

deisIMDB Relatório

1. Introdução

Este projeto explora o tema do cinema, recorrendo aos dados extraídos do IMDB. Pretendemos criar uma solução que utilize bases de dados relacionais (SQL), de forma a aplicar os princípios fundamentais do modelo relacional e da linguagem SQL. O objetivo deste projeto é desenvolver um sistema de gestão de informação sobre filmes que permita consultar e analisá-los.

2. Identificação de Entidades

- **Movies:** Descrição - Representa os filmes.
Atributos - movieId, movieName, movieDuration, movieBudget, movieReleaseDate.

- **MovieVotes:** Descrição - Armazena dados relacionados às avaliações dos filmes.
Atributos - movieRating, movieRatingCount.

- **Actor:** Descrição - Representa os atores que participam nos filmes
Atributos - actorID, actorName, actorGender.

- **Director:** Descrição - Representa os diretores responsáveis pelos filmes.
Atributos - directorId, directorName.

2.1. Relacionamentos entre as Entidades:

- **"contém" (1:N):** Um filme contém uma ou várias avaliações (MovieVotes), mas cada avaliação pertence a um único filme.

- **"participam" (1:N):** Um filme pode ter vários atores, mas um ator pode participar em apenas um filme.

- **"gere" (1:N):** Um diretor gera um filme, e cada filme pode ser gerido por vários diretores.

- **"têm" (N:N):** Um filme pode pertencer a vários gêneros, e um gênero pode incluir vários filmes.

3. Modelo Entidade-Relacional (Com uso de Notação Chen)

Figura 1 - (Encontra-se no final)

Durante o desenho do diagrama de Entidade-Relacional decidiu-se retirar os 'movieIds' nas tabelas "MovieVotes", "Actor" e "Director", como também a tabela MovieGenres pois estes irão ser criados quando ocorrer a transposição para o modelo físico, de maneira a respeitar as regras do mesmo.

4. Metodologia

Este projeto está desenvolvido em três etapas principais:

Modelagem Conceitual: Utilizando a notação de Chen, identificar as principais entidades - Movies, Actor, Director, Genre e MovieRating - e os seus relacionamentos.

Modelagem Lógica: As entidades foram processadas e normalizadas, garantindo que cada uma possuísse uma chave primária e relações, e que as relações N:N fossem representadas por tabelas associativas (MovieActor, MovieDirector, MovieGenre).

Modelagem Física: O modelo foi implementado em SQL, com a definição de chaves primárias, restrições de integridade e exclusões em cascata.

4.1. Modelo Físico 3FN:

De maneira a respeitar algumas regras da integridade dos dados, foram feitas estas alterações como resultado para o modelo físico:

Figura 2 - (Se encontra-se no final)

Na relação 1:1(Movie <--> MovieVotes): Significa que cada registo numa tabela está relacionado com, no máximo, um registo noutra tabela e vice-versa. No contexto deste modelo relacional, isto acontece quando: um filme tem exatamente um rating, e esse rating pertence apenas a esse filme. Com isto, a criação de um movieId garante que um rating está sempre associado a um filme, não podendo haver assim ratings "solto" ou filmes com múltiplos ratings.

Na relação N:N(restantes): Um registo numa tabela pode estar relacionado com muitos registos noutra tabela, e o inverso também é possível. Por exemplo: um filme pode ter vários atores, e cada ator pode estar em vários filmes. Foi criada uma tabela extra (de associação) "movieactor", "moviedirector" e "moviegenre" porque o modelo relacional não permite guardar listas diretas dentro de atributos também permite garantir a flexibilidade para adicionar ou remover ligações e evitar a redundância. Nestas tabelas as colunas principais (e únicas) são chaves estrangeiras e chaves primárias (formando assim uma chave composta) para as duas tabelas principais.

Nestas tabelas as colunas principais (e únicas) são chaves estrangeiras e chaves primárias (formando assim uma chave composta) para as duas tabelas principais.

5. Resultados e Discussão

A implementação final da base de dados apresenta uma estrutura sólida e eficiente.

Cada entidade é projetada de forma a garantir a integridade dos dados. As relações N:N foram representadas por tabelas intermediárias, assegurando flexibilidade nas consultas.

O modelo físico foi testado com comandos SQL que validaram as restrições e garantiram a consistência referencial.

O Diagrama da Figura 1 mostra o modelo conceptual do sistema.

