

BASES DE DADOS

Book Lending

Entrega Final

Turma 3 – Grupo 3:

João Martins - up201704748@fe.up.pt

Leonor Sousa - up201705377@fe.up.pt

Maria Ferreira - up201704684@fe.up.pt

Prof.^a Carla Alexandra Teixeira Lopes

Índice

Descrição do Tema	3
Diagrama UML.....	4
Diagrama UML Corrigido.....	4
Esquema Relacional	5
Restrições	8
Interrogações	11
Gatilhos	11

Descrição do Tema

Book lending é uma iniciativa que pretende incentivar as pessoas a ler e a partilhar, poupando o ambiente e dinheiro. Baseia-se em duas plataformas diferentes: um site online e postos físicos.

Para utilizar o sistema, o utilizador deverá registar-se no site, fornecendo o seu nome, data de nascimento, país de origem, email e uma password. No site, poderá consultar todos os livros que fazem parte da rede e o seu estado atual: disponível ou não e, se disponível, a sua localização atual. Para além disso, o utilizador poderá ver todo o seu histórico de empréstimos, assim como, escrever avaliações aos livros que requisitou e ao serviço de empréstimo. Todas as avaliações estão também públicas para toda a comunidade registada no site.

Nos postos físicos, denominados de pontos de troca, é possível efetuar requisições e entregas de livros, assim como fazer dádivas à rede. A requisição e a entrega podem ou não ser efetuados no mesmo ponto de troca. Para isso, o utilizador não tem que fazer qualquer tipo de reserva, embora seja necessário estar registado no site, de forma a ser possível manter todo o histórico da rede.

Deste modo, relativamente a cada livro, é guardado o seu nome, a sua data de publicação e informações sobre o seu autor (nome, data de nascimento, pseudónimo e nacionalidade) e o sobre o seu editor (nome e nacionalidade). Relativamente a cada empréstimo, são guardadas as suas datas de início e de fim, assim como o feedback fornecido pelo utilizador que lhe está associado. Cada empréstimo está associado a um livro, a um utilizador e aos pontos de troca onde foram efetuados a requisição e a devolução. As dádivas estão associadas a um utilizador, a um livro e ao ponto de troca onde foram efetuadas, guardando-se também a data respetiva. Um utilizador pode também associar-se a um livro, através de uma avaliação, que se caracteriza por um rate (quantitativo) e a um comentário (qualitativo), embora para isso tenha que ter requisitado o livro. Um ponto de troca é caracterizado pela sua localização (morada e país), pelo número atual de livros e pela sua capacidade (número máximo de livros que comporta).

