

## **TRABALHO REALIZADO POR:**

Catarina da Silva Guimarães | up202307420

Inês Rocha de Oliveira | up202305418

Sara Alves Garcia | up202306877

# **BASE DE DADOS**

A aplicação de uma base de dados no contexto de uma plataforma do tipo Goodreads

# ÍNDICE

INTRODUÇÃO.....	3
5.1.1: MODELO CONCEITUAL.....	4
5.1.2: ESQUEMA RELACIONAL INICIAL.....	5
5.1.3: INTEGRAÇÃO DE INTELIGÊNCIA ARTIFICIAL.....	6
5.1.4: ESQUEMA RELACIONAL FINAL.....	7
5.1.5: ANÁLISE DE DEPENDÊNCIAS FUNCIONAIS E FORMAS NORMAIS.....	8
5.1.6: INTEGRAÇÃO DE INTELIGÊNCIA ARTIFICIAL.....	13
5.1.7: ANÁLISE FINAL DE DEPENDÊNCIAS FUNCIONAIS E FORMAS NORMAIS.....	14
5.1.9: INTEGRAÇÃO DE INTELIGÊNCIA ARTIFICIAL NO CREATE.....	19
5.1.12: INTEGRAÇÃO DE INTELIGÊNCIA ARTIFICIAL NO POPULATE.....	21
CONCLUSÃO.....	21

# INTRODUÇÃO

Num mundo onde a procura pelo conhecimento se torna cada vez mais presente, surge a necessidade da criação de uma plataforma capaz de gerir e potencializar o acesso a grande maioria dos livros (uma das maiores fontes de informação a que qualquer um pode ter acesso) que foram sendo publicados ao longo de toda a nossa história.

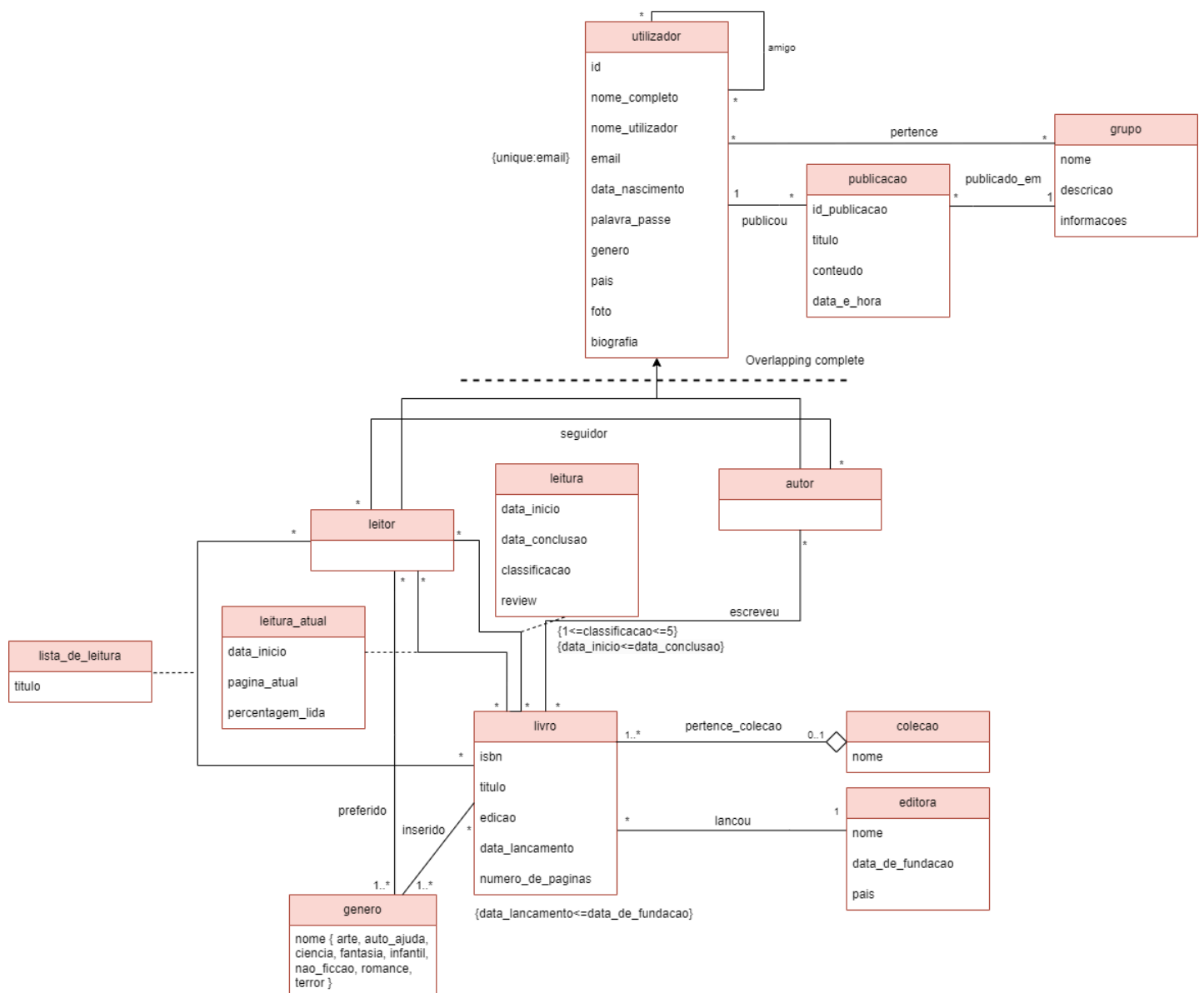
Tal plataforma permitiria aos seus usuários uma maior organização das suas leituras (permitindo que estas fossem armazenadas nas mais variadas listas de leitura) e também um universo bastante vasto de títulos ainda por descobrir.