6. Conclusão

O projeto atingiu com sucesso aquilo que foi proposto, resultando num modelo relacional consistente e funcional. A normalização garantiu que não existam redundâncias e o cumprimento das regras de integridade.

O sistema está pronto para as futuras etapas.

7. Contributions

Students:

Paulo Sérgio Leite Ribeiro - a22304812

Francisco Pinto - a22207132

Joseph Vickers a22408678

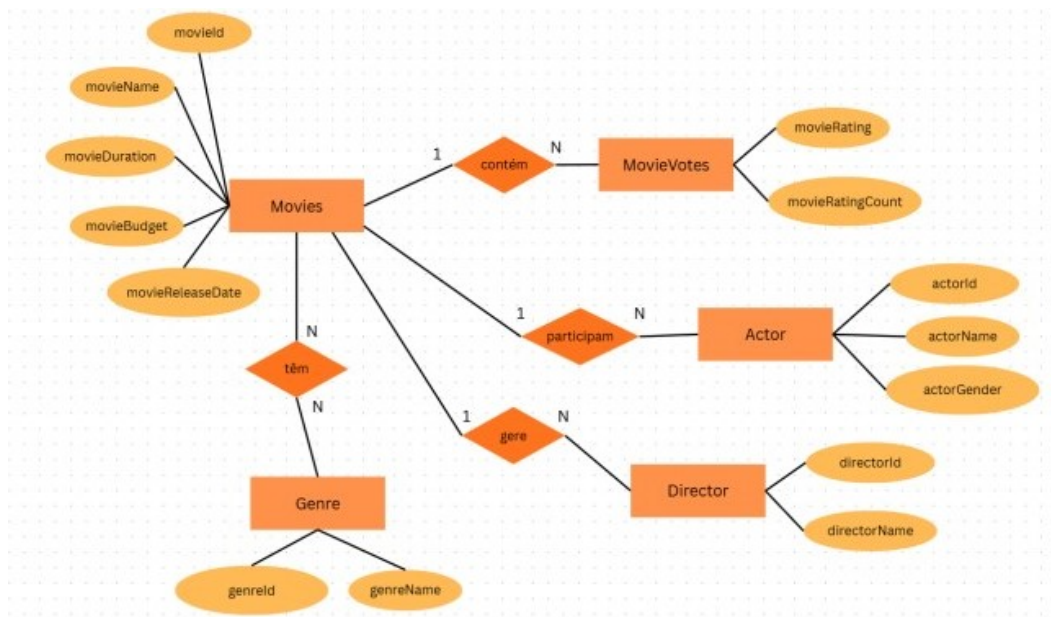


Figure 1. Example image

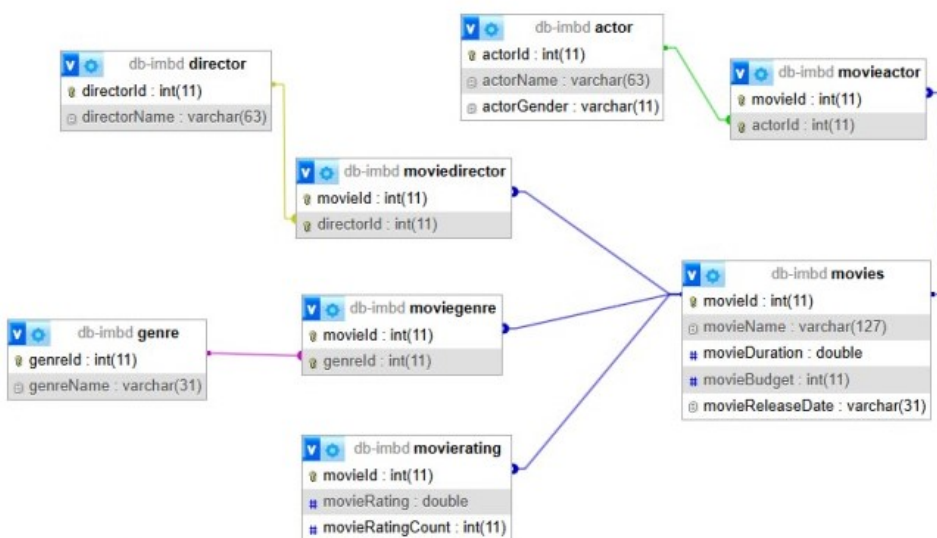


Figure 2. Example image