

**UNIVERSIDADE FEDERAL DE SÃO CARLOS CAMPUS
SOROCABA – CIÊNCIA DA COMPUTAÇÃO**

MARCELO PIRRO - RA: 800510

GUILHERME CAMPOS BORTOLETTO - RA: 801477

**FASE INTERMEDIÁRIA II DO PROJETO PRÁTICO DO GRUPO
10**

Tema – Biblioteca

Entrega da fase final do projeto do
Grupo 10 apresentado para a disciplina
Banco de Dados.

Professor responsável: Prof^ª. Dra.
Sahudy Montenegro González.

ÍNDICE

1. DESCRIÇÃO DO PROBLEMA	4
1.1. Consultas	5
2. PROJETO CONCEITUAL	6
2.1. Tabelas de metadados	6
3. PROJETO LÓGICO	11
3.1. Mapeamento	11
3.2. Restrições de integridade	13
3.3. Dependências Funcionais	13
4. PROJETO FÍSICO DE BANCO DE DADOS	14
4.1. Criação do Banco de Dados	14
4.2. Alimentação do Banco de Dados	15
4.3. Consultas	16
4.4. Triggers	19
4.4.1. Trigger 1: Atualizar número de exemplares	19
4.4.2. Trigger 2: Verificar se o empréstimo é possível	20
4.5. Considerações finais	21
BIBLIOGRAFIA	23

LISTAS DE ILUSTRAÇÕES

FIGURAS

Figura 1. Diagrama Entidade-Relacionamento.....	6
Figura 2. Diagrama do Modelo Relacional.....	11
Figura 3. Tipo relacionamento Ternário.....	21

TABELAS

Tabela 1. Tipos de atributos por tipo-entidade do <i>DER</i>	7
Tabela 2. Opções de estado civil do cliente.....	9
Tabela 3. Possíveis gêneros do cliente.....	9
Tabela 4. Idiomas dos livros.....	9
Tabela 5. Categorias dos livros.....	10
Tabela 6. Nacionalidade dos autores dos livros nos acervos.....	10

1. DESCRIÇÃO DO PROBLEMA

Uma Biblioteca deseja desenvolver um sistema que mantenha as informações dos usuários cadastrados na biblioteca, dos livros no acervo, dos empréstimos realizados e dos funcionários contratados.

Sobre os usuários deve-se armazenar um nome, um número de telefone (cada usuário deve informar dois números para contato), um endereço (composto por logradouro, nome da rua, bairro, número e CEP), CPF, um código de identificação único na biblioteca (*ID*), sua idade (apenas maiores de 16 anos podem se cadastrar e realizar empréstimos), seu estado civil (podendo ser solteiro (a), casado (a), viúvo (a), divorciado (a) e separado (a)) e seu gênero (masculino, feminino ou outro (a)).

Um empréstimo ocorre quando existe um usuário, um funcionário e pelo menos um exemplar de algum livro, porém nem todo usuário e funcionário realizam um empréstimo e nem todo exemplar é emprestado. Todo empréstimo possui armazenado a data em que foi realizado e uma data prevista para a devolução do exemplar.

Um funcionário possui um número de identificação próprio na biblioteca (*ID*), um nome, um CPF, um gênero (masculino, feminino ou outro (a)), uma idade (apenas maiores de 18 anos podem ser contratados) e dois números de telefone registrados no sistema para contato.

Cada usuário pode emprestar diversos exemplares por vez e sempre que um empréstimo for realizado é necessário armazenar a data de retirada e a data de devolução prevista para que o cliente devolva o exemplar.

Um livro pode possuir vários exemplares e cada exemplar possui um código único de identificação de acordo com o total de exemplares presentes do livro na biblioteca e um nível de conservação que varia de 0 a 5 para representar o estado de conservação do exemplar.

Um livro possui um título, um ano de publicação, um número total de páginas, um número de exemplares que é calculado de acordo com os exemplares cadastrados no acervo da biblioteca, um Padrão Internacional de Numeração de Livro (*ISBN*) único para identificação, uma categoria (ciência, esporte, biografia, história, ficção científica, romance, suspense, fantasia, infantil, drama, poesia, religião, ação e aventura ou outro (a)) que serve para organizar os livros no acervo (cada livro possui apenas uma categoria), uma editora, um idioma em que está escrito (inglês, português, espanhol ou outro) e um ou mais autores.

Cada livro possui apenas uma editora, e uma editora pode possuir diversos livros presentes no acervo. Uma editora é identificada por seu CNPJ, e possui um nome, um número de telefone opcional para contato (o sistema deve permitir armazenar um número para contato caso ele exista) e um endereço (composto por logradouro, nome da rua, bairro, número e CEP) da sede da editora.

Um autor, obrigatoriamente, deve possuir ao menos um livro cadastrado, um nome único que o identifica na biblioteca (*ID*) e uma nacionalidade (brasileiro, estadunidense, canadense, alemão, francês, português, espanhol ou outro (a)).

1.1. Consultas

Marcelo Pirro

1. Quantos Autores estão cadastrados?
2. Qual livro possui o maior número de exemplares?
3. Qual categoria possui mais livros ?
4. Selecionar o nome dos autores que escreveram um livro publicado em um determinado ano.
5. Listar o título do livro e o nome da editora que o publicou, ordenados por ano de publicação em ordem crescente.