Uma plataforma onde leitores de todos os cantos do mundo se podem reunir para discutir e partilhar as suas opiniões sobre as mais variadas obras, ideal para conhecer pessoas novas, cujos interesses se possam alinhar. Onde autores são também capazes de ter acesso a uma ligação mais direta com o seu público, estando apenas à distância de um clique de todas as avaliações deixadas pelos usuários da plataforma na página do seu livro.

Assim sendo, decidimos desenvolver uma base de dados que poderia, eventualmente, ser aplicada numa plataforma como esta que acabámos de descrever. Como referência para o nosso trabalho, usamos uma aplicação já existente, de nome *Goodreads*. O uso desta mostrou-se essencial para uma melhor compreensão do projeto e das possíveis aplicações que poderíamos implementar ao longo do desenvolvimento deste projeto.

Neste relatório, pretendemos abordar de uma forma aprofundada elementos técnicos desta mesma base de dados, avaliando qual será a melhor abordagem a ser seguida.

### 5.1.1: MODELO CONCEITUAL



**ALGUMAS DAS ALTERAÇÕES MAIS MARCANTES FEITAS AO UML ANTERIORMENTE ENTREGUE:**

- a alteração do nome de todas as classes e de seus atributos, para o formato `snake_case`;
- atributos removidos das classes leitor e autor, algumas por redundância, outras por simplesmente não se mostrarem necessárias;
- houve a remoção da classe `notificacoes`, dado que esta não tinha uma funcionalidade específica dentro da aplicação deste uml;
- procedemos à aplicação de algumas restrições anteriormente não referidas.

## 5.1.2: ESQUEMA RELACIONAL INICIAL

amigo (id->leitor, id->leitor)  
autor (id->utilizador)  
colecao (nome)  
editora (nome, pais, data\_de\_fundação)  
escreveu (id ->autor, isbn->livro)  
genero (nome)  
grupo (nome, descricao, informacoes)  
inserido (isbn -> livro, nome->genero)  
lancou (isbn->livro, nome->editora)  
leitor (id->utilizador)  
leitura\_atual (isbn->livro, id->leitor, data\_inicio, pagina\_atual, percentagem\_lida)  
leitura (isbn->livro, id->leitor, data\_inicio, data\_conclusao, classificacao, review)  
leitura\_atual (isbn->livro, id->leitor, data\_inicio, pagina\_atual, percentagem\_lida)  
lista\_de\_leitura (isbn->livro, id->leitor, titulo)  
livro (isbn, titulo, edicao, data\_lançamento, numero\_de\_paginas)  
pertence\_colecao (isbn->livro, nome->colecao)  
pertence\_grupo (id->utilizador, nome->grupo)  
preferido (id->leitor, nome->genero)  
publicacao (id->utilizador, nome->grupo, titulo, conteudo, data\_e\_hora)  
publicado\_em (id\_publicacao -> publicacao, nome->grupo)  
publicou (id\_publicacao -> publicacao, id->utilizador)  
seguidor (id->leitor, id->autor)  
utilizador (id, nome\_completo, nome\_utilizador, email, data\_nascimento, password, genero, pais, foto, biografia)

### NOTA:

No desenvolvimento do nosso modelo relacional, optámos por adotar o estilo E/R em vez do estilo object-oriented. A principal razão para esta escolha foi a clareza e simplicidade que o modelo E/R oferece, especialmente num contexto relacional. O estilo E/R permite mapear diretamente as entidades e os relacionamentos entre elas, usando tabelas que representam as classes e chaves estrangeiras para conectar essas classes de maneira clara e eficiente.

## 5.1.3: INTEGRAÇÃO DE INTELIGÊNCIA ARTIFICIAL

Recorrendo ao uso da ferramenta de inteligência artificial ChatGPT, procedemos ao envio de uma foto do nosso UML de modo a que o sistema pudesse avaliar possíveis alterações a serem feitas. Também pedimos para ele transformar o UML fornecido num esquema relacional. Contudo, neste segundo pedido, a ferramenta falhou miseravelmente: o ChatGPT não foi capaz de fornecer uma resposta correta ao nosso pedido. Não sublinhou as chaves primárias nem concedeu os atributos corretos a certas classes.

