



UNIVERSIDADE DE SÃO PAULO

ICMC - Instituto de Ciências Matemáticas e Computação

StickerBOOM 

Dê um boom na sua coleção de álbuns e figurinhas.

SCC0640 - Base de Dados

Prof. Dra. Elaine Parros M. de Sousa

PAE: André Moreira Souza

Carlos Henrique Hannas de Carvalho	NºUSP: 11965988
Guilherme Azevedo Escudeiro	NºUSP: 11345600
Leonardo Hannas de Carvalho Santos	NºUSP: 11800480
Lucas Carvalho Freiburger Stapf	NºUSP: 11800559

23 de agosto de 2023

Sumário

1	Introdução	1
2	Modelo Entidade-Relacionamento (MER)	2
2.1	Levantamento de requisitos	2
2.2	Principais funcionalidades	4
2.3	Análise das Agregações presentes no MER	5
2.4	Análise dos ciclos presentes no MER	6
2.5	Diagrama do Modelo Entidade-Relacionamento	9
3	Correções e Mudanças em relação à Parte 1	10
3.1	Correções	10
3.2	Mudanças	11
4	Modelo Relacional	12
4.1	Diagrama Modelo Relacional	13
4.2	Anotações do Modelo Relacional	14
4.3	Justificativas sobre o mapeamento proposto	15
5	Correções e Mudanças em relação à Parte 2	24
5.1	Correções	24
5.2	Mudanças	25
6	Implementação do Sistema	26
7	<i>Scripts</i> das Consultas	27
7.1	Consulta 1	27
7.2	Consulta 2	27
7.3	Consulta 3	28
7.4	Consulta 4	29
7.5	Consulta 5	30

8	Aplicação	31
9	Conclusão	35

1 Introdução

Em ano de Copa do Mundo é muito comum, não somente entre crianças, mas também entre jovens e adultos, a coleção do álbum de figurinhas do evento quadrienal. Neste ano em específico, o álbum da Copa do Mundo do Qatar conta com 670 figurinhas, sendo que há cinco delas por pacote, que possui um preço fixo de quatro reais. Ademais, para a compra apenas dos livros ilustrados, os preços são R\$44,90 e R\$ 12,00 para capa dura e capa tradicional, respectivamente.

Ao se contabilizar todos os gastos, Gilcione Nonato Costa, como referenciado em [1], doutor em Matemática pela Universidade Federal de Minas Gerais, fez os cálculos, partindo-se de algumas hipóteses, do preço total para se completar o álbum. Em suma, o resultado em que se chegou foi o valor de R\$3178,00, preço equivalente a 794 pacotes ou 3973 figurinhas necessárias para a obtenção do álbum completo - cenário em que não há trocas de figurinhas entre os colecionadores.

Dado que o valor final é incompatível com a realidade de muitos colecionadores, devido aos elevados preços, a solução lógica é a promoção de trocas de figurinhas repetidas entre os colecionadores. Segundo Costa, analisando-se uma amostra de dez amigos trocando figurinhas entre si, o preço final para cada colecionador cairia para R\$900,00 reais - valor consideravelmente inferior aos R\$3178,00 originais.

Mediante a essa situação, o propósito deste projeto é auxiliar os integrantes da comunidade de colecionadores de álbuns a completarem suas coleções, provendo-lhes, de maneira rápida e eficaz, a possibilidade de trocas e vendas de seus itens com os demais membros da comunidade. Finalmente, essa aplicação não se restringe apenas aos álbuns de Copa do Mundo, visto que quaisquer outros álbuns de figurinhas podem ser cadastrados no sistema.

2 Modelo Entidade-Relacionamento (MER)

Padronizou-se os componentes do MER da seguinte maneira: os conjuntos de entidades são identificadas por letras em maiúsculo (ENTIDADE), enquanto os atributos são caracterizados pelos colchetes ([atributos]). Por fim, os conjuntos de relacionamentos serão referenciados em *itálico* neste documento.

2.1 Levantamento de requisitos

O sistema comporta uma entidade USUÁRIO, representado por [CPF], [Nome], [Endereço] e [Cargo] - o [Cargo] subdivide a entidade em outras duas: ADMINISTRADOR e COLECIONADOR. O primeiro *cadastra e remove* algumas entidades no sistema, em uma determinada [Data/Hora], como ÁLBUM, FIGURINHA e BANCA. Além disso, ele também é capaz de realizar a operação de remoção de alguma AVALIAÇÃO devido a um [Motivo] específico ou BANIMENTO de algum COLECIONADOR, com certa [Duração], em uma determinada [Data/Hora].

A entidade BANCA, caracterizada pelo [CNPJ] e [Endereço], realiza vendas de ÁLBUM e FIGURINHA. O ÁLBUM é identificado pelo [ISBN], [Título] e [NroFigurinhas] - sendo que esse último atributo é derivado, através da relação entre as entidades ÁLBUM e FIGURINHA. Vale ressaltar que as entidades ÁLBUM e FIGURINHA referem-se, na verdade, a um tipo específico. Por exemplo, existe apenas uma instância do álbum da Copa de Mundo de 2006, assim como existe apenas uma instância do álbum do Brasileiro 2022 ou da figurinha do jogador Neymar. Por fim, há um relacionamento de posse entre COLECIONADOR e ÁLBUM - isso permite que COLECIONADOR possua mais de uma instância da entidade ÁLBUM.

O PACOTE_FIG, entidade que se refere aos pacotes de figurinhas, é vendido por uma BANCA. Ele é identificado pelo atributo [CodBarras], além do atributo [QntFigurinhas], que indica a quantidade de figurinhas dentro do pacote. Para contextualizar, caso um COLECIONADOR tenha o ÁLBUM da Copa do Mundo de 2022, e ele deseje

adquirir pacotes de figurinhas associados a este tipo de ÁLBUM, cada pacote possui um código de barras associado, além de conter uma quantidade pré-definida de figurinhas desconhecidas.

O COLECIONADOR tem uma [Reputação] e há uma relação de posse entre ele e a entidade FIGURINHA. Essa relação possui um atributo [NroFigurinhas], que é calculado a partir de um determinado [Identificador] da FIGURINHA. O colecionador consegue, em aplicação, atualizar o valor do atributo [NroFigurinhas] em sua posse a qualquer momento. Isso viabiliza ao COLECIONADOR comprar e/ou trocar figurinhas por fora do sistema.

Por fim, o COLECIONADOR também pode realizar alguma NEGOCIAÇÃO sobre sua respectiva FIGURINHA. A NEGOCIAÇÃO, caracterizada pela [Data/Hora] e [Quantidade] que será negociada, pode ser feita em duas diferentes operações: VENDA ou TROCA. Por exemplo, caso o usuário possua figurinhas repetidas e deseja vendê-las ou trocá-las a fim de se obter monetização ou figurinhas inéditas para a sua coleção, o sistema oferece suporte ao usuário, através das operações de VENDA e TROCA.

Uma troca envolve, necessariamente, uma ou mais figurinhas de cada colecionador. Por exemplo, o *Colecionador A* pode trocar três de suas figurinhas por outras duas figurinhas do *Colecionador B*. Uma venda, por sua vez, envolve dois colecionadores: o primeiro vende uma ou mais figurinhas de sua coleção ao segundo.