Guilherme Bortoletto

6. Listar todos os autores que não sejam brasileiros, caso exista.
7. Quantos exemplares estão emprestados ?
8. Listar todas as Editoras com mais de dois livros, se existir.
9. Selecionar o título e o nome da editora de todos os livros emprestados por um determinado usuário.

2. PROJETO CONCEITUAL

Para a construção do diagrama Entidade-Relacionamento utilizamos o site *Diagrams*, o diagrama é representado pela Figura 1 abaixo e encontra-se disponível no link: l1nk.dev/ov2L1.

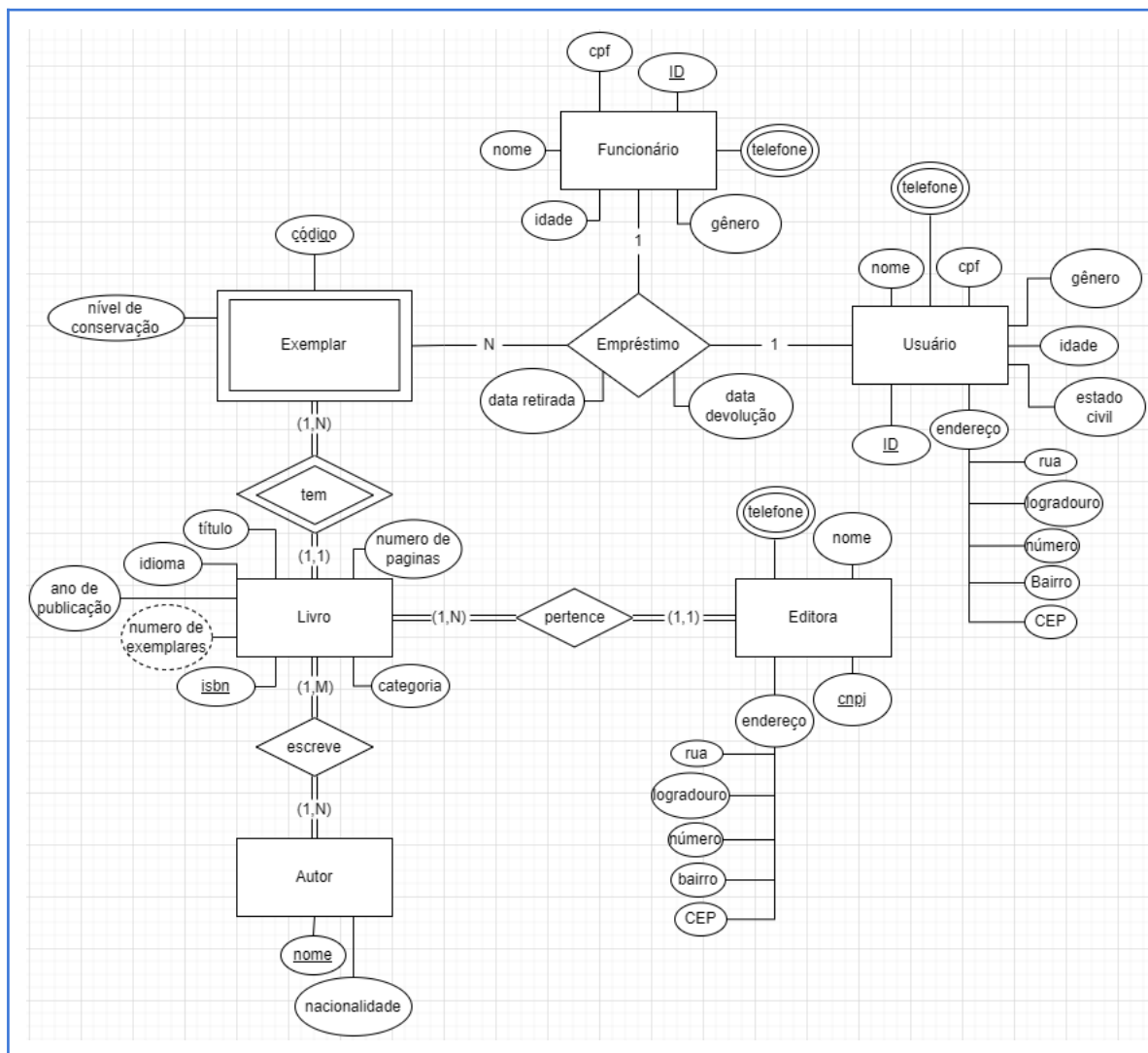


Figura 1. Diagrama Entidade-Relacionamento

Anotação semântica: O identificador parcial na entidade *Exemplar* é um número incremental gerado a partir do total de exemplares disponíveis do mesmo livro na biblioteca.

2.1. Tabelas de metadados

A Tabela 1 abaixo representa os tipos de atributos por tipo-entidade do *DER*, em seguida encontram-se as Tabelas 2, 3, 4, 5 e 6 que representam os

catálogos para os atributos de estado civil, gênero, idioma, categoria e nacionalidade, respectivamente.

