



Conceitos e Aplicações Básicas para Modelagem de Dados

Por
Alvaro Caetano Pimentel Sobrinho

Resumo

Este manual é uma continuação da proposta de discussão de acesso aberto, iniciada no artigo “**Open Access: Ser ou não ser?**”, que está disponível no endereço <http://artigocientifico.tebas.kinghost.net/pesquisadores/?mnu=2&smnu=5&id=29536>, e corresponde à segunda parte do material que foi disponibilizado primeiramente no manual “**Free Access: O MySQL na prática**”, também no mesmo endereço acima, o qual apresenta a implementação de uma base de dados MySQL. Neste manual serão apresentados os métodos básicos para a modelagem de dados que incluem tópicos de abstração, modelagem lógica, física e normalização. Além disso, é composto de uma série de exercícios resolvidos que podem ser utilizados para treinar, e serem discutidos, a cada parte avançada. A última parte da proposta de disseminação de conhecimentos será a apresentação de uma apostila de uso das ferramentas CASE **MySQL Workbench** e **MySQL Query Browser**.

Palavras Chave: Modelagem de dados, Normalização, Abstração

Abstract

This manual is a continuation of proposal open access discussion that began in the article “**Open Access: To be or not?**”, and it corresponds to the second part of the material that is available first in the manual “**Free Access: The MySQL in the practice**” which presents the construction of a MySQL database. In this manual will be presented the basic methods for the data modeling that include abstraction topics, logical\physical modeling and normalization. Moreover, is composed of a series of exercises that can be used to train, and to be argued, to each advanced part. The last part of the proposal of dissemination of knowledge will be the presentation of use of the tools CASE **MySQL Workbench** and **MySQL Query Browser**.

Key Words: Data modeling, Normalization, Abstraction

Tópicos	
1.	Introdução 5
2.	Conceitos Básicos 5
2.1	Bases de Dados 5
2.2	Sistemas de Gerenciamento de Bases de Dados 5
2.3	Sistemas de Bases de Dados 5
2.4	Objetivos de um Sistema de Bases de Dados 6
3.	Abstração de Dados 7
4.	Modelos de Dados 7
4.1	Modelo Conceitual/Lógico de Dados 7
4.1.1	Obtenção de um Modelo de Dados 7
4.1.2	Modelo de Dados Baseado em Objetos 9
4.1.3	Modelo de Dados Baseado em Registros 9
4.1.3.1	Modelo de Dados Relacional 9
4.1.3.2	Modelo de Dados em Rede 10
4.1.3.3	Modelo de Dados Hierárquico 10
5.	Definição e Manipulação de Dados 11
5.1	DDL 11
5.2	DML 11
5.3	DLL 11
6.	Funções em Sistemas de Bancos de Dados 12
6.1	Usuários 12
6.2	Administrador 12
6.2.1	Administração Dados (AD) 12
6.2.2	Administração de Bases de Dados (ABD) 12
7.	Modelo Entidade Relacionamento (MER) 13
7.1	Definição 13
7.2	Objetivo 13
7.3	Fases do Projeto de Bases de Dados 13
7.3.1	CDM – Conceptual Data Model 13
7.3.2	LDM – Logical Data Model 13
7.3.3	PDM – Physical Data Model 13

8.	Entidades e Conjuntos-Entidade	13
	8.1 Componentes do Diagrama E-R (Peter Chen):	13
9.	Atributos	14
	9.1 Domínio de um Atributo	14
10.	Relacionamentos	15
11.	Restrições de Mapeamento (Cardinalidade)	15
	11.1 Um-para-um	15
	11.2 Um-para-muitos	16
	11.3 Muitos-para-muitos	16
12.	Exercícios	17
13.	Projeto de Chaves	19
14.	Auto-Relacionamento	19
15.	Agregação	20
16.	Generalização – Especialização	21
	16.1 Total e Parcial	21
	16.2 Não-Exclusiva	21
	16.3 Múltipla	22
17.	Relacionamentos de Grau Superior a 2	22
18.	Dependência Existencial e Entidades Fracas	23
19.	Exercícios	25
20.	Normalização	27
	20.1 Primeira Forma Normal (1FN)	28
	20.2 Segunda Forma Normal (2FN)	28
	20.3 Terceira Forma Normal (3FN)	30
21.	Exercícios	31
22.	Soluções	
	22.1 Soluções dos exercícios 12.1 a 12.7	33
	22.2 Soluções dos exercícios 19.1 a 19.8	35
	22.3 Soluções dos exercícios 21.1 a 21.4	39

Introdução

A importância da informação para a tomada de decisões nas organizações tem impulsionado o desenvolvimento dos sistemas de processamento de informações.

A definição de informação pode ser dada, restringindo-se o seu conceito à informática, como um dado que possui “unicidade”, “atomicidade” e que já tenha sido tratado, de tal maneira que agregue significante e significado representando o objeto de origem, real ou abstrato. Vale lembrar que a Ciência da Informação possui uma definição mais abrangente de informação, mas para o estudo de bases de dados, neste nível, essa mais restrita é absolutamente satisfatória.

Algumas ferramentas pertinentes ao tratamento da informação:

- processadores de texto (editoração eletrônica),
- planilhas (cálculos com tabelas de valores),
- Sistemas de Gerenciamento de Bancos de Dados - SGBD

1. Conceitos Básicos

2.1 Bases de Dados

Uma base de dados pode ser definida como uma coleção de dados inter-relacionados representando informações sobre um domínio específico.

Exemplos: lista telefônica, controle do acervo de uma biblioteca, sistema de controle dos recursos humanos de uma empresa.

2.2 Sistemas de Gerenciamento de Bases de Dados - SGBD

O SGBD, do inglês *Database Management System* – DBMS – É um *software* com recursos específicos para facilitar a manipulação das informações dos bancos de dados e o desenvolvimento de programas aplicativos. Além disso permite o armazenamento de grandes volumes de dados, estruturados em registros e tabelas, com recursos para acesso e processamento das informações.

Exemplos: Oracle, MySQL, SQL Server, InterBase, Ingres, Paradox*, Access*, DBase*.

* *Desktop Database Management Systems*

2.3 Sistemas de Bases de Dados

É um sistema de manutenção de registros por computador, envolvendo quatro componentes principais:

- dados,
- hardware,
- software e
- usuários.

O sistema de bancos de dados pode ser considerado como uma sala de arquivos eletrônica. Existe uma série de métodos, técnicas e ferramentas que visam sistematizar o desenvolvimento de sistemas de bancos de dados.

