

Conceitos Básicos de modelagem de dados

Se você pretende desenvolver aplicações que usam banco de dados relacionais deverá possuir os conceitos básicos sobre modelagem de dados. Não importa se sua aplicação é muito simples ; a correta modelagem dos seus dados irá com certeza tornar sua aplicação mais robusta e mais fácil de manter.

O propósito deste artigo é fornecer os conceitos básicos sobre modelagem de dados. Este assunto daria centenas de livros por isto estarei sendo o mais direto e o objetivo possível de forma a que você possa aplicar de imediato os conceitos aprendidos. Como o título já diz serão conceitos básicos e sobre banco de dados relacionais.

Qual o objetivo da modelagem de dados ? Por que modelar ?

- Representar o ambiente observado
- Documentar e normalizar
- Fornecer processos de validação
- Observar processos de relacionamentos entre objetos

Modelar implica em construir modelos então como fazer isto ? Podemos definir as etapas envolvidas na construção de modelos em :

1 - **Modelo conceitual** - Representa as regras de negócio sem limitações tecnológicas ou de implementação por isto é a etapa mais adequada para o envolvimento do usuário que não precisa ter conhecimentos técnicos. Neste modelo temos :

- Visão Geral do negócio
- Facilitação do entendimento entre usuários e desenvolvedores
- Possui somente as entidades e atributos principais
- Pode conter relacionamentos n para m.

2- **Modelo Lógico** - Leva em conta limites impostos por algum tipo de tecnologia de banco de dados. (banco de dados hierárquico , banco de dados relacional ,etc.). Suas características são :

- Deriva do modelo conceitual e via a representação do negócio
- Possui entidades associativas em lugar de relacionamentos n:m
- Define as chaves primárias das entidades
- Normalização até a 3a. forma normal
- Adequação ao padrão de nomenclatura
- Entidades e atributos documentados

3- **Modelo Físico** - Leva em consideração limites impostos pelo SGBD (Sistema Gerenciador de Banco de dados) e pelos requisitos não funcionais dos programas que acessam os dados. Características:

- Elaborado a partir do modelo lógico
- Pode variar segundo o SGBD
- Pode ter tabelas físicas (log , lider , etc.)
- Pode ter colunas físicas (replicação)

Precisamos definir agora entidade e atributo. O que são e o que representam ?

Uma **Entidade** pode ser definida como qualquer coisa do mundo real , abstrata ou concreta , na qual se deseja guardar informações. (*Tabela , File, etc..*). Exemplos de entidades : Cliente , Produto , Contrato , Vendas , etc.

Um **atributo** é tudo o que se pode relacionar como propriedade da entidade. (*coluna , campo , etc,..*). Exemplos de atributos : Código do Produto (Entidade Produto) , Nome do Cliente (Entidade Cliente).

Nota : Chama-se Domínio o conjunto de valores possíveis do atributo.

Obs: Nenhum modelo é suficientemente claro se não for acompanhado de uma definição formal dos elementos , fazemos isto através do Dicionário de Dados . Lembre-se , conceitos que podem ser triviais a quem esta modelando podem não ser para pessoas leigas no assunto. Assim o dicionário de dados tem o objetivo de deixar claro qualquer informação que seja de valia para o processo de compreensão e unificação de conceitos.

Para que fique claro vamos fazer um exercício simples: Definir uma entidade que represente as informações de uma Pessoa e descrever seus atributos.

Podemos definir a entidade **Pessoa** que irá representar as informações de uma pessoa. Abaixo temos a representação da entidade e de alguns de seus atributos feitos no **ERWin**.

Pessoa

Número sequencial identificador de uma Pessoa
Texto do Nome da Pessoa
Número do CPF da Pessoa
Texto do Endereço da Pessoa
Texto do Nome do Pai da Pessoa
Texto do Nome da Mãe da Pessoa
Texto do Telefone da Pessoa
Texto do Email da Pessoa

Ao lado temos a representação feita no ERWin da Entidade Pessoa e de alguns de seus atributos.