Diagrama UML

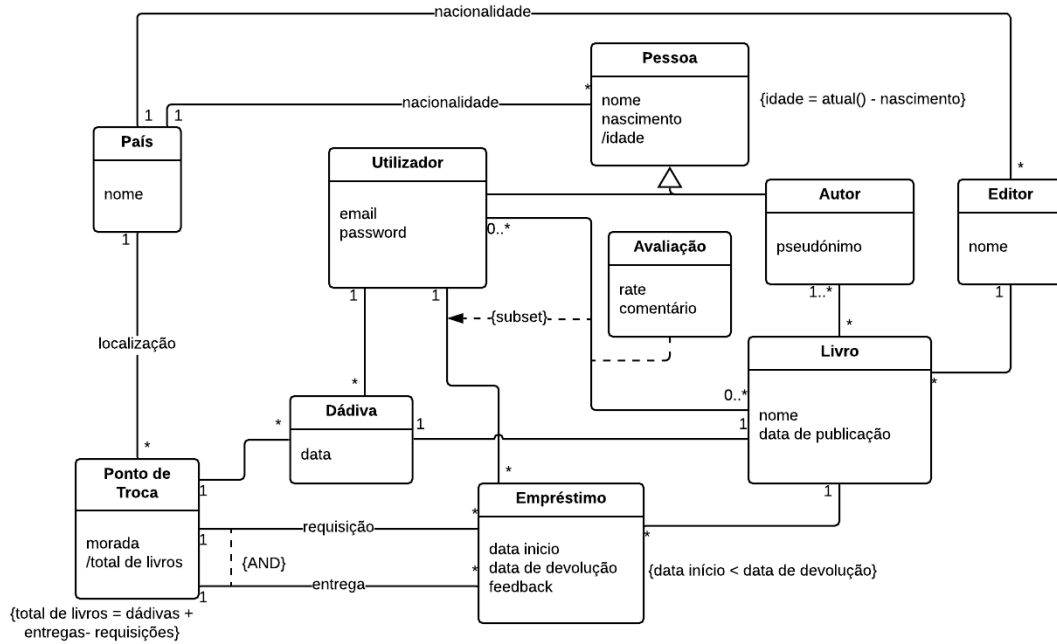
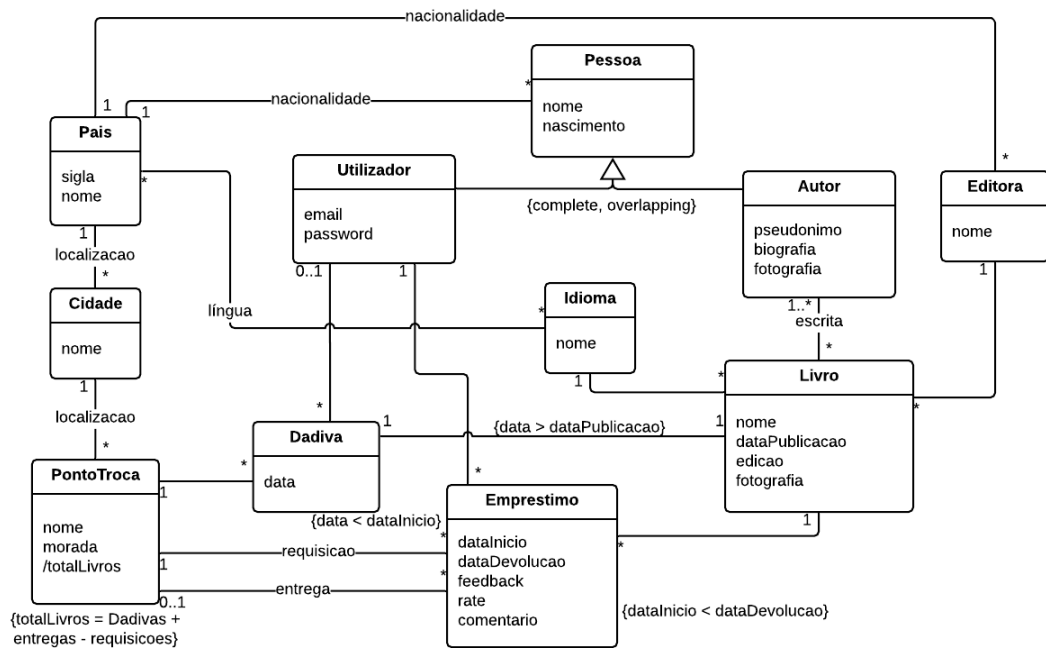


Diagrama UML Corrigido



Esquema Relacional

Esquema relacional, dependências funcionais e análise das formas normais

Pais(sigla, nome)

sigla -> nome

nome -> sigla

- $\{sigla\}^+ = \{sigla, nome\} \Rightarrow sigla \text{ é chave primária}$
- $\{nome\}^+ = \{sigla, nome\} \Rightarrow nome \text{ é chave}$

Cidade(idCidade, nome, pais -> Pais)

idCidade -> nome, pais

nome, pais -> idCidade

- $\{idCidade\}^+ = \{idCidade, nome, pais\} \Rightarrow idCidade \text{ é chave primária}$
- $\{nome, pais\}^+ = \{idCidade, nome, pais\} \Rightarrow \{nome, pais\} \text{ é chave composta}$
- pais é chave estrangeira para Pais

Idioma(nome)

- $\{nome\}^+ = \{nome\} \Rightarrow nome \text{ é chave primária}$

Lingua(pais -> Pais, idioma -> Idioma)

- $\{pais, idioma\}^+ = \{pais, idioma\} \Rightarrow \{pais, idioma\} \text{ é chave composta primária}$
- pais é chave estrangeira para Pais
- idioma é chave estrangeira para Idioma