A resposta obtida relativamente ao primeiro pedido apresentou algumas alterações a serem aplicadas. Começou por referir que atributos estavam bem colocados e quais deviam ser alterados. Classes como utilizador, leitor, autor, livro, editora, genero, leitura, leitura\_atual, seguidor, grupo e publicação foram corretamente aplicadas e não estavam carentes de alterações. Por outro lado, classes como colecao, lista\_de\_leitura e amigo (segundo a ferramenta), precisavam de ser modificadas. Optamos por aplicar a alteração sugerida para a classe amigo, introduzindo nomes de colunas diferentes para evitar uma eventual confusão entre os dois ids.

```
12. Amigo (Friendship)
• Logical, but you might want to use different column names for the two ID->Leitor foreign keys to avoid confusion. For example:

rust Copiar código

Amigo (ID1->Leitor, ID2->Leitor)
```

As alterações sugeridas para as outras duas classes eram descabidas: na classe lista\_de\_leitura, a ferramenta sugeriu tirar um atributo relevante para a classe (o atributo titulo) e na classe colecao sugeriu alterar o nome de um dos atributos para nome\_da\_editora, o que acreditamos se tratar de uma alteração desnecessária para a eficiência da base de dados.

```
10. Lista_de_leitura (Reading List)
• Accurate, but consider if the titulo column is necessary here, as the ISBN already identifies the book uniquely. It might be better to simplify it to:

rust Copiar código

Lista_de_leitura (ISBN->Livro, ID->Leitor)
```

```
7. Coleção (Collection)
• Coleção (nome) is defined correctly but lacks a foreign key for the publisher (Editora), which is essential in the UML. Update it to:

rust Copiar código

Coleção (nome, nome_editora -> Editora)
```

## 5.1.4: ESQUEMA RELACIONAL FINAL

amigo (id1->utilizador, id2->utilizador)

autor (id->utilizador)

colecacao (nome)

editora (nome, pais, data\_de\_fundacao)

escreveu (id ->autor, isbn->livro)

genero (nome)

grupo (nome, descricao, informacoes)

inserido (isbn -> livro, nome->genero)

lancou (isbn->livro, nome->editora)

leitor (id->utilizador)

leitura\_atual (isbn->livro, id->leitor, data\_inicio, pagina\_atual, percentagem\_lida)

leitura (isbn->livro, id->leitor, data\_inicio, data\_conclusao, classificacao, review)

lista\_de\_leitura (isbn->livro, id->leitor, titulo)

livro (isbn, titulo, edicao, data\_lancamento, numero\_de\_paginas)

pertence\_colecacao (isbn->livro, nome->colecacao)

pertence\_grupo (id->utilizador, nome->grupo)

preferido (id->leitor, nome->genero)

publicacao (id\_publicacao, titulo, data\_e\_hora, conteudo)

publicado\_em (id\_publicacao -> publicacao, nome->grupo)

publicou (id\_publicacao -> publicacao, id->utilizador)

seguidor (id1->leitor, id2->autor)

utilizador (id, nome\_completo, nome\_utilizador, email, data\_nascimento,

palavra\_passe, genero, pais, foto, biografia)

## 5.1.5: ANÁLISE DE DEPENDÊNCIAS FUNCIONAIS E FORMAS NORMAIS

amigo:

(id1, id2) -> não tem dependências funcionais não triviais

autor:

id -> não tem dependências funcionais não triviais

colecção:

nome -> não tem dependências funcionais não triviais

editora:

nome -> país, data\_de\_fundação

escreveu:

(id, isbn) -> não tem dependências funcionais não triviais

género:

nome -> não tem dependências funcionais não triviais

grupo:

nome -> descrição, informações

inserido:

(isbn, nome) -> não tem dependências funcionais não triviais

lançou:

isbn -> editora

leitor:

id -> não tem dependências funcionais não triviais

leitura:

(isbn, id) -> data\_início, data\_conclusão, classificação, review

leitura\_atual:

(isbn, id) -> data\_início, data\_conclusão, classificação, review

lista\_de\_leitura:

(isbn, id, título) -> não tem dependências funcionais não triviais



livro:

isbn -> titulo, edicao, data\_lancamento, numero\_de\_paginas

pertence\_colecao:

isbn -> nome

pertence\_grupo:

(id, nome) -> não tem dependências funcionais não triviais

preferido:

(id, nome) -> não tem dependências funcionais não triviais

publicacao:

id\_publicacao -> titulo, data\_e\_hora, conteudo

publicado\_em:

id\_publicacao -> nome

publicou:

id\_publicacao -> id

seguidor:

(id1, id2) -> não tem dependências funcionais não triviais

utilizador:

id -> nome\_completo, nome\_utilizador, email, data\_nascimento, password, genero, pais, foto, biografia

## RESTRIÇÕES

Com o objetivo de garantir a integridade dos dados armazenados, assegurando que estes são consistentes e estão corretos e em conformidade com as regras definidas para o sistema, utilizamos diferentes tipos de restrições, como por exemplo CHECK, UNIQUE, NOT NULL, DEFAULT e chaves primárias/estrangeiras.