Tipo-Entidade	Atributo	Tipo	Restrição
usuário	ID	Identificador ▾	obrigatório
	nome	Monovalorado ▾	obrigatório
	telefone	Simples ▾ Multivalorado ▾	obrigatório
	CPF	Simples ▾ Unico ▾	obrigatório, apenas dígito
	idade	Simples ▾	obrigatório, > de 16 anos
	estado civil	Monovalorado ▾	obrigatório, tabela 2
	gênero	Monovalorado ▾	obrigatório, tabela 3
	endereço	Composto ▾	obrigatório
	endereço_log radouro	Monovalorado ▾	obrigatório
	endereço_ rua	Monovalorado ▾	obrigatório
	endereço_ numero	Simples ▾	obrigatório, apenas dígitos
	endereço_ bairro	Monovalorado ▾	obrigatório
	endereço_ CEP	Simples ▾	obrigatório, apenas dígitos
funcionário	ID	Identificador ▾	obrigatório
	nome	Monovalorado ▾	obrigatório
	idade	Simples ▾	obrigatório, > 18 anos
	CPF	Simples ▾ Unico ▾	obrigatório, apenas dígitos

Tipo-Entidade	Atributo	Tipo	Restrição
	gênero	Monovalorado ▾	obrigatório, tabela 3
	telefone	Simples ▾ Multivalorado ▾	obrigatório, apenas dígitos
exemplar	código	Id parcial ▾	obrigatório
	nível de conservação	Simples ▾	obrigatório, entre 0 e 5
livro	ISBN	Identificador ▾	obrigatório
	título	Monovalorado ▾	obrigatório
	ano de publicação	Simples ▾	obrigatório, formato 4 dígitos
	número de exemplares	Simples ▾ Calculado ▾	obrigatório, =0
	categoria	Monovalorado ▾	obrigatório, tabela 5
	número de páginas	Simples ▾	obrigatório, >=1
	idioma	Monovalorado ▾	obrigatório, tabela 4
editora	CNPJ	Identificador ▾	obrigatório, apenas dígitos
	telefone	Simples ▾ Monovalorado ▾	opcional
	nome	Monovalorado ▾	obrigatório
	endereço	Composto ▾	obrigatório
	endereço_logradouro	Monovalorado ▾	obrigatório
	endereço_ rua	Monovalorado ▾	obrigatório
	endereço_numero	Simples ▾	obrigatório, apenas dígitos
	endereço_bairro	Monovalorado ▾	obrigatório

Tipo-Entidade	Atributo	Tipo	Restrição
autor	endereço_cep	<input type="text" value="Simples"/>	obrigatório, apenas dígitos
	ID	<input type="text" value="Identificador"/>	obrigatório
	nome	<input type="text" value="Monovalorado"/>	obrigatório
	nacionalidade	<input type="text" value="Monovalorado"/>	obrigatório, tabela 6

Tabela 1. Tipos de atributos por tipo-entidade do DER.

Estado civil

Solteiro (a)
Casado (a)
Divorciado (a)
Viúvo (a)
Separado (a)

Tabela 2. Opções de estado civil do cliente.

Gênero

Masculino
Feminino
Outro (a)

Tabela 3. Possíveis gêneros do cliente.

Idioma

Português
Inglês
Espanhol
Outro

Tabela 4. Idiomas dos livros.

Categoria

Ciência
Esportes
Biografia
História
Ficção científica
Romance
Suspense
Fantasia
Infantil
Drama
Poesia
Religião
Ação e Aventura
Outro

Tabela 5. Categorias dos livros.

Nacionalidade

Brasileiro
Estadunidense
Canadense
Alemão
Francês
Português
Espanhol
Outro (a)

Tabela 6. Nacionalidade dos autores dos livros nos acervos.

3. PROJETO LÓGICO

Para a construção do modelo relacional foi utilizada a ferramenta *Diagrams*, ele pode ser visualizado pela Figura 2 abaixo ou através do *link*: encr.pw/dQIqZ.