Note que na definição dos atributos eu estou definindo a natureza do tipo de atributo. Exemplos de tipos de natureza: Texto , Número , Indicador(sim/não) , Código, etc.

Alguns atributos são obrigatórios outros são opcionais.

Nome é obrigatório pois toda pessoa deve ter um nome
Telefone é opcional pois nem toda pessoa possui um telefone

Então podemos fazer as seguintes definições:

Atributo obrigatório - è aquele que para uma instância de uma entidade ou relacionamento **deve** possuir um valor. (NOT NULL)

Atributo opcional - É aquele que para uma instância da entidade ou relacionamento **pode** possuir um valor. (NULL)

Podemos ainda classificar os atributos como :

Atributo Identificador - (#) - Atributo capaz de identificar exclusivamente cada ocorrência de uma entidade. Também conhecido como chave Primária ou Primary Key

(PK). Ex: Código do Cliente , Código do Produto , etc.(O símbolo # é usado para representar a chave primária em algumas notações)

Chave Candidata - Atributo ou grupamento de atributos que têm a propriedade de identificar unicamente uma ocorrência da entidade . Pode vir a ser uma chave Primária. *A chave candidata que não é chave primária também chama-se chave Alternativa.*

Características de uma Chave Primária :

a - NÃO PODE haver duas ocorrências de uma mesma entidade com o mesmo conteúdo na Chave Primária

b - A chave primária não pode ser composta por atributo opcional , ou seja , atributo que aceite nulo.

c - Os atributos identificadores devem ser o conjunto mínimo que pode identificar cada instância de um entidade.

d - Não devem ser usadas chaves externas. (Atributos sobre os quais você não tem controle. Ex: CPF)

e - Cada atributo identificador da chave deve possui um tamanho reduzido

f - Não deve conter informação volátil.

Ao criar modelos geralmente temos diversas entidades cada uma com diversos atributos que podem se relacionar entre si. Vamos definir como podem ser estes relacionamentos.

O que é um relacionamento ?

Um relacionamento pode ser entendido como uma associação entre instâncias de Entidades devido a regras de negócio. Normalmente ocorre entre instâncias de duas ou mais Entidades , podendo ocorrer entre instâncias da mesma Entidade (auto-relacionamento).

Por que o relacionamento é necessário ?

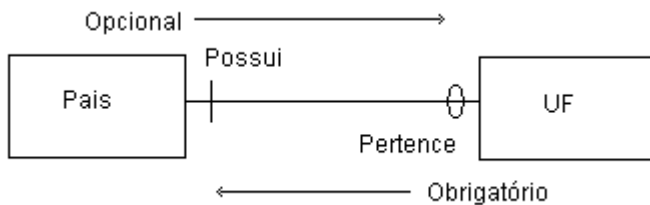
- Quando existem várias possibilidades de relacionamento entre o par das entidades e se deseja representar apenas um
- Quando ocorrer mais de um relacionamento entre o par de entidades
- Para evitar ambiguidade
- Quando houver auto-relacionamento

Para definir o número de ocorrências de uma entidade usamos o conceito de **Cardinalidade.**

A **Cardinalidade** indica quantas ocorrências de uma Entidade participam no mínimo e no máxima do relacionamento.

Cardinalidade Mínima - define se o relacionamento entre duas entidades é obrigatório ou não.

Ex: Abaixo temos a entidade Pais e a Entidade UF.

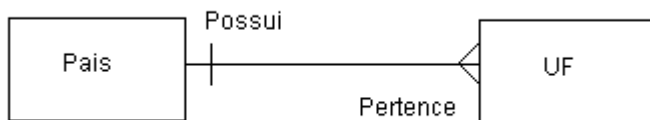


Um país possui no mínimo **ZERO** UF (Existem países que não possuem Estados . Ex: Vaticano)

Uma UF pertence pelo menos a **UM** País.

Nota: O nome UF talvez não seja mais apropriado. A entidade representa um estado ou subdivisão equivalente em um País

Cardinalidade Máxima - define a quantidade máxima de ocorrências da Entidade que pode participar do Relacionamento. Deve ser maior que zero. Ex: Abaixo temos a entidade País e a Entidade UF novamente.



