Movie Database

The goal of the database is to be used as a personal movie database where a user can store movies that they have watched and are yet to watch. A table called "Movies" will store movies where each entry will have these properties: viewed_date, movie_id, year, title, run_time, rating, soundtrack_rating. If viewed_date is null this implies that the user has not yet viewed the movie. movie_id is the primary key here. A table called Director will store all the names of the directors and will give each a unique id (dir_id). dir_id is the primary key here. A table called Stars will do the same for the actors/actresses. actor_id is the primary key here. A table called Genre will store all the types of genres and assign an id (genre_id) to each entry. genre_id is the primary key here.

A table called Directed_by will store a movie_id along with a dir_id. This will connect the movie table to the director table hence show all the directors of each movie. The primary key is (movie_id, dir_id). A table called Performed_by will store a movie_id along with an actor_id. This will connect the movie table to the Stars table hence show all the stars of each movie. The primary key is (movie_id, actor_id). A table called Genre_Is will store a movie_id along with a genre_id. This will connect the movie table to the Genre table hence show all the genres of each movie. The primary key is (movie_id, genre_id).

Using queries we can create tables that show movies that belong to a certain genre, decade, by a specific director, rating of x or higher, etc.

Entity-Relationship Diagram

Director entity
- name
- dir id (key)

Director — M — *directed by* — N — **Movie**

Movie entity attributes:
- viewed_date
- movie id (key)
- year
- title
- run time
- rating
- soundtrack rating

Movie — M — *performed by* — N — **Star**

Star entity
- name
- actor id (key)

Movie — M — *GenreIs* — N — **Genre**

Genre entity
- name
- genre id (key)