

Projeto Base de Dados

André Silva
up201906559

João Aires
up201905175

Pedro Vieira
up201905272

April 2021

1 Requisitos

Os requisitos para a base de dados sobre a gestão de um banco de dados sobre histórico de visualização de animes são:

1. Animes, users, voice actors, sources que têm associados códigos (únicos).
2. Cada voice actor tem também nome, data de nascimento, um possível email e a informação sobre o sexo. Tem de ter pelo menos um contrato feito com um estúdio, sendo que desse contrato não têm de resultar animes.
3. É guardado os participantes (voice actor) em cada anime. Sendo que um voice actor pode participar em múltiplos animes.
4. É registado os trabalhadores para um estúdio (para simplificação apenas voice actors). Cada voice actor faz contrato com um estúdio, com um salário associado para um ou múltiplos animes desse estúdio. No entanto, um estúdio pode oferecer um contrato a um voice actor sem ter em mente que ele participe num anime.
5. Cada estúdio, para além do Id, tem um nome, e pode ter feito vários animes, sendo que todo o estúdio tem de ter feito pelo menos um anime. Um estúdio pode ter vários headquarters em múltiplas localizações.
6. Na mesma localização podem existir vários estúdios.
7. Cada anime tem também um nome, único, uma data em que começou a ser transmitido, sendo que esta pode ser uma data futura, indicando que o anime ainda não

começou. Se já foi transmitido o episódio final então existe uma data de fim. Têm também associado, possivelmente, múltiplos géneros, bem como o seu rating pelos users. Este atributo rating é opcional pois pode nenhum user ter dado rating aquele anime, ou no caso de ainda não ter estreado, não terá certamente rating. Todo o anime é feito apenas por um estúdio, sendo que tem sempre de ser feito por alguém. Todos os animes têm de ter origem numa source. Para além disso, cada anime tem possivelmente múltiplos episódios, e cada episódio pode ter uma duração fixa ou variável. Todo o anime têm ainda de ter voice actors a participar.

8. Cada source tem uma data de quando começou a ser feita, que tem de ser anterior ou igual à do anime que adapta, e uma data de fim se já foi finalizada. Uma source apenas pode originar um anime e um anime apenas tem origem numa source. Cada source pode ser de unicamente de um tipo.
9. Cada source foi criada por possivelmente vários autores.
10. Os users têm também um nome, e um login que é único. Têm também uma data de nascimento, um sexo e possivelmente um número de telemóvel associado. É necessário um email para identificação que não tem de ser único entre users. No perfil de cada user tem também o seu avg rating e o seu rating mais alto e mais baixo já alguma vez dado a um anime.
11. Cada user tem uma lista de animes (muito provavelmente vários) que já viu, que pretende ver no futuro, e animes que está a ver "no momento", ou seja ainda não os acabou de ver.
12. É registado ainda os ratings que cada user deu a um anime de modo a fomentar o processo de ratings para animes e para os perfis dos utilizadores.

2 Modelo ER

Mapeando praticamente diretamente dos requisitos, foram criadas 4 entidades-tipo e 1 entidade-frac. As 4 entidades correspondentes são *USER*, *ANIME*, *VOICE_ACTOR*, *STUDIO* e *SOURCE*. Os atributos de cada entidade são:

- **USER**: UserId (atributo chave); Name; Login; BirthDate; Age (atributo derivado); Sex; Phone (atributo opcional); Email; AvgRating; HighestRating; LowestRating.
- **ANIME**: AnimeId (atributo chave); Name; StartDate; FinishedDate (atributo opcional); NumEpisodes; Genre (atributo multivalor); Rating (atributo opcional).

- **SOURCE**: SourceId (atributo chave); DateStarted; DateFinished (atributo opcional); Type; Creator (atributo multivalor).
- **VOICE_ACTOR**: VoiceActorId (atributo chave); Name; BirthDate; Email (atributo opcional); Age (atributo derivado); Sex;
- **STUDIO**: StudioId (atributo chave); Name; Location (atributo multivalor) contém vários Address{Street; Floor (atributo opcional), PostCode} (atributo composto).

