

Descrição das componentes do diagrama e o Modelo Lógico Relacional:

Cada subtítulo em *itálico* é equivalente a uma tabela do modelo relacional, exceto se alguma entidade possui atributos múltiplos, e compõe de quatro partes: descrição de cada Entidade/Relacionamento/Atributo do diagrama, um mapeamento para o modelo relacional com <nome_da_tabela>::<atributos>, a descrição técnica de cada atributo da tabela final e as restrições de integridade.

Todo mapeamento do modelo ER para o modelo lógico relacional é realizado de acordo com a tabela 5.1 encontrada no livro de Heuser, capítulo 5.

Legenda do mapeamento para modelo lógico:

- Sublinhado indica chave primária
- Underscore indica chave estrangeira
- Asterisco indica que atributo admite valor nulo

Country:

Entidade Country: possibilita anunciar a países específicos

country_id: identificador do país.

country_name: nome do país. Útil para emissão de relatórios. Único.

country::(COUNTRYID, COUNTRYNAME)

Atributo	Tipo	Tamanho	Exemplo
COUNTRYID	Char	3	BRA
COUNTRYNAME	Varchar	30	BRASIL

User, PseudoUser, UserPremium, Is From, Library e Owns:

Entidade User: usuário base. Sem ele não há anunciantes, tampouco aplicação.

user_id: identificador padrão.

user_name: nome de usuário. Identificador do usuário, usado para login. Único.

user_email: confirmação de cadastro, notificações diretas. Único.

user_password: necessário para acesso ao aplicativo.

Relacionamento Is From: Países devem existir sem usuários para poder definir potenciais locais de expansão de acordo com o interesse de anunciantes. Útil também para gerar relatórios de número de usuários por país, inclusive os países cadastrados sem usuários.

Entidade PseudoUser: entidade auxiliar para funcionalidades de pré-assinatura.

Entidade UserPremium: usuário com benefícios. Fonte de renda para a aplicação.

Entidade Library: Ficamos em dúvida quanto à necessidade da entidade. Gostaríamos de ter algo que identificasse as listas de playlists e músicas sem estar relacionado à identificação direta do usuário. Parece ajudar a modularizar o raciocínio, apesar de superficial.

Relacionamento Owns: Segue da mesma justificativa de Library.

_user::(USERID, USERNAME, EMAIL, PASS, PREMIUMSTATE, _COUNTRYID, LIBID)

Atributo	Tipo	Tamanho	Exemplo
USERID	Char	10	SR38A9C5BR
USERNAME	Varchar	20	salvDali
EMAIL	Varchar	40	salv_dali@domain.com
PASS	Varchar	20	12thisAintMaPass!321
PREMIUMSTATE	Numerico	1	0
COUNTRYID	Char	3	BRA
LIBID	Char	10	LIB1054357

PREMIUMSTATE é uma enumeração indicando o tipo de User, como segue: 0 - User; 1 - PseudoUser; 2 - UserPremium.

Restrições de integridade:

COUNTRYID:

Delete: Restrict. Caso existem usuários de um dado país, este deve constar na base de dados.

Update: Cascade.

Is To Firm:

Relacionamento Is To Firm: Necessário para encontrar o(s) usuário(s) a sofrer(em) modificações.

holder_id: necessário para identificar o responsável pelo contrato. Único.

waitingConfirmation::(_USERID, _CONTRACTID)

Atributo	Tipo	Tamanho	Exemplo
USERID	Char	10	SR38A9C5BR
CONTRACTID	Char	10	CT38B978BR
HOLDERID	Char	10	SR38A9C5BR
PLANID	Char	3	PRE

Descrição: Quando o usuário decidir aderir a um plano, uma entrada nesta tabela e em contrato são criadas. A entrada é removida quando o primeiro pagamento for efetuado ou quando a data para o primeiro pagamento expirar.