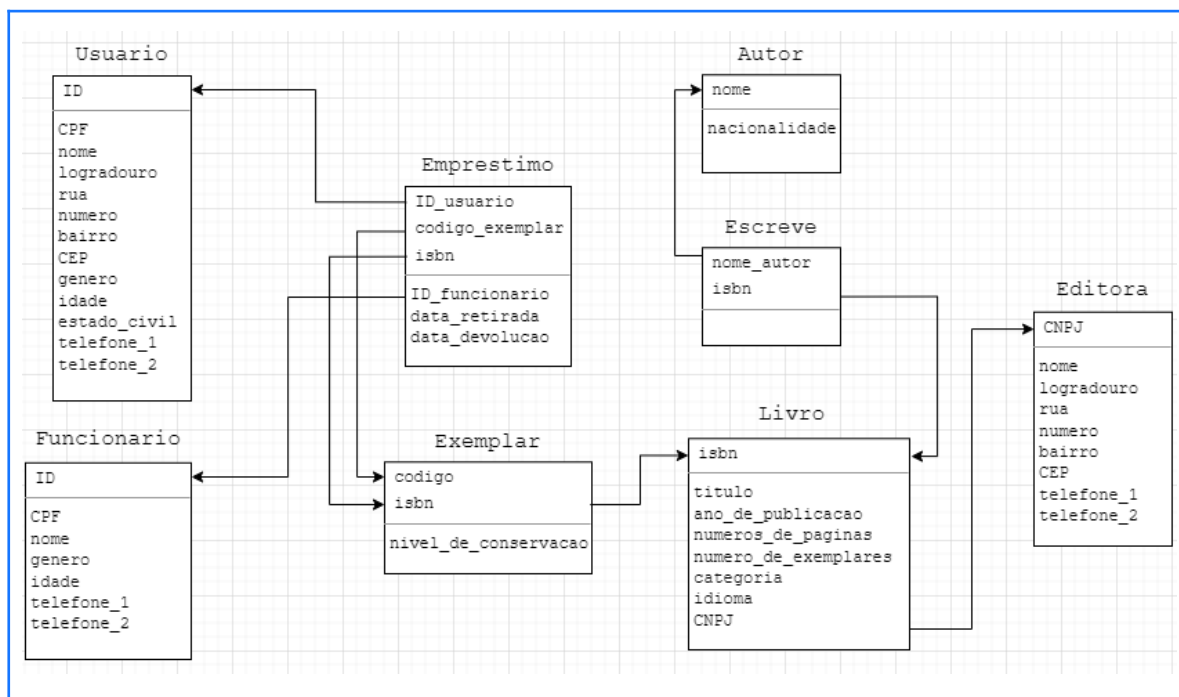


Figura 2. Diagrama do Modelo Relacional

3.1. Mapeamento

O mapeamento foi feito de forma manual, e o resultado obtido atende a 3FN, a seguir serão apresentados os procedimentos realizados para o mapeamento.

O primeiro passo realizado foi o mapeamento de entidades fortes, *Funcionario*, *Usuario*, *Livro*, *Autor* e *Editora*. Todos eles possuem apenas um atributo primário que compõem suas chaves primárias (PK), seguido de seus atributos.

O segundo passo foi mapear as entidades fracas, no caso a entidade *Exemplar*, sua chave primária (PK) é composta por seu identificador parcial e o identificador da entidade *Livro*.

O próximo passo foi realizar o mapeamento dos relacionamentos do tipo 1:N, para esse caso a alternativa preferível de mapeamento é a de adição de colunas na entidade de cardinalidade N. Para o relacionamento entre *Livro* e *Editora*, na entidade *Livro* temos uma chave estrangeira (FK) que

corresponde a chave primária de `Editora` (`CNPJ`), indicando o relacionamento entre as entidades. Para o relacionamento entre `Exemplar` e `Livro`, como `Exemplar` é uma entidade fraca, o relacionamento já está explícito uma vez que a chave primária de `Exemplar` (`codigo; isbn`) possui a chave primária da entidade `Livro` (`isbn`).

Em seguida foi realizado o mapeamento dos relacionamentos tipo N:M, para esse caso a alternativa preferível de mapeamento é a criação de uma tabela própria. A tabela `Escreve` irá representar o relacionamento entre `Autor` e `Livro` e a chave primária (PK) dessa nova tabela será formada pela chave primária de `Autor` (`nome`) e a chave primária de `Livro` (`isbn`).

`Usuario`, `Funcionário` e `Editora` possuem um atributo multivalorado (`telefone`). `Usuarios` e `Funcionarios` devem ter 2 números de telefone e a `Editora` pode ou não possuir algum, tendo a possibilidade de registrar até dois números. Para esses casos foi criado dois atributos monovalorados (`telefone_1` e `telefone_2`) para cada valor de atributo multivalorado ao invés de criar uma nova tabela de relacionamento a fim de evitar junções.

O último procedimento foi realizar o mapeamento do relacionamento ternário, para o relacionamento ternário a cardinalidade em questão é 1:1:N (Terceiro caso). Para que haja o relacionamento `Emprestimo`, é necessário que 1 funcionário esteja disponível e 1 usuário solicite o empréstimo de um ou vários exemplares de livros. Para esse cenário foi criado uma nova tabela `Emprestimo` onde a chave primária será o identificador da entidade de cardinalidade N e o identificador de alguma das entidades de cardinalidade 1 (`Funcionario` ou `Usuario`), e o identificador da entidade de cardinalidade 1 que não for utilizado na chave primária será uma chave estrangeira (FK). Para esse caso a chave primária da tabela que representa o relacionamento ternário será formada pelo identificador de `Exemplar` (`isbn` e `codigo`) e o identificador de `Usuario` (`ID_usuario`), e o identificador de `Funcionario` (`ID_funcionario`) será uma chave estrangeira (FK).

O atributo calculado `numero_de_exemplares` de um livro representa a quantidade de exemplares disponíveis, de um determinado livro, para a locação e é calculado de acordo com o total de exemplares cadastrados do mesmo livro.

3.2. Restrições de integridade

Restrições de Integridade são usadas para garantir a consistência e a exatidão dos dados. A integridade de unicidade de chave garante a identificação única de cada uma das tuplas de uma tabela.

A integridade referencial, garante a identificação do relacionamento entre tabelas, uma chave estrangeira referencia uma outra tabela, estabelecendo uma ligação entre a tabela em que está presente e a tabela a qual está referenciando. A inserção de valores para a coluna da chave estrangeira só será possível se o valor existir na tabela a qual ele referencia.

A integridade da entidade garante que nenhuma chave pode ser nula, já integridade de domínio garante que todos os dados em um campo contenham valores válidos. A integridade do domínio é definida pelo tipo de dados (como número inteiro, caractere ou decimal) e pelo comprimento permitido dos dados. Dessa forma, o modelo relacional elaborado não infringe e respeita todas as restrições de integridade, garantindo a consistência e a exatidão dos dados.