### NOT NULL:

Esta restrição foi utilizada em casos onde certo atributo deverá obrigatoriamente ter um valor atribuído.

- Na criação de um **utilizador**, os atributos **id**, **nome\_completo**, **nome\_utilizador**, **email**, **data\_nascimento** e **palavra\_passe** não devem ser nulos, já que estas informações são essenciais para a identificação de cada utilizador.
- Já na classe **livro**, atributos como **isbn**, **titulo**, **edicao**, **data\_lancamento** e **numero\_de\_paginas** também não podem ser nulos, por serem características essenciais de um livro.
- Da mesma forma, em classes como **grupo**, **editora**, **genero** e **colecao**, o atributo **nome** será não nulo.
- Em **publicacao**, o **id\_publicacao**, **titulo**, **conteudo** e **data\_e\_hora** também não podem ser nulos. Nas relações de **publicação**, **publicou** e **publicado\_em**, o **id\_publicacao** nunca deverá ser nulo, tal como os restantes atributos destas relações, **id** do utilizador no caso de **publicou** e **nome** do grupo no caso de **publicado\_em**.
- Nas relações **amigo** e **seguidor**, nenhum dos **id**'s em questão poderá ser nulo, já que para que uma destas relações exista é necessária a ligação de dois utilizadores.
- Para as relações **pertence\_grupo** e **pertence\_colecao**, os atributos **id** e **isbn**, respetivamente, tal como o atributo **nome** que para ambos não deverá ser nulo.
- Em classes como **escreveu** e **lancou**, o **isbn** do livro em questão deverá ser não nulo, tal como o **id** do autor e o **nome** da editora respetivamente.
- As classes de relação **leitura\_atual** e **leitura** têm vários atributos, não podendo o **isbn** e **id** do leitor ser nulos em nenhuma delas. O mesmo acontece no caso da classe de relação **lista\_de\_leitura**, que adicionalmente deverá contar com um **nome** não nulo.
- Por fim, nas relações de género **preferido** e livro **inserido** num determinado género, o nome do género não deverá ser nulo, tal como o **id** do utilizador em **preferido** e **isbn** do livro em **inserido**.

#### CHECK:

Esta restrição garante que os valores atribuídos a um determinado atributo satisfaçam uma condição lógica definida, assegurando que os dados inseridos estejam em conformidade com as regras de integridade e os requisitos específicos do sistema.

- Também na classe **utilizador**, no que diz respeito ao atributo **genero** este poderá escolher apenas entre 'FEMININO', 'MASCULINO', 'OUTRO'.
- Na classe **genero**, o atributo **nome** terá de pertencer a uma das opções: 'ARTE', 'AUTO AJUDA', 'CIENCIA', 'FANTASIA', 'INFANTIL', 'NAO FICCAO', 'ROMANCE' e 'TERROR'.

- Já nas relações **amigo** e **seguidor**, o atributo **id1** e **id2** terão de ser diferentes, uma vez que um utilizador não pode ser nem amigo de si próprio nem pode seguir-se a si próprio.
- Em **leitura\_atual**, a **percentagem** terá de estar compreendida entre os 0% e os 100%.
- Na classe **leitura**, A **classificação** que um utilizador poderá atribuir a um livro será uma das seguintes: '☆☆☆☆☆', '★★★★☆', '★★★★☆', '★★★★☆', '★★★★☆' ou '★★★★★'.

### PRIMARY KEY:

Esta restrição assegura que cada registo numa tabela seja único e identificável de forma inequívoca, garantindo que o valor atribuído ao atributo (ou conjunto de atributos) que compõe a chave primária não se repita em nenhuma outra linha da tabela. Esta restrição é essencial para manter a integridade dos dados, pois evita a duplicação de registos e assegura que cada entrada possa ser identificada de maneira exclusiva, facilitando operações de busca, atualização e relacionamento com outras tabelas no banco de dados.

- Nas classes **utilizador**, **leitor** e **autor** a primary key é o **id**.
- Já na classe **livro**, a primary key é o **isbn**.
- **nome** é a primary key das classes **editora**, **coleccion**, **genero** e **grupo**.
- Em **publicacao** a primary key é o **id\_publicacao**.
- A relação **amigo** tem como primary key a combinação dos **id**'s de 2 utilizadores diferentes.
- A relação **seguidor** tem como primary key a combinação dos **id**'s um de um leitor e outro de um autor.
- A primary key da relação **pertence\_grupo** é a combinação do **id** do utilizador e o **nome** do grupo a que o utilizador pertence
- A primary key da relação **pertence\_colecao** é a combinação do **isbn** do livro e o **nome** da coleccion do qual o livro faz parte
- A relação **escreveu** tem como primary key a combinação do **isbn** do livro escrito e o **id** do autor do mesmo
- Em **lançou** a primary key é a união do **isbn** do livro e do **nome** da editora que o publicou
- A relação **inserido** tem como primary key a combinação do **isbn** do livro e do **nome** do género em que este se insere
- A primary key da relação **preferido** trata-se de uma combinação do **id** correspondente ao leitor com o **nome** do género assinalado como favorito
- Relações como **publicou** e **publicado\_em** têm ambas como primary key o **id\_publicacao**
- As relações **leitura\_atual**, **leitura** e **lista\_de\_leitura** têm como primary key a combinação do **isbn** do livro lido e do **id** do leitor

## DEFAULT:

Esta restrição define um valor padrão para um atributo, garantindo que, caso nenhum valor seja fornecido durante a inserção de dados, o valor predefinido seja automaticamente atribuído. Isto ajuda a manter a consistência dos dados e evita valores nulos quando não forem informados explicitamente.