Restrições de integridade:

CONTRACTID:

Delete: Restrict. Um contrato só pode ser removido da base de dados seguindo um conjunto de passos bem definidos, onde seus relacionamentos serão removidos primeiro.

Update: Restrict. Não nos interessa modificar contratos firmados.

USERID:

Delete: Cascade. Caso o usuário seja removido, este deve ser removido de toda base de dados.

Update: Restrict. Não nos interessa modificar identificadores de usuários.

HOLDERID:

Delete: Cascade. Caso o usuário seja removido, este deve ser removido de toda base de dados. Deve ser criado um gatilho para remover o contrato e pagamento pendentes.

Update: Restrict. Não nos interessa modificar identificadores de usuários.

PLANID:

Delete: Restrict. Um plano não pode ser apagado se houver algum contrato onde conste. Caso se queira remover um plano que consta em uma assinatura, pode-se apenas modificar sua disponibilidade de modo a não permitir novas assinaturas.

Update: Restrict.

Subscribes:

Relacionamento Subscribes: indica usuários vinculados a um contrato.

holder_id: o usuário responsável pelo contrato. Útil para entrar em contato caso seja necessário, ou oferecer melhorias de plano. Único.

subscription::(_CONTRACTID, _USERID, _HOLDERID, _PLANID)

Atributo	Tipo	Tamanho	Exemplo
CONTRACTID	Char	10	CT38B978BR
USERID	Char	10	SR38A9C5BR
HOLDERID	Char	10	SR38A9C5BR
PLANID	Char	3	PRE

HOLDERID é o ID do User responsável pela assinatura.

Restrições de integridade:

CONTRACTID:

Delete: Restrict. Um contrato só pode ser removido da base de dados seguindo um conjunto de passos bem definidos, onde seus relacionamentos serão removidos primeiro.

Update: Restrict. Não nos interessa modificar contratos firmados.

USERID:

Delete: Cascade. Caso o usuário seja removido, este deve ser removido de toda base de dados.

Update: Restrict. Não nos interessa modificar identificadores de usuários.

HOLDERID:

Delete: Cascade. Caso o usuário seja removido, este deve ser removido de toda base de dados. Há risco de inconsistências pois um usuário pode ser responsável por um contrato que abrange outros usuários. Ideal seria tratar este caso com um gatilho OU garantir que os efeitos indesejados estejam descritos em contrato.

Update: Restrict. Não nos interessa modificar identificadores de usuários.

PLANID:

Delete: Restrict. Um plano não pode ser apagado se houver algum contrato onde conste. Caso se queira remover um plano que consta em uma assinatura, este deve ser mantido e sua disponibilidade atualizada de modo a não permitir novas assinaturas.

Update: Restrict.

Plan:

Entidade Plan: Planos devem existir independentemente de estarem vinculados a um usuário e contrato.

plan_id: visa facilitar as consultas que relacionem planos a contratos ou usuários.

plan_name: nome do plano. Útil para apresentações. Único.

plan_price: valor associado à assinatura mensal em dólares.

available: indica se o plano segue disponível para novas assinaturas. Para que contratos sigam contendo planos válidos.

plan::(PLANID, PLANNAME, PRICE, AVAILABLE)

Atributo	Tipo	Tamanho	Exemplo
PLANID	Char	3	PRE
PLANNAME	Varchar	20	Premium
PRICE	Numeric	6,2	16.90
AVAILABLE	Booleano	1	T

Contains:

Relacionamento Contains: Necessário para funcionalidades da aplicação do ponto de vista do usuário. Codifica as playlists salvas pelo usuário.

** A funcionalidade de criação de playlists pelo usuário foi suprimida.

lib_playlist::(_LIBID, _PLAYID)

Atributo	Tipo	Tamanho	Exemplo
LIBID	Char	10	LIB1054357
PLAYID	Char	15	PL3740192837461

Restrições de integridade:

LIBID:

Delete: Cascade. Uma biblioteca só é apagada junto à remoção do usuário, não nos interessando mantê-la neste caso.