3.3. Dependências Funcionais

As dependências funcionais são dadas por:

Usuário:

$\{ID\} \rightarrow \{CPF, nome, logradouro, rua, numero, bairro, CEP, genero, idade, estado_civil, telefone_1, telefone_2\}$

Funcionário:

$\{ID\} \rightarrow \{CPF, nome, genero, idade, telefone_1, telefone_2\}$

Autor:

$\{nome\} \rightarrow \{nacionalidade\}$

Livro:

$\{isbn\} \rightarrow \{titulo, ano_de_publicacao, numeros_de_paginas, numero_de_exemplares, categoria, idioma, CNPJ_editora\}$

Editora:

```
{CNPJ} → {nome, logradouro, rua, numero, bairro, CEP,
telefone_1, telefone_2}
```

Escreve:

```
{nome_autor, isbn}
```

Exemplar:

```
{codigo, isbn} → {nivel_de_conservacao}
```

Empréstimo:

```
{ID_usuario, codigo_exemplar, isbn} → {ID_fundionario,
data_retirada, data_devolucao}
```

Como mencionado anteriormente, o resultado do mapeamento obtido atende a Terceira Forma Normal (3FN), isso pode ser verificado pois não temos a presença de atributos multivalores, dependências parciais ou transitivas, todas as dependências são dependências totais.

Para que a normalização esteja na Terceira Forma Normal (3FN) ela necessariamente precisa estar na Segunda e na Primeira Forma.

A Primeira Forma Normal (1FN) exige que todos os atributos sejam atômicos, ou seja, não são permitidos atributos multivalorados, (como era o caso do telefone inicialmente).

A Segunda Forma Normal (2FN) exige que a primeira forma seja atendida e não existam dependências parciais, ou seja, os atributos devem depender da chave primária, ou seja, em caso de chave primária com dois ou mais atributos primários, os demais atributos da tabela devem depender inteiramente dos atributos que compõem a chave e não apenas de um deles.

A Terceira Forma Normal (3FN) exige que a segunda forma seja atendida e não existam Dependências Parciais, ou seja, atributos não chave não podem depender de outro atributo não chave.

Com isso concluímos que o modelo obtido atende os requisitos, portanto encontra-se na Terceira Forma Normal (3FN).

4. PROJETO FÍSICO DE BANCO DE DADOS

4.1. Criação do Banco de Dados

Os scripts de criação do banco de dados se encontram anexados no arquivo **ddl.sql**.

Para garantir as restrições estabelecidas anteriormente (Tabela 1) foram adotadas: cláusulas **CHECK** para verificar restrições que exigiam pertencer

a um determinado intervalo, como por exemplo, idade do usuário e do funcionário precisam ser maior que um valor pré definido, além deles `numeros_de_paginas`, `numero_de_exemplares`, `nivel_de_conservacao` também usaram cláusulas `CHECK` para verificar se atendem aos valores estabelecidos (Tabela 1). Para garantir que valores obrigatórios sejam de fatos obrigatórios foi utilizado a cláusula `NOT NULL`, da mesma forma, para valores únicos foi utilizado a cláusula `UNIQUE` e para definir um valor padrão, como no caso do `numero_de_exemplares` de um exemplar, foi utilizado a cláusula `DEFAULT`.

A cláusula `DEFAULT` em `numero_de_exemplares`, define o padrão como zero, uma vez que é um atributo calculado através de um trigger, não tem a necessidade de ser inserido.

Para chaves estrangeiras foram utilizadas as cláusulas: `ON UPDATE CASCADE` e `ON DELETE CASCADE`.

`ON UPDATE CASCADE` e `ON DELETE CASCADE` são cláusulas que definem o comportamento do banco de dados quando um registro referenciado por uma chave estrangeira é atualizado ou excluído. `ON UPDATE CASCADE` especifica que quando um registro na tabela pai é atualizado, todos os registros correspondentes na tabela filha também serão alterados. `ON DELETE CASCADE` especifica que quando um registro na tabela pai é excluído, todos os registros correspondentes na tabela filha também serão excluídos automaticamente.

Para garantir que os atributos atendam os domínios definidos pelas tabelas 2, 3, 4, 5, 6 (Seção 2) foram criados tipo `ENUM` para garantir que só sejam inseridos as opções definidas previamente.

Para Gênero criou-se um tipo `GENERO AS ENUM ('Masculino', 'Feminino', 'Outro(a)')`.

Para Estado civil criou-se o tipo `ESTADO_CIVIL AS ENUM ('Solteiro(a)', 'Casado(a)', 'Divorciado(a)', 'Viúvo(a)', 'Separado(a)')`.

Para Categoria criou-se um tipo `CATEGORIA AS ENUM ('Ciência', 'Esportes', 'Biografia', 'História', 'Ficção científica', 'Romance', 'Suspense', 'Fantasia', 'Infantil', 'Drama', 'Poesia', 'Religião', 'Ação e Aventura', 'Outro')`.

Para Nacionalidade criou-se um tipo `NACIONALIDADE AS ENUM ('Brasileiro', 'Estadunidense', 'Canadense', 'Alemão', 'Francês', 'Português', 'Espanhol', 'Outro(a)')`.