Pessoa(idPessoa, nome, nascimento, pais -> Pais)

idPessoa -> nome, nascimento, pais

- $\{idPessoa\}^+ = \{idPessoa, nome, nascimento, pais\} \Rightarrow idPessoa \text{ é chave primária}$
- pais é chave estrangeira para Pais

Utilizador(idUtilizador -> Pessoa, email, password)

idUtilizador -> email, password

email -> idUtilizador

- $\{idUtilizador\}^+ = \{idUtilizador, email, password\} \Rightarrow idUtilizador \text{ é chave primária}$
- $\{email\}^+ = \{idUtilizador, email, password\} \Rightarrow email \text{ é chave}$
- idUtilizador é chave estrangeira para Pessoa

Autor(id -> Pessoa, pseudonimo, biografia, fotografia)

id -> pseudonimo, biografia, fotografia

- $\{id\}^+ = \{id, pseudonimo, biografia, fotografia\} \Rightarrow id \text{ é chave primária}$
- id é chave estrangeira para Pessoa

Livro(id, nome, dataPublicacao, edicao, fotografia, editora -> Editora, idioma -> Idioma)

id -> nome, dataPublicacao, edicao, fotografia, editora, idioma

- {id}⁺ = {id, nome, dataPublicacao, edicao, fotografia, editora, idioma} ⇔ id é chave primária
- editora é chave estrangeira para Editora
- idioma é chave estrangeira para Idioma

Escrita(livro -> Livro, autor -> Autor)

- {livro, autor}⁺ = {livro, autor} ⇔ {livro, autor} é chave primária composta
- livro é chave estrangeira para Livro
- autor é chave estrangeira para Autor

Emprestimo(id, dataInicio, dataDevolucao, feedback, rate, comentario, livro -> Livro, utilizador -> Utilizador, pontoRequisicao -> PontoTroca, pontoEntrega -> PontoTroca)

id -> dataInicio, dataDevolucao, feedback, rate, comentario, livro, utilizador, pontoRequisicao, pontoEntrega

utilizador, livro, dataInicio -> id

- {id}⁺ = {id, dataInicio, dataDevolucao, feedback, rate, comentario, livro, utilizador, pontoRequisicao, pontoEntrega} ⇔ id é chave primária
- {utilizador, livro, dataInicio}⁺ = {id, dataInicio, dataDevolucao, feedback, rate, comentario, livro, utilizador, pontoRequisicao, pontoEntrega} ⇔ {utilizador, livro, dataInicio} é chave composta
- livro é chave estrangeira para Livro
- utilizador é chave estrangeira para Utilizador
- pontoRequisicao é chave estrangeira para PontoTroca
- pontoEntrega é chave estrangeira para PontoTroca

Editora(id, nome, pais -> Pais)

id -> nome, pais

- {id}⁺ = {id, nome, pais} ⇔ id é chave primária
- pais é uma chave estrangeira para Pais

Dadiva(id, data, livro -> Livro, utilizador -> Utilizador, pontoTroca -> PontoTroca)

id -> data, livro, utilizador, pontoTroca

livro -> id

- {id}⁺ = {id, data, livro, utilizador, pontoTroca} ⇔ id é chave primária
- {livro}⁺ = {id, data, livro, utilizador, pontoTroca} ⇔ livro é chave
- livro é chave estrangeira para Livro
- utilizador é chave estrangeira para Utilizador
- pontoTroca é chave estrangeira para PontoTroca

PontoTroca(id, nome, morada, totalLivros, cidade -> Cidade)

id -> nome, morada, totalLivros, cidade

morada -> id

- $\{id\}^+ = \{id, nome, morada, totalLivros, cidade\} \Rightarrow id$ é chave primária
- $\{morada\}^+ = \{id, nome, morada, totalLivros, cidade\} \Rightarrow morada$ é chave
- cidade é chave estrangeira para Cidade

O domínio de cada atributo contém apenas valores atômicos, e o valor de cada atributo contém apenas um único valor do domínio. Desta maneira, respeita a primeira forma normal.

