



Projeto de BDAD
2019 - 2020

Grupo 702

Jéssica Nascimento - up201806723

Rafael Cristino - up201806680

Xavier Pisco - up201806134

Índice

Contexto	2
Diagrama UML revisto	4
Esquema Relacional	5
Análise das Dependências Funcionais	6
Análise das Formas Normais	7
Lista e Forma de Implementação de Restrições	9
Interrogações à Base de Dados	12
Gatilhos Implementados	13

Contexto

O **IMBd_{ad}** (Internet Movie Base de Dados) é uma base de dados destinada a guardar informações sobre **filmes (Movie)**, **séries (Series)** e os seus **atores e produtores** (ambos da classe **Celebrity**).

Para além desta informação existem os **utilizadores (User)** da plataforma, que podem criar **listas (List)** e **avaliar (Review)**. Cada **utilizador** é reconhecido pelo seu email e pode ter uma foto de perfil e é habilitado a criar **listas** de **filmes** e/ou **séries** que são identificadas por um nome e, conforme a escolha do utilizador, podem ser públicas ou privadas.

Uma **Review** é caracterizada por título, texto e pontuação. Ela relaciona-se com o utilizador que a criou e com a produção ou episódio a que se refere (tem que estar relacionada obrigatoriamente ou com uma produção ou com um episódio). A partir da pontuação de todas as reviews sobre uma determinada produção ou episódio, calcula-se a pontuação do/a mesmo/a.

Tanto os **utilizadores** como as **celebridades** são classes filhos/filhas da classe **person (Person)**, a qual guarda informação sobre o seu nome e a sua data de nascimento.

Cada celebridade pode participar num **filme (Movie)** ou num **episódio de uma série (Episode)** com qualquer **Role** (através de relações entre as classes, ao que melhor se aplicar - se apenas participar num episódio de uma série, apenas se relacionará com esse episódio).

Estas relações distinguem-se da seguinte forma: uma participação num episódio de uma série é caracterizada por uma relação tripla entre **Episode**, **Celebrity** e **Role** (se o role for relacionado com atuar no episódio, o nome da personagem fica guardado na classe de associação da relação como **characterName**); a participação num filme é caracterizada por uma relação tripla entre **Movie**, **Celebrity** e **Role** (se o role for relacionado com atuar no filme, o nome da personagem fica guardado na classe de associação da relação como **characterName**);

