

Projeto de BDAD

Sistema de compra e venda de livros

Grupo 404

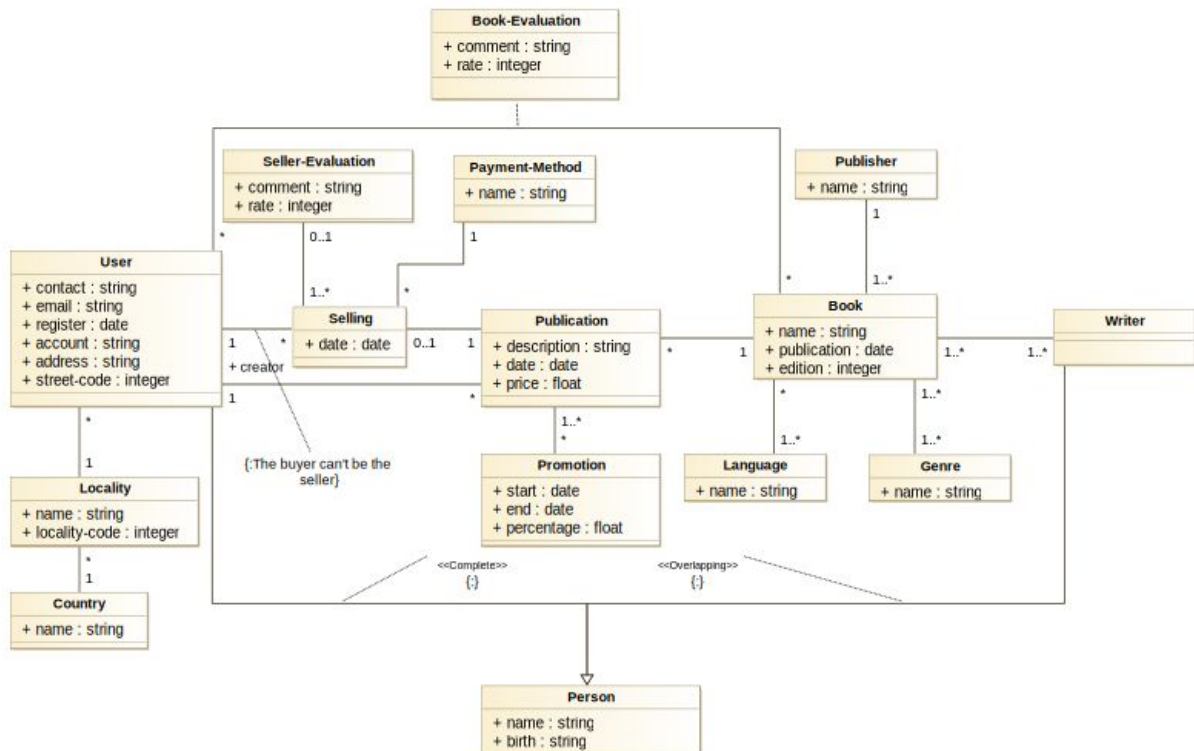
Alexandre Almeida de Abreu Filho - up201800168

Juliane de Lima Marubayashi - up201800175

João Castro Pinto - up201806667

Porto, 2020

Modelo UML



Contexto

O projeto modela um sistema de compra e venda de livros online, onde os próprios utilizadores realizam as vendas e compras.

User

A classe user representa os vendedores e compradores de livros, tem como atributos o contacto, email, data de registo no sistema, conta bancária, e morada (que é representada por house-number, street, postal-code e Locality).

Selling

Selling representa uma compra dos itens de uma publicação. Quem faz a compra é o utilizador que está diretamente associado à classe “Selling”. O vendedor está indiretamente associado a selling através da publicação. O criador associado à publicação é quem está a fazer a venda.

Quando se faz uma compra, é possível avaliar o utilizador que vendeu o livro. Esta informação guarda-se no objeto “Seller-Valuation”, que tem como atributos uma nota de 1 a 5, e um comentário sobre o vendedor.

Tem como atributos a data de realização da compra e o método de pagamento (indicado pela classe Method)

Publication

Quando se coloca um livro à venda, faz-se através de uma publicação. Portanto, terá de estar associado a um utilizador quando é criado, ao livro a que se está a vender, e às promoções que são ou foram aplicadas a ele. Se um livro for vendido, ou seja, se alguém comprar o livro que está à venda na publicação, cria-se um objeto “Selling”.

Promotion

Uma publicação específica pode ter uma promoção inicial e mais tarde pode-se adicionar outra, portanto uma publicação pode ter várias promoções aplicadas e assim a relação é de um para muitos.

Book-Valuation

A fim de ter uma avaliação fiel a reputação do livro, um cliente não necessariamente precisa comprá-lo para o avaliar. Assim, mesmo que determinado livro nunca tenha sido vendido no site, este ainda pode possuir um histórico de avaliações e, portanto, compradores futuros irão possuir um parâmetro de qualidade.

Book

A classe Book representa uma obra e tem como atributos o nome, a data de publicação a edição, o idioma, o gênero, o autor e a editora.

Person

Person é uma generalização da classe “User” e “Writer”, sendo “Writer” a classe que representa um escritor de um livro.