Ambas operações de VENDA e de TROCA são caracterizadas por um [Local], sendo que a primeira ainda possui um [Valor] para vender o cromo. Ao final de cada ação, um COLECIONADOR pode realizar uma AVALIAÇÃO referente a uma operação que foi realizada com outro COLECIONADOR. A AVALIAÇÃO, por fim, é caracterizada pela [Data/Hora], [Nota], [Comentário] e [Tipo]. O [Tipo], por sua vez, subdivide a AVALIAÇÃO em AVAL_VENDA (referente à VENDA) ou AVAL_TROCA (referente à TROCA). Além disso, a AVALIAÇÃO serve de base para cálculo da [Reputação] do COLECIONADOR, quando ele é avaliado.

2.2 Principais funcionalidades

Há diversas funcionalidades que descrevem a interação e a colaboração entre as entidades do sistema, como pode ser observado no diagrama MER do item (2.5).

A entidade USUÁRIO especializa-se em ADMINISTRADOR e COLECIONADOR, uma vez que ambos possuem atributos comuns. Todavia, a especialização ocorre devido às diferenças de relacionamentos que cada um pode executar, como descrito anteriormente no item (2.1). Ademais, a entidade BANCA participa de relacionamento de venda das entidades ÁLBUM e/ou FIGURINHA. Dessa forma, então, as principais funcionalidades do sistema, bem como as consultas, estão listadas a seguir:

- ADMINISTRADOR

- Banimento de um COLECIONADOR que possa cometer irregularidades no sistema, como a difamação de algum outro COLECIONADOR;
- Remoção de avaliações de aspectos intolerável, com relação à VENDA e/ou TROCA de FIGURINHAS;
- Cadastro e/ou remoção de ÁLBUM e/ou de figurinhas do sistema;
- Cadastro de uma BANCA, onde o COLECIONADOR pode adquirir seu ÁLBUM e/ou figurinhas.

- COLECIONADOR

- Negociar suas figurinhas por outras de interesse. A NEGOCIAÇÃO pode ser feita em instantes distintos, isto é, os colecionadores podem negociar por várias vezes, até que alguma operação, de VENDA e/ou TROCA seja concluída ou não.
- O COLECIONADOR pode registrar no sistema avaliações sobre um outro COLECIONADOR registrado no sistema, após a operação de VENDA e/ou TROCA de figurinhas ser feita;

- BANCA
 - Venda de ÁLBUM de figurinhas;
 - Venda de figurinhas, por unidade ou pacotes.

2.3 Análise das Agregações presentes no MER

1. Agregação NEGOCIAÇÃO

Há duas entidades que fazem parte desta agregação: COLECIONADOR e FIGURINHA. Todo COLECIONADOR possui suas respectivas figurinhas e elas podem ser negociadas em operações de VENDA ou TROCA com outra instância da entidade COLECIONADOR. Em toda NEGOCIAÇÃO feita, há atributos referentes a [Data/Hora] e [Quantidade] de FIGURINHA de uma dada instância. Na operação de VENDA, são relacionados a agregação NEGOCIAÇÃO, cujos componentes são o par COLECIONADOR vendedor das instâncias de FIGURINHA, bem como a entidade COLECIONADOR, que exercerá a compra. A agregação VENDA é caracterizada por um [Valor] e [Local] da operação. Por fim, ao comprador é possível realizar uma AVALIAÇÃO referente à VENDA da qual participou (AVAL_VENDA) registrando [Data/Hora], [Nota] e [Comentário]. A agregação TROCA é composta por duas entidades agregadas NEGOCIAÇÃO, ou seja, envolve dois pares (COLECIONADOR, FIGURINHA). Ademais, possui como atributo o [Local] de realização da operação. Ao final da TROCA, é possível realizar uma AVALIAÇÃO (AVAL_TROCA) em relação ao COLECIONADOR, com o qual foi feita a TROCA.

2. Agregação ÁLBUM VIRTUAL

A agregação ÁLBUM VIRTUAL reúne as entidades COLECIONADOR e ÁLBUM. Assim, dada uma instância de COLECIONADOR que possui um ÁLBUM, necessariamente, o usuário possuirá um único ÁLBUM VIRTUAL, devido à agre-

gação presente no diagrama. Portanto, a chave do **ÁLBUM VIRTUAL** será composta pela chave do **COLECIONADOR**, que seria o [CPF] de **USUÁRIO**, além da chave da entidade **ÁLBUM**, o [ISBN]. O principal objetivo da existência dessa agregação é proporcionar ao **COLECIONADOR** uma melhor administração sobre seu **ÁLBUM** - isto é, controlar quais figurinhas já estão coladas em seu **ÁLBUM** físico. Dada uma instância de **ÁLBUM VIRTUAL**, ela deverá, necessariamente, possuir o mesmo conjunto de figurinhas que a instância **ÁLBUM** que a gerou.

2.4 Análise dos ciclos presentes no MER

1. Ciclo **COLECIONADOR** → **FIGURINHA**

O ciclo envolvendo tais conjuntos de entidades envolve os conjuntos de relacionamentos *Possui* e *Negocia*. O relacionamento *Possui* indica a posse entre um determinado tipo de **FIGURINHA** e um **COLECIONADOR**. O relacionamento *Negocia* fecha o ciclo. Dessa forma, para que o **COLECIONADOR** negocie uma **FIGURINHA**, isto é, trocá-la ou vendê-la, é necessário, antes, a averiguação de que, de fato, tal **COLECIONADOR** possua o cromo para negociá-lo. Essa verificação será feita em aplicação. Não foi possível quebrar esse ciclo de dependência, visto que esta foi a única maneira encontrada para relacionar o **COLECIONADOR** com as suas figurinhas e as suas respectivas quantidades (de cada tipo de **FIGURINHA**).

2. Ciclo **NEGOCIAÇÃO** → **COLECIONADOR**

Este ciclo é regido pelo relacionamento *Venda*. Nele, são relacionados uma entidade de **COLECIONADOR** com uma de **NEGOCIAÇÃO**, a qual envolve o par (**COLECIONADOR**, **FIGURINHA**). O **COLECIONADOR** deste par venderá o cromo para outra instância da entidade **COLECIONADOR**. O ciclo em questão pode ser problemático, visto que a instância do **COLECIONADOR** comprador do cromo não pode ser a mesma da instância do **COLECIONADOR** vendedor do produto.

Em outras palavras, um COLECIONADOR não pode vender uma FIGURINHA para ele mesmo. Essa questão será tratada em aplicação.

3. Ciclo NEGOCIAÇÃO

Formado pelo auto-relacionamento *Troca* da entidade NEGOCIAÇÃO, este ciclo não pode ser quebrado uma vez que a troca é baseada neste relacionamento. Um problema gerado e que deve ser tratado em aplicação é que uma *Troca* não deve ser realizada por uma mesma instância de COLECIONADOR.

4. Ciclo COLECIONADOR \rightarrow ÁLBUM \rightarrow FIGURINHA

Trata-se de um ciclo natural do sistema e não implica problemas para o sistema em análise.