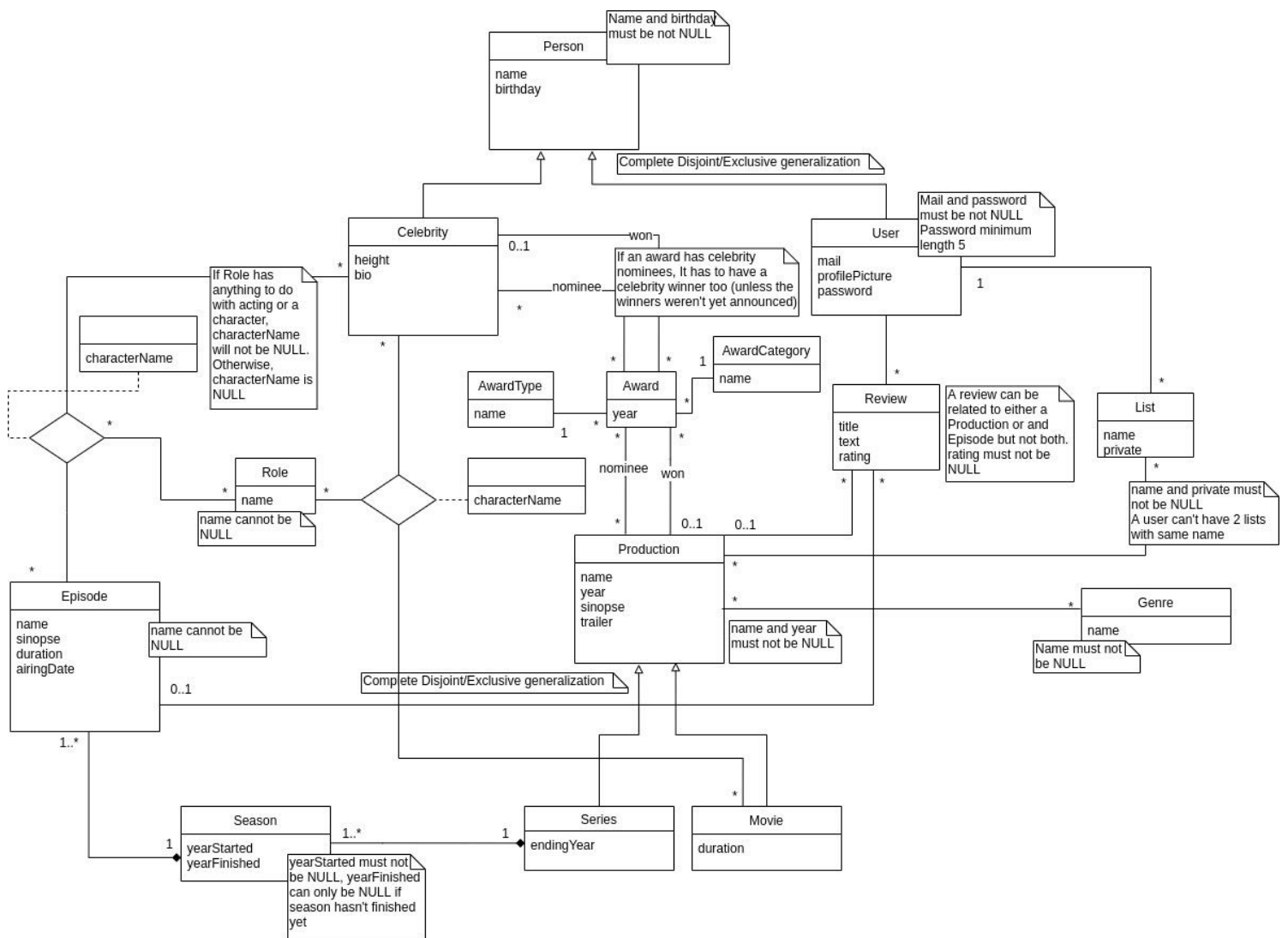
Os **filmes** e as **séries** são classes filhos/filhas da classe **produção (Production)** a qual guarda informação sobre o seu nome, ano, sinopse e o trailer e cada um está relacionado com uma ou mais **géneros (Genre)**. Por sua vez, os

filmes têm o atributo duração enquanto as **séries** têm o ano de término e estão associadas à classe **temporada (Season)**. Esta classe é composta pelos atributos ano de começo e ano de término e associa-se também aos **episódios (Episode)** que dela fazem parte, os quais são compostos pelo seu nome, sinopse, duração e data de estreia.

Por fim, as classes **prémio (Award)**, **tipo de prémio (AwardType)** e **categoria do prémio (AwardCategory)** são utilizadas para guardar informações sobre os prémios ganhos pelos **filmes, séries e celebridades** sendo que o tipo se refere ao nome do prémio, por exemplo, Óscar, e a categoria à sua categoria, por exemplo, melhor ator.

A classe **Award** relaciona-se com exatamente um **AwardType** e uma **AwardCategory** e possui o atributo year (Ano em que o prémio foi vencido). Para além disso, relaciona-se com uma ou nenhuma produção e com uma ou nenhuma celebridade (caso o prémio for, por exemplo, melhor ator, este relaciona-se com uma celebridade e uma produção; se for melhor filme apenas se relaciona com a produção; se não se relacionar nem com uma produção nem com uma celebridade, significa que o vencedor ainda não foi anunciado (no entanto, podem já haver nomeados)) através da relação **won** que identifica os vencedores do prémio. Com a relação **nominee**, o prémio relaciona-se com todas as celebridades e produções que foram/são nomeadas para o receber.