Do ponto 1. tiramos que estas entidades-tipo tem um Id único que funcionará como chave. Do ponto 2. tiramos as informações sobre a entidade **VOICE_ACTOR**, do ponto 5. e 6. tiramos sobre o a entidade **STUDIO**. Os atributos das entidades tipo **USER** e **ANIME** vêm do ponto 10. e do ponto 7.. Já os atributos entidade **SOURCE** vêm diretamente dos pontos 8. e 9..

A entidade fraca deriva do ponto 7.. Visto que um anime têm uma quantidade variável de episódios, podemos traduzir isto por uma entidade fraca em que o dependente (episódio) tem uma chave parcial que é o número do episodio e um atributo, a duração de cada episódio. Isto permite-nos ter uma quantidade arbitraria de episódios, com durações arbitrárias, ligados a um anime sem problema de não conseguir identificar qual episódio pertence a qual anime.

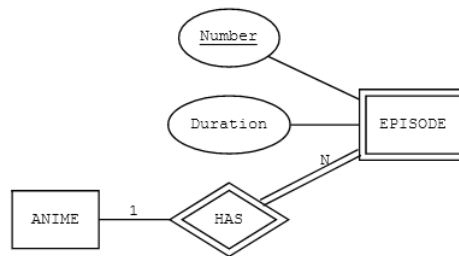


Figure 1: Entidade Fraca para identificar episódios a animes.

Os requisitos do ponto 11. podem ser traduzidos por relações entre a entidade-tipo **USER** e a entidade-tipo **ANIME**. Estas 3 relações são todas parciais quer de **USER** para **ANIME**, quer de **ANIME** para **USER**, pois nem todos os users têm de ter visto, estar a ver ou pretender ver algum anime, nem um anime tem de estar a ser visto, ter sido visto

ou ser pretendido ver por algum user. Quanto à cardinalidade todas são M:N pois um user pode ter vários animes sobre qualquer um dos *status* anteriores e um anime pode estar em qualquer dos *status* anteriores por vários users (e.g vários users querem ver o mesmo anime e esse anime tem varios users que o querem ver).

No ponto 12. dos requisitos, como é guardado os diferentes ratings que cada user dá a um anime, podemos traduzir isto por uma relação M:N, com participação parcial para ambas as entidades-tipo com um atributo rating. As entidades-tipo participantes são *USER* e *ANIME* e como é exposto pela relação descrita, um user pode dar vários ratings a vários animes, e um anime receber vários ratings de vários users, não sendo obrigado um user a dar rating a um anime, nem um anime a ter um rating de algum user.

Do ponto 5., tiramos que um estúdio pode ter feito vários animes e que tem de ter pelo menos um anime feito. Do ponto 7. é dito que um anime apenas pode ser feito por um estúdio. Assim traduzimos isto por uma relação que envolve as entidades-tipo *STUDIO* e *ANIME* com cardinalidade 1:M com participação total de ambas as entidades. Um estúdio pode fazer vários animes, mas tem de ter feito pelo menos 1, ou seja todo o estúdio tem de estar nesta relação, e um anime é feito apenas por um estúdio, sendo que um anime tem de ser feito por alguém, todos os animes estão na relação.

Do ponto 7. e 8. podemos tirar uma relação entre *SOURCE* e *ANIME* 1:1 com participação total da parte de *ANIME* e parcial de *SOURCE*, pois nunca é dito que toda a source tem de originar um anime. Assim, traduzimos que cada source origina um e um só anime (ou não origina nenhum), e um anime tem de ter origem numa e numa só source.

Para finalizar dos ponto 3. 4. e 7., tiramos que um voice actor tem de ter tido ou ter um contrato com um estúdio e um estúdio tem de ter (ou ter tido) pelo menos um contrato com um voice actor, que, por sua vez podem participar em múltiplos animes. Tiramos ainda que todos os animes têm de ser participados por 1 ou mais voice actors. Assim podemos estabelecer uma relação N:M, entre as entidades tipo *VOICE_ACTOR* e *ANIME* com participação parcial para *VOICE_ACTOR* e total para *ANIME*, pois nunca é referido que um *VOICE_ACTOR* tem de obrigatoriamente participar em um anime. Tiramos ainda que existe uma relação N:M total de ambas as partes com o atributo *wage* para as entidades *STUDIO* e *VOICE_ACTOR* que descreve os possíveis contratos entre voice actors e studios.

3 Relational

As 4 entidades-tipo foram diretamente mapeadas nas tabelas correspondentes com os seus atributos, onde a chave primária é o atributo chave do modelo ER. A entidade *fraca* tornou-se uma tabela com a entrada *Duration*, e tem como chave primária a combinação da sua chave parcial com a chave externa *AnimeId* que aponta naturalmente para o atributo da tabela *ANIME* com o mesmo nome. De facto este tipo de chave externa é tão comum que a partir daqui e salvo menção específica, se uma chave externa com o nome, por exemplo *someName* for referenciada, deve ser interpretada como referindo à tabela que têm a entrada com mesmo nome, *someName*, como chave primária.

Para as relações sobre o *status* de um user para com um anime, foram traduzidas em 3 tabelas: *PLAN_TO_WATCH*, *WATCHING*, *COMPLETED*. Cada uma destas tabelas é composta por duas entradas que em conjunto formam as chaves primárias da sua tabela. Estas duas entradas são também duas chaves externas *UserId* e *AnimeId*. Esta combinação de chaves para chaves primárias permite traduzir a multiplicidade subjacente às relação correspondentes no modelo ER.

A relação sobre ratings no modelo ER foi também traduzida numa tabela *RATES*, com a dupla de chaves externas *UserId* e *AnimeId* a funcionar como chaves primárias, pela mesma razão da combinação de chaves nas tabelas do paragrafo acima. Esta tabela absorve também o atributo *Rating* da relação.

Quanto às relações descritas no ultimo paragrafo da secção 2, foram traduzidas da seguinte forma. A relação de empregador-empregado para os voice actors e os estúdios foi traduzida numa tabela independente que absorveu o atributo *wage* da relação, e que tem como chaves primárias a dupla de chaves externas *StudioId* e *VoiceActorId*, permitindo que vários voice actors tenham contrato no mesmo estúdio. A relação de participação de voice actors em animes foi também traduzida numa tabela, composta apenas pela combinação de chaves externas *VoiceActorId* e *AnimeId* como chaves primárias da mesma, permitindo a participação de vários voice actors no mesmo anime.

O facto de os estúdios terem várias localizações, representados por um atributo multivalor composto na entidade-tipo *STUDIO* foi traduzido em duas tabelas. Uma tabela *LOCATION* que tem como chave primária a dupla de chaves externas *StudioId* e *AddressId*, para permitir que o mesmo estúdio tenha várias localizações. Para os atributos de um address, foi criada a tabela *ADDRESS* que tem uma chave primária *AddressId*. Os restantes atributos de address foram também mapeados nessa tabela.

Para traduzir o atributo multivalor da entidade-tipo *SOURCE* foi usada também uma tabela independente onde os únicos atributos são o nome do autor e o *SourceId*, ambos

chaves primárias para que um autor possa fazer várias sources.

Finalmente para traduzir as relações da entidade-tipo *ANIME* com a *SOURCE* e *STUDIO* foram acrescentadas duas chaves externas *SourceId* e *StudioId* que apontam para os sítios óbvios, à tabela *ANIME*.

Para representar a possível multiplicidade de géneros de um anime, representado pelo atributo multivalor no modelo ER, foi acrescentado uma tabela *GENRE* que tem um *label* a dizer o tipo de género, e um *AnimeId* que é chave externa. Estas duas entradas na tabela formam a chave primária da mesma. Assim, vários animes podem ter o mesmo género e um anime pode ter vários géneros.

4 Dados

- ***ADDRESS***: 22 entradas.
- ***ANIME***: 51 entradas.
- ***COMPLETED***: 496 entradas.
- ***CREATOR***: 50 entradas.
- ***EMPLOYES***: 248 entradas.
- ***EPISODE***: 3452 entradas.
- ***GENRE***: 258 entradas.
- ***LOCATION***: 26 entradas.
- ***PARTICIPATED***: 248 entradas.
- ***PLAN_TO_WATCH***: 100 entradas.
- ***RATES***: 501 entradas.
- ***SOURCE***: 51 entradas.
- ***STUDIO***: 23 entradas.
- ***USER***: 100 entradas.
- ***VOICE_ACTOR***: 59 entradas.
- ***WATCHING***: 100 entradas.