

Bases de Dados



Omazon

Modelo Conceptual

Estudantes & Autores:

Turma 4 Grupo 2

André Moreira up201904721@fe.up.pt

Fernando Rego up201905951@fe.up.pt

Nuno Alves up201908250@fe.up.pt

Índice

1. Contexto	3
2. Especificação	3
3. Restrições	4
4. Modelo Conceptual	5

1. Contexto

A **Amazon** é uma empresa de vendas online que visa disponibilizar diversas categorias, com vários produtos. Esses produtos mantêm informação das vendas e avaliações de cada produto, para que os clientes possam ter mais facilidade em escolher entre diversos produtos.

Para isso, vai-se criar uma base de dados em que se guarda toda a informação dos clientes e empregados, dos seus respectivos produtos, reviews correspondentes e também encomendas.

2. Especificação

São guardados dois tipos de utilizadores na base de dados, clientes e funcionários. A classe **Person** é uma generalização das classes **Client** e **Employee**. Para esta, é guardada informação relativa ao nome, morada, data de nascimento, nif, telefone e email.

Cada cliente pode ter associado **Review's** que tenha feito a produtos. Estas guardam informação relativa ao título, descrição e avaliação numérica (compreendido entre 1 e 5). Note-se que o utilizador pode fazer várias avaliações ao mesmo produto. Também pode estar associado a vários carrinhos que guardam informação sobre os produtos comprados relativos a uma certa encomenda.

Os produtos guardam informações como nome e descrição. Cada produto pode estar associado a vários armazéns, havendo stock diferente para cada uma das relações com os armazéns. Também pode ser calculada a avaliação numérica média e o número de reviews associadas a esse produto. Estes produtos também guardam informação quanto à categoria e ao fabricante. Existe também uma relação entre **Product** e **Cart**, **CartQuantity**, que além de guardar informação da quantidade de cada produto no carrinho, este também guarda um preço. Este *price* é útil pelo facto de que os preços flutuam ao longo do tempo, e convém guardar informação dos preços na altura da compra, para o utilizador ter uma noção correta sobre os produtos que comprou no passado.

Pelo facto de haver múltiplas instâncias do carrinho associadas ao mesmo cliente, enquanto uma compra não for terminada, os produtos serão adicionados ao carrinho mais recente (através do atributo *date*).

Uma compra só pode ser efetuada se todos os produtos no carrinho tiverem stock em pelo menos um armazém. Ao efetuar a compra, o carrinho ficará associado a uma **Order**

para guardar informação dos produtos que foram comprados. Esta pode ter vários estados (encomendada, paga, enviada...).

O pagamento pode ser efetuado através de **MBWay** ou **CreditCard**, sendo guardados os dados necessários a cada um em conjunto com o valor e data.

3. Restrições

- $\{rating \in [1,5]\}$ (em **Rating**): O valor de *rating* de **Review** é um inteiro entre 1 e 5.
- $\{product.stock > 0\}$ (entre **Cart** e **Order**): Não é possível comprar um produto que não tenha stock necessário para responder ao pedido do utilizador (*ammount* em **CartQuantity**) nos armazéns (*ammount* em **Stock**).
- $\{order \text{ is paid}\}$ (entre **Order** e **Shipment**): A encomenda tem de estar paga (**Payment**) para que possa ser enviado o produto.
- $\{ammount \geq 0\}$ (em **Stock**): Não pode haver stock negativo

4. Generalizações

- **Person**: *Complete* e *Overlapping*, visto que se está na guardado na base de dados tem de ser ou cliente, ou funcionário (complete). Além disso, considera-se que um funcionário pode ser também um cliente (overlapping), guardando assim a sua informação apenas uma vez, para não haver duplicação de dados.
- **Payment**: *Complete* e *Disjoint*, visto que tem de ser pelo menos um dos métodos de pagamento (complete) e pode ser apenas usado um desses métodos (disjoint).

5. Modelo Conceptual