País possui no máximo Várias (mais de uma) UF

Juntando as duas cardinalidade temos o modelo lógico abaixo:



País pertence no mínimo a ZERO UF e no máximo a VÁRIOS UF

UF pertence no máximo e no mínimo a UM País.

Agora vamos definir os tipos de cardinalidade quanto ao relacionamento:

Cardinalidade **UM para UM** :

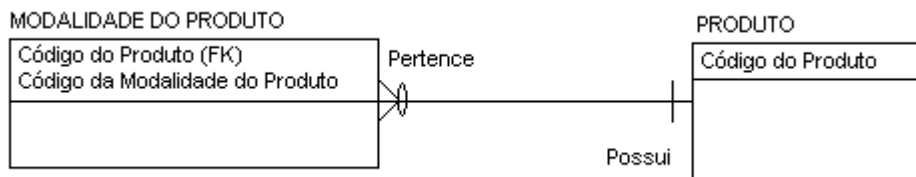


PESSOA pode ser no mínimo um CLIENTE. (opcional)

CLIENTE É uma PESSOA.(Obrigatório)

Nota: No relacionamento Um para Um temos o lado opcional e o lado obrigatório . A chave primária se desloca em direção ao lado opcional. No exemplo acima o descolamento seria da entidade CLIENTE para a entidade PESSOA.

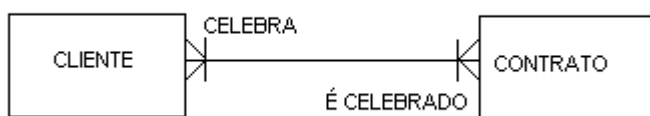
Cardinalidade **UM para N**.



PRODUTO possui nenhum ou muitas modalidade de produto
MODALIDADE DE PRODUTO pertence a um produto.

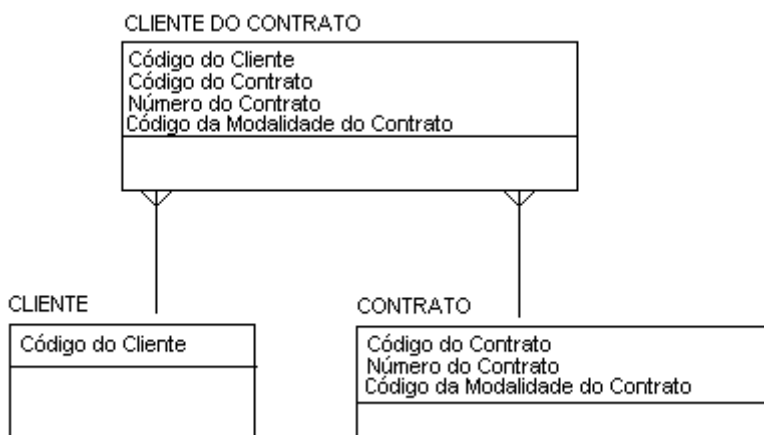
Nota : A cardinalidade UM para N leva a chave primária do lado UM para o lado N. Neste caso o atributo recebe o nome de chave estrangeira ou **Foreign Key** (FK). Chave Estrangeira é a chave primária de uma entidade que aparece em outra entidade em virtude do relacionamento.

Cardinalidade **N para N**.



CLIENTE celebra um ou vários Contratos
 CONTRATO é celebrado por um ou vários clientes

A cardinalidade N para N leva para o modelo lógico a necessidade de definição de mais um entidade. Chamamos isto de **ASSOCIATIVA**. Para o exemplo acima teríamos:



A Entidade CLIENTE DO CONTRATO é necessária para que possamos identificar o contrato de um determinado cliente. Em toda Cardinalidade N para N temos a ASSOCIATIVA.

Normalização

Normalização é o conjunto de regras que visa minimizar as anomalias de modificação dos dados e dar maior flexibilidade em sua utilização.

Por que Normalizar ?