5. Ciclo ÁLBUM VIRTUAL \rightarrow ÁLBUM \rightarrow FIGURINHA

Este é um ciclo de dependência, dado que o relacionamento *Contém* entre instâncias de ÁLBUM VIRTUAL e FIGURINHA só é válido caso a FIGURINHA pertença à instância relativa à entidade ÁLBUM em questão.

6. Ciclo COLECIONADOR \rightarrow AVALIAÇÃO \rightarrow VENDA/TROCA

Tem-se um ciclo de dependência ao se considerar os relacionamentos *Realiza* e *Refere-se*, uma vez que para que um COLECIONADOR realize uma AVALIAÇÃO de uma TROCA, ele deve ser um dos participantes da troca em questão e, para realizar uma AVALIAÇÃO de uma VENDA, o colecionador deve ser o comprador no processo de venda.

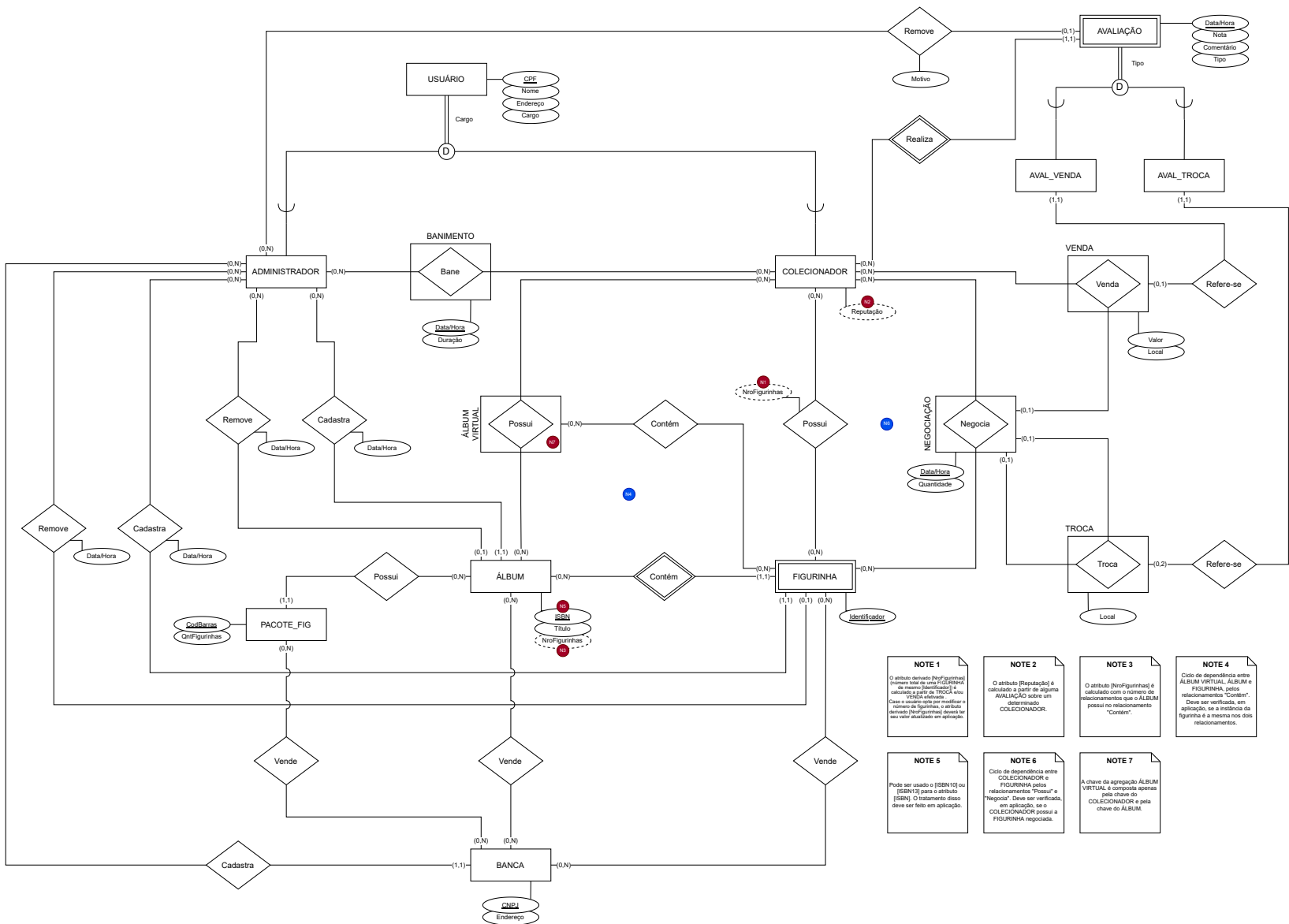
7. Ciclo BANCA \rightarrow ÁLBUM \rightarrow FIGURINHA

Este ciclo se trata de uma naturalidade no sistema. A entidade BANCA não necessariamente precisa vender uma FIGURINHA de um determinado ÁLBUM, ou vice-versa. Por exemplo, ela vende a figurinha pertencente ao álbum X e isso não implica que ela venderá a instância desse álbum.

8. **Ciclo BANCA \rightarrow ÁLBUM \rightarrow PACOTE_FIG**

É um ciclo natural do sistema. Assim, uma BANCA pode vender pacotes de figurinhas de um determinado ÁLBUM, porém não vender tal instância de ÁLBUM.

2.5 Diagrama do Modelo Entidade-Relacionamento



3 Correções e Mudanças em relação à Parte 1

Com relação à primeira entrega, foram realizadas as seguintes correções/mudanças no projeto:

3.1 Correções

1. O Note 1 do MER foi modificado. Nele, foi acrescentado o fato de o usuário poder modificar seu número de figurinhas. Com isso, o valor do atributo [NRO-Figurinhas] deverá ser atualizado em aplicação. Esse caso pode ocorrer quando um COLECIONADOR opte por vender, comprar ou trocar figurinhas com outros colecionadores ainda não cadastrados na base de dados modelada.
2. Foi adicionado um exemplo a fim de se deixar mais claro as operações de venda e de troca.
3. Ao tratar sobre as principais funcionalidades do sistema, ficou subjetivo o significado da entidade USUÁRIO, porém, de forma sucinta, indicamos as especializações (herança) desta entidade e as respectivas funcionalidades relacionadas ao COLECIONADOR e ao ADMINISTRADOR.
4. Foi esclarecido o fato de um COLECIONADOR poder possuir apenas um único ÁLBUM VIRTUAL para uma dada instância de ÁLBUM. Assim, a agregação ÁLBUM VIRTUAL não precisará conter uma chave adicional. Ela apenas herdar as chaves das entidades que fazem parte da agregação, ou seja, as chaves de COLECIONADOR e de ÁLBUM.
5. Adicionou-se um novo estudo de ciclo entre as entidades BANCA, ÁLBUM e FIGURINHA. Essa análise trata-se do caso em que uma BANCA pode vender figurinhas de um determinado álbum, apesar de não necessariamente vender essa mesma instância de álbum, ou vice-versa.

6. Com a adição da entidade PACOTE_FIG e de seus respectivos relacionamentos, foi adicionada a análise do ciclo do qual a nova entidade PACOTE_FIG faz parte.