Definição do esquema relacional

Publication(id, description, date, price, idUser->User, ISBN->Book)

Publisher(id, name)

Selling(idPublication->Publication, idUser->User, date, idPayment->Payment-Method, evaluation->Seller-Evaluation)

Genre(id, name)

Language(code, name)

Person(id, name, birth)

Locality(locality-code, name, nameCountry->Country)

Country(code, name)

User(idPerson->Person, contact, email, register, account, address, street-code, locality-code->Locality)

Writer(idPerson->Person)

Book(ISBN, name, publication, edition, idPublisher->Publisher)

Promotion(id, start, end, percentage)

Payment-Method(id, name)

Seller-Evaluation(id, comment, rate)

Book-Evaluation(idPerson->User, ISBN->Book, comment, rate)

Relações de associação:

Book-Writer(idWriter->Writer, ISBN->Book)

Book-Genre(ISBN->Book, idGenre->Genre)

Book-Language(ISBN->Book, codeLanguage->Language)

Publication-Promotion(idPublication->Publication, idPromotion->Promotion)

Análise de Dependências Funcionais e Formas Normais

Publication(id, description, date, price, idUser, ISBN)

id -> description, date, price, idUser, ISBN

Publisher(id, name)

id -> name

Selling(idPublication, idUser, date, idPayment, evaluation)

idPublication -> idUser, date, idPayment, evaluation

Genre(id, name)

id -> name

name -> id

Language(code, name)

code -> name

name -> code

Person(id, name, birth)

id -> name, birth

Locality(locality-code, name, nameCountry)

locality-code -> name, nameCountry

Country(code, name)

code -> name

name -> code

User(idPerson, contact, email, register, account, address, street-code, locality-code)

idPerson -> contact, email, register, account, address, street-code, locality-code

email -> idPerson, contact, register, account, address, street-code, locality-code

contact -> idPerson, email, register, account, address, street-code, locality-code

account -> idPerson, contact, email, register, address, street-code, locality-code

address -> locality-code, street-code

Writer(idPerson)

(*nothing*)

Book(ISBN, name, publication, edition, idPublisher)

ISBN -> name, publication, edition, idPublisher

Promotion(id, start, end, percentage)

id -> start, end, percentage

start, end, percentage -> id

Payment-Method(id, name)

id -> name

name -> id

Seller-Evaluation(id, comment, rate)

id -> comment, rate

comment, rate -> id

Book-Evaluation(idPerson, ISBN, comment, rate)

idPerson, ISBN -> comment, rate

Book-Writer(idWriter, ISBN)

(*nothing*)

Book-Genre(ISBN, idGenre)

(*nothing*)

Book-Language(ISBN,codeLanguage)

(*nothing*)

Publication-Promotion(idPublication, idPromotion)

(*nothing*)

Conclusão

Atributos que seguem as formas normais

3 Forma Normal:

Todas as dependências pertencem a 3 forma normal, exceto o **User**.

BCNF:

Todas as dependências pertencem a BCNF, exceto o **User**.

Restrições

As restrições foram aplicadas com CHECK são:

- *edition* e *ISBN* de **Book** têm de ser um número natural;
- *price* de **Publication** tem de ser positivo;
- *rate* de **SellerEvaluation** tem de ser um número entre 0 e 5 inclusive;
- *rate* de **BookEvaluation** tem de ser um número entre 0 e 5 inclusive;
- *end* tem que ser após *start* em uma **Promotion**, além de que *percentage* tem que ser um número maior que 0 e menor ou igual a 100;
- *contact* em **User** precisa ser um inteiro maior que 10000000, a fim de ser um número válido

A restrição NOT NULL foi aplicada nos seguintes casos:

- Em todos os atributos *name* pertencentes às tabelas **Country**, **Publisher**, **PaymentMethod**, **Genre**, **Language**, **Locality**, **Person**, **Book**.
- Em todos os atributos do tipo DATE à exceção do atributo *end* da tabela **Promotion**.
- Nos atributos *contact*, *email*, *register* (tipo DATE), *account*, *address* e *streetCode* pertencentes à tabela **User**.
- Nos atributos *name*, *publication* (tipo DATE) e *edition* pertencentes à tabela **Book**.
- Nos atributos *registerDate* (tipo DATE) e *price* pertencentes à tabela **Publication**.
- No atributo *rate* da tabela **SellerEvaluation**.
- Nos atributos *start*(tipo DATE) e *percentage* pertencentes à tabela **Promotion**.
- No atributo *rate* da tabela **BookEvaluation**.

Interrogações

Fáceis

1. Qual o isbn e a média das classificações dos livros mais bem avaliados pelos habitantes de Portugal?
2. Qual o preço médio de venda de cada um dos livros?
3. Qual mês tem maior número de promoções?

Médias

4. Qual o id, o nome e quantidade de livros em promoção da editora que possui menos livros em promoção?
5. Quais as percentagens mais comuns aplicadas em promoções (intervalos de 10 valores) e quantas vezes cada percentagem é utilizada?
6. Quais os id's dos utilizadores que possuem publicações de livros com average rating maior ou igual a 4 do idioma francês?
7. Qual o id dos usuários que tiveram todos os seus livros vendidos e quantos livros venderam?

Difíceis

8. Qual a diferença da nota média das compras entre utilizadores de Portugal e de fora?
9. Qual o idioma mais vendido em cada país e sua percentagem das vendas nesse país?
10. Quais os utilizadores que compraram um livro em comum?

Gatilhos

1. Comprador de uma publication não por ser o Vendedor
2. Quando o user é removido da base de dados, suas publicações não vendidas são removidas.
3. Um utilizador só pode ser removido se não tiver feito nenhuma compra e nenhuma venda. Ou seja, se um utilizador não tiver feito compras, mas tiver publicações não vendidas, ele pode ser removido.