- 1º) Minimização de redundâncias e inconsistências;
- 2º) Facilidade de manipulações do Banco de Dados;
- 3º) Facilidade de manutenção do Sistema de Informações

Para que você compreenda melhor vou dar um exemplo. Vamos supor que você criou uma entidade **Funcionários** para armazenar as informações dos funcionários de um empresa e que o resultado físico final seja a tabela mostrada abaixo.

Funcionarios : Tabela					
	Codigo	Nome	Cargo	Setor	QuantidadeFuncionarios
	1	Miriam	Gerente	Vendas	23
	2	Jefferson	Programador	Suporte	20
	3	Jessica	Analista	Compras	15
	4	Janice	Programadora	Suporte	16
	5	Mario	Gerente	Design	9

Se você olhar bem para a tabela acima vai ter que concordar comigo que ele sofre das seguintes anomalias:

- **Anomalia de Exclusão** - O que acontece se você excluir o funcionário de código igual a 3 ? O Setor vai ser excluído junto e ai você dançou..

Anomalia de Alteração - O nome do Setor Suporte mudou para Apoio . Você vai ter alterar o nome em todos os registros da tabela. Dançou novamente...

- **Anomalia de Inclusão** - Foi contratado um novo funcionário para o Setor Suporte. Você vai ter que incluir um funcionário ao campo - QuantidadeFuncionarios - em todas as ocorrências com setor de nome SUPORTE. Dançou mais uma vez...

Para poder resolver o dilema acima temos que **NORMALIZAR** a entidade. Para isto aplicamos as **formas normais** a saber:

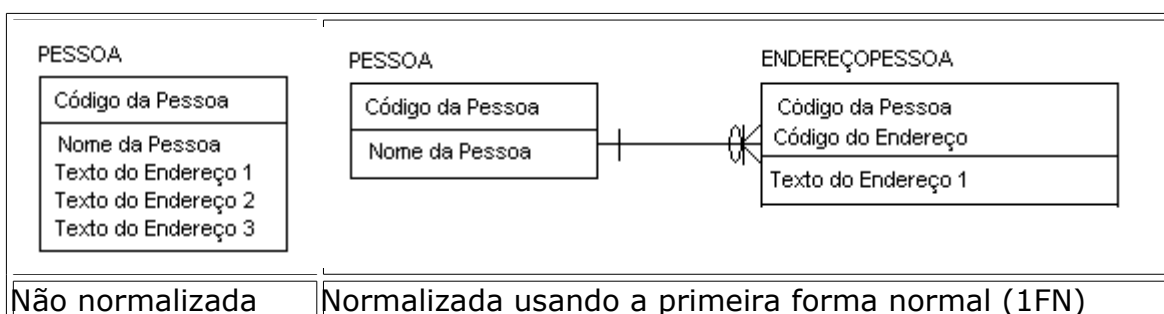
1- **Primeira Forma Normal -(1FN)-** Uma relação está na 1FN se somente todos os domínios básicos contiverem somente valores atômicos (não contiver grupos repetitivos). Para atingir esta forma normal devemos eliminar os grupos de repetição. Como ?0

Procedimentos:

- Identificar a chave primária da entidade;
- Identificar o grupo repetitivo e excluí-lo da entidade;
- Criar uma nova entidade com a chave primária da entidade anterior e o grupo repetitivo.

A chave primária da nova entidade será obtida pela concatenação da chave primária da entidade inicial e a do grupo repetitivo.

Abaixo temos um exemplo de como efetuar a normalização para a primeira forma normal:



2- **Segunda Forma Normal -(2FN)-** Uma relação R está na 2FN se e somente se ela estiver na primeira e todos os atributos não chave forem totalmente dependentes da chave primária (dependente de toda a chave e não apenas de parte dela).

Procedimentos:

- Identificar os atributos que não são funcionalmente dependentes de toda a chave primária.
- Remover da entidade todos esses atributos identificados e criar uma nova entidade com eles.

A chave primária da nova entidade será o atributo do qual os atributos do qual os atributos removidos são funcionalmente dependentes.