3.2 Mudanças

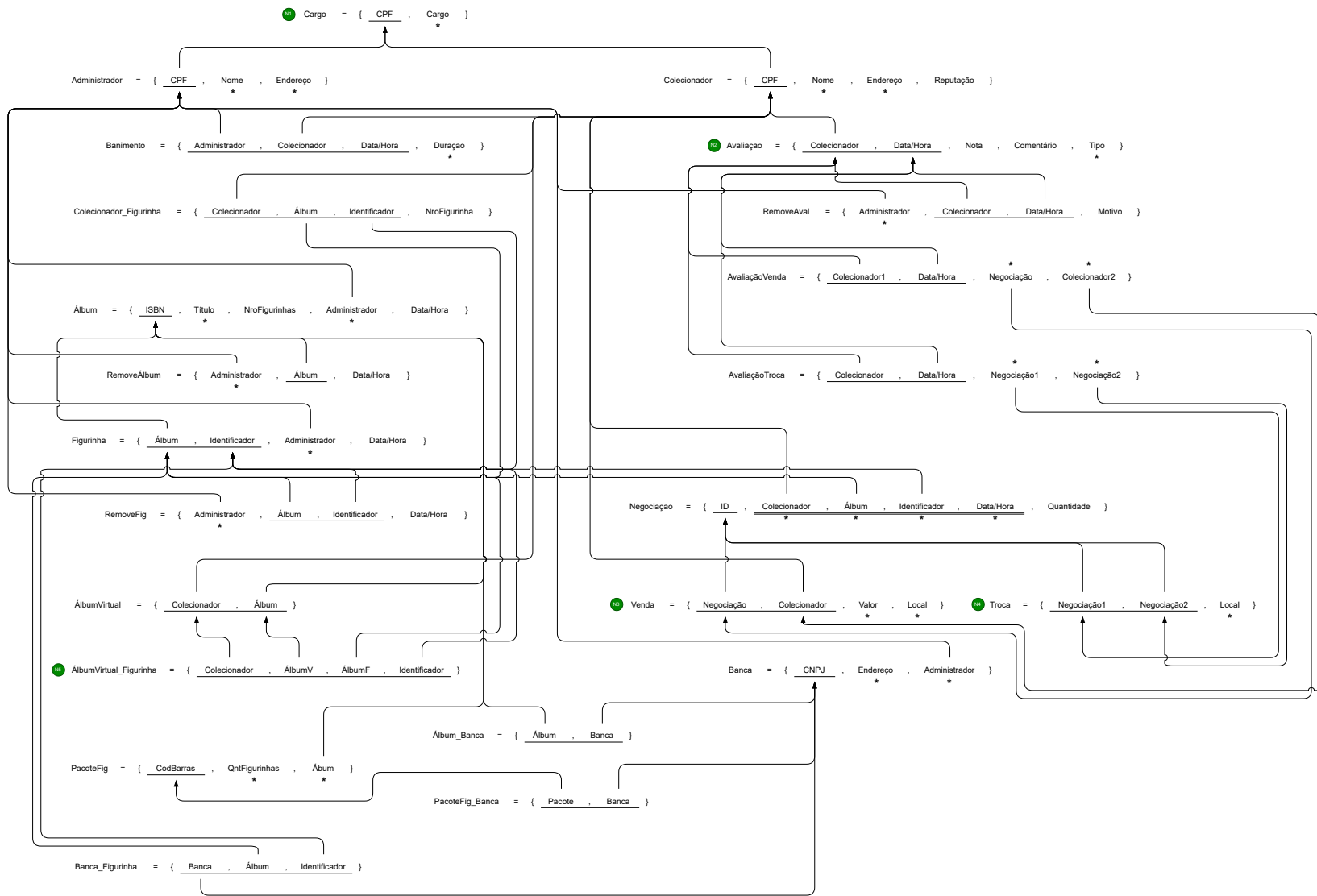
1. Realizou-se algumas mudanças na formatação do documento.

4 Modelo Relacional

Nesta seção trata-se sobre o Mapeamento do Modelo Relacional referente ao tema proposto. Essa forma de mapeamento é um intermédio entre o Mapeamento Entidade-Relacionamento (MER) e a implementação do banco de dados.

Dessa forma, apresentou-se a esquematização do Modelo Relacional referente ao item (2.5). Além disso, há análises da solução adotada para o Modelo Relacional - as análises, por sua vez, da solução envolvem uma breve descrição dela, vantagens, desvantagens e outras alternativas que poderiam ser adotadas..

4.1 Diagrama Modelo Relacional



* NOT NULL

4.2 Anotações do Modelo Relacional

NOTE 1

O mapeamento adotado para a especialização da entidade USUÁRIO em ADMINISTRADOR e COLECIONADOR garante apenas a disjunção, não garantindo a especialização total, que deverá ser garantida na implementação.

NOTE 2

Este mapeamento não garante disjunção e especialização total, sendo necessário, portanto, resolver este problema na aplicação.

NOTE 3

É necessário garantir que os dois usuários envolvidos na venda sejam diferentes.

NOTE 4

É necessário garantir que as duas negociações sejam diferentes. Há ainda outro problema que são pares (Neg1, Neg2) e (Neg2, Neg1), que são entendidos como iguais para o sistema. Essas situações devem ser lidadas na aplicação.

NOTE 5

É necessário verificar se o atributo "ÁlbumV" é igual ao atributo "ÁlbumF" para garantir que uma figurinha não esteja sendo colada em um álbum que não pertença.

4.3 Justificativas sobre o mapeamento proposto

1. Especialização AVALIAÇÃO

- Solução adotada: Criou-se tabelas de AVALIAÇÃO, AVAL_VENDA e AVAL_TROCA porque as três entidades possuem relacionamentos distintos entre si. Isto é, há relacionamentos de cada uma delas com outras entidades do sistema.
- Vantagens: Facilidade para mapear os relacionamentos das entidades, genéricas e específicas, bem como uma maior transparência para consulta de dados, se necessário.
- Desvantagens: Criou-se três tabelas diferentes: uma para a entidade geral e outra para cada conjunto de entidades específicas. Essa solução demanda mais recursos computacionais, como maior demanda de armazenamento, devido ao número de tabelas necessárias para solucionar o problema proposto. Além disso, o mapeamento adotado não garante a especialização total e nem a disjunção.
- Alternativas: Alternativamente, pode-se mapear da seguinte maneira: mapear todas as especializações em uma única tabela, a fim de economizar espaço de memória. Todavia, isso dificultaria o mapeamento dos demais relacionamentos entre as entidades.

2. Especialização USUÁRIO

- Solução adotada: A entidade USUÁRIO não possui relacionamento algum com as demais. Dessa forma, não é necessário criar uma tabela para USUÁRIO, uma vez que serve apenas para intermédio de características comuns dos conjuntos de entidades específicas (CEEs). As tabelas criadas foram as seguintes: uma tabela para cada uma das CEEs (ADMINISTRADOR e COLECIONADOR) e uma tabela que reúne o atributo chave e "*Cargo*" do USUÁRIO - esse último atributo é o critério de distinção das especializações.