- Na classe **utilizador**, as restrições default encontram-se aplicadas nos atributos como o **genero** do leitor e o **pais** onde este se encontra, sendo o default '-' para ambos
- Na relação **leitura**, o único atributo que apresenta uma restrição do tipo default é a **classificacao** que um leitor dá a um livro aquando do término da sua leitura (o default é '☆☆☆☆☆')

## FOREIGN KEY:

A restrição foreign key estabelece uma relação entre duas tabelas, garantindo que o valor de um atributo numa tabela corresponda a um valor válido na chave primária de outra tabela.

- Tanto **leitor** quanto **autor** têm como foreign key os **id**'s que referenciam o **id** do **utilizador**
- Na relação **amigo**, existem 2 foreign keys, uma delas no **id1** que referencia o **id** de utilizador e o outro em **id2** que também referencia o **id** de um outro **utilizador**
- Em **pertence\_grupo** as foreign keys são **id** e **nome** que referenciam respetivamente o **id** do **utilizador** e o **nome** do **grupo** a que o utilizador pertence
- Na relação **pertence\_colecao** as foreign keys são o **isbn** que referencia o isbn do **livro** e o **nome** que referencia o nome da **coleção**
- Na relação **escreveu** as foreign keys são o **isbn** que referencia o isbn do **livro** e o **id** que referencia o id da **autor**
- A relação **lancou** possui duas foreign keys: **isbn** que referencia o isbn do livro e **nome** que referencia o nome da editora responsável pela publicação do respetivo livro
- Na relação **inserido**, as duas foreign keys que podemos encontrar são: o **isbn** que referencia o isbn específico de um dado livro e o **nome** que referencia o nome do género em que o livro de insere
- A relação **preferido** conta com as duas foreign keys **id**, que referencia o id do leitor, e **nome**, que referencia o nome do género preferido pelo leitor
- As duas foreign keys que podemos encontrar na relação **publicado\_em** tratam-se de: **id\_publicacao** que referencia o id da publicação e **nome** que referencia o nome do grupo onde a publicação foi feita
- A relação **publicou**, por sua vez, apresenta duas foreign keys: o **id\_publicacao** que referencia o id de uma publicação e o **id** que referencia o id de um utilizador

- Numa relação como a **seguidor** podemos encontrar duas foreign keys como: **id1** que referencia o id do leitor e o **id2** que referencia o id do autor
- Nas relações **leitura\_atual**, **leitura** e **lista\_de\_leitura**, podemos averiguar a existência de duas foreign keys: o **isbn** que referencia o isbn de um determinado livro, e o **id** que referencia o id de um leitor

**NOTA:** para todas as foreign keys abordadas nos tópicos anteriores foram aplicadas as devidas restrições *on update* e *on delete*.

## 5.1.6: INTEGRAÇÃO DE INTELIGÊNCIA ARTIFICIAL

Uma vez com a análise de dependências funcionais e formas normais já feita, dirigimo-nos à ferramenta de inteligência artificial de nome ChatGPT de maneira a encontrar uma forma de aprimorar o nosso esquema já feito. Por mais que o recurso à tecnologia não tenha sido muito frutífero, chegámos a uma conclusão à qual não tínhamos chegado anteriormente: o atributo **email** necessitava de sofrer uma restrição, tornando-se *unique* e assim, deixando de ser capaz de estar presente em mais do que um perfil dentro da plataforma.

## 5.1.7: ANÁLISE FINAL DE DEPENDÊNCIAS FUNCIONAIS E FORMAS NORMAIS

### ALTERAÇÕES REALÇADAS A ROSA

amigo:

(id1, id2) -> não tem dependências funcionais não triviais

autor:

id-> não tem dependências funcionais não triviais

colecão:

nome -> não tem dependências funcionais não triviais

editora:

nome -> país, data\_de\_fundação

escreveu:

(id, isbn) -> não tem dependências funcionais não triviais

gênero:

nome -> não tem dependências funcionais não triviais

grupo:

nome-> descrição, informações

inserido:

(isbn, nome) -> não tem dependências funcionais não triviais

lançou:

isbn -> editora

leitor:

id -> não tem dependências funcionais não triviais

leitura:

(isbn, id) -> data\_início, data\_conclusão, classificação, review

leitura\_atual:

(isbn, id) -> data\_inicio, data\_conclusao, classificacao, review

lista\_de\_leitura:

(isbn, id, titulo) -> não tem dependências funcionais não triviais

livro:

isbn -> titulo, edicao, data\_lancamento, numero\_de\_paginas

pertence\_colecao:

isbn -> nome

pertence\_grupo:

(id, nome) -> não tem dependências funcionais não triviais

preferido:

(id, nome) -> não tem dependências funcionais não triviais

publicacao:

id\_publicacao -> titulo, data\_e\_hora, conteudo

publicado\_em:

id\_publicacao -> nome

publicou:

id\_publicacao -> id

seguidor:

(id1, id2) -> não tem dependências funcionais não triviais

utilizador:

id -> nome\_completo, nome\_utilizador, email, data\_nascimento, password, genero, pais, foto, biografia

email -> id, nome\_completo, nome\_utilizador, data\_nascimento, password, genero, pais, foto, biografia

## RESTRIÇÕES

Com o objetivo de garantir a integridade dos dados armazenados, assegurando que estes são consistentes e estão corretos e em conformidade com as regras definidas para o sistema, utilizamos diferentes tipos de restrições, como por exemplo CHECK, UNIQUE, NOT NULL, DEFAULT e chaves primárias/estrangeiras.

### NOT NULL:

Esta restrição foi utilizada em casos onde certo atributo deverá obrigatoriamente ter um valor atribuído.

- Na criação de um **utilizador**, os atributos **id**, **nome\_completo**, **nome\_utilizador**, **email**, **data\_nascimento** e **palavra\_passe** não devem ser nulos, já que estas informações são essenciais para a identificação de cada utilizador.
- Já na classe **livro**, atributos como **isbn**, **titulo**, **edicao**, **data\_lancamento** e **numero\_de\_paginas** também não podem ser nulos, por serem características essenciais de um livro.
- Da mesma forma, em classes como **grupo**, **editora**, **genero** e **colecacao**, o atributo **nome** será não nulo.
- Em **publicacao**, o **id\_publicacao**, **titulo**, **conteudo** e **data\_e\_hora** também não podem ser nulos. Nas relações de **publicação**, **publicou** e **publicado\_em**, o **id\_publicacao** nunca deverá ser nulo, tal como os restantes atributos destas relações, **id** do utilizador no caso de **publicou** e **nome** do grupo no caso de **publicado\_em**.
- Nas relações **amigo** e **seguidor**, nenhum dos **id**'s em questão poderá ser nulo, já que para que uma destas relações exista é necessária a ligação de dois utilizadores.
- Para as relações **pertence\_grupo** e **pertence\_colecacao**, os atributos **id** e **isbn**, respetivamente, tal como o atributo **nome** que para ambos não deverá ser nulo.
- Em classes como **escreveu** e **lancou**, o **isbn** do livro em questão deverá ser não nulo, tal como o **id** do autor e o **nome** da editora respetivamente.
- As classes de relação **leitura\_atual** e **leitura** têm vários atributos, não podendo o **isbn** e **id** do leitor ser nulos em nenhuma delas. O mesmo acontece no caso da classe de relação **lista\_de\_leitura**, que adicionalmente deverá contar com um **nome** não nulo.
- Por fim, nas relações de género **preferido** e livro **inserido** num determinado género, o nome do género não deverá ser nulo, tal como o **id** do utilizador em **preferido** e **isbn** do livro em **inserido**.



### UNIQUE:

Esta restrição garante que o valor atribuído a um determinado atributo seja único em toda a tabela, evitando duplicados e assegurando que cada registro possa ser identificado de forma distinta por aquele campo.

- Na classe **utilizador**, o **email** só deverá pertencer a uma conta.

### CHECK:

Esta restrição garante que os valores atribuídos a um determinado atributo satisfaçam uma condição lógica definida, assegurando que os dados inseridos estejam em conformidade com as regras de integridade e os requisitos específicos do sistema.

- Também na classe **utilizador**, no que diz respeito ao atributo **genero** este poderá escolher apenas entre 'FEMININO', 'MASCULINO', 'OUTRO'.
- Na classe **genero**, o atributo **nome** terá de pertencer a uma das opções: 'ARTE', 'AUTO AJUDA', 'CIENCIA', 'FANTASIA', 'INFANTIL', 'NAO FICCAO', 'ROMANCE' e 'TERROR'.
- Já nas relações **amigo** e **seguidor**, o atributo **id1** e **id2** terão de ser diferentes, uma vez que um utilizador não pode ser nem amigo de si próprio nem pode seguir-se a si próprio.
- Em **leitura\_atual**, a **percentagem** terá de estar compreendida entre os 0% e os 100%.
- Na classe **leitura**, A **classificação** que um utilizador poderá atribuir a um livro será uma das seguintes: '★☆☆☆☆', '★★☆☆☆', '★★★☆☆', '★★★★☆' ou '★★★★★'.