Update: Restrict. Não nos interesse modificar o identificador de bibliotecas.

PLAYID:

Delete: Cascade. Playlist sendo deletada, deve deletar as tuplas formadas por aquela playlist.

Update: Restrict. Uma playlist não muda de identificador após recebê-lo.

Hold References:

Relacionamento Hold References: Necessário para funcionalidades da aplicação do ponto de vista do usuário. Codifica as músicas salvas pelo usuário.

lib_music::(_LIBID, _MUSICID)

Atributo	Tipo	Tamanho	Exemplo
LIBID	Char	10	LIB1054357
MUSICID	Char	15	MUS837401728492

Restrições de integridade:

LIBID:

Delete: Cascade. Uma biblioteca só é apagada junto à remoção do usuário, não nos interessando mantê-la neste caso.

Update: Restrict. Não nos interesse modificar o identificador de bibliotecas.

MUSICID:

Delete: Cascade. No momento que uma música é removida, suas informações devem ser removidas da base de dados.

Update: Restrict. Não nos interessa modificar identificadores de músicas.

Playlist e Classify Playlist:

Entidade Playlist: Guarda um conjunto de músicas.

playlist_id: Necessário para que haja uma chave única.

playlist_name: Nome da playlist

is_public: Determina se a playlist é privada ou pública (pode ser exibida a outros usuários em uma busca).

Relacionamento Classify Playlist: Categorização da playlist de acordo com submissões de usuários e possivelmente algoritmos de inteligência artificial. Útil para o usuário.

playlist::(PLAYID, PLAYNAME, ISPUBLIC, *_GENRE1, *_GENRE2)

Atributo	Tipo	Tamanho	Exemplo
PLAYID	Char	15	PL3740192837461
PLAYNAME	Varchar	50	The 60's are back
ISPUBLIC	Boolean	1	T
GENRE1	Char	5	POP00
GENRE2	Char	5	ROCK0

Restrições de integridade:

GENRE1 e GENRE2:

Delete: Set Null. Uma vez que admite valor nulo, quando um gênero for deletado todas as tuplas que o contém irão ser atualizadas com valor nulo para esse gênero.

Update: Cascade.

Music e Classify Music:

Entidade Music: O objeto final da aplicação do ponto de vista do usuário.

music_id: Necessário para que haja uma chave única.

music_name: Nome da música

Relacionamento Classify Music: Categorização da música de acordo com a submissão do artista e possivelmente algoritmos de inteligência artificial. Útil para o usuário.

music::(MUSICID, MUSICNAME, *_GENRE1, *_GENRE2)

Atributo	Tipo	Tamanho	Exemplo
MUSICID	Char	15	MUS837401728492
MUSICNAME	Varchar	30	Singing in the rain
GENRE1	Char	5	HOUS0
GENRE2	Char	5	

Restrições de integridade:

GENRE1 e GENRE2:

Delete: Set Null. Uma vez que admite valor nulo, quando um gênero for deletado todas as tuplas que o contém irão ser atualizadas com valor nulo para esse gênero.

Update: Cascade.

Includes:

Relacionamento Includes: Codifica músicas listadas em uma Playlist.

playlist_music::(_PLAYID, _MUSICID)

Atributo	Tipo	Tamanho	Exemplo
PLAYID	Char	15	PL3740192837461
MUSICID	Char	15	MUS837401728492

Restrições de integridade:

PLAYID:

Delete: Cascade. Playlist sendo deletada, deve deletar as tuplas formadas por aquela playlist.

Update: Restrict. Uma playlist não muda de identificador após recebê-lo.

MUSICID:

Delete: Cascade. No momento que uma música é removida, suas informações devem ser removidas da base de dados.

Update: Restrict. Não nos interessa modificar identificadores de músicas.