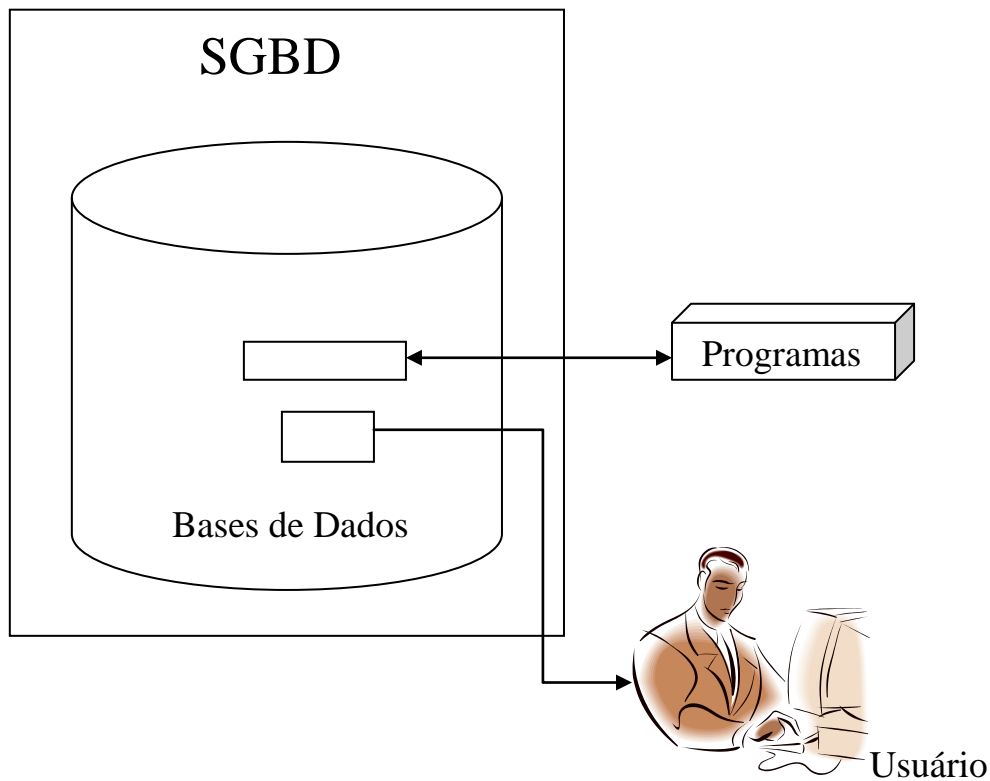
2.4 Objetivos de um Sistema de Bases de Dados

- Isolar os usuários dos detalhes mais internos do banco de dados (abstração de dados).
- Prover independência de dados às aplicações (estrutura física de armazenamento e à estratégia de acesso).

Vantagens:

- rapidez na manipulação e no acesso à informação,
- redução do esforço humano (desenvolvimento e utilização),
- disponibilização da informação no tempo necessário,
- controle integrado de informações distribuídas fisicamente,
- redução de redundância e de inconsistência de informações,
- compartilhamento de dados,
- aplicação automática de restrições de segurança,
- redução de problemas de integridade

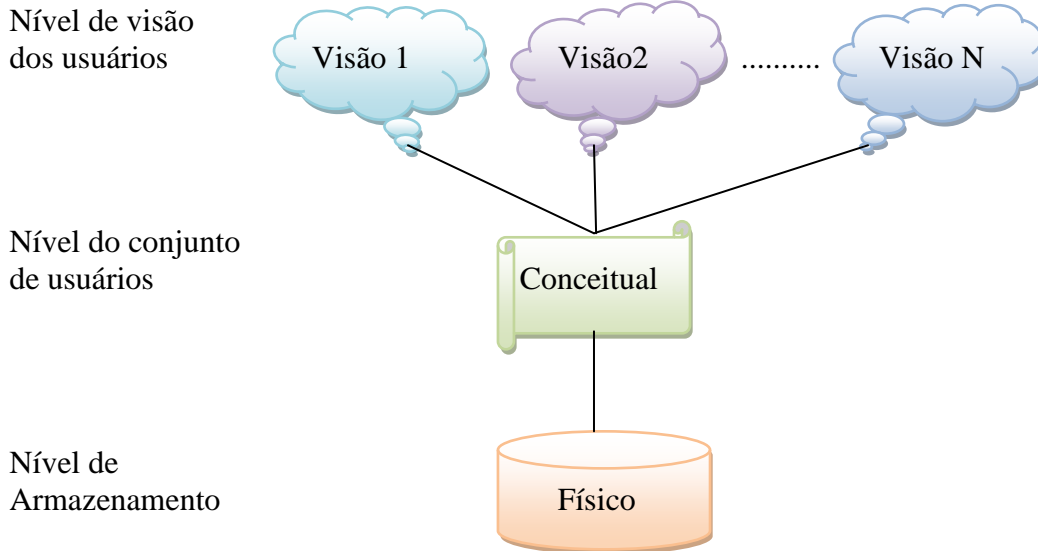
2.5 Componentes de um Sistema de Bases de Dados



2. Abstração de Dados

O sistema de bancos de dados deve fornecer uma visão abstrata de dados para os usuários.

A abstração se dá em três níveis:



Nível de visões do usuário: descreve partes do banco de dados, de acordo com as necessidades de cada usuário, individualmente.

Nível Conceitual/Lógico: descreve quais dados estão/serão armazenados e seus relacionamentos. Neste nível, o banco de dados é descrito através de estruturas relativamente simples, que podem envolver estruturas complexas no nível físico.

Nível Físico: nível mais baixo de abstração. Descreve como os dados estão realmente armazenados, englobando estruturas complexas de baixo nível.

3. Modelo de Dados

É a representação gráfica e textual de ESTRUTURAS, OPERADORES e das REGRAS, que definem os dados, assim como as restrições de consistência e integridade

Estruturas

São formadas de Regras (asserções que regulam o funcionamento da estrutura) e Operadores que são comandos que permitem a manipulação das estruturas segundo um comportamento estabelecido.

4.1 Modelo Conceitual/Lógico de Dados (CDM/LDM)

É uma atividade desenvolvida em fases variadas do processo metodológico de desenvolvimento de sistemas, com a finalidade de levantar informações (REQUISITOS FUNCIONAIS) para a obtenção do modelo de dados.