Para Idioma criou-se um tipo `IDIOMA AS ENUM ('Português', 'Inglês', 'Espanhol', 'Outro')`.

Para as restrições que definiam um formato específico ou que fossem inseridos apenas dígitos, como no caso de telefone, CEP, CPF, ISBN, CNPJ e ANO foram criados domínios para garantir que essa integridade fosse respeitada.

4.2. Alimentação do Banco de Dados

Os scripts de inserção de dados no banco de dados se encontram anexados no arquivo **dados.sql**.

A inserção dos dados foi feita de forma fiel, e todos os dados apresentados de usuário e funcionário são dados fictícios, como telefone_1, telefone_2, rua etc.

Os dados inseridos no banco referente aos dados dos livros, autores e editoras, são dados reais, como por exemplo, ISBN, CNPJ, autor_do_livro, ano_de_publicação etc.

O Script respeita a ordem de inserção, **lembrando que a inserção deve ser realizada após a criação dos triggers**.

4.3. Consultas

Os scripts das consultas de dados no banco de dados se encontram anexados no arquivo **consultas.sql**.

Consulta 01: Quantos Autores estão cadastrados?

Consultar quantos autores estão cadastrados no sistema, e retornar a quantidade em uma tabela com o atributo "qtd_autores". Responsável: Marcelo Pirro.

AR

$$\pi_{\text{qtd_autores}} (\rho_{\text{t(qtd_autores)}} (\mathcal{F} \text{ COUNT(nome)(autor)}))$$

SQL

```
SELECT COUNT(nome) AS qtd_autores FROM autor;
```

Consulta 02: Qual livro possui o maior número de exemplares?

Consultar qual dos livros possui o maior número de exemplares cadastrados, retornar uma tabela com o livro e a quantidade cadastrada. Responsável: Marcelo Pirro.

AR

$$\begin{aligned} & t \leftarrow \rho_{\text{t(max_num_ex)}} (\mathcal{F} \text{ MAX(numero_de_exemplares) (livro)}) \\ & \pi_{\text{titulo, numero_de_exemplares}} (\sigma_{\text{numero_de_exemplares} = \text{t.max_num_ex}} (\text{livro})) \end{aligned}$$

SQL

```
SELECT titulo, numero_de_exemplares
FROM livro
WHERE numero_de_exemplares = (
```



```

SELECT MAX(numero_de_exemplares)
FROM livro
);

```

Consulta 03: Qual categoria possui mais livros ?

Verificar qual categoria possui maior quantidade de livros cadastrados, retornar a categoria e o total de livros cadastrados na mesma. Responsável: Marcelo Pirro.

AR

```

t1 ← categoria, total_livros  $\mathcal{F}$  COUNT(categoria)(livro)
t2 ←  $\mathcal{F}$  MAX(total_livros)(t1)
 $\Pi$  categoria, total_livros (t1  $\bowtie$  t1.total_livros = t2.maximo t2)

```

SQL

```

SELECT categoria, total_livros
FROM (
    SELECT categoria, COUNT(*) AS total_livros
    FROM livro
    GROUP BY categoria
)t
JOIN (
    SELECT MAX (total_livros) AS MAXIMO
    FROM (
        SELECT categoria, COUNT(*) AS total_livros
        FROM livro
        GROUP BY categoria
    )t
)m
ON t.total_livros = m.maximo;

```

Consulta 04: Listar todos os autores que não sejam brasileiros, caso exista.

Consultar os autores que não são Brasileiros e retornar seu nome e nacionalidade. Responsável: Marcelo Pirro.

AR

```

 $\Pi$  nome, nacionalidade ( $\sigma$  nacionalidade  $\neq$  'Brasileiro' (autor))

```

SQL

```
SELECT nome, nacionalidade FROM autor
WHERE nacionalidade <> 'Brasileiro';
```

Consulta 05: Quantos exemplares estão emprestados ?

Calcular a quantidade de exemplares emprestados. Responsável: Marcelo Pirro.

AR

```
ρ t(total_emprestimos) (Σ COUNT(*) (emprestimo))
```

SQL

```
SELECT COUNT(*) AS total_emprestados FROM emprestimo;
```

Consulta 06: Listar todas as Editoras com mais de dois livros, se existir.

Listar as Editoras com mais de dois livros e a respectiva quantidade de livros de cada uma delas. Responsável: Guilherme.

AR

```
σ total_livros > 2 (ρ t2(editora.nome, total_livros) ( editora.nome Σ COUNT(*)
(editora ⋈ livro))
```

SQL

```
SELECT editora.nome, COUNT(*) AS total_livros
FROM editora
NATURAL JOIN livro
GROUP BY editora.nome
HAVING COUNT(*) > 2;
```

Consulta 07: Selecionar o nome dos autores que escreveram um livro publicado em um determinado ano.

Listar os autores que escreveram um livro que foi publicado no ano '<ano>'. Responsável: Guilherme.

AR

```
t1 ← autor ⋈ autor.nome = escreve.nome_autor escreve ⋈ escreve.isbn =
livro.isbn livro
π nome (σ livro.ano_de_publicacao='2022' (t1))
```

SQL

```

SELECT autor.nome
FROM autor
INNER JOIN escreve ON autor.nome = escreve.nome_autor
INNER JOIN livro ON escreve.isbn = livro.isbn
WHERE livro.ano_de_publicacao = '<ano>';

```

Consulta 08: Selecionar o título e o nome da editora de todos os livros emprestados por um determinado usuário.