Diagrama UML



Esquema Relacional

Person (id, name, birthday)

Celebrity (personId -> Person, height, bio)

User (personId -> Person, mail, profilePicture, password)

Production (id, name, year, sinopse, trailer)

Movie (productionId -> Production, duration)

Series (productionId -> Production, endingYear)

Season (id, seriesId -> Series, yearStarted, yearFinished)

Episode (id, seasonId -> Season, name, sinopse, duration, airingDate)

Role (id, name)

MovieRole (movieId -> Movie, celebId -> Celebrity, roleId -> Role, characterName)

EpisodeRole (episodId -> Episode, celebId -> Celebrity, roleId -> Role, characterName)

Review (id, userId -> User, title, text, rating)

ReviewProduction (reviewId -> Review, productionId -> Production)

ReviewEpisode (reviewId -> Review, episodId -> Episode)

List (id, name, userId -> User, private)

BelongsToList (listId -> List, productionId -> Production)

Genre (id, name)

BelongsToGenre (productionId -> Production, genreId -> Genre)

Award (id, year, productionId -> Production, awardTypeId -> awardType, awardCategoryId -> AwardCategory, celebId -> Celebrity)

AwardType (id, name)

AwardCategory (id, name)

NomineeProduction (awardId -> Award, productionId -> Production)

NomineeCelebrity (awardId -> Award, celebId -> Celebrity)

Análise das Dependências Funcionais

Person: {id} -> {name, birthday}

Celebrity: {personId} -> {height, bio}

User: {personId} -> {mail, profilePicture, password}

{mail} -> {personId, profilePicture, password}

Production: {id} -> {name, year, sinopse, trailer}

Movie: {productionId} -> {duration}

Series: {productionId} -> {endingYear}

Season: {id} -> {seriesId, yearStarted, yearFinished}

Episode: {id} -> {seasonId, name, sinopse, duration, airingDate}

Role {id} -> {name}

{name} -> {id}

MovieRole {movieId, celebId, roleId} -> {characterName}

EpisodeRole {episodeId, celebId, roleId} -> {characterName}

Review: {id} -> {userId, title, text, rating}

ReviewProduction: {reviewId} -> {productionId}

ReviewEpisode: {reviewId} -> {episodeId}

List: {id} -> {name, userId, private}

{name, userId} -> {id, private}

BelongsToList: {listId, productionId} -> {}

Genre: {name} -> {id}

BelongsToGenre: {productionId, genreId} -> {}

Award {id} -> {year, productionId, awardTypeId, awardCategoryId, celebId}

{year, awardTypeId, awardCategoryId} -> {id, productionId, celebId}

AwardType {id} -> {name}

{name} -> {id}

AwardCategory {id} -> {name}

{name} -> {id}

NomineeProduction {awardId, productionId} -> {}

NomineeCelebrity {awardId, celebId} -> {}

Análise das Formas Normais

Analisando os fechos de cada dependencia:

Person: {id}+ = {id, name, birthday}

Celebrity: {personId}+ = {personId, height, bio}

User: {personId}+ = {personId, mail, profilePicture, password}
{mail}+ = {personId, mail, profilePicture, password}

Production: {d}+ = {id, name, year, sinopse, trailer}

Movie: {productionId}+ = {productionId, duration}

Series: {productionId}+ = {productionId, endingYear}

Season: {id}+ = {id, seriesId, yearStarted, yearFinished}

Episode: {id}+ = {id, seasonId, name, sinopse, duration, airingDate}

Role {id}+ = {id, name}

{name}+ = {id, name}

MovieRole {movieId, celebId, roleId}+ = {productionId, celebId, roleId, characterName}

EpisodeRole {episodeId, celebId, roleId}+ = {episodeId, celebId, roleId, characterName}

Review: {id}+ = {id, userId, title, text, rating}

ReviewProduction: {reviewId}+ = {reviewId, productionId}

ReviewEpisode: {reviewId}+ = {reviewId, episodeId}

List: {id}+ = {id, name, userId, private}

{name, personId}+ = {id, name, userId, private}

BelongsToList: {listId, productionId}+ = {listId, productionId}

Genre: {id}+ = {id, name}

{name}+ = {id, name}

BelongsToGenre: {productionId, genreId}+ = {productionId, name}

Award {id}+ = {id, year, productionId, awardTypeId, awardCategoryId, celebId}

{year, awardTypeId, awardCategoryId}+ = {id, year, productionId, awardTypeId, awardCategoryId, celebId}

AwardType {id}+ = {id, name}

{name}+ = {id, name}

AwardCategory {id}+ = {id, name}

{name}+ = {id, name}

NomineeProduction {awardId, productionId}⁺ = {awardId, productionId}

NomineeCelebrity {awardId, celebId}⁺ = {awardId, celebId}

Podemos concluir que todas as relações respeitam a Terceira Forma Normal uma vez que seguem todas as seguintes regras:

- Todos os atributos contêm valores atômicos e os seus valores são singulares (1NF);
- Todos os atributos membros que fazem parte de uma chave são primos (2NF);
- Não existem dependências transitivas, ou seja, um atributo depende exclusivamente de uma chave primária (3NF).

