Basics of database systems

Project – Database design

Lappeenranta-Lahti University of Technology LUT
Software Engineering

Basics of database systems
Spring 2025

TABLE OF CONTENTS

1 DEFINITION

My “Movie Database” project is developed to manage movie-related data, providing versatile information about the movies, production companies, actors, directors and reviews. Movie studios and other film industry professionals can utilize this database for various purposes, such as tracking the details and attributes of movies.

The following database queries must be implemented:

1. List the information of a specific movie
2. List all movies produced by a production company
3. List all genre of a movie
4. Show all the actors that are in a specific movie
5. Show all reviews for a specific movie
6. List all movies directed by a specific director

2 MODELING

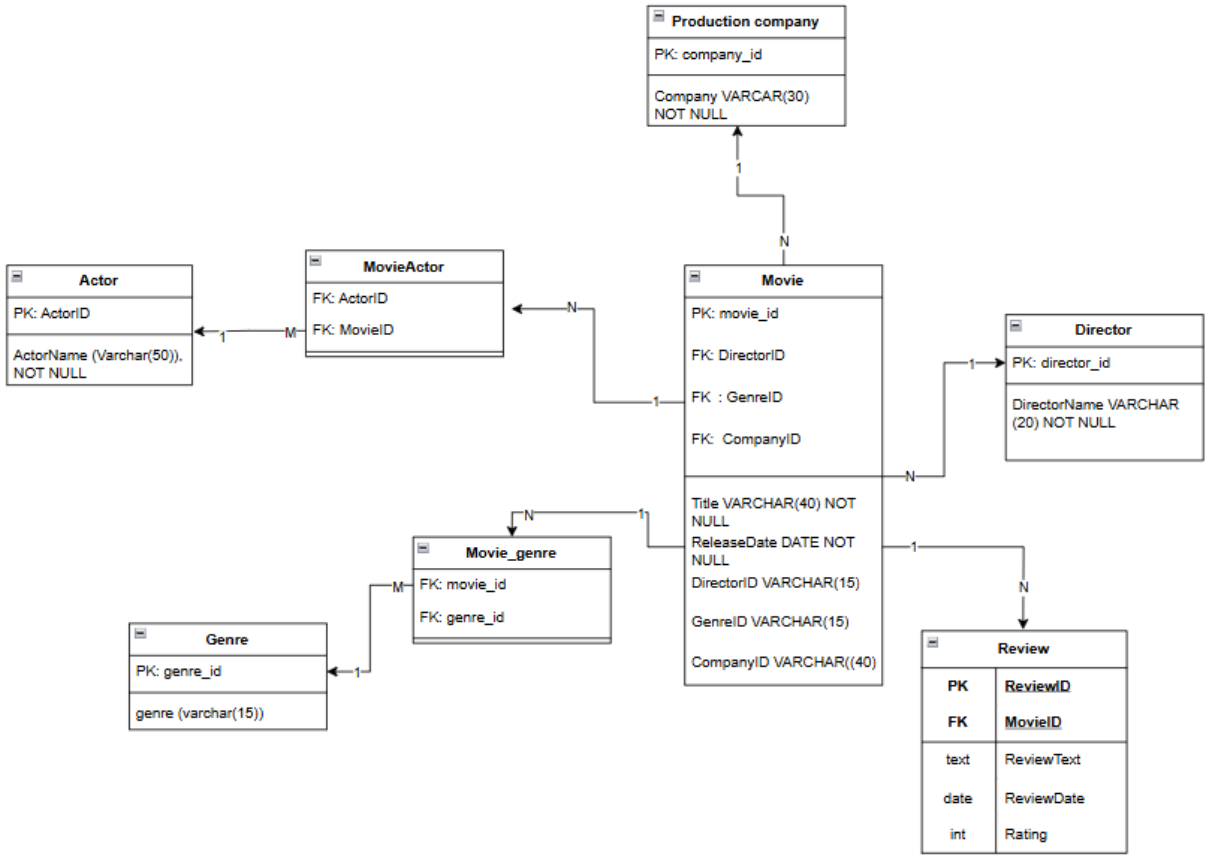

Figure 1: Data model

Figure 1 shows the database. There are six entities without counting the junction tables (Genre and actor) to handle many-to-many relationships. Other entities have one-to-many cardinality The entities show the concepts of this movie database, while the relationships define how the concepts are connected. Description of the data model:

1) Director and movie:
   Cardinality: One-to-Many (1:N)
   - A Director directs many movies, while a movie has only one director

2) ProductionCompany and movie:

Cardinality: One-to-Many (1:N)

- A production company can produce many movies, while a movie is produced by one company

3) Movie and Genre:
   Cardinality: Many-to-Many (N:M)
   - A movie can belong to multiple genres, and a genre can include multiple movies. Many-to-Many relationship is managed with the moviegenre junction table

4) Movie and Actor:
   Cardinality: Many-to-Many (N:M)
   - A movie can have many actors, and an actor can act in many movies. This is managed by the MovieActor junction table.

5) Movie and Review:
   Cardinality: One-to-Many (1:N)
   - A movie can have many reviews, while a review is done to only one movie.

3 DATABASE IMPLEMENTATION

Director:

- DirectorID is the primary key, cannot be null (NOT NULL).
- DirectorName cannot be null (NOT NULL).

Genre:

- GenreID is the primary key, cannot be null (NOT NULL).
- Genre cannot be null (NOT NULL).

ProductionCompany:

- CompanyID is the primary key, cannot be null (NOT NULL).

- CompanyName cannot be null (NOT NULL):

Movie:

- MovieID is the primary key, cannot be null.
- Title cannot be (NOT NULL).
- ReleaseDate cannot be null (NOT NULL)
- Foreign key references to Director, ProductionCompany and GenreID:
  - ON UPDATE CASCADE
  - ON DELETE CASCADE

Actor:

- ActorID is the primary key, cannot be null (NOT NULL).
- ActorName cannot be null (NOT NULL).

Review:

- ReviewID is the primary key, cannot be null.
- ReviewText cannot be null (NOT NULL), it has a default value 'No review provided'.
- Rating is between 1 and 10 (IMDB style) (CHECK constraint).
- ReviewDate cannot be null (NOT NULL)
- No duplicate reviews for the same movie with the help of a unique constraint (same date) (MovieID, ReviewDate).
- Foreign key references to Movie:
  - ON UPDATE CASCADE
  - ON DELETE CASCADE

MovieGenre:

- MovieID and GenreID are composite primary keys that cannot be null (foreign key MovieID references to Movie and GenreID references to Genre
  - ON UPDATE CASCADE
  - ON DELETE CASCADE

MovieActor:

- MovieID and ActorID are composite primarykeys that cannot be null. Foreign key MovieID references to Movie and ActorID references to Actor