Listar o título e as respectivas editoras dos livros emprestados para um determinado usuário <ID>. Responsável: Guilherme.

AR

$$\begin{aligned}
 &t1 \leftarrow \text{emprestimo} \bowtie \text{emprestimo.isbn} = \text{livro.isbn} \text{ livro} \bowtie \text{livro.cnpj} = \text{editora.cnpj} \text{ editora} \\
 &\pi_{\text{livro.titulo, editora.nome}} (\sigma_{\text{emprestimo.ID_usuario}=01} (t1))
 \end{aligned}$$

SQL

```

SELECT livro.titulo, editora.nome
FROM emprestimo
INNER JOIN livro ON emprestimo.isbn = livro.isbn
INNER JOIN editora ON livro.cnpj = editora.cnpj
WHERE emprestimo.ID_usuario = <ID>;

```

Consulta 09: Liste o título do livro e o nome da editora que o publicou, ordenados por ano de publicação em ordem crescente.

Dado os livros cadastrados, listar o título e a editora de cada um deles, ordenados em ordem crescente por ano de publicação. Responsável: Guilherme.

AR

$$\pi_{\text{livro.titulo, editora.nome}} (\text{livro} \bowtie \text{livro.cnpj} = \text{editora.cnpj} \text{ editora})$$

SQL

```

SELECT livro.titulo, editora.nome
FROM livro
INNER JOIN editora ON livro.cnpj = editora.cnpj
ORDER BY livro.ano_de_publicacao ASC;

```

4.4. Triggers

Os scripts das consultas de dados no banco de dados se encontram anexados no arquivo **triggers.sql**.

4.4.1. Trigger 1: Atualizar número de exemplares

O campo `numero_de_exemplares` na entidade `livro` é calculado automaticamente, ou seja, sempre que um novo exemplar de um livro é inserido ou removido, o número total de exemplares de um livro precisa ser atualizado. O cenário proposto condiz com o minimundo apresentado, pois a qualquer momento a biblioteca pode adquirir um exemplar novo, ou até mesmo precisar retirar de circulação algum exemplar antigo que tenha estragado ou se perdido por exemplo.

O Trigger `atualizar_numero_exemplares` chama a função `atualizar_numero_exemplares()` antes de uma inserção ou uma remoção na tabela `exemplar`, o que representa um novo cadastro de algum exemplar, ou a remoção de um cadastro antigo. A função vai verificar se um novo exemplar foi inserido ou removido, no caso de inserção a função vai incrementar o `numero_de_exemplares` do `livro` com o mesmo `isbn` que o exemplar inserido. No caso de remoção, a função fará o processo inverso, ou seja, vai decrementar o `numero_de_exemplares` do `livro` com o mesmo `isbn` que o exemplar for removido.

4.4.2. Trigger 2: Verificar se o empréstimo é possível

O relacionamento empréstimo possui como chave primária os atributos `ID_usuario`, de usuário e os atributos `isbn` e `codigo_exemplar` da entidade `exemplar`. Ou seja, embora uma chave primária não possa ser repetida podem haver algumas falhas, isso porque o `isbn` e o código do exemplar não podem ser únicos, pois vários exemplares de um mesmo livro podem ser emprestados, ou seja, todos eles terão o mesmo `isbn`, e vários exemplares que possuem o mesmo código, porém `isbn` diferentes, podem ser emprestados também. Isso porque o código do exemplar é referente ao `numero_de_exemplares` de um livro, ou seja, pode acontecer um erro de um outro usuário realizar o empréstimo de um exemplar que já está emprestado. Por exemplo:

```
INSERT INTO emprestimo VALUES(01, '9786555980868', 01, 02, '2023-04-01', '2023-04-15');
```

Os três primeiros valores representam a chave primária do empréstimo em questão, sendo respectivamente o `ID` do `usuário`, o `isbn` e o `código do exemplar`.

se for inserido um novo empréstimo em que esses três primeiros campos sejam iguais, tal operação não será permitida, pois para esse caso a chave primária será a mesma. porém se o seguinte empréstimo for realizado:

```
INSERT INTO emprestimo VALUES(02, '9786555980868', 01, 02,
'2023-04-01', '2023-04-15');
```

O empréstimo será inserido na tabela isso porque como o `ID` do `usuario` mudou, a chave primária não é a mesma, entretanto o exemplar que está sendo emprestado já está com outro usuário, com isso a tupla é uma informação falsa, o que pode gerar que o sistema apresente muito mais empréstimos do que quantidade de exemplares disponíveis, uma vez que o exemplar pode ser emprestado várias vezes.

Para evitar esse problema o trigger `emprestimo_trigger` irá ser acionado sempre antes que ocorra um empréstimo, quando for acionado ele vai chamar a função `verificar_emprestimo()`. Essa função vai verificar se já consta na tabela “empréstimo” um exemplar com o mesmo `codigo` e um mesmo `isbn` do exemplar que está sendo emprestado, caso exista, significa que esse exemplar já está emprestado para outro usuário, nesse caso o sistema irá lançar uma exceção, caso não exista significa que o empréstimo é possível e o novo empréstimo será inserido na tabela. Com isso, além de verificar e garantir que o exemplar não seja emprestado duas vezes ao mesmo tempo, garantimos que não sejam emprestados mais exemplares do que os disponíveis, garantindo dessa forma forma confiança e validade dos dados apresentados na tabela.