Dado que confirmamos que está na primeira forma normal e nenhum atributo não primo da relação é funcionalmente dependente de um subconjunto próprio de uma chave candidata, também se encontra na segunda forma normal.

Dado que se verifica a segunda forma normal e todos os atributos não primos da relação dependem funcionalmente de todas as chaves candidatas da relação de forma não transitiva, podemos afirmar que também se verifica a terceira forma normal.

Como em todas as dependências funcionais, o conjunto dos atributos do lado esquerdo corresponde a uma chave (ou superchave) da relação, verifica-se também a BCNF. Tanto a BCNF como a 3NF garantem que não há perdas de informação, sendo que a 3NF garante a preservação de todas as dependências funcionais.

Restrições

Pais(sigla, nome)

- sigla PRIMARY KEY CHECK (LENGTH(sigla) == 2) - cada país tem uma sigla única; sigla tem um tamanho igual a 2 letras
- nome NOT NULL, UNIQUE - cada país tem que ter uma nome obrigatoriamente; dois países não podem ter o mesmo nome

Cidade(idCidade, nome, pais->Pais)

- idCidade PRIMARY KEY - cada cidade tem um id único
- nome NOT NULL - cada cidade tem que ter uma nome obrigatoriamente
- pais NOT NULL, REFERENCES Pais - pais identifica o país onde se localiza a cidade; cada cidade tem que ter uma país associado obrigatoriamente
- UNIQUE(nome, pais) - num país, não existem duas cidades com o mesmo nome

Idioma(nome)

- nome PRIMARY KEY - cada Idioma tem um nome único

Lingua(pais->Pais, idioma->Idioma)

- pais NOT NULL, REFERENCES Pais - pais identifica um país onde se fala a língua
- idioma NOT NULL, REFERENCES Idioma - idioma identifica o idioma relacionado com a língua
- PRIMARY KEY (pais, idioma) - cada tuplo país, idioma é único

Pessoa(idPessoa, nome, nascimento, pais->Pais)

- idPessoa PRIMARY KEY - cada pessoa tem um idPessoa próprio
- nome NOT NULL - o nome da pessoa é obrigatório
- pais NOT NULL REFERENCES Pais -> cada pessoa esta associada ao seu país de nascença

Utilizador(idUtilizador->Pessoa, email, password)

- idUtilizador PRIMARY KEY REFERENCES Pessoa - cada utilizador tem um idUtilizador único; o id do utilizador permite identificar a pessoa a que corresponde
- email UNIQUE NOT NULL - cada utilizador tem um email único
- password NOT NULL - password é obrigatória

Autor(id->Pessoa, pseudonimo, biografia, fotografia)

- id PRIMARY KEY REFERENCES Pessoa - cada Pessoa tem um id único; o id do utilizador permite identificar a pessoa a que corresponde

Livro(id, nome, dataPublicacao, edicao, fotografia, editora->Editora, idioma->Idioma)

- id PRIMARY KEY - cada livro, mesmo que igual, tem um id único (cópias do mesmo livro tem ids diferentes)
- editora NOT NULL REFERENCES Editora - cada livro tem uma editora associada
- idioma NOT NULL REFERENCES Idioma - cada livro está escrito num idioma
- fotografia UNIQUE, NOT NULL - dado que é uma forma de obtenção do id do livro

- nome NOT NULL - um livro tem obrigatoriamente que ter um nome

Escrita(livro->Livro, autor->Autor)

- livro NOT NULL, REFERENCES Livro - cada escrita tem um livro associado
- autor NOT NULL, REFERENCES autor - cada escrita tem um autor associado
- PRIMARY KEY (livro, autor) - cada par livro, autor é único

Emprestimo(id, dataInicio, dataDevolucao, feedback, rate, comentario, livro->Livro, utilizador->Utilizador, pontoRequisicao->PontoTroca, pontoEntrega->PontoTroca)