- Vantagens: Não foi necessário criar uma tabela para entidade USUÁRIO. Além disso, o modelo adotado para o mapeamento em questão garante a disjunção.
- Desvantagens: A especialização total não é garantida com esse mapeamento.
- Alternativas: Uma alternativa seria o mapeamento da entidade geral USUÁRIO contendo a chave "*CPF*" e os atributos "*Nome*", "*Endereço*" e "*Cargo*". Além disso, uma tabela seria criada para ADMINISTRADOR contendo apenas a chave e outra para COLECIONADOR contendo também a chave e "*Reputação*".

3. Entidade fraca FIGURINHA

- Solução adotada: Criou-se uma tabela de FIGURINHA, sendo identificada por uma chave composta (composição das chaves da entidade forte e entidade fraca). Mapeou-se também o relacionamento *Cadastra* da FIGURINHA por um ADMINISTRADOR.
- Vantagens: Facilidade para mapear todos os relacionamentos em que a entidade fraca participa.
- Desvantagens: Não identificou-se desvantagens para esse tipo de mapeamento.
- Alternativas: Não identificou-se alternativas de mapeamento.

4. Agregação NEGOCIAÇÃO

- Solução adotada: Para esta agregação, optou-se por elaborar uma tabela com os campos "*Colecionador*", "*Álbum*", "*Identificador*", "*Data/Hora*" e "*Quantidade*", sendo que a chave primária composta seria formada pelos quatro primeiros campos. No entanto, dado que essa chave ficaria extremamente extensa, optou-se por substituí-la por um ID sintético "*ID*". Assim, a tabela de NEGOCIAÇÃO possui uma chave primária simples de campo "*ID*", além

de uma chave secundária composta dos campos "*Colecionador*", "*Álbum*", "*Identificador*" e "*Data/Hora*", sendo que cada campo da chave secundária não pode assumir valores nulos.

- Vantagens: A maior vantagem da utilização do ID sintético seria a facilidade nas consultas e aos acessos à tabela da agregação NEGOCIAÇÃO. Além disso, uma outra vantagem é a economia de espaço em tabelas que referenciam à NEGOCIAÇÃO.
- Desvantagens: A adição de um novo campo (ID sintético) na tabela da agregação implica uma maior espaço de armazenamento.
- Alternativas: Uma alternativa imediata seria a não utilização do ID sintético como chave primária e manter apenas a chave composta por "*Colecionador*", "*Álbum*", "*Identificador*" e "*Data/Hora*" como primária.

5. Agregação VENDA

- Solução adotada: A tabela dessa agregação é constituída pelos campos "*Negociação*", "*Colecionador*", "*Valor*" e "*Local*". Utilizou-se chave composta pelos dois primeiros campos; os últimos dois, são campos NOT NULL. É importante ressaltar que o campo "*Colecionador*" deve ser diferente do campo "*Colecionador*" de "*Negociação*". Ou seja, os colecionadores envolvidos na VENDA devem ser diferentes entre si.
- Vantagens: Viabilizou-se a realização de vendas de figurinhas no sistema.
- Desvantagens: A criação de uma nova agregação ocupa um maior espaço de armazenamento.
- Alternativas: Não foram identificadas alternativas de mapeamento.

6. Agregação TROCA

- Solução adotada: A tabela da agregação TROCA é composta pelos campos "*Negociação1*", "*Negociação2*" e "*Local*". A chave da agregação é composta pelas "*Negociação1*" e "*Negociação2*" e devem ser necessariamente distintas para a ocorrência de uma troca; o par ("*Negociação1*", "*Negociação2*") é igual ao par ("*Negociação2*", "*Negociação1*") - tratamento é feito em aplicação. Por fim, o campo "*Local*" deve ser não nulo.
- Vantagens: Foi viabilizada a realização de trocas no sistema.
- Desvantagens: A criação de uma nova agregação implica maior espaço de armazenamento.
- Alternativas: Não foram identificadas alternativas para esse caso.

7. Agregação BANIMENTO

- Solução adotada: Optou-se pela solução clássica de mapeamento de uma agregação. Assim, a tabela de BANIMENTO é composta pelas entidades agregadas, ADMINISTRADOR e COLECIONADOR, bem como pelos atributos próprios da agregação. A tabela, então, é composta pelos seguintes campos: "*Administrador*", "*Colecionador*", "*Data/Hora*" e "*Duração*", em que a chave primária é composta pelos três primeiros campos. Por fim, a ocorrência de um banimento implica uma duração; logo, o campo "*Duração*" deve assumir algum valor NOT NULL.
- Vantagens: Como o atributo "*Data/Hora*" faz parte da chave dessa agregação, é possível armazenar um histórico de banimentos.
- Desvantagens: A criação de uma nova tabela para a agregação implica um maior espaço de armazenamento.
- Alternativas: Não foram identificadas alternativas para esse mapeamento.

8. Agregação ÁLBUM VIRTUAL

- Solução adotada: Optou-se pela solução clássica de mapeamento de uma agregação. Assim, ela é composta pelos seguintes campos: "*Colecionador*" e "*Álbum*", em que ambos são chave composta da entidade agregada. Cabe ressaltar que ÁLBUM VIRTUAL não possui uma chave adicional, pois, dado um COLECCIONADOR e um ÁLBUM, existe apenas um único ÁLBUM VIRTUAL relacionado.
- Vantagens: Conseguiu-se relacionar um único ÁLBUM VIRTUAL a um par (COLECCIONADOR, ÁLBUM).
- Desvantagens: Criou-se uma nova tabela, o que implica à ocupação de um maior espaço de armazenamento.
- Alternativas: Não foram identificadas outras alternativas de mapeamento.

9. Relacionamento "*Cadasta*" entre ADMINISTRADOR e ÁLBUM

- Solução adotada: Utilizou-se o mapeamento padrão de relacionamentos com cardinalidade N:1. Isto é, neste caso, o relacionamento de *Cadasta* foi mapeado na tabela de ÁLBUM.
- Vantagens: A multiplicidade do relacionamento é preservada.
- Desvantagens: Não identificou-se desvantagens em mapear dessa maneira.
- Alternativas: Uma alternativa seria a criação de uma nova tabela para esse conjunto de relacionamento.

10. Relacionamento "*Vende*" entre BANCA e PACOTE_FIG, entre BANCA e ÁLBUM e entre BANCA e FIGURINHA

- Solução adotada: Trata-se de relacionamentos N:N, então há uma tabela para cada um deles. Os atributos de cada tabela são as chaves das entidades em que estão relacionadas entre si. Por fim, as tabelas *Álbum_Banca*, *Pacote-*

Fig_Banca e *Banca_Figurinha*, provenientes dos relacionamentos *Vende*, são identificadas por chaves primárias compostas.

- Vantagens: Preservação da cardinalidade nos relacionamentos.
- Desvantagens: A criação de novas tabelas ocupa maior espaço de armazenamento.
- Alternativas: Não identificou-se alternativas de mapeamento para esse caso.