### PRIMARY KEY:

- Nas classes **utilizador**, **leitor** e **autor** a primary key é o **id**.
- Já na classe **livro**, a primary key é o **isbn**.
- **nome** é a primary key das classes **editora**, **coleccion**, **genero** e **grupo**.
- Em **publicacao** a primary key é o **id\_publicacao**.
- A relação **amigo** tem como primary key a combinação dos **id**'s de 2 utilizadores diferentes.
- A relação **seguidor** tem como primary key a combinação dos **id**'s um de um leitor e outro de um autor.
- A primary key da relação **pertence\_grupo** é a combinação do **id** do utilizador e o **nome** do grupo a que o utilizador pertence
- A primary key da relação **pertence\_colecao** é a combinação do **isbn** do livro e o **nome** da coleccion do qual o livro faz parte
- A relação **escreveu** tem como primary key a combinação do **isbn** do livro escrito e o **id** do autor do mesmo
- Em **lancou** a primary key é a união do **isbn** do livro e do **nome** da editora que o publicou

- A relação **inserido** tem como primary key a combinação do **isbn** do livro e do **nome** do género em que este se insere
- A primary key da relação **preferido** trata-se de uma combinação do **id** correspondente ao leitor com o **nome** do género assinalado como favorito
- Relações como **publicou** e **publicado\_em** têm ambas como primary key o **id\_publicacao**
- As relações **leitura\_atual**, **leitura** e **lista\_de\_leitura** têm como primary key a combinação do **isbn** do livro lido e do **id** do leitor

#### DEFAULT:

- Na classe **utilizador**, as restrições default encontram-se aplicadas nos atributos como o **genero** do leitor e o **pais** onde este se encontra, sendo o default '-' para ambos
- Na relação **leitura**, o único atributo que apresenta uma restrição do tipo default é a **classificacao** que um leitor dá a um livro aquando do término da sua leitura (o default é '☆☆☆☆☆')

#### FOREIGN KEY:

- Tanto **leitor** quanto **autor** têm como foreign key os **id's** que referenciam o **id** do **utilizador**
- Na relação **amigo**, existem 2 foreign keys, uma delas no **id1** que referencia o **id** de utilizador e o outro em **id2** que também referencia o **id** de um outro **utilizador**
- Em **pertence\_grupo** as foreign keys são **id** e **nome** que referenciam respetivamente o **id** do **utilizador** e o **nome** do **grupo** a que o utilizador pertence
- Na relação **pertence\_colecao** as foreign keys são o **isbn** que referencia o isbn do **livro** e o **nome** que referencia o nome da **coleção**
- Na relação **escreveu** as foreign keys são o **isbn** que referencia o isbn do **livro** e o **id** que referencia o id da **autor**
- A relação **lancou** possui duas foreign keys: **isbn** que referencia o isbn do livro e **nome** que referencia o nome da editora responsável pela publicação do respetivo livro
- Na relação **inserido**, as duas foreign keys que podemos encontrar são: o **isbn** que referencia o isbn específico de um dado livro e o **nome** que referencia o nome do género em que o livro se insere
- A relação **preferido** conta com as duas foreign keys **id**, que referencia o id do leitor, e **nome**, que referencia o nome do género preferido pelo leitor
- As duas foreign keys que podemos encontrar na relação **publicado\_em** tratam-se de: **id\_publicacao** que referencia o id da publicação e **nome** que referencia o nome do grupo onde a publicação foi feita

- A relação **publicou**, por sua vez, apresenta duas foreign keys: o **id\_publicacao** que referencia o id de uma publicação e o **id** que referencia o id de um utilizador
- Numa relação como a **seguidor** podemos encontrar duas foreign keys como: **id1** que referencia o id do leitor e o **id2** que referencia o id do autor
- Nas relações **leitura\_atual**, **leitura** e **lista\_de\_leitura**, podemos averiguar a existência de duas foreign keys: o **isbn** que referencia o isbn de um determinado livro, e o **id** que referencia o id de um leitor

**NOTA:** para todas as foreign keys abordadas nos tópicos anteriores foram aplicadas as devidas restrições *on update* e *on delete*.

## 5.1.9: INTEGRAÇÃO DE INTELIGÊNCIA ARTIFICIAL NO CREATE

Para o `create1.sql` utilizamos maioritariamente os conceitos que adquirimos em aula, contudo, quando procuramos melhorar o nosso código para o `create2.sql`, as ferramentas de inteligência artificial revelaram-se bastante úteis.

Por exemplo, o chatGPT auxiliou-nos na aplicação das restrições **ON DELETE CASCADE** e **ON UPDATE CASCADE**, explicando em que situações estes se poderiam adequar ao nosso trabalho.



In your project, **ON DELETE CASCADE** and **ON UPDATE CASCADE** should be used carefully to maintain data consistency and automate the handling of dependencies between tables. Here's when they are most appropriate:

### **ON DELETE CASCADE:**

Use this when the deletion of a parent record should automatically result in the deletion of all associated child records. This is particularly useful in cases where child records are meaningless or irrelevant without their parent.