Exemplo:

Sejam as entidades :

Arquivo de Notas Fiscais (Num. NF, Série, Código do Cliente, Nome do cliente, Endereço do cliente, Total Geral da Nota)

Arquivo de Vendas (Num. NF, Código da Mercadoria, Descrição da Mercadoria, Quantidade vendida, Preço de venda e Total da venda)

NotaFiscal : Tabela		
	Nome do campo	Tipo de dados
?	NumeroNotaFiscal	Número
	Serie	Texto
	CodigoCliente	Número
	NomeCliente	Texto
	EnderecoCliente	Texto
	Total	Texto

Vendas : Tabela		
	Nome do campo	Tipo de dados
?	NumeroNotaFiscal	Número
?	CodigoMercadoria	Número
	Descricao	Texto
	QuantidadeVendida	Número
	PrecoVenda	Moeda
	Total	Moeda

Normalizando para segunda forma normal (2FN):

Arquivo de Notas Fiscais (Num. NF, Série, Código do Cliente, Nome do cliente, Endereço do cliente, Total Geral da Nota)

Arquivo de Vendas (Num. NF, Código da Mercadoria, Quantidade vendida e Total da Venda)

Arquivo de Mercadorias (Código da Mercadoria, Descrição da Mercadoria, Preço de venda)

NotaFiscal : Tabela		
	Nome do campo	Tipo de dados
?	NumeroNotaFiscal	Número
	Serie	Texto
	CodigoCliente	Número
	NomeCliente	Texto
	EnderecoCliente	Texto
	Total	Texto

Vendas : Tabela		
	Nome do campo	Tipo de dados
?	NumeroNotaFiscal	Número
?	CodigoMercadoria	Número
	QuantidadeVendida	Número
	Total	Moeda

Mercadorias : Tabela		
	Nome do campo	Tipo de dados
?	CodigoMercadoria	Número
	Descricao	Texto
	PrecoVenda	Moeda

Como resultado desta etapa, houve um desdobramento do arquivo de **Vendas** (o arquivo de Notas Fiscais, não foi alterado, por não possuir chave composta) em duas estruturas a saber:

Primeira estrutura (**Arquivo de Vendas**): Contém os elementos originais, sendo excluídos os dados que são dependentes apenas do campo Código da Mercadoria.

Segundo estrutura (**Arquivo de Mercadorias**): Contém os elementos que são identificados apenas pelo Código da Mercadoria, ou seja, independentemente da Nota Fiscal, a descrição e o preço de venda serão constantes.

3- **Terceira Forma Normal** -(2FN)- Uma relação R está na 3FN se somente estiver na 2FN e todos os atributos não chave forem dependentes não transitivos da chave primária (cada atributo for funcionalmente dependente apenas dos atributos componentes da chave primária ou se todos os seus atributos não chave forem independentes entre si).

Procedimentos:

- Identificar todos os atributos que são funcionalmente dependentes de outros atributos não chave;
- Removê-los e criar uma nova entidade com os mesmos.

A chave primária da nova entidade será o atributo do qual os atributos removidos são funcionalmente dependentes.

Estrutura na segunda forma normal (2FN):

Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente, Nome do cliente, Endereço do cliente, Total Geral da Nota)

Arquivo de Vendas (Num. NF, Código da Mercadoria, Quantidade vendida e Total da venda desta mercadoria)

Arquivo de Mercadorias (Código da Mercadoria, Descrição da Mercadoria, Preço de venda)

Estrutura na terceira forma normal (3FN):

Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente e Total Geral da Nota)

Arquivo de Vendas (Num. NF, Código da Mercadoria, Quantidade vendida e Total da venda desta mercadoria)

Arquivo de Mercadorias (Código da Mercadoria, Descrição da Mercadoria, Preço de venda)

Arquivo de Clientes (Código do Cliente, Nome do cliente, Endereço do cliente)

Como resultado desta etapa, houve um desdobramento do arquivo de Notas Fiscais, por ser o único que possuía campos que não eram dependentes da chave principal (Num. NF), uma vez que independente da Nota Fiscal, o Nome, Endereço são inalterados. Este procedimento permite evitar inconsistência nos dados dos arquivos e economizar espaço por eliminar o armazenamento freqüente e repetidas vezes destes dados. A cada nota fiscal comprada pelo cliente, haverá o armazenamento destes dados e poderá ocorrer divergência entre eles.