- PRIMARY KEY id - cada empréstimo tem um id única
- dataInicio NOT NULL - o empréstimo tem obrigatoriamente que ter uma data de início
- rate CHECK (rate>=0 and rate<=5) - o rate é sempre um número entre 0 e 5 (inclusivé)
- livro NOT NULL, REFERENCES Livro - cada empréstimo tem um livro associado
- utilizador NOT NULL, REFERENCES Utilizador - cada empréstimo tem um utilizador associado
- pontoRequisicao NOT NULL, REFERENCES PontoTroca - cada empréstimo tem um ponto de requisição associado
- pontoEntrega REFERENCES PontoTroca
- UNIQUE (utilizador, livro, dataInicio) - o utilizador conjuntamente com o livro e a dataInicio permitem identificar o empréstimo em questão
- CHECK (dataInicio < dataDevolucao) - a data de início do empréstimo tem que ser sempre inferior à data de devolução do livro
- CHECK ((dataDevolucao IS NOT NULL and pontoEntrega IS NOT NULL) or (dataDevolucao IS NULL and pontoEntrega IS NULL and feedback IS NULL and rate IS NULL and comentario IS NULL)) - a existência de uma data de devolução implica a existência de um pontoEntrega e vice-versa e a inexistência de um implica também a inexistência do outro; só é possível dar feedback, rate e comentário depois de se devolver o livro;

Editora(id, nome, pais->Pais)

- id PRIMARY KEY - cada editora tem um id único
- nome NOT NULL - uma editora tem que ter um nome
- pais REFERENCES Pais - cada editora tem sede num determinado país

Dadiva(id, data, livro->Livro, utilizador-> Utilizador, pontoTroca->PontoTroca)

- id PRIMARY KEY - cada id é único
- data NOT NULL - cada dádiva é efetuada numa data específica
- livro UNIQUE, NOT NULL, REFERENCES Livro - um livro só pode ser doado uma vez; o livro tem, obrigatoriamente de estar registado para todas as operações futuras
- utilizador REFERENCES Utilizador - o livro é doado por um utilizador, que pode ser anónimo
- pontoTroca NOT NULL, REFERENCES PontoTroca - para requisições futuras é preciso saber a sua localização

PontoTroca(id, nome, morada, totalLivros, cidade->Cidade)

- id PRIMARY KEY - cada ponto de troca é único, tendo todos ids diferentes

- morada UNIQUE, NOT NULL - para todos os utilizadores poderem encontrar o ponto de troca
- totalLivros NOT NULL DEFAULT 0 - para saber os livros que lá se encontram (será implementado futuramente)
- cidade NOT NULL, REFERENCES Cidade - cada ponto de troca está localizado numa cidade

Outras Restrições

- Dativa.Data < Emprestimo.DataInicio
- Dativa.Data > Livro.DataPublicacao
- A generalização associada a Pessoa é do tipo complete, overlapping

restrições não foram ainda implementadas, visto que o conteúdo lecionado ainda não o permite fazer.

Interrogações

1. Quais os utilizadores que requisitaram todos os livros de um determinado autor?
2. Qual o utilizador com maior número de dádivas?
3. Quais os emails dos pares de utilizadores que já leram um livro em comum?
4. Qual é a rate média de cada livro emprestado por ordem decrescente?
5. Qual é o idioma mais comum para cada livro (par nome-autor)?
6. Qual os livros que estão emprestados?
7. Qual é o livro com mais empréstimos no sistema?
8. Qual é o livro que não é requisitado a mais tempo? (Poderá haver livros que nunca foram requisitados e dados à mais tempo que o último a ser requisitado.)
9. Quais os utilizadores que não devolveram um livro?
10. Qual é o nome do país com mais pontos de troca?

Extra:

11. Quantos livros com o mesmo nome e autor existem?
12. Qual é o ponto de troca com mais livros disponíveis?

Gatilhos

1. Sempre que é feita uma dádiva ou devolvido um livro num ponto de troca, o total de livros disponíveis incrementa.
2. Sempre que é feito uma requisição de um livro num ponto de troca, o total de livros disponíveis decrementa.
3. Se houver uma tentativa de fazer uma dádiva de um livro cuja data seja anterior à data de publicação do livro esta não é realizada. Se houver uma tentativa de fazer um empréstimo de um livro cuja data de início seja anterior à data da dádiva desse mesmo livro este não é realizada.