11. Relacionamento "*Remove*" entre ADMINISTRADOR e ÁLBUM, entre ADMINISTRADOR e FIGURINHA e entre ADMINISTRADOR e AVALIAÇÃO

- Solução adotada: O campo da entidade ADMINISTRADOR nas tabelas é NOT NULL. Vale ressaltar que ele não faz parte das chaves no mapeamento das relações, viabilizando ao ADMINISTRADOR a execução de várias remoções, e não apenas uma. Com isso, a chave da tabela *RemoveAval* é composta de "*Colecionador*" e "*Data/Hora*". A chave da tabela *RemoveÁlbum* é composta apenas pelo campo "*Álbum*". Por fim, a chave da tabela *RemoveFig* é composta por "*Álbum*" e "*Identificador*".
- Vantagens: Evita-se replicação de valores nulos em relação ao mapeamento tradicional.
- Desvantagens: Maior espaço de armazenamento ocupado, devido à criação de novas tabelas.
- Alternativas: Mapear o relacionamento *Remove*, entre ADMINISTRADOR e FIGURINHA, dentro da tabela de FIGURINHA, com a desvantagem da ocorrência de campos nulos. Isso prejudica a otimização do uso de memória, porém aumenta a rapidez da realização de consultas.

12. Relacionamento "*Refere-se*" entre AVAL_VENDA e VENDA e entre AVAL_TROCA e TROCA

- Solução adotada: Realizou-se o mapeamento tradicional de cardinalidade 1:N entre AVAL_TROCA e TROCA. Bem como o mapeamento padrão de cardinalidade 1:1 entre AVAL_VENDA e VENDA. Dessa maneira, o mapeamento dos relacionamentos "*Refere-se*" entre AVAL_TROCA e TROCA e "*Refere-se*" entre AVAL_VENDA e VENDA foram feitos nas tabelas de AVAL_TROCA e AVAL_VENDA, respectivamente.
- Vantagens: Preservação da multiplicidade nos relacionamentos.
- Desvantagens: Criação de novas tabelas acarreta um maior espaço de armazenamento.
- Alternativas: Não identificou-se novas alternativas para esse mapeamento.

13. Relacionamento "*Contém*" entre ÁLBUM VIRTUAL e FIGURINHA

- Solução adotada: Utilizou-se o mapeamento tradicional para relacionamentos com cardinalidade N:N. Logo, criou-se uma nova tabela para mapear o relacionamento *Contém*. Nesta tabela, os campos são: "*Colecionador*", "*ÁlbumV*", "*ÁlbumF*" e "*Identificador*", em que os dois primeiros campos fazem referência ao ÁLBUM VIRTUAL, enquanto os dois últimos referenciam a FIGURINHA. É necessário garantir, em aplicação, que os campos de "*ÁlbumV*" e "*ÁlbumF*" sejam iguais, pois isso garante que uma FIGURINHA de um determinado ÁLBUM não seja alocada incorretamente, isto é, alocar a FIGURINHA em um ÁLBUM do qual ela não faz parte.
- Vantagens: Preservação da multiplicidade.
- Desvantagens: O mapeamento dessa forma pode garantir inconsistência de dados, devido à falta de tratamento em aplicação.
- Alternativas: Não encontrou-se alternativas para esse tipo de mapeamento, devido à multiplicidade.

14. Relacionamento "*Possui*" entre PACOTE_FIG e ÁLBUM

- Solução adotada: Utilizou-se o mapeamento convencional para relacionamentos de cardinalidade 1:N, com participação total de PACOTE_FIG em ÁLBUM. Isto é, o relacionamento entre as entidades é mapeado na tabela de PACOTE_FIG.
- Vantagens: Preservação da cardinalidade no relacionamento.
- Desvantagens: Não encontrou-se desvantagens para esse tipo de mapeamento, devido à multiplicidade do relacionamento.
- Alternativas: Uma alternativa seria a criação de uma nova tabela para esse conjunto de relacionamento.

15. Relacionamento "*Possui*" entre COLECIONADOR e FIGURINHA

- Solução adotada: Realizou-se o mapeamento convencional de relacionamento com cardinalidade N:N. Ou seja, criou-se uma nova tabela para mapear o relacionamento. Além disso, o atributo derivado [NroFigurinhas] foi armazenado nessa mesma tabela.
- Vantagens: Preservação da multiplicidade nos relacionamentos.
- Desvantagens: Não encontrou-se desvantagens para esse tipo de mapeamento, uma vez que é necessário a criação de uma nova tabela devido à multiplicidade do relacionamento.
- Alternativas: Não identificou-se alternativas de mapeamento para esse caso.

16. Atributo derivado [NroFigurinhas] da entidade ÁLBUM e do relacionamento "*Possui*" entre COLECIONADOR e FIGURINHA

- Solução adotada: O atributo [NroFigurinhas] representa a quantidade de um tipo de figurinhas que um COLECIONADOR possui. Logo, ele foi mapeado na tabela de "Colecionador_Figurinha". Trata-se de um atributo derivado,

pois ele é calculado após algumas operações, como TROCA e VENDA. Além disso, o atributo pode ser modificado pelo USUÁRIO na aplicação.

- Vantagens: Atualização constante da quantidade de um tipo de figurinha do COLECIONADOR.
- Desvantagens: Como é um valor que pode ser modificado pelo próprio COLECIONADOR, pode ocorrer inconsistências. Como a diferença de quantidades reais de figurinhas e quantidades cadastradas por ele (COLECIONADOR).
- Alternativas: A alternativa é feita através de cálculos sob demandas, tratados na aplicação.

17. Atributo derivado [Reputação] da entidade COLECIONADOR

- Solução adotada: Esse atributo é associado ao COLECIONADOR, para indicar a [Reputação] dele. Esse atributo é calculado de acordo com as avaliações de outros colecionadores sobre VENDA e/ou TROCA de figurinhas daquele em que se diz à respeito.
- Vantagens: Há um comportamento dinâmico desse atributo, devido à constante inserção de novas avaliações. Portanto, o valor do atributo é alterado com frequência. Por se tratar de um atributo derivado, o valor pode ser atualizado pelo sistema a qualquer momento.
- Desvantagens: O atributo precisa ser redefinido sempre que houver uma nova inserção de avaliação.
- Alternativas: A alternativa é feita através de cálculos sob demandas, tratados na aplicação.

5 Correções e Mudanças em relação à Parte 2

Com relação à segunda entrega, foram realizadas as seguintes correções/mudanças no projeto:

5.1 Correções

1. Corrigiu-se a especialização de USUÁRIO no item (2) de (4.3), através da remoção de um item em "desvantagem". Essa remoção foi realizada, pois a descrição antes apresentada não tratava-se de um problema - a descrição tratava-se apenas de uma maneira de mapeamento relacional de especialização de disjunção.
2. A alternativa de mapeamento descrita no item (3) de (4.3) não era uma alternativa da entidade fraca FIGURINHA. A alternativa refere-se ao conjunto de relacionamento *Remove* entre ADMINISTRADOR e FIGURINHA: item (11) de (4.3). Fez-se a atualização das alternativas, conforme foi descrito.
3. Adicionou-se uma outra vantagem no item (4) de (4.3). A vantagem adicionada é em relação à economia de espaço em tabelas que referenciam à NEGOCIAÇÃO.
4. Adicionou-se uma alternativa de mapeamento no item (9) de (4.3). A alternativa trata-se da criação de uma nova tabela para o relacionamento.
5. Adicionou-se uma alternativa de mapeamento no item (14) de (4.3). A alternativa trata-se da criação de uma nova tabela para o relacionamento.
6. Corrigiu e atualizou-se a alternativa de mapeamento do item (16) em (4.3).
7. Corrigiu e atualizou-se a alternativa de mapeamento do item (17) em (4.3).