As estruturas alteradas e o motivo das alterações :

- Primeira estrutura (**Arquivo de Notas Fiscais**): Contém os elementos originais, sendo excluído os dados que são dependentes apenas do campo Código do Cliente (informações referentes ao cliente).
- Segundo estrutura (**Arquivo de Clientes**): Contém os elementos que são identificados apenas pelo Código do Cliente, ou seja, independente da Nota Fiscal, o Nome, Endereço serão constantes.

Após a normalização, as estruturas dos dados estão projetadas para eliminar as inconsistências e redundâncias dos dados, eliminando desta forma qualquer problema de atualização e operacionalização do sistema. A versão final dos dados poderá sofrer alguma alteração, para atender as necessidades específicas do sistema, a critério do analista de desenvolvimento durante o projeto físico do sistema

Exercícios sobre modelagem de dados

Vejamos a seguir alguns exercícios práticos onde iremos colocar toda esta teoria para funcionar em casos quase reais.

Exercício

1

Vou começar com um bem simples : De acordo com as regras , normalize as estruturas abaixo.

– • **Relação de Funcionário:**

- -MATRÍCULA DO FUNCIONÁRIO
- -NOME DO FUNCIONÁRIO
- -DATA DO NASCIMENTO
- **-DEPENDENTE**
 - -CÓDIGO DO DEPENDENTE
 - -NOME DO DEPENDENTE
- **-CURSO**
 - -CÓDIGO DO CURSO
 - -NOME DO CURSO
 - -ANO DO CURSO

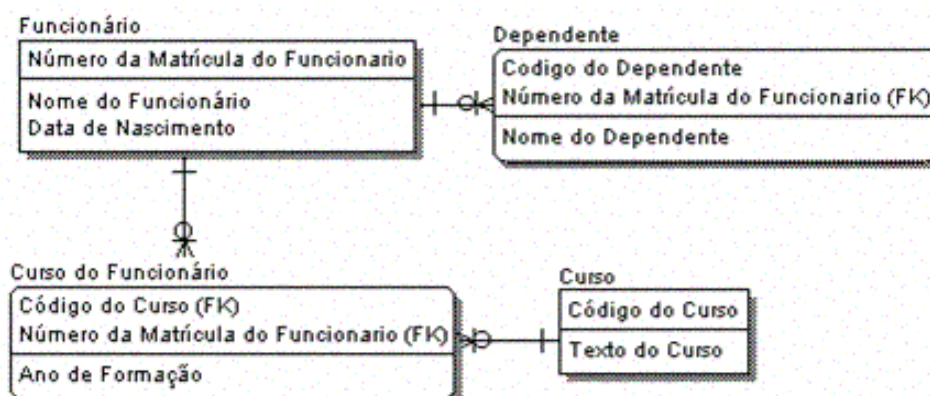
–

Regras do negócio :

- Um funcionário pode ter mais de um dependente
- Um funcionário pode fazer mais de um curso

Esta é simples. Veja abaixo uma das soluções:

Resposta:



–

Exercício

2:

Você acabou de fundar sua empresa de consultoria , a **Maicrousofiti** Consultoria , e seu primeiro trabalho e desenvolver um sistema para cadastro de clientes você recebeu o cliente uma lista com os dados que deverão compor o sistema , com base nesta lista normalize a estrutura de dados de acordo com as formas normais.

Lista de informações que deverão compor o sistema cadastro de clientes:

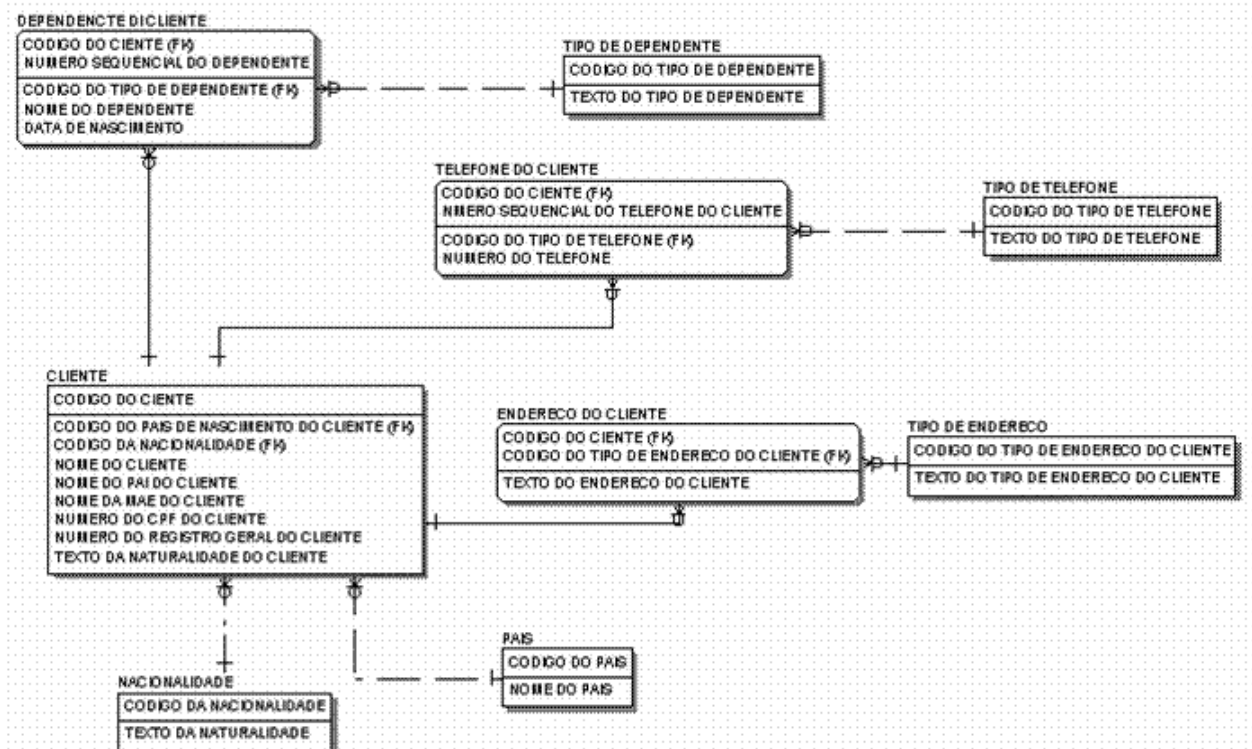
Nome

Nome do Pai
 Nome da Mãe
 Endereço
 Telefone1
 Telefone2
 Número do Fax
 Número do Celular
 Telefone do trabalho
 Data de Nascimento
 Naturalidade
 Nacionalidade
 Endereço de correspondência
 Nome do filho 1
 idade do filho 1
 Nome do filho 2
 idade do filho 2
 Nome do filho 3
 idade do filho 3
 Nome do filho 4
 idade do filho 4
 Nome do Cônjuge
 Número do CPF
 Número da carteira de identidade

Antes de ver a resposta sugiro que pelo menos tente esboçar alguma tentativa de resolver a questão.

Que tal deixar a solução para um próximo artigo ? Brincadeira ! Uma das possíveis soluções é dada abaixo:

Resposta



Exercício

3

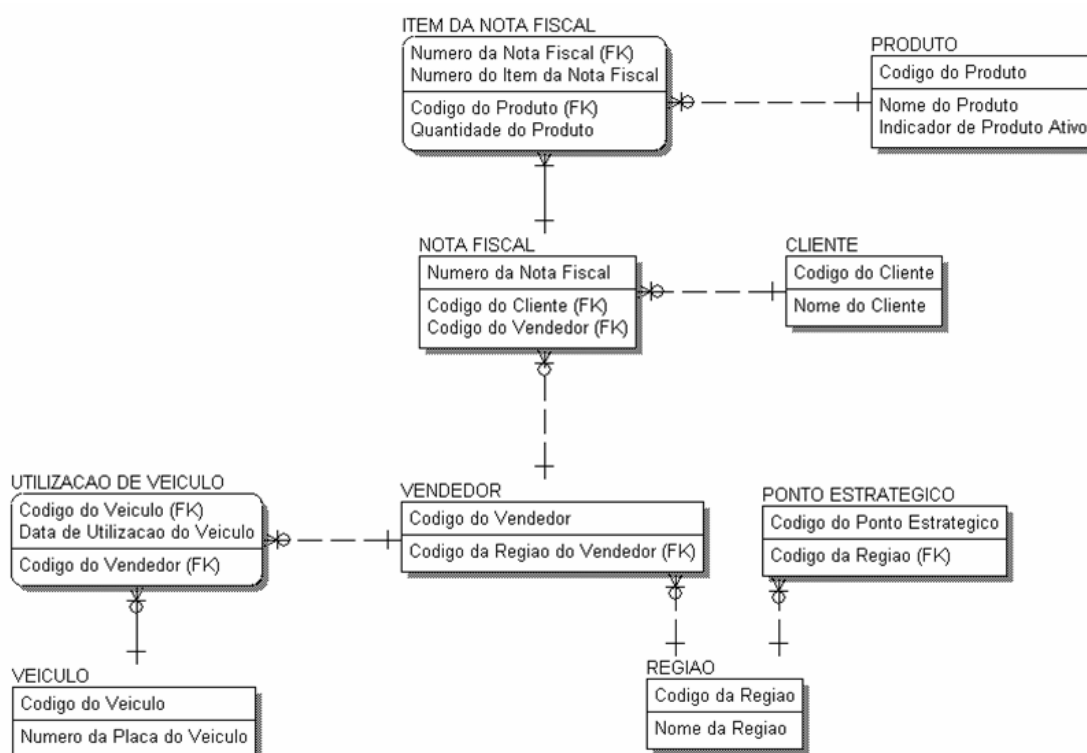
Para deixar você ainda mais afiado vamos a outra situação.

Você deve representar usando o modelo lógico a situação descrita a seguir:

- O Departamento de Vendas da Indústria Beleza Ltda, após estudos de mercado, verificou que para atingir seus objetivos seria necessário adquirir frota de veículos próprios para motorizar seus vendedores. O mercado consumidor foi dividido em regiões de venda; foram estabelecidos percursos de entrega abrangendo pontos estratégicos dessas regiões e vendedores foram designados para cobrir estes percursos. Um sistema deve ser construído para administração da nova sistemática de vendas adotada pela empresa. Após entrevistas com o gerente da área, foram obtidas as seguintes informações:
 - - cada região é identificada por um código;
 - - uma região é composta de vários pontos estratégicos;
 - - as regiões não têm pontos estratégicos em comum;
 - - o vendedor tem a responsabilidade de cobrir uma região;
 - - uma região pode ser coberta por vários vendedores;
 - - a cada dia, um veículo fica sob responsabilidade de um vendedor;
 - - um vendedor pode vender quaisquer itens ativos da tabela de produtos;
 - - o vendedor é responsável pela identificação de cada cliente consumidor na nota fiscal;
 - - a nota fiscal contendo identificação do vendedor, itens e quantidades vendidas é exigida para comprovação da venda

E então ? Quer ver como seria uma das soluções? Então espia...

Resposta



Para encerrar aqui vai a última questão.

Exercicio

4:

De acordo com as regras , normalize as estruturas abaixo.

- **Relação de Programadores:**

- Numero da Matrícula
- Nome do Programador
- Setor
- Nível (1,2,3)
- Descrição do Nível (1 - Júnior, 2 - Pleno, 3 - Senior)

- **Programas**

- Codigo do Programa
- Nome do Programa
- Tempo Estimado
- Nível de Dificuldade (1, 2 ou 3)
- Descrição da Dificuldade (Fácil, Médio, Difícil)

Regras

do

negócio:

- Um programa pode ser feito por mais de um Programador;
- Um programador pode fazer um ou mais programas;
- O Nível de dificuldade do programa depende do tempo estimado para a elaboração do mesmo;

Resposta:

