

Anny Karolliny de Oliveira Silva - **20211214010002**;

Antonio Carlos Gomes Trindade - **20211214010032**;

Terezinha Dantas Lucena de Medeiros - **20211214010011**.

Etapa 2 - Transformação entre modelos

- É necessário levar em conta o diagrama de entidade-relacionamento desenvolvido da etapa 1 do projeto.

Desenvolvimento

CLIENTE - efetua - PEDIDO: **ADIÇÃO DE COLUNA**

Cliente (cpfCliente, nomeCliente, telefone, dataNas, endereco)

Pedido (idPedido, valorPedido, quantPedido, dataPedido, cpfCliente)

cpfCliente referencia Cliente

VENDEDOR - emite - PEDIDO: **TABELA PRÓPRIA**

Vendedor (rgVendedor, nomeVendedor, emailVendedor)

Pedido (idPedido, valorPedido, quantPedido, dataPedido)

emite (rgVendedor, idPedido)

rgVendedor referencia Vendedor

idPedido referencia Pedido

PEDIDO - contém - PRODUTO: **TABELA PRÓPRIA**

Pedido (idPedido, valorPedido, quantPedido, dataPedido)

Produto (idProduto, valorProduto)

contem (idPedido, idProduto)

idPedido referencia Pedido

idProduto referencia Produto

FORNECEDOR - disponibiliza - PRODUTO: **TABELA PRÓPRIA**

Fornecedor (cnpj, dataEntrega)

Produto (idProduto, valorProduto)

disponibiliza (cnpj, idProduto)

cnpj referencia Fornecedor

idProduto referencia Produto

Passo-a-passo da transformação em cada um dos casos:

Durante a transformação do modelo conceitual para o modelo lógico, vale perceber que as entidades do modelo conceitual se comportam como tabelas, levando em consideração os seus atributos, onde serão as colunas. Com relação a nomenclatura, deve-se manter nomes curtos nas colunas e atentar-se caso possivelmente se repita em outras tabelas, inserindo o sufixo com o nome da tabela. A cardinalidade também é **extremamente importante** na tradução dos relacionamentos e seus atributos. Sendo assim, existem diferentes estratégias para transformação dos relacionamentos, sendo elas: tabela própria, adição de coluna e fusão de tabelas. Serão aplicadas dependendo da cardinalidade existente.

No caso de **CLIENTE - efetua - PEDIDO**, temos a cardinalidade de (1,1) e (1, n), respectivamente. É um relacionamento de 1:N, levando em conta a cardinalidade máxima dos dois casos. É necessário perceber qual é a melhor estratégia que deverá ser utilizada. Nesse caso, entre as três opções de estratégias, **ADIÇÃO DE COLUNA** é a mais indicada. O processo ocorre da seguinte forma: transforma-se a primeira entidade do relacionamento em uma tabela, repetindo os seus respectivos atributos existentes. A segunda entidade (lado N do relacionamento) presente também se transforma em uma tabela, repetindo os seus atributos e adicionando outros: um atributo para referenciar a primeira tabela, onde, no nosso caso, foi o **cpfCliente**, e um atributo presente no próprio relacionamento, caso exista. Em CLIENTE - efetua - PEDIDO não há nenhum atributo presente no relacionamento **efetua**, por isso não foi adicionado.

Observação: a primeira tabela (Cliente) possui **cpfCliente** como chave primária, por isso o atributo cpfCliente presente na segunda tabela está referenciando CLIENTE.

Já em **VENDEDOR - emite - PEDIDO**, temos a cardinalidade de (1, n) e (0, n), respectivamente. Diferente do caso CLIENTE - efetua - PEDIDO, é um relacionamento de N:N, respeitando as cardinalidades máximas dos dois casos. Sendo assim, percebe-se que a melhor estratégia a ser utilizada é a **TABELA PRÓPRIA**, justamente por ser a mais adequada. O processo ocorre da seguinte forma: transforma-se a primeira entidade em uma tabela, repetindo os seus atributos. Acontece a mesma situação com a segunda entidade. Por fim, transforma-se o relacionamento entre as duas entidades também em uma tabela, adicionando um atributo para referenciar a primeira tabela, outro para referenciar a segunda tabela e um terceiro atributo caso exista algum no relacionamento. Caso não exista, coloca-se somente os dois para referenciar as duas tabelas criadas, como ocorreu no nosso caso.

Observação: a primeira tabela (Vendedor) possui **rgVendedor** como chave primária, enquanto a segunda tabela (Pedido) possui **idPedido**. O relacionamento emite não possui nenhum atributo e, por isso, é transformado em uma tabela somente com os atributos rgVendedor e idPedido, onde rgVendedor referencia VENDEDOR e idPedido referencia PEDIDO. Vale lembrar que esses atributos constituem uma chave primária composta.

Em **PEDIDO - contém - PRODUTO**, temos a cardinalidade de (0, n) e (1, n), respectivamente. Também é um relacionamento de N:N, como o anterior. Sendo assim, a melhor estratégia a ser utilizada é a **TABELA PRÓPRIA**, por ser a mais adequada no relacionamento de N:N. O processo é o mesmo ocorrido anteriormente. Transforma-se as entidades (pedido e produto) em tabelas, repetindo os seus respectivos atributos, incluindo as chaves primárias. O relacionamento (contém) também é transformado em uma tabela, onde os seus atributos serão a chave primária presente na primeira tabela (pedido) e na segunda tabela (produto). Caso esse relacionamento já possua algum atributo, ele também será adicionado na tabela. No nosso caso, o relacionamento contém não possui nenhum atributo, então foram adicionadas somente as chaves primárias das duas tabelas criadas (pedido e produto), referenciando cada uma delas.

Observação: a tabela PEDIDO possui **idPedido** como chave primária, enquanto a tabela PRODUTO possui **idProduto**. Na tabela CONTEM existem somente os atributos idPedido e idProduto, pois esse relacionamento já não possuía nenhum atributo. Esses atributos adicionados fazem parte de uma chave primária composta.

Concluindo, em **FORNECEDOR - disponibiliza - PRODUTO**, temos a cardinalidade de (1, n) e (0, n), respectivamente. Assim como nos outros dois casos, é um relacionamento de N:N e a estratégia mais adequada a ser utilizada é **TABELA PRÓPRIA**. Ocorre a mesma situação presente nos dois casos anteriores: transforma-se a entidade FORNECEDOR em uma tabela, repetindo os seus respectivos atributos. A entidade PRODUTO também é transformada em tabela e os seus atributos prevalecem. Por fim, o relacionamento disponibiliza torna-se uma terceira tabela e nela haverá a chave primária pertencente a primeira tabela, a chave primária pertencente a segunda, além dos seus próprios atributos (caso exista). Nesse caso, o relacionamento disponibiliza não possui nenhum atributo, por isso não foi adicionado. As chaves primárias da primeira e segunda tabela referenciam cada uma delas.

Observação: a chave primária da tabela FORNECEDOR é o **cnpj**, enquanto a da tabela PRODUTO é o **idProduto**. Dessa forma, foram adicionados a tabela DISPONIBILIZA, referenciando FORNECEDOR e PRODUTO. Como não existia nenhum atributo inicialmente, pertenceram somente esses dois. Sendo assim, vale destacar que fazem parte de uma chave primária composta.