Album e Classify Album:

Entidade Album: Codifica um álbum publicado.

album_id: Necessário para que haja uma chave única.

album_name: Nome do album

Relacionamento Classify Album: Categorização do álbum de acordo com uma submissão manual e possivelmente algoritmos de inteligência artificial. Útil para o usuário.

album:: (ALBUMID, NAME, *_GENRE1, *_GENRE2)

Atributo	Tipo	Tamanho	Exemplo
ALBUMID	Char	12	ALB019274659
NAME	Varchar	40	Tropicália
GENRE1	Char	5	TROPI
GENRE2	Char	5	BRPOP

Restrições de integridade:

GENRE1 e GENRE2:

Delete: Set Null. Uma vez que admite valor nulo, quando um gênero for deletado todas as tuplas que o contém irão ser atualizadas com valor nulo para esse gênero.

Update: Cascade.

Features:

Relacionamento Features: Codifica a participação de músicas em álbuns.

album_music::(_ALBUMID, _MUSICID)

Atributo	Tipo	Tamanho	Exemplo
ALBUMID	Char	12	ALB019274659
MUSICID	Varchar	15	MUS837401728492

Restrições de integridade:

ALBUMID:

Delete: Cascade. Um álbum pode ser removido da base de dados por completo.

Update: Restrict. Não nos interessa modificar identificador de álbum.

MUSICID:

Delete: Restrict. Um álbum é uma publicação, e como tal deve ser tratado atomicamente.

Update: Restrict. Não nos interessa modificar identificadores de músicas.

Artist e Classify Artist:

Entidade Artist: Um artista. Útil ao usuário.

artist_id: Necessário para que haja uma chave única.

artist_name: Nome do artista

Relacionamento Classify Artist: Categorização do artista de acordo com uma submissão manual e possivelmente algoritmos de inteligência artificial. Útil para o usuário.

artist::(ARTID, ARTNAME, *_GENRE1, *_GENRE2)

Atributo	Tipo	Tamanho	Exemplo
ARTID	Char	11	ART83740293
ARTNAME	Varchar	30	Johann Sebastian Bach
GENRE1	Char	5	CLAS0
GENRE2	Char	5	

Restrições de integridade:

GENRE1 e GENRE2:

Delete: Set Null. Uma vez que admite valor nulo, quando um gênero for deletado todas as tuplas que o contém irão ser atualizadas com valor nulo para esse gênero.

Update: Cascade.

Takes Part:

Relacionamento Takes Part: Codifica a participação de artistas em um álbum. Útil para o usuário.

artist_album::(_ALBUMID, _ARTID)

Atributo	Tipo	Tamanho	Exemplo
ALBUMID	Char	12	ALB019274656
ARTID	Char	11	ART83740292

Restrições de integridade:

ALBUMID:

Delete: Cascade. Um álbum pode ser removido da base de dados por completo.

Update: Restrict. Não nos interessa modificar identificador de álbum.

ARTID:

Delete: Restrict. Um artista não pode ser removido sem que antes sejam removidos todos seus álbuns.

Update: Restrict. Não nos interessa modificar identificadores de artistas.

Composes:

Relacionamento Composes: Codifica autorias de um artista.

artist_music::(_ARTID, _MUSICID)

Atributo	Tipo	Tamanho	Exemplo
ARTID	Char	11	ART83740292
MUSICID	Char	15	MUS837401728498

Restrições de integridade:

ARTID:

Delete: Restrict. Um artista não pode ser removido sem que antes sejam removidos todas suas músicas.

Update: Restrict. Não nos interessa modificar identificadores de artistas.

MUSICID:

Delete: Cascade.

Update: Restrict. Não nos interessa modificar identificadores de músicas.

Genre:

Entidade Genre:

genre_id: Identificador único.

genre_name: Nome a ser apresentado ao usuário ou em relatórios.

genre::(GENREID, GENRENAME)

Atributo	Tipo	Tamanho	Exemplo
GENREID	Char	5	POP00
GENRENAME	Varchar	30	Pop

Contract, Plan Contract, Temp Plan Contract, Advertising Contract e Bound To:

Entidade Contract: Necessária para identificar de forma generalizada as componentes de um contrato.

contract_id: Identificador único de um contrato.

value: Valor mensal pelo serviço prestado.

Entidade Plan Contract: Um contrato já firmado entre um usuário e o fornecedor da aplicação.

sign_date: Data em que o contrato foi efetivado por ambas as partes.

expire_date: Data de vencimento do contrato.

Entidade Temp Plan Contract: Entidade auxiliar para controles de pré-contratação de planos.

Entidade Advertising Contract: Contrato firmado entre a prestadora e um anunciante.

sign_date: Data em que o contrato foi efetivado por ambas as partes.

expire_date: Data de vencimento do contrato.

Relacionamento Bound To: Identifica a qual documento um contrato está vinculado.

contract::(CONTRACTID, *SIGNDATE, *EXPDATE, DOCID, TYPE)

Atributo	Tipo	Tamanho	Exemplo
CONTRACTID	Char	10	CT38B978BR
VALUE	Float	7	00019.90
SIGNDATE	Date	1	10-06-2018
EXPDATE	Date	1	10-12-2018
DOCID	Char	10	DOC12485BR
TYPE	Numerico	1	2

Descrição: um contrato pode ser criado sem ser confirmado (apenas para TYPE := 0). Por conta disto, SIGNDATE e EXPDATE podem ser nulos. TYPE é uma enumeração para codificar a hierarquia, como segue: 0 - Temp Plan Contract; 1 - Plan Contract; 2 - Advertising Contract.

Payments e Billing:

Entidade Payments: Registra todos os pagamentos efetuados.

payment_id: Identificador único de um registro de pagamento.

received: Verdadeiro se o pagamento foi recebido. Útil para gatilhos e relatórios.

value: Valor do pagamento. Útil para relatórios.

date_received: Data de recebimento. Útil para relatórios.

Relacionamento Billing: Codifica a relação entre um contrato e múltiplos pagamentos.

generation_date: Data em que a guia de pagamento foi gerada.

expire_date: Data em que a guia de pagamento expira. Necessária para gatilhos.

payment::(PAYID, VALUE, RECEIVED, GENDATE, EXPDATE, *DATERECEIV, *_CONTRACTID)

Atributo	Tipo	Tamanho	Exemplo
PAYID	Char	15	PAY839203928394
VALUE	Numeric	6,2	00019.90
RECEIVED	Booleano	1	True
GENDATE	Date	1	12-10-2018
EXPDATE	Date	1	12-11-2018
DATERECEIV	Date	1	13-10-2018
CONTRACTID	Char	10	CT38B978BR

Funcionalidade: Se EXPDATE for alcançada e RECEIVED for False, avalia o tipo de contrato. Se for Temp Plan Contract, apaga o contrato e muda o tipo de usuário para User. Caso contrário, notifica o responsável pelo contrato.

Gatilho: Quando RECEIVED for atualizado para True, verifica o tipo de contrato. Se for Temp Plan Contract, modifica contract.TYPE para Plan Contract e o(s) usuário(s) envolvido(s) para UserPremium. Caso contrário, não faz nada.

Restrições de integridade:

CONTRACTID:

Delete: Set Null. Queremos manter o histórico de pagamentos mesmo que o contrato esteja depreciado.

Update: Restrict. Não nos interessa modificar identificadores de contratos.

