie) and Directors (directors_idDirector).

    - A M:N relationship is formed between Movies and Actors, where both are mandatory for the connection to exist. This relationship is implemented with an intersection table, Movies_has_Actors, where the intersection table has Foreign Keys for both Movies (movies_idMovie) and Actors (actors_idActor).
    - A M:1 relationship is formed between Movies and Genres, where both elements are mandatory for the connection to exist. This relationship is implemented with a Foreign Key of idGenre inside Movies.
    - A 1:M relationship is formed between Movies and AudienceReviews, where Movies are mandatory for AudienceReviews but AudienceReviews are optional for Movies. This relationship is implemented with a Foreign Key of idMovie inside AudienceReviews.

Directors: stores information about a director who created (at least) one of our featured Top 50 movies

- idDirector: INT, auto_increment, unique, not NULL, PK
- firstName: VARCHAR(45), not NULL
- lastName: VARCHAR(45), not NULL
- middleName: VARCHAR(45)
- Relationship: A M:N relationship is formed between Directors and Movies, where both are mandatory for the connection to exist. This relationship is implemented with an intersection table, Movies_has_Directors, where the intersection table has Foreign Keys for both Movies (movies_idMovie) and Directors (directors_idDirector).

Actors: records the details of actors featured in any of our Top 50 movies

- idActor: INT, auto_increment, unique, not NULL, PK
- firstName: VARCHAR(45), not NULL
- lastName: VARCHAR(45), not NULL
- middleName: VARCHAR(45)

- Relationship: A M:N relationship is formed between Actors and Movies, where both are mandatory for the connection to exist. This relationship is implemented with an intersection table, Movies_has_Actors, where the intersection table has Foreign Keys for both Movies (movies_idMovie) and Actors (actors_idActor).

Genres: stores the different genres that our Top 50 movies may fall into

- idGenre: INT, auto_increment, unique, not NULL, PK
- category: VARCHAR(30), not NULL
- Relationship: A 1:M relationship is formed between Genres and Movies, where both elements are mandatory for the connection to exist. This relationship is implemented with a Foreign Key of idGenre inside Movies.

AudienceReviews: stores the different audience reviews attached to each of our Top 50 movies

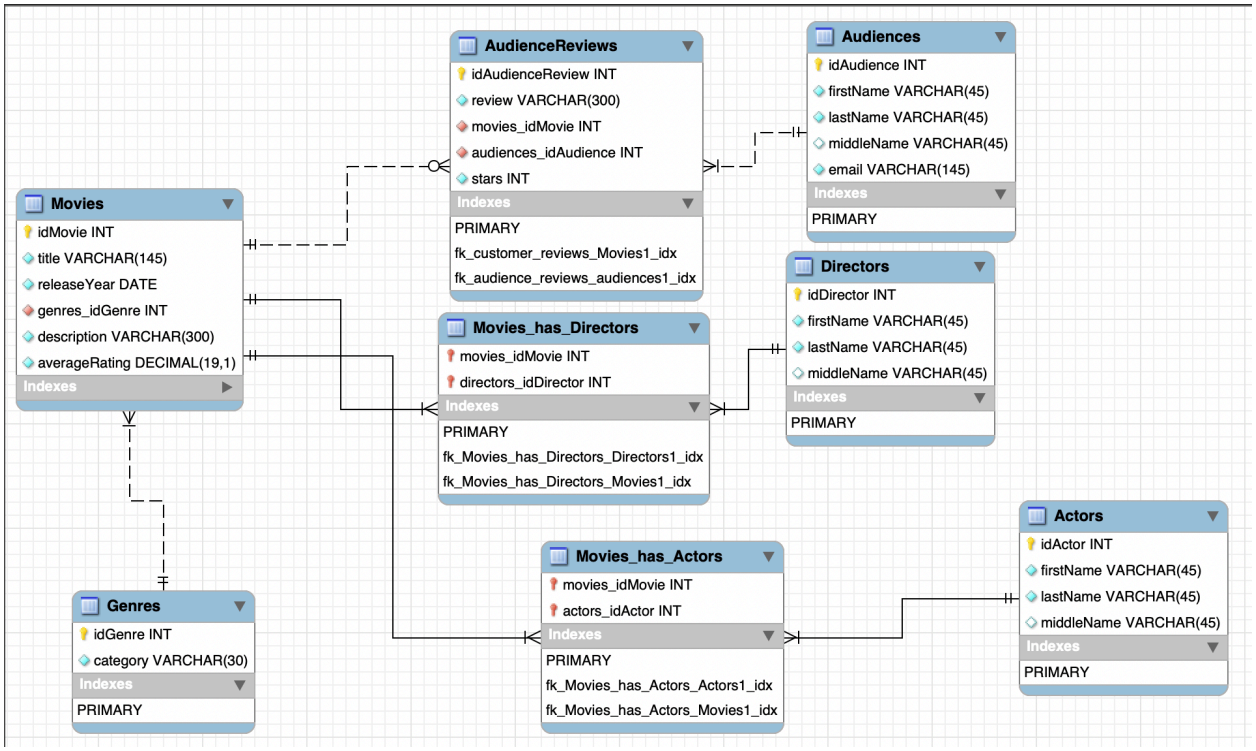
- idAudienceReview: INT, auto_increment, unique, not NULL, PK
- review: VARCHAR(300), not NULL
- movies_idMovie: INT, not NULL, FK
- audiences_idAudience: INT, not NULL, FK
- stars: INT, not NULL
- Relationship:
    - A M:1 relationship is formed between AudienceReviews and Audiences, where both are mandatory for the connection to exist. This relationship is implemented with a Foreign Key of idAudience inside AudienceReviews
    - A M:1 relationship is formed between AudienceReviews and Movies, where AudienceReviews are optional for Movies but Movies are mandatory for AudienceReviews. This relationship is implemented with a Foreign Key of idMovie inside AudienceReviews.

Audiences: records the information of the audience who left a review on one of our Top 50 movies

- idAudience: INT, auto_increment, unique, not NULL, PK
- firstName: VARCHAR(45), not NULL
- lastName: VARCHAR(45), not NULL
- middleName: VARCHAR(45)
- email: VARCHAR(145), not NULL
- Relationship: a 1:M relationship is formed between Audience and AudienceReviews, where both are mandatory for the connection to exist. This relationship is implemented with a Foreign Key of idAudience inside AudienceReviews

---

Entity-Relationship Diagram:



Citations: N/A