4.5. Considerações finais

A relação ternária ‘Empréstimo’ apresenta cardinalidade 1:1:N, isso pode ser observado na Figura 1 da Seção 2, dado isso se enquadra no terceiro caso:

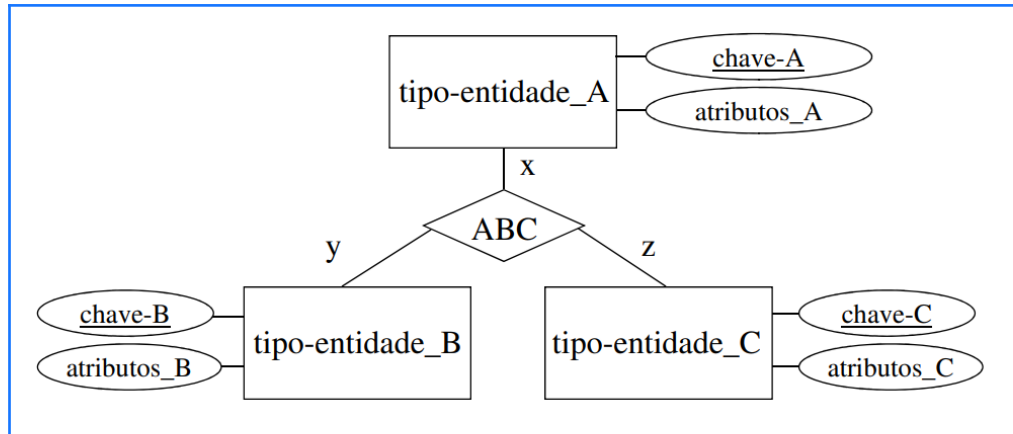


Figura 3. Tipo relacionamento Ternário

Fonte: http://wiki.icmc.usp.br/images/a/aa/Mat05_Mapeamento.pdf

Terceiro Caso: $x=1$, $y=1$, $z=m$

- ABC (chaves_A, chaves_B, chaves_C)
- ABC (chaves_A, chaves_B, chaves_C)

Para esse cenário, as chaves primárias devem ser composta pela chave da entidade da cardinalidade n , e uma das chaves da uma das outras entidades de cardinalidade 1. No banco de dados em questão apresentado na Figura 1, a entidade de cardinalidade N é a entidade Exemplar, cuja chave primária é o `isbn` e o `código`, pois o exemplar é uma entidade fraca, ou seja, apenas o `código` não é o suficiente para identificar unicamente um exemplar. Logo a chave primária desse relacionamento deve ser composto pela chave primária do exemplar (`isbn` e `código`) + a chave primária de uma das outras entidades, no caso foi selecionado a entidade usuário (`ID`). A chave primária da terceira entidade que não foi selecionada, será um atributo da tabela, sendo assim uma chave estrangeira.

Outra consideração final diz respeito a criação de dos tipo enum, foi feito dessa forma para que as informações ficassem mais organizadas, além do fato que alguns campos como `genero`, se repetem em mais de uma entidade.

BIBLIOGRAFIA

GONZALEZ, M. Sahudy. Modelos Conceituais de Dados: MER. **Ambiente de Banco de Dados do Ava2**, 2023. Disponível em: <https://ava2.ead.ufscar.br/course/view.php?id=20508§ion=2#tabs-tree-start>.

Acesso em: 10 de janeiro de 2023.

GONZALEZ, M. Sahudy. VideoAula do dia 16 de março de 2021, Google Drive. 16 de março de 2023. Disponível em : [BD \(2021-03-16 at 10:14 GMT-7\) - Google Drive](#)

GONZALEZ, M. Sahudy. Instruções de desenvolvimento do projeto e para escrita do relatório. **Ambiente de Banco de Dados do Ava2**, 2023. Disponível em: <https://ava2.ead.ufscar.br/course/view.php?id=20508§ion=2#tabs-tree-start>.

Acesso em: 12 de janeiro de 2023.

GONZALEZ, M. Sahudy. Instruções para formatação dos relatórios. **Ambiente de Banco de Dados do Ava2**, 2023. Disponível em: <https://ava2.ead.ufscar.br/course/view.php?id=20508§ion=2#tabs-tree-start>.

Acesso em: 12 de janeiro de 2023.

GONZALEZ, M. Sahudy. Mapeamento do Modelo Entidade-Relacionamento para o Modelo Relacional. **Ambiente de Banco de Dados do Ava2**, 2023. Disponível em: https://ava2.ead.ufscar.br/pluginfile.php/874850/mod_page/content/20/CC_BD_aula_DERparaDED.pdf?time=1675098069173. Acesso em: 10 de janeiro de 2023.

CIFERRI, D. A. Cristina. Mapeamento do Modelo Entidade-Relacionamento para o Modelo Relacional. Disponível em: http://wiki.icmc.usp.br/images/a/aa/Mat05_Mapeamento.pdf. Acesso em: 25 de fevereiro de 2023.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de Gerenciamento de Banco de Dados**. 3. ed. São Paulo: McGrawHill, 2008. ISBN 978-85-7726-027-0.