

Exercício de Modelagem Conceitual e Normalização de Banco de Dados para Compras em um Site

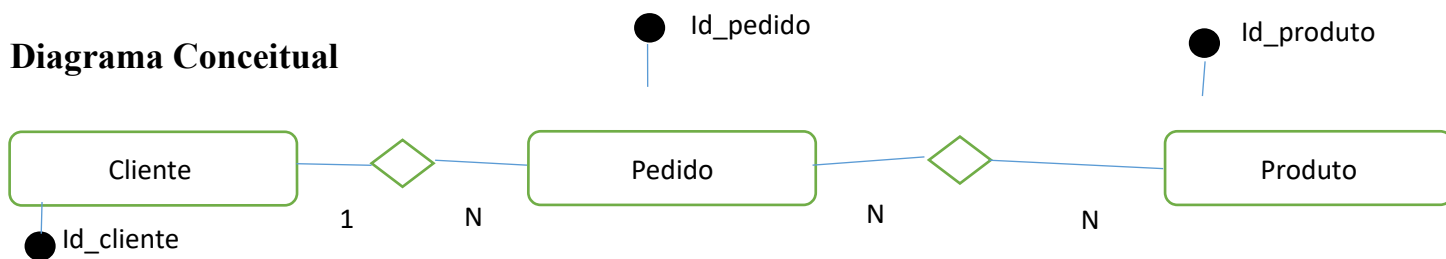
Cenário:

Você está desenvolvendo o banco de dados para um site de compras online. O sistema deve gerenciar os produtos, os clientes que fazem as compras e os pedidos realizados.

Requisitos do sistema:

1. O site deve armazenar informações de clientes (nome, email, endereço e uma chave).
 2. O site deve manter o catálogo de produtos (nome, descrição, preço e uma chave).
 3. Cada cliente pode realizar vários pedidos.
 4. Cada pedido contém um ou vários itens da lista de produtos.
 5. Cada item da lista de produtos em um pedido deve ter a quantidade solicitada.
 6. Banco de dados é Sqlite
-
- I. Monte o DER para este sistema.
 - II. Forneça o projeto lógico das tabelas.
 - III. Cuide para que as relações do projeto lógico estejam normalizadas na 1FN, 2FN e 3FN.
 - IV. Explique por que cada tabela do modelo lógico está normalizada.

Diagrama Conceitual



Cliente (id_cliente, nome, email, endereço)

Produto (id_produto, nome, descrição, preço)

Pedido (id_pedido, id_cliente, data_pedido).

Projeto Lógico

Item_pedido (id_pedido, id_produto, quantidade) esta tabela resolve a cardinalidade vários para vários entre pedido e produto

1. Cliente

- **ID_Cliente** (PK)
- Nome text
- Email text
- Logradouro text
- Complemento text
- Cod_municipio text
- Cod_uf text
- CEP

O atributo endereço foi desmembrado em logradouro, complemento, cod_municipio, cod_uf e cep para deixar de ser multivalorado.

2. Produto

- **ID_Produto** inteiro (PK)
- Nome text
- Descrição text
- Preço real

3. Pedido

- **ID_Pedido** inteiro (PK)
- Dia int
- Mes int
- Ano int
- **ID_Cliente** inteiro (FK para cliente)

4. Item_Pedido

- **ID_Pedido** int (FK para Pedido)
- **ID_Produto** int (FK para produto)
- Quantidade int check (Quantidade>0)

5. UF

cod_uf int (PK)
Nome_uf text not null

6. Municipio

Cod_municipio int (PK)
Nome_municipio text not null
Cod_uf int (FK para UF)

1ª Forma Normal (1FN):

- Elimina valores repetidos ou múltiplos valores em uma única coluna. As tabelas já estão na 1ª FN, pois todos os atributos contêm valores atômicos (únicos e indivisíveis). Todos as tabelas atendem este requisito. A tabela Cliente possui endereço.

2ª Forma Normal (2FN):

- Uma tabela está na 2ª FN se estiver na 1ª FN e todos os atributos que não são chaves dependem inteiramente da chave primária. Todos as tabelas atendem este requisito. A tabela pedido apesar de ter chave composta se cada um de seus atributos sozinhos não define quantidade.

3ª Forma Normal (3FN):

- Para estar na 3ª FN, a tabela deve estar na 2ª FN e não deve haver dependência transitiva (ou seja, nenhum atributo não chave deve depender de outro atributo não chave). A primeira tabela tem cod_municipio que define cod_uf, mas já está normalizada.