Examples in your project:

1. **UTILIZADOR and LEITOR/AUTOR:** If a user (UTILIZADOR) is deleted, their corresponding entries in the LEITOR and AUTOR tables should also be deleted.
2. **UTILIZADOR and AMIGO:** If a user is deleted, any friendship records in the AMIGO table involving that user should also be removed.
3. **LIVRO and DEPENDENT TABLES:** If a book (LIVRO) is deleted, its references in related tables like LEITURA, LEITURA\_ATUAL, INSERIDO, or PERTENCE\_COLECAO should also be deleted.

## ON UPDATE CASCADE:

Use this when the modification of a primary key in a parent table should automatically update the foreign keys in child tables to maintain referential integrity. This is important in scenarios where primary keys may change (though this should be rare).

Examples in your project:

1. **UTILIZADOR:** If the ID of a user is updated (e.g., for technical reasons like restructuring IDs), all dependent tables such as LEITOR, AUTOR, AMIGO, or PERTENCE\_GRUPO should update their references accordingly.
2. **LIVRO:** If the ISBN of a book is updated, references in tables like LEITURA, INSERIDO, and PERTENCE\_COLECAO should be automatically updated to reflect the change.

## Key Considerations:

- **ON DELETE CASCADE** is particularly useful when dealing with dependent data that should not exist without its parent, ensuring clean data removal.
- **ON UPDATE CASCADE** should be used with caution since primary keys are rarely updated. If you expect frequent updates to primary keys, reconsider their design.
- Avoid these cascades in cases where child records must persist for auditing or historical purposes, as cascading could result in unintentional data loss.

Além disso, o chatGPT sugeriu uma alteração interessante na tabela leitura, mais especificamente no atributo classificacao: em vez de ter '-' como o nosso default value, a ferramenta de inteligência artificial sugeriu utilizar já um formato de estrelas para este.

Acabamos por implementar esta sugestão, apenas mudando a classificação nula para 5 estrelas vazias (sem avaliação) em vez de uma estrela preenchida e 4 vazias (o que faria com que qualquer leitura sem avaliação tivesse automaticamente uma classificação de 1 estrela)



Your **LEITURA** table definition is mostly correct, but there are a couple of minor issues that should be addressed:

1. **Missing comma after **CLASSIFICACAO** column:** You need to add a comma after the **CHECK** constraint on **CLASSIFICACAO**, just before the **REVIEW** column.
2. **Reviewing default values:** For the **CLASSIFICACAO** column, you have a default value of **'-'**, which is a placeholder. However, since this column will likely hold star ratings, you may want to consider using a default value such as **'★★★★★'** instead of **'-'** to avoid invalid values.
3. **Primary Key:** The primary key constraint is correctly defined as the combination of **ISBN** and **ID**, ensuring each reader's record for a book is unique.

## 5.1.12: INTEGRAÇÃO DE INTELIGÊNCIA ARTIFICIAL NO POPULATE

Mais uma vez, para a criação do ficheiro `populate1.sql`, utilizamos em grande parte os conhecimentos que adquirimos durante as aulas teóricas e teórico-práticas de Bases de Dados.

Este ficheiro inicial tinha, contudo, um volume muito pequeno relativamente ao pretendido, com aproximadamente 3 linhas por tabela, problema que as ferramentas de inteligência artificial foram muito úteis para resolver. Apesar de por vezes se confundir um pouco com chaves estrangeiras, trocando conteúdos como nomes de autores e ISBNs, o chatGPT foi muito útil para criar um `populate` mais extenso, o que se traduziu na criação do ficheiro final com aproximadamente 10 linhas por tabela, o `populate2.sql`.

## CONCLUSÃO

A realização deste trabalho permitiu-nos aprofundar os nossos conhecimentos em vários temas relacionados com a cadeira de Bases de Dados, como por exemplo UML Data Modelling, Relational Models e a sua conversão a partir de UMLs, e também nos permitiu ficar mais à vontade com programação em SQLite.

Ao longo do nosso projeto utilizamos diversas ferramentas de inteligência artificial, como por exemplo ChatGPT e Preplexity, que apesar de por vezes se revelarem pouco úteis ou até prejudiciais para o nosso trabalho, também nos deram ideias e sugeriram alterações que se revelaram muito contributivas para o mesmo. Este processo também nos permitiu melhorar as nossas técnicas no uso da inteligência artificial, tal como uma comunicação mais clara e eficiente.

Este projeto proporcionou-nos a oportunidade de aplicar teoria na prática, melhorando a nossa capacidade de resolver problemas reais de modelagem e gestão de dados.

Em suma, este projeto proporcionou-nos a oportunidade de aplicar teoria na prática, melhorando a nossa capacidade de resolver problemas reais de modelagem e gestão de dados. Ao longo do mesmo, fomos tornando cada vez mais capazes de compreender como uma base de dados de uma plataforma do tipo Goodreads funcionaria no mundo real, desde o seu conceito mais simples até o seu conceito mais elaborado, explorando todas as opções de funcionamento da sua interface.