5.2 Mudanças

1. Na tabela AVALIAÇÃO_VENDA havia dois atributos com o mesmo nome ([Colecionador]). Os nomes dos atributos foram modificados para [Colecionador1] e [Colecionador2].

6 Implementação do Sistema

O protótipo da plataforma StickerBOOM foi desenvolvido na linguagem *Java*, utilizando o *OpenJDK 19* como kit de desenvolvimento e o *Maven* como gerenciador de projetos. O Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido foi o *Oracle SQL*. O repositório do projeto pode ser acessado através do link: <https://github.com/LucasStapf/stickerboom>.

Montou-se o esquema da base de dados na linguagem SQL, isto é, criação de tabelas e inserção de dados em cada uma delas, com os devidos tratamentos de consistência. Além disso, fez-se cinco consultas de diferentes complexidades para testes no sistema.

7 *Scripts* das Consultas

7.1 Consulta 1

Objetivo: Selecionar a tupla da tabela AVALIAÇÃO_VENDA referente ao maior valor de VENDA.

Essa consulta é baseada em duas tabelas: AVALIAÇÃO_VENDA e VENDA. Inicialmente, busca-se a conectividade e a relação entre essas tabelas através das chaves estrangeiras, pela funcionalidade de junção interna. A condição de busca ocorre na tabela VENDA, no atributo [Valor]. Nesse atributo, seleciona-se a função agregada 'MAX()' para retornar a instância de venda de maior valor.

Por fim, satisfazendo as condições e correlacionando as tabelas como foi descrito anteriormente, a busca retorna o que foi proposto inicialmente: a tupla de AVALIAÇÃO_VENDA referente ao maior valor de VENDA.

Abaixo, encontra-se o código na extensão *.sql*:

```
1 SELECT AV.COLECIONADOR_1, AV.DATA_HORA, AV.NEGOCIACAO,  
2       AV.COLECIONADOR_2, V.VALOR  
3 FROM AVALIACAO_VENDA AV JOIN VENDA V  
4       ON AV.NEGOCIACAO = V.NEGOCIACAO  
5       AND AV.COLECIONADOR_2 = V.COLECIONADOR  
6 WHERE V.VALOR = (SELECT MAX(VALOR) FROM VENDA);
```

7.2 Consulta 2

Objetivo: Selecionar os atributos [Nota] e [Comentário] da relação AVALIAÇÃO referente à VENDA de maior valor.

Essa consulta envolve três tabelas: VENDA, AVALIAÇÃO_VENDA e AVALIAÇÃO. Inicialmente, busca-se a conectividade entre as tabelas VENDA e AVALIAÇÃO_VENDA, através das chaves estrangeiras, pela funcionalidade de junção interna. Além disso,

utiliza-se a função agregada 'MAX()' no atributo [Valor] de VENDA, para determinar a instância mais cara de venda.

Em seguida, através de uma outra junção interna entre as tabelas AVALIAÇÃO e AVALIAÇÃO_VENDA, correlacionando corretamente as chaves estrangeiras de cada uma delas, foi possível averiguar a [Nota] e o [Comentário] da instância de avaliação da venda em que envolvia o maior valor.

Abaixo, encontra-se o código na extensão *.sql*:

```
1  SELECT NOTA, COMENTARIO
2      FROM VENDA V JOIN AVALIACAO_VENDA AV
3      ON V.NEGOCIACAO = AV.NEGOCIACAO AND
4      V.VALOR = (SELECT MAX(VALOR) FROM VENDA)
5      JOIN AVALIACAO A
6      ON A.COLECIONADOR = AV.COLECIONADOR_1 AND
7      A.DATA_HORA = AV.DATA_HORA;
```

7.3 Consulta 3

Objetivo: Selecionar todas as figurinhas de copas e suas respectivas quantidades vendidas pela BANCA localizada na Rua "Thiago Silva".

Essa consulta é baseada nas tabelas BANCA_FIGURINHA, BANCA e ÁLBUM. Inicialmente usa-se o comando 'SELECT' para fazer a filtragem da instância de álbum e figurinha e a respectiva quantidade - o último atributo é calculado através da função agregada 'COUNT()'.

Assim, a consulta partiu da junção entre as tabelas BANCA_FIGURINHA e BANCA, a fim de se acessar o atributo [Endereço] da relação BANCA, além de se ter todas as figurinhas vendidas pela BANCA de endereço na Rua "Thiago Silva, 3". A seguir, foi feita a última junção com a tabela ÁLBUM para que fosse possível procurar por todos os títulos de ÁLBUM com a palavra "*Cup*". Finalmente, a consulta foi agrupada pelos

atributos identificadores de FIGURINHA e o operador 'COUNT()' foi utilizado para se efetuar a contagem.

Abaixo, encontra-se o código na extensão *.sql*:

```
1 SELECT BF.ALBUM, BF.IDENTIFICADOR, COUNT(*)
2 FROM BANCA_FIGURINHA BF JOIN BANCA B
3 ON BF.BANCA = B.CNPJ
4 JOIN ALBUM A ON BF.ALBUM = A.ISBN
5 WHERE UPPER(B.ENDERECO) = 'RUA THIAGO SILVA, 3' AND
6 UPPER(A.TITULO) LIKE '%CUP%'
7 GROUP BY BF.ALBUM, BF.IDENTIFICADOR;
```

7.4 Consulta 4

Objetivo: Selecionar a média de completude de todas as instâncias de ÁLBUM_VIRTUAL cadastradas no sistema.

A lógica dessa consulta se baseou no cálculo da relação entre o número de figurinhas não repetidas sob a posse dos colecionadores e o número total de figurinhas, considerando todas as instâncias dos álbuns virtuais.

Assim, buscou-se inicialmente o número total de figurinhas não repetidas na posse dos colecionadores e, tal número foi armazenado na variável TOTAL_FIG. Em seguida, foram percorridos todos os álbuns cadastrados no sistema que continham colecionadores e figurinhas relacionados. Com isso, o número máximo de figurinhas possíveis foi armazenado na variável MAX_FIG. Finalmente, a porcentagem de completude de cada ÁLBUM foi calculada como sendo a razão entre TOTAL_FIG e MAX_FIG.

Abaixo, encontra-se o código na extensão *.sql*:

```
1 SELECT AL.ISBN, AL.TITULO, TOTAL_FIG / MAX_FIG AS "Completude do Album"
2 FROM (SELECT AVF.ALBUM_V, COUNT(*) AS "TOTAL_FIG"
```



```

3      FROM ALBUM_VIRTUAL_FIGURINHA AVF
4      GROUP BY AVF.ALBUM_V) TABELA_MF
5      JOIN (SELECT A.ISBN, SUM(A.NROFIGURINHAS) AS "MAX_FIG"
6             FROM ALBUM A JOIN ALBUM_VIRTUAL AV
7             ON A.ISBN = AV.ALBUM
8             GROUP BY A.ISBN) TABELA_TF
9      ON TABELA_MF.ALBUM_V = TABELA_TF.ISBN
10     JOIN ALBUM AL
11     ON AL.ISBN = TABELA_TF.ISBN;

```

7.5 Consulta 5

Objetivo: Selecionar os atributos [CPF] e [Nome] dos colecionadores que colecionam todos os álbuns cadastrados no sistema.