Em todas as relações, exceto as listas e os prêmios, para cada dependência, o lado esquerdo corresponde a uma superchave. Assim sendo, para além da Terceira Forma Normal, estas relações encontram-se na Forma Normal Boyce-Codd. As listas e os prêmios estão apenas na Terceira Forma Normal

Lista e Forma de Implementação de Restrições

Person:

- id PRIMARY KEY
- name, birthday NOT NULL

Celebrity:

- personId PRIMARY KEY
- personId FOREIGN KEY
- personId ON DELETE CASCADE
- personId ON UPDATE CASCADE

User:

- personId PRIMARY KEY
- personId FOREIGN KEY
- personId ON DELETE CASCADE
- personId ON UPDATE CASCADE
- mail UNIQUE
- mail NOT NULL
- password NOT NULL
- password CHECK (length > 5)

Production:

- id PRIMARY KEY
- name, year NOT NULL

Movie:

- productionId PRIMARY KEY
- productionId FOREIGN KEY
- productionId ON DELETE CASCADE
- productionId ON UPDATE CASCADE

Series:

- productionId PRIMARY KEY
- productionId FOREIGN KEY
- productionId ON DELETE CASCADE
- productionId ON UPDATE CASCADE

Season:

- id PRIMARY KEY
- seriesId FOREIGN KEY
- seriesId NOT NULL
- seriesId ON DELETE CASCADE
- seriesId ON UPDATE CASCADE
- yearStarted NOT NULL
- yearFinished >= yearStarted

Episode:

- id PRIMARY KEY
- seasonId FOREIGN KEY
- seasonId NOT NULL
- seasonId ON DELETE CASCADE
- seasonId ON UPDATE CASCADE
- name NOT NULL

Role:

- id PRIMARY KEY
- name NOT NULL
- name UNIQUE

MovieRole:

- movieId, celebId, roleId PRIMARY KEY
- movieId, celebId, roleId FOREIGN KEY
- movieId NOT NULL
- movieId ON DELETE CASCADE
- movieId ON UPDATE CASCADE
- celebId NOT NULL
- celebId ON DELETE CASCADE
- celebId ON UPDATE CASCADE
- roleId NOT NULL
- roleId ON DELETE CASCADE
- roleId ON UPDATE CASCADE

EpisodeRole:

- episodeId, celebId, roleId PRIMARY KEY
- episodeId NOT NULL
- episodeId ON DELETE CASCADE
- episodeId ON UPDATE CASCADE
- celebId NOT NULL
- celebId ON DELETE CASCADE
- celebId ON UPDATE CASCADE
- roleId NOT NULL
- roleId ON DELETE CASCADE
- roleId ON UPDATE CASCADE

Review:

- id PRIMARY KEY
- userId FOREIGN KEY
- userId NOT NULL
- userId ON DELETE CASCADE
- userId ON UPDATE CASCADE
- rating NOT NULL CHECK(rating > 0 and rating <= 10)

ReviewProduction:

- reviewId PRIMARY KEY
- reviewId, productionId FOREIGN KEY
- reviewId NOT NULL
- reviewId ON DELETE CASCADE
- reviewId ON UPDATE CASCADE
- productionId NOT NULL
- productionId ON DELETE CASCADE
- productionId ON UPDATE CASCADE

ReviewEpisode:

- reviewId PRIMARY KEY
- reviewId, episodeId FOREIGN KEY
- episodeId NOT NULL
- reviewId NOT NULL

List:

- id PRIMARY KEY
- name NOT NULL
- private NOT NULL
- userId FOREIGN KEY
- userId NOT NULL
- userId ON DELETE CASCADE
- userId ON UPDATE CASCADE
- userId, name UNIQUE

BelongsToList:

- listId, productionId PRIMARY KEY
- listId, productionId FOREIGN KEY
- listId NOT NULL
- listId ON DELETE CASCADE
- listId ON UPDATE CASCADE
- productionId NOT NULL
- productionId ON DELETE CASCADE
- productionId ON UPDATE CASCADE

Genre:

- Id PRIMARY KEY
- name UNIQUE
- name NOT NULL

BelongsToGenre:

- productionId, genreId PRIMARY KEY
- productionId, genreId FOREIGN KEY
- productionId NOT NULL
- productionId ON DELETE CASCADE
- productionId ON UPDATE CASCADE
- genreId NOT NULL
- genreId ON DELETE CASCADE
- genreId ON UPDATE CASCADE

Award:

- id PRIMARY KEY
- year NOT NULL
- awardTypeId, awardCategoryId FOREIGN KEY
- awardTypeId NOT NULL
- awardTypeId ON DELETE CASCADE

- awardTypeId ON UPDATE CASCADE
- awardCategoryId NOT NULL
- awardCetegoryId ON DELETE CASCADE
- awardCategoryId ON UPDATE CASCADE
- productionId ON DELETE CASCADE
- productionId ON UPDATE CASCADE
- celebId ON DELETE SET NULL
- celebId ON UPDATE CASCADE
- Year, awardTypeId, awardCategoryId UNIQUE
- If ProductionId is NULL then celebId must be NULL

AwardType:

- id PRIMARY KEY
- name NOT NULL
- name UNIQUE

AwardCategory:

- Id PRIMARY KEY
- name NOT NULL
- name UNIQUE

NomineeProduction:

- awardId, productionId PRIMARY KEY
- awardId, productionId FOREIGN KEY
- awardId NOT NULL
- awardId ON DELETE CASCADE
- awardId ON UPDATE CASCADE
- productionId NOT NULL
- productionId ON DELETE CASCADE
- productionId ON UPDATE CASCADE

NomineeCelebrity:

- awardId, celebId PRIMARY KEY
- awardId, celebId FOREIGN KEY
- awardId NOT NULL
- awardId ON DELETE CASCADE
- awardId ON UPDATE CASCADE
- celebId NOT NULL
- celebId ON DELETE CASCADE
- celebId ON UPDATE CASCADE

Interrogações à Base de Dados

1. Celebridades que ganharam dois ou mais prémios num mesmo filme ou série. Esse prémios pode ter sido ganhos no mesmo ano ou não (no caso das séries).
2. Segunda celebridade que ganhou mais prémios (sem utilizar o max()). No caso de existirem mais do que uma celebridade com o mesmo número de prémios mostra todas as que tiverem esse número de prémios.
3. Celebridades que participaram simultaneamente em filmes e séries, considerando participar simultaneamente ser participar num filme que saiu no mesmo ano em que saiu uma temporada da série, na qual participou.
4. Celebridades que venceram pelo menos um Óscar, filmes em que os ganharam e as suas categorias.
5. Obter a contagem de prémios pelo género que corresponde à produção por ordem decrescente.
6. Obter pares de utilizadores que deram a mesma pontuação a uma dada produção.
7. Obter a pontuação de uma determinada produção, tendo em conta todas as reviews que foram feitas sobre a mesma. No caso das séries, contar também com as reviews dos episódios.
8. Obter os pares de utilizadores que partilham uma produção nas suas listas. Mostra os utilizadores e o nome da produção em questão.
9. Atores que participaram em todos os episódios de uma série.
10. Celebridades que já assumiram pelo menos 3 papéis diferentes.

Gatilhos Implementados

1. Verificar se quando se adiciona um *ProductionRole* ou um *EpisodeRole* onde a *Role* é *Character* essa personagem tem nome. Se a role for '%character%' (interpreta uma personagem), não tiver nome e se tratar de um *EpisodeRole*, verificar nos episódios anteriores se a mesma celebridade participa com o mesmo *Role*. Se sim, preencher o *CharacterName* com o *CharacterName* dos episódios anteriores.
2. Garantir a nomeação de uma celebridade ou produção antes de permitir que esta ganhe um prémio.
3. Garantir que quando uma celebridade é nomeada existe uma nomeação para esse mesmo prémio para a produção em que a celebridade participou. Garantir que quando uma celebridade vence um prémio, a mesma participou na produção que o venceu.