Document:

Entidade Document:

document_id: Identificador único.

document_location: Local onde o documento está armazenado. Único.

document::(DOCID, DOCLOC)

Atributo	Tipo	Tamanho	Exemplo
DOCID	Char	10	DOC12485BR
DOCLOC	Varchar	30	./data/Docs/ myDOC12485BR.pdf

Advertiser:

Entidade Advertiser: Representa anunciantes contratantes.

advertiser_id: Identificador único. Útil para consultas.

advertiser_name: Nome do anunciante.

contact_info: Informações de contato. Podem ser múltiplos valores.

advertiser::(ADVID, ADVNAME)

Atributo	Tipo	Tamanho	Exemplo
ADVID	Char	10	ADV759289
ADVNAME	Varchar	30	Crina Brilhante SA

adv_contact::(_ADVID, CONTACT)

Atributo	Tipo	Tamanho	Exemplo
ADVID	Char	10	ADV759289
CONTACT	Varchar	50	Rua dos bobos nr 0

Restrições de integridade:

ADVID:

Delete: Cascade. Se o anunciante for removido da base de dados, não nos interessa manter suas informações de contato.

Update: Restrict. Não nos interessa modificar identificadores de anunciantes.

Signs:

Relacionamento Signs: Um anunciante pode ter múltiplos contratos para exibição de seus diferentes anúncios de acordo com seus variados interesses publicitários.

signed::(_CONTRACTID, _ADVID)

Atributo	Tipo	Tamanho	Exemplo
CONTRACTID	Char	10	CT38B978BR
ADVID	Char	10	ADV740592839

Restrições de integridade:

CONTRACTID:

Delete: Restrict. O contrato com anunciantes só é removido da base de dados através de um gatilho por conta da data de fim de contrato.

Update: Restrict. Não nos interessa modificar identificadores de contrato.

ADVID:

Delete: Restrict. Um anunciante só é removido da base de dados através de uma transação onde as informações de contrato e serão tratadas primeiro.

Update: Restrict. Não nos interesse modificar identificadores de anunciantes.

Advertise:

Entidade Advertise: Representa os anúncios do Udd.

advertise_id: Identificador único do anúncio.

shown_count: O número de vezes que o anúncio foi apresentado. Útil para relatórios.

clicked_count: Número de vezes em que foi clicado. Útil para relatórios.

advertise::(ADID, SHOWCOUNT, CLICKCOUNT)

Atributo	Tipo	Tamanho	Exemplo
ADID	Char	10	AD840293847
SHOWCOUNT	Numerico	8	00000250
CLICKCOUNT	Numerico	8	00000010

Reaches:

Relacionamento Reaches: Codifica relação de usuários que clicou em um dado anúncio.

clickers::(_ADID, _USERID)

Atributo	Tipo	Tamanho	Exemplo
ADID	Char	10	AD840293847
USERID	Char	15	SR38A9C5BR

Restrições de integridade:

ADID:

Delete: Cascade. Quando um anúncio for removido, suas informações serão desconsideradas.

Update: Restrict. Não nos interessa modificar identificadores de anúncio.

USERID:

Delete: Set Null.

Update: Restrict. Não nos interessa modificar identificadores de usuário.

Covers:

Relacionamento Covers: Um contrato cobre um dado número de anúncios em um conjunto de países a serem apresentados.

contract_cover::(COVERID, _ADID, _COUNTRYID, *_CONTRACTID)

Atributo	Tipo	Tamanho	Exemplo
COVERID	Numerico	10	8375018395
CONTRACTID	Char	10	CT38B978BR
COUNTRYID	Char	3	FRA
ADID	Char	10	AD840293847

Descrição: o contrato pode assumir valores nulos por conta de anúncios não vinculados a contratos como, por exemplo, anúncios da aplicação.

Restrições de integridade:

ADID:

Delete: Cascade. Quando um anúncio for removido, sua atuação deve ser removida.

Update: Restrict. Não nos interessa modificar identificadores de anúncio.

CONTRACTID:

Delete: Restrict. Um contrato só pode ser removido dada uma sequência bem definida de passos, onde seus relacionamentos serão removidos primeiro.

Update: Restrict. Não nos interessa modificar identificadores de contrato.

COUNTRYID:

Delete: Restrict. Um país não pode ser removido se houver um contrato vinculado a ele.

Update: Cascade.