Nesta consulta, empregou-se um exemplo de divisão relacional, o qual foi implementado com base na operação de diferença entre conjuntos, através do operador 'MINUS'.

Abaixo, encontra-se o código na extensão *.sql*:

```

1  SELECT DISTINCT AV1.COLECIONADOR, C.NOME FROM ALBUM_VIRTUAL AV1
2  JOIN COLECIONADOR C ON AV1.COLECIONADOR = C.CPF
3  WHERE NOT EXISTS (
4      SELECT ISBN FROM ALBUM
5      MINUS
6      SELECT AV2.ALBUM FROM ALBUM_VIRTUAL AV2
7      WHERE AV1.COLECIONADOR = AV2.COLECIONADOR);

```

8 Aplicação

A aplicação foi feita em uma interface gráfica simples, para que o usuário final não tivesse nenhuma dificuldade em realizar as operações disponíveis.



Figura 1: Tela inicial da aplicação.

As seguintes funcionalidades estão disponíveis na aplicação:

- Login: Um usuário, se registrado no banco de dados e com cargo de colecionador, é capaz de logar na aplicação.
- Registrar: Um usuário é capaz de se registrar no sistema, inserindo seu CPF, Nome e Endereço.
- Listar álbuns disponíveis no sistema: Listagem de todos os álbuns, não colecionados pelo usuário, que estão disponíveis no sistema.
- Listar álbuns do usuário: Listagem de todos os álbuns do colecionador.

- Filtrar listas de álbuns: Usuário é capaz de filtrar a lista de álbuns que é mostrado na tela. O filtro por título realiza uma busca no banco de dados por todos os álbuns que tenham a *substring* passada nos seus respectivos títulos. O filtro por ISBN realiza uma busca por todos os álbuns que tenham o início do ISBN igual ao do filtro.

A seguir estão alguns trechos de códigos, encontrados na classe *Queries* (*src/main/java/br/com/stickerboom/database/Queries.java*), da aplicação com declarações em SQL. Tais trechos representam algumas das funcionalidades encontradas no programa.

O código abaixo representa uma busca na lista de álbuns do colecionador, filtrando os resultados pelo título e pelo ISBN.

```

1  PreparedStatement pStmt =
        con.prepareStatement(
2      "SELECT * FROM ALBUM " + "WHERE ISBN IN
        " +
3      "(SELECT AV.ALBUM FROM ALBUM_VIRTUAL AV
        " +
4      "WHERE ? = AV.COLECIONADOR) " +
5      "AND UPPER(TITULO) LIKE
        UPPER((CONCAT(CONCAT('%', ?), '%'))))
        " +
6      "AND TO_CHAR(ISBN) LIKE
        CONCAT(TO_CHAR(?), '%')");
7
8  pStmt.setString(1, collector.getCPF());
9  pStmt.setString(2, title);
10 pStmt.setLong(3, ISBN);
11
12 ResultSet resultSet = pStmt.executeQuery();

```

Neste trecho é realizada a inserção de novos colecionadores no sistema. Todo o

tratamento de *commit* e *rollback* é tratado.

```
1  con = DBConnection.getConnection();
2  con.setAutoCommit(false);
3
4  PreparedStatement pStmt =
    con.prepareStatement(
5      "INSERT INTO CARGO VALUES (?, 'USER')");
6
7  pStmt.setString(1, cpf);
8
9  result = pStmt.executeUpdate();
10
11 if (result != 1) {
12     con.rollback();
13     con.setAutoCommit(true);
14     return result;
15 }
16
17 pStmt = con.prepareStatement(
18     "INSERT INTO COLECIONADOR VALUES " +
19     "(?, ?, ?, DEFAULT)");
20
21 pStmt.setString(1, cpf);
22 pStmt.setString(2, name);
23 pStmt.setString(3, address);
24
25 result = pStmt.executeUpdate();
26
27 if (result != 1) {
28     con.rollback();
29     con.setAutoCommit(true);
30     return result;
31 }
```

32

33 `con.commit()` ;

9 Conclusão

O ano de 2022 foi marcado pela Copa do Mundo no Qatar. Em anos de Copa, é comum a coleção de álbuns de figurinhas, que marcam o evento quadrienal. Devido aos elevados preços dos itens colecionáveis no Brasil, muito se fala sobre as trocas de figurinhas repetidas entre os colecionadores. Mediante a essa situação, esse projeto, STICKERBOOM, propõe-se a auxiliar os integrantes da comunidade de colecionadores a completarem seus respectivos álbuns com certa facilidade e rapidez, não se limitando apenas aos álbuns relacionados à Copa do Mundo, mas sim aos diversos livros de cromo que existem no mercado.

O projeto foi dividido em três grandes e importantes tópicos sobre a criação de um banco de dados. Os tópicos são Modelo Entidade-Relacionamento (MER), Mapeamento Relacional e implementação do sistema. Inicialmente estudou-se sobre as possíveis entidades que poderiam compor o esquema e os relacionamentos envoltivos entre elas. Feita a análise anterior, montou-se o MER para facilitar a visualização do sistema a ser implementado futuramente. A elaboração do diagrama Entidade-Relacionamento não foi uma tarefa trivial - a discussão em grupo e atendimentos de dúvida foram momentos cruciais para que possíveis problemas fossem finalmente solucionados.

Após a construção do MER, traduziu-se o diagrama para o Modelo Relacional - intermédio entre o Modelo Entidade-Relacionamento e a implementação do banco de dados, como estudado em aulas teóricas. O Mapeamento Relacional do projeto STICKERBOOM seguiu as boas práticas de solução, a fim de garantir as formas de normalização e evitar possíveis inconsistências de dados. Por fim, através de um Relacional consistente, pode-se implementar o banco de dados na linguagem SQL - a implementação envolveu a criação de tabelas com as devidas restrições (chaves primárias, secundárias, estrangeiras, *checks* e etc.), inserção de dados em cada um dos esquemas e algumas consultas às tabelas. As buscas de dados foram pensadas a fim de fazer sentido no contexto de aplicação, mas que possuísem complexidades média-alta.

Pode-se concluir que o projeto possui excelência naquilo que foi proposto inicialmente - vínculo entre o conteúdo de sala e a implementação de um sistema que envolve uma ação comunitária. Verificou-se certas dificuldades durante a execução do projeto, mas as sessões de dúvidas foram cruciais para dar continuidade ao bom andamento da prática. Por fim, verifica-se também a importância dessa aplicação, que envolve a implementação de um banco de dados de cenário real.

Referências

- [1] Marcello De Vico. Quanto custa completar o álbum de figurinhas da copa? É mais do que parece. URL: <https://www.uol.com.br/esporte/futebol/ultimas-noticias/2022/07/20/quanto-custa-completar-o-album-de-figurinhas-da-copa-2022.htm>.