4.1.1 Obtenção de um Modelo de Dados

Decomposição funcional;
Análise de eventos;
Análise das regras de negócio



Decomposição funcional (Análise da hierarquia de Funções-Processos-Atividades)

Identificação de entidades externas ou de depósitos de dados relacionados a cada processo que sugere entidades de dados que irão compor o modelo (conceito de EVENTO trazido pela Análise Essencial)

Análise de eventos – são fatos que provocam estímulos para os quais os sistemas deverão conter processos de reposta.

Análise das regras de negócio – É o estudo de análise dos dados, processos, objetos e suas correlações. Divide-se em:

Fatos – asserção que estabelece que um objeto possua determinadas propriedades ou desempenha determinado papel.

Ex.: `CLIENTE emite FATURA`
`CLIENTE possui endereço`
`CX-INFO atua em New York (sujeito verbo objeto)`

Restrições – são limitações colocadas aos fatos para a diminuição de ocorrências válidas naquele universo

Ex.: `Um PEDIDO deve ser colocado para um CLIENTE cadastrado`

Regras de derivação – são condições colocadas sobre fatos e que podem gerar outros fatos.

Ex.: `Se um PEDIDO for efetuado até o dia X ele será processado no mês seguinte.`

Entidade – Representação abstrata de um objeto do mundo real (PESSOAS LUGARES COISAS), e podem ser identificadas, por similaridade, com análise sintática nas linguagens naturais:

Retângulo – Cj-Entidade (sujeito / objeto da sentença)

Elipse – Atributo (adjetivo)

Losango – Cj-Relacionamento (verbo)

Linha – Entidade com Relacionamento e Atributo a Entidade (advérbios e adjuntos)

Conjunto-Entidade – Grupo de entidades que possuem características semelhantes

Ex.: `Conjunto-entidade FORNECEDOR`

Relacionamento – Estrutura que indica a associação de elementos de duas ou mais entidades.

Atributos de relacionamento – Depende de todos os conjuntos-entidade associados entre si.

Cardinalidade – Restrições de Mapeamento

Um-para-um (1:1): Entidade em A associada a uma, e somente uma, entidade em B

Um-para-muitos (1:N): Entidade em A associada a qualquer nº de entidades em B

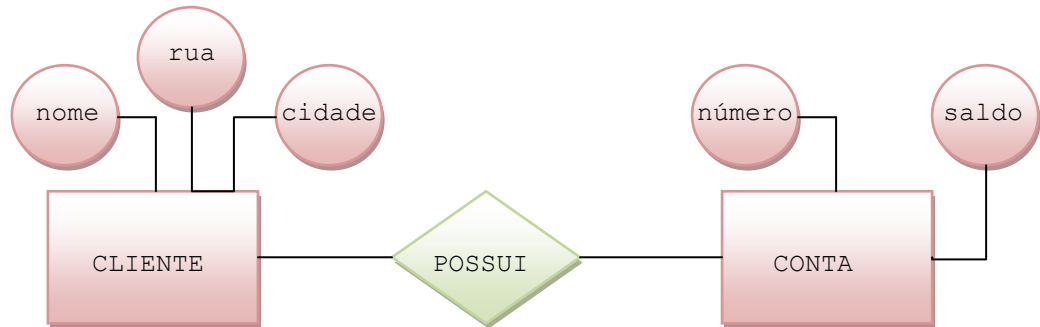
Muitos-para-muitos (N:N): Entidade em A associada a qualquer nº de entidades em B e vice-versa.



4.1.2 Modelo de Dados Baseado em Objetos

Define a descrição dos dados nos níveis conceitual/lógico e de visões de usuários. No modelo orientado a objetos o código executável é parte integrante do modelo de dados.

Ex.: Entidade-Relacionamento.



4.1.3 Modelo de Dados Baseado em Registros

- Define a descrição dos dados nos níveis conceitual/lógico e de visões de usuários;
- O banco de dados é estruturado em registros de formatos fixos, de diversos tipos;
- Cada tipo de registro tem sua coleção de atributos;
- Há linguagens específicas para expressar consultas e atualizações no banco de dados.

Ex.: Modelo Relacional,
Modelo Hierárquico.

4.1.3.1 Modelo de Dados Relacional

No Modelo Relacional, dados e relacionamentos entre dados são representados por tabelas, cada uma com suas colunas específicas. Tanto os dados quanto os relacionamentos são representados por tabelas.

Prescinde de estruturas de índice eficientes e hardware adequado para alcançar desempenho viável em situações práticas.

Ex.: Modelo Relacional

Tabela Cliente

cód -cliente	nome	rua	cidade
015	Alvaro	Laura Teles	Rio de Janeiro
021	Mário	Laranjeiras	Rio de Janeiro
037	Giselle	Ipê	Campos do Jordão

Tabela Conta

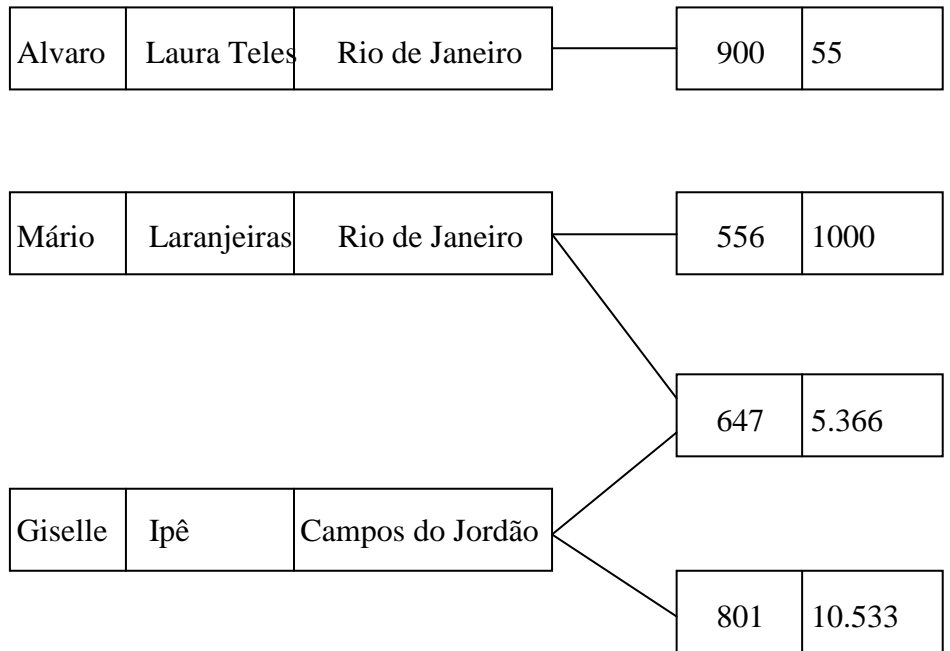
nro-conta	saldo
900	55
556	1.000
647	5.366
801	10.533

Tabela Cliente-Conta

cód -cliente	nro-conta
015	900
021	556
021	647
037	801
037	647

4.1.3.2 Modelo de Dados em Rede

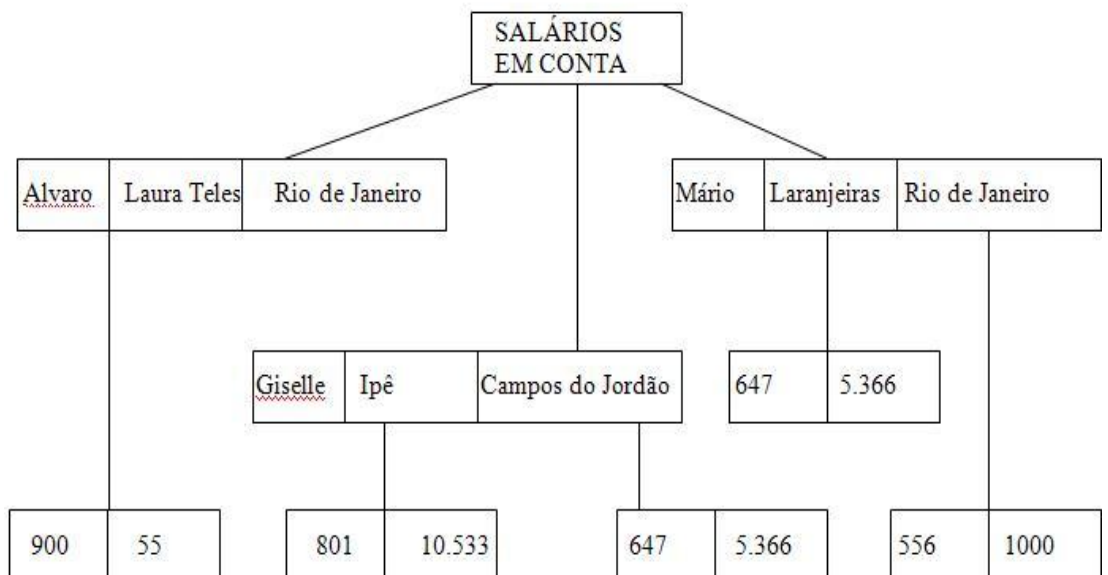
Os dados são representados por coleções de registros e os relacionamentos por elos.



4.1.3.3 Modelo de Dados Hierárquico

Os dados e relacionamentos são representados por registros e ligações, respectivamente.

Os registros são organizados como coleções arbitrárias de árvores.



5 Definição e Manipulação de Dados

Os bancos de dados possuem uma estrutura que armazena as informações pertinentes a tabelas, views, registros, definições de dados e atributos. Apesar de possuírem nomes ou identificações diferentes, todo SGBD possui um esquema e um subesquema.

Esquema do Banco de Dados

O esquema armazena o “projeto geral” (estrutura) do banco de dados e possui as seguintes características:

- não muda com frequência;
- há um esquema para cada nível de abstração e um subesquema para cada visão de usuário.

A construção de um esquema acontece através de uma linguagem apropriada para a transformação de um modelo conceitual/lógico em um modelo físico. A transcrição ocorre por meio de uma ferramenta CASE – *Computer Aided Software Engineer* (algo como: linguagem de construção para auxiliar à computação). Esta linguagem é conhecida como DDL – *Data Definition Language*.

5.1 DDL – Linguagem de Definição de Dados

Permite especificar o esquema do banco de dados, através de um conjunto de definições de dados. A compilação dos comandos em DDL é armazenada no dicionário (ou diretório) de dados. Neste ponto são criados os metadados que irão compor o esquema do projeto que estiverem em transcrição.

Obs.: Metadados, por definição, são dados que definem outros dados. Assim os comandos compilados são metadados dos comandos originais.

5.2 DML – Linguagem de Manipulação de dados

A manipulação dos dados em uma base acontece através de:

- recuperação da informação armazenada,
- inserção de novas informações,
- exclusão de informações,
- modificação de dados armazenados.

Para que os eventos acima sejam possíveis é necessário utilizar uma linguagem apropriada para interagir com a base de tal maneira que, através da leitura do esquema, consiga entender o dado e a sua posição na tabela. Esta linguagem é conhecida como DML – *Data Manipulation Language* – Linguagem de Manipulação de Dados que permite ao usuário acessar ou manipular os dados, vendo-os da forma como foram definidos no nível de abstração mais alto do modelo de dados utilizado.

O processo de interação no DBMS é feito através de um conjunto de procedimentos chamado de *QUERY*. Portanto, uma consulta (“*query*”) é um conjunto de comandos que requisitam uma recuperação de informação na base de dados. Por essa razão, a parte de uma DML que envolve recuperação de informação é chamada linguagem de consulta.

5.3 DLL – Biblioteca de Comunicação Dinâmica

As *queries* podem ser construídas diretamente no DBMS, ou podem fazer uso de uma linguagem de alto nível (orientada a objeto ou não) para efetuar a interação com a base de dados. Neste caso, alguns programas fazem uso de uma DLL – *Dynamic Link Library* – Biblioteca de Comunicação Dinâmica, que é instalada no computador, e nada mais é do que códigos que vários aplicativos compartilham e possibilita significativa redução de memória do computador..

6 Funções em Sistemas de Bancos de Dados

Dentre as inúmeras funções humanas em relação ao uso de uma base de dados duas se destacam na interação com as bases.

6.1 Usuários

Realizam operações de manipulação de dados.

- programadores de aplicações,
- usuários sofisticados,
- usuários especializados,
- usuários “ingênuos”.

6.2 Administrador

Pessoa (ou grupo) responsável pelo controle do sistema de banco de dados.

- Administrador de Dados
- Administrador do SGBD

6.2.1 Administração de Dados (AD)

- definição e atualização do esquema do banco de dados.
- definição dos atributos e elementos de dados
- definição do CDM

6.2.2 Administrador de Bases de Dados (ABD)

- definição da estrutura de armazenamento e a estratégia (ou método) de acesso,
- concessão de autorização para acesso a dados,
- definição de controles de integridade,
- definição de estratégias para cópia de segurança e recuperação,
- monitoramento do desempenho,
- execução de rotinas de desempenho,
- modificação da organização física.

7 Modelo Entidade Relacionamento (MER)

7.1 Definição

Modelo baseado na percepção do mundo real, que consiste em um conjunto de objetos, básicos chamados entidades, e nos relacionamentos entre esses objetos.

Entidade: é uma representação abstrata de um objeto do mundo real.

Ex.: O fornecedor Pedro, com código F1

7.2 Objetivo

Facilitar o projeto de bases de dados possibilitando a especificação da estrutura lógica geral do banco de dados

7.3 Fases do Projeto de Bases de Dados

7.3.1 CDM – Conceptual Data Model

Representa o modelo de dados conceitual (2º nível de abstração) que será expresso através de um Diagrama Entidade-Relacionamento. Normalmente é desenvolvido dentro de uma ferramenta CASE (ERWIN, MySQL Workbench, Power Designer, Rose, etc.)

7.3.2 LDM – Logical Data Model

O modelo de dados lógico é a construção lógica das entidades, atributos e seus relacionamentos. Neste ponto surge a identificação da DDL que vai servir para a implementação física do modelo. A construção da DDL será automática, se o modelo estiver construído em uma ferramenta CASE. Em alguns SGBDs, tal como o DB2 em computadores de grande porte, a geração da DDL é construída manualmente por meio de utilitário próprio.

7.3.3 PDM – Physical Data Model

Ao se executar a DDL gerada no LDM, o SGBD cria fisicamente o modelo lógico elaborado do CDM. Essa construção é identificada como modelo de dados Físico

8 Entidades e Conjuntos-Entidade

A estrutura lógica geral de um banco de dados pode ser expressa graficamente por um Diagrama Entidade-Relacionamento

8.1 Componentes do Diagrama E-R (Peter Chen):

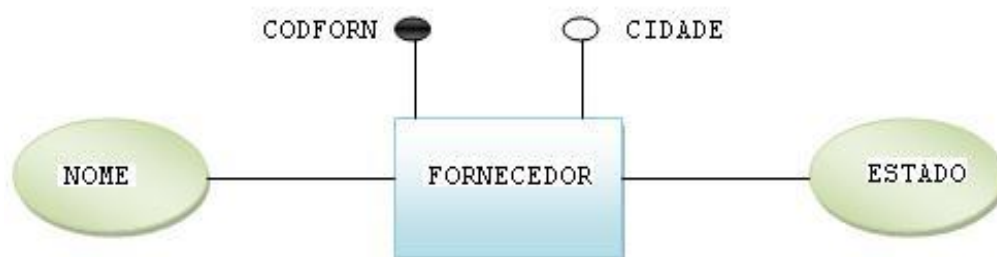
Retângulos: representam conjuntos-entidade

Elipses: representam atributos e quando estão em negrito, significam chave primária – *primary key* (pk) ou chave estrangeira – **foreign key** (fk)

Losangos: representam conjuntos-relacionamento

Triângulos: representam especializações ou generalizações de uma entidade

Linhas: ligam atributos a conjuntos-entidade e conjuntos-entidade a conjuntos-relacionamento



CODFORM foi escolhida como **PK** porque só há um código para identificar o fornecedor

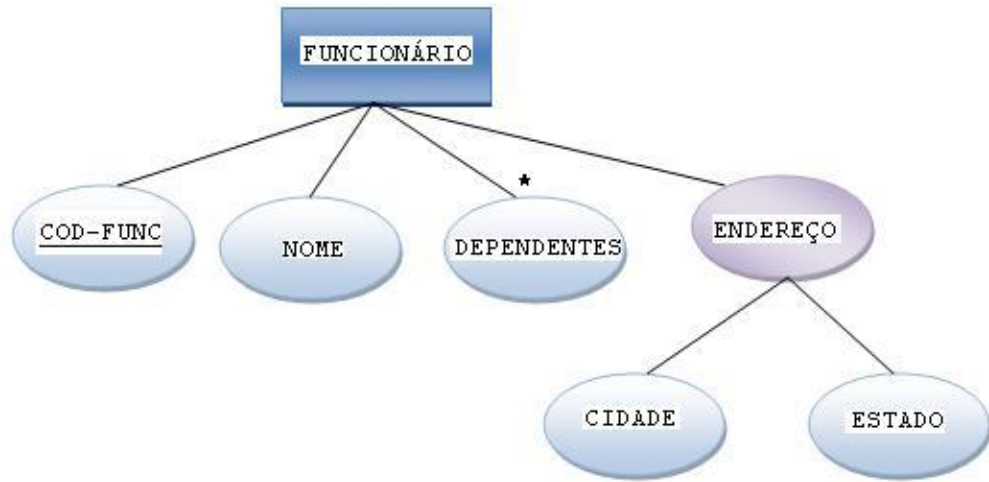
alvaro_pimentel@uol.com.br



9 Atributos

São elementos de dados que contêm as descrições que possibilitam definir uma entidade.

Ex.:



Obs: Nenhum modelo é suficientemente claro se não for acompanhado de uma definição formal dos elementos e isto é feito através do Dicionário de Dados. Ao se modelar deve-se sempre ter em mente que conceitos, aparentemente, triviais a quem está modelando podem não ser para outros profissionais. Assim, o dicionário de dados tem o objetivo definir as qualificações, compreender e unificar conceitos dos dados.

9.1 Domínio de um Atributo

Os atributos devem ser divididos em grupos de tal maneira que possibilite, já durante a sua definição, identificar fatores que vão auxiliar na definição da base de dados e no desenvolvimento das aplicações. Tomando como base o exemplo acima, foram definidos, a seguir, alguns grupos que permitem identificar os atributos:

Monovalorado: cada elemento do conjunto-entidade só pode possuir um único valor.

Ex.: Nome

Composto: diz-se do atributo que é formado por um ou mais sub-atributos.

Ex.: Endereço

No exemplo, poder-se-ia, ainda, dividir o endereço em complemento, comercial e residencial.

Multivalorado: é assim definido quando uma única entidade tem diversos valores para este atributo. Quando está representado no plural é facilmente identificado, porém, em e-mail ou telefone, por estarem no singular, pode criar dúvidas quanto à qualificação.

Ex.: Dependentes

Determinante: identifica cada entidade de um conjunto-entidade (também conhecido com atributo chave: matrícula, cpf, cnpj)

Ex.: Cod-Func

Domínio: refere-se a um conjunto de valores permitidos para o atributo como:

Ex.: Sexo {M, F} Estado civil {CASADO, SOLTEIRO, DIVORCIADO, VIÚVO, DESQUITADO}

10 Relacionamentos

Chama-se de relacionamento a estrutura que indica a associação de elementos de duas ou mais entidades.



Entretanto, em alguns casos, é necessário que os atributos estejam inseridos em uma entidade de **junção** (ou **associativa**) para que o relacionamento fique mais bem definido. Neste caso, o **atributo de relacionamento** é aquele que permite efetuar a associação dos conjuntos-entidade.

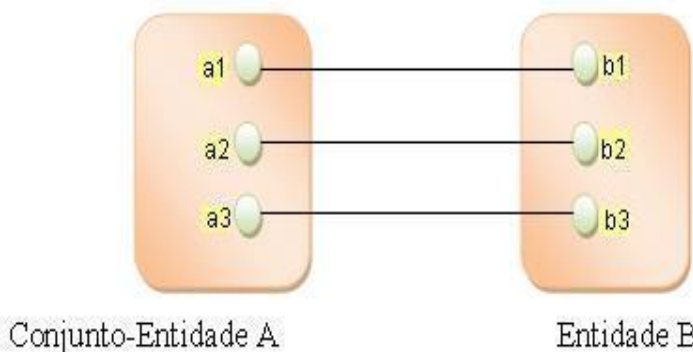


11 Restrições de Mapeamento (Cardinalidade)

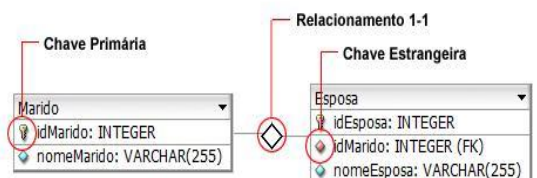
Os relacionamentos entre as entidades de um MER podem ser limitados através de valores mínimos e máximos que pode haver entre elas. Este limite é chamado de cardinalidade que é um número que expressa o número de ocorrências de determinada entidade em relação à outra.

A cardinalidade máxima, exprime o número máximo de ocorrências que uma entidade pode assumir ao se relacionar com outra. A cardinalidade mínima, determina o menor valor possível que pode haver no relacionamento das entidades. As possibilidades de relacionamentos podem ser infinitas, uma vez que os valores máximos e mínimos são expressos através de números cardinais. Porém, existem padrões que abrangem grande parte desses limites e são conhecidos como:

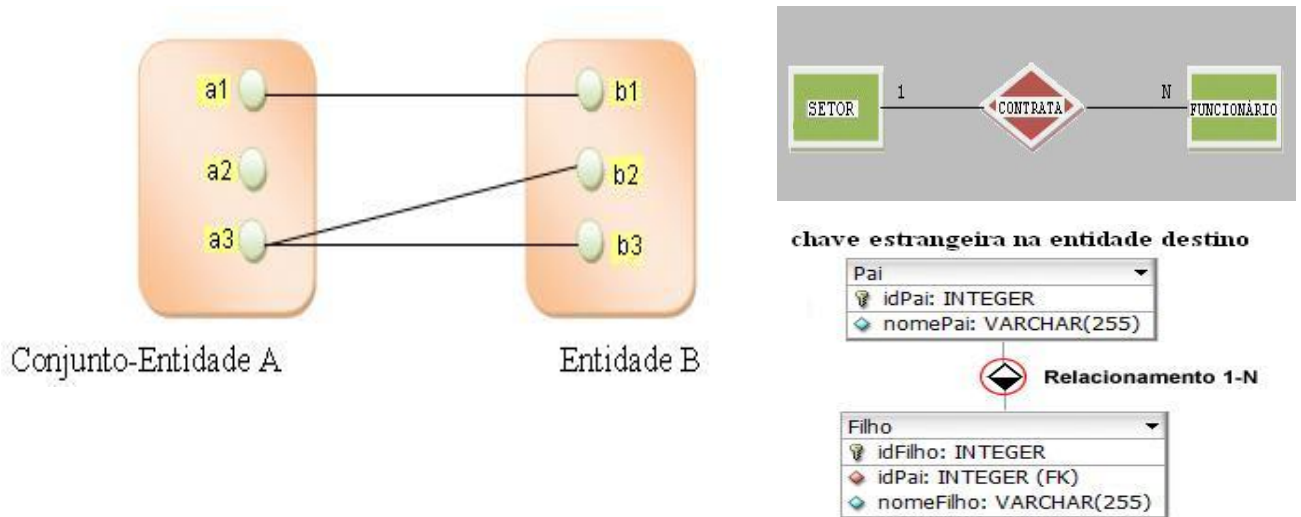
11.1 Um-para-um: um elemento de A está associado a somente um elemento da entidade B. Da mesma forma, uma ocorrência de B pode se associar a somente uma ocorrência em A



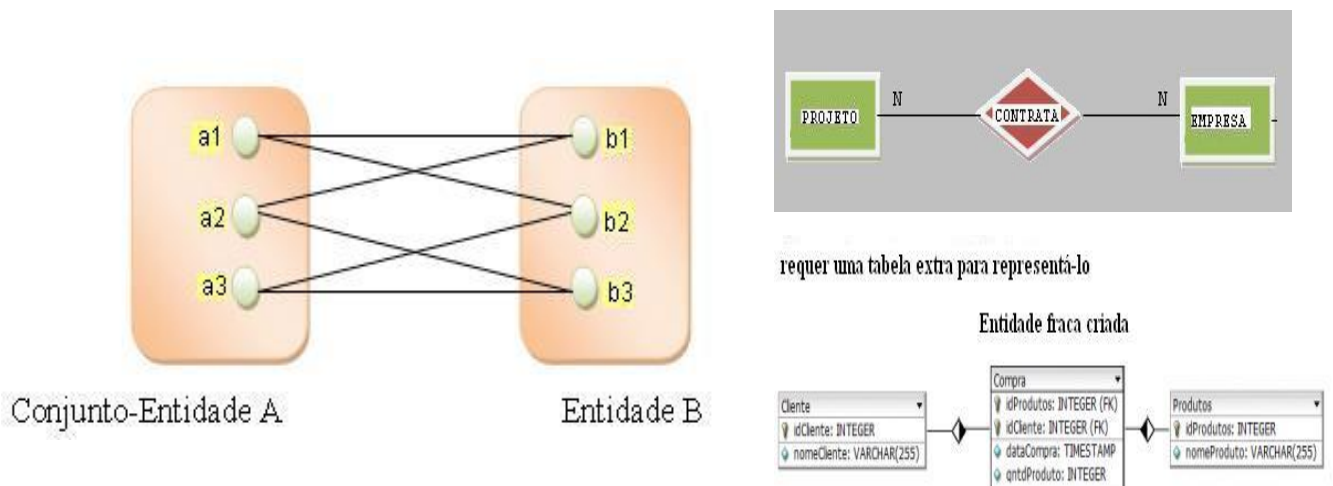
chave estrangeira em uma das entidades



11.2 Um-para-muitos: um elemento em A está associado a qualquer número de elementos em B, enquanto uma ocorrência em B está associada somente uma ocorrência em A.



11.3 Muitos-para-muitos: Um elemento em A está associado a qualquer número de elementos em B, e um elemento em B está associado a qualquer número de elementos em A.

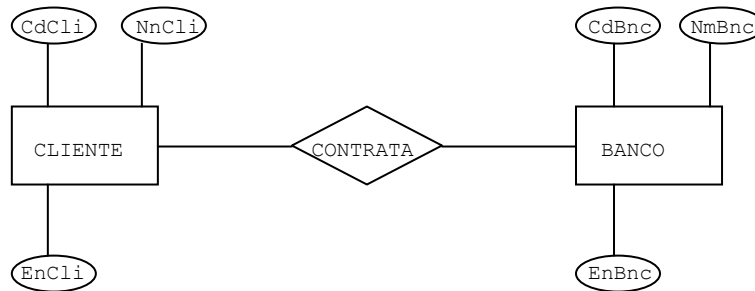


12 Exercícios

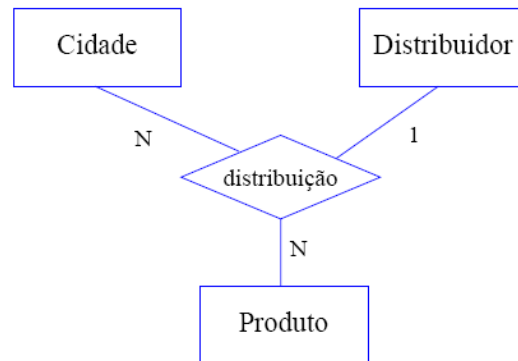
12.1. Modele o relacionamento de um cliente com um banco considerando os atributos (CLIENTE – cpf, nome, idade, endereço; BANCO – código do banco. Ignore a aplicação das formas normais

12.2. Modele o relacionamento dos atletas com os esportes de competição (ATLETA – nome, idade, peso, país; ESPORTE – tipo, material, local). Ignore a aplicação das formas normais

12.3. Transcreva para linguagem lógica a modelagem conceitual a seguir.

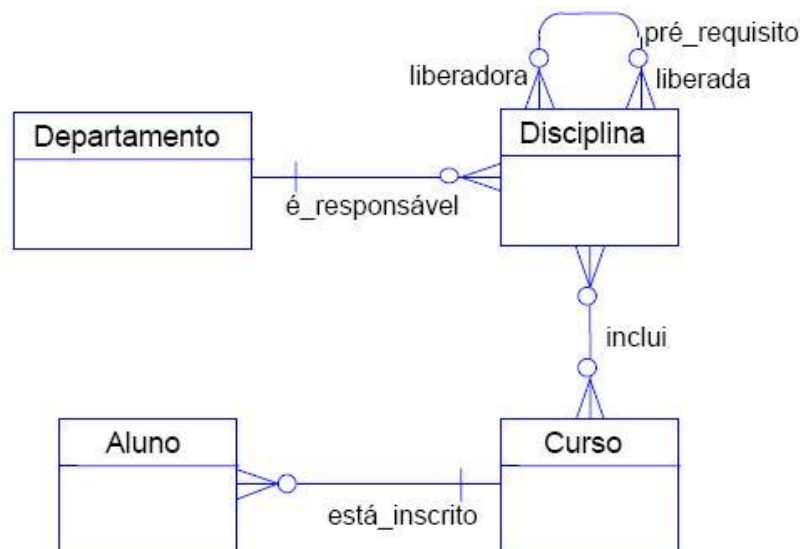


12.4. Mostre como o diagrama abaixo pode ser representado por relacionamentos binários.



12.5. Modifique o diagrama abaixo para especificar o seguinte:

- Um curso não pode estar vazio, isto é, deve possuir alguma disciplina em seu currículo.
- Um aluno pode fazer mais de um curso.



12.6 Construa um diagrama E-R para um hospital com um conjunto de pacientes e um conjunto de médicos. Registros de diversos testes realizados são associados a cada paciente.

18

18

12.7 Construa um diagrama E-R para uma companhia de seguros de automóveis com um conjunto de clientes, onde cada um possui certo número de carros. Cada carro tem um número de acidentes associados a ele.



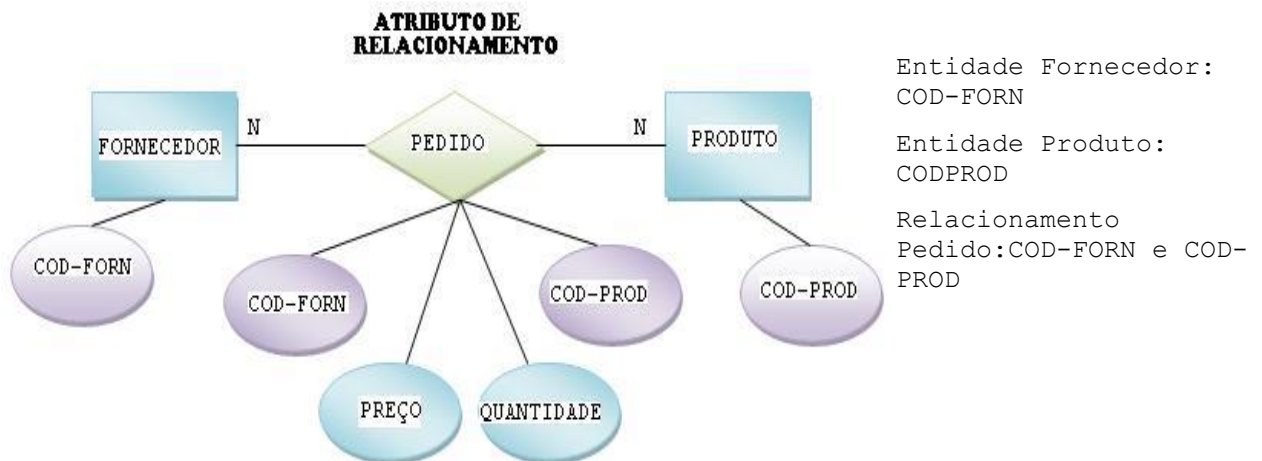
13 Projeto de Chaves

Chave: é um conjunto de um ou mais atributos que, tomados coletivamente, possibilita identificar uma única ocorrência na entidade

Integridade de Entidade: Nenhum atributo que participe da chave de um conjunto-entidade deve aceitar valores nulos.

Aspectos Relevantes

- ✓ A questão fundamental do projeto de chaves é reduzir ao máximo os efeitos de redundância.
- ✓ A alteração dos valores de campos constituintes da chave primária ou a remoção de uma entidade de um conjunto-entidade pode ocasionar problemas de integridade referencial.



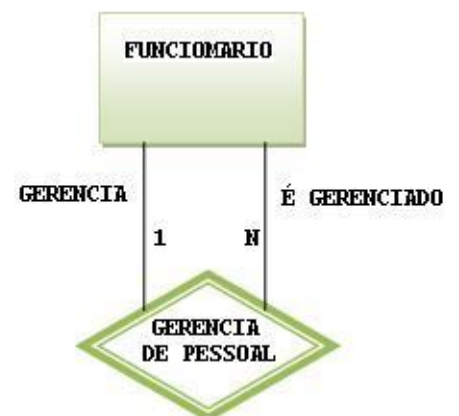
14 Auto-Relacionamento

Um **auto-relacionamento** acontece quando os elementos de uma entidade se relacionam com eles mesmos. Ou seja, um item de uma entidade se relaciona com outro item (ou com o mesmo) dessa mesma entidade. Apesar de parecer algo difícil de acontecer há casos, bem comuns, que identificam este tipo de relação. Por exemplo, uma ocorrência da entidade Funcionário, Gerencia e é Gerenciada por si própria. Assim se dá, ainda, na classe Militar em que o Oficial é um Soldado. Este auto-relacionamento é também conhecido como **RELACIONAMENTO RECURSIVO** ou **RECURSIVIDADE**. Um fator importante a ser observado é que as relações entre os elementos pode ser 1:N ou N:N, e possui as mesmas características de um relacionamento binário (relação entre duas entidades). O relacionamento recursivo deve ser, sempre, entendido e representado nos dois sentidos (EMPREGADO GERENCIA e É GERENCIADO)

A implementação de um auto-relacionamento, através de um SQL, é feita da seguinte forma (considere que a tabela FUNCIONARIO é pré-existente e o EMPREGADO foi promovido a GERENTE):

```
alter table FUNCIONARIO
add constraint EMPR_EMPR_FK foreign key (GERENTE)
references FUNCIONARIO (EMPREGADO);
```

Note que a constraint EMPR_EMPR_FK relaciona a coluna GERENTE como uma chave estrangeira (Foreign Key) da coluna EMPREGADO na tabela FUNCIONARIO¹

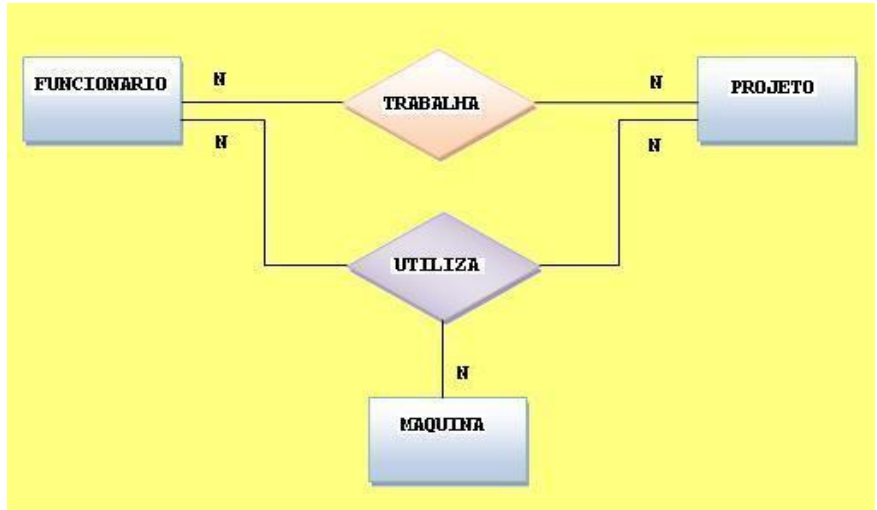


¹ Esta definição foi inserida no endereço <http://pt.wikipedia.org/wiki/Auto-relacionamento>

alvaro_pimentel@uol.com.br

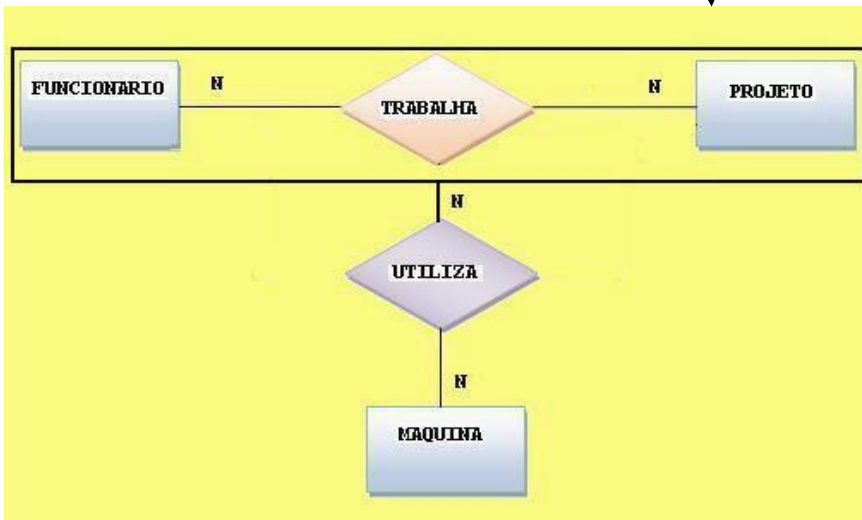
15 Agregação

- ⌘ Uma limitação do MER é não poder expressar relacionamentos entre relacionamentos.
- ⌘ Agregação é uma abstração através da qual os relacionamentos são tratados como entidades de nível superior. Abaixo se observa a existência de dois relacionamentos distintos entre as entidades FUNCIONARIO e PROJETO.



Apesar de parecer assustador, a solução é simples e possibilita uma leitura mais clara do novo diagrama que será desenhado

Após a agregação, uma nova entidade (que pode ser rotulada/nomeada) é representada contendo as entidades FUNCIONARIO-TRABALHA-PROJETO.



Com esta solução, a limitação inicial de não poder relacionar TRABALHA e UTILIZA ficou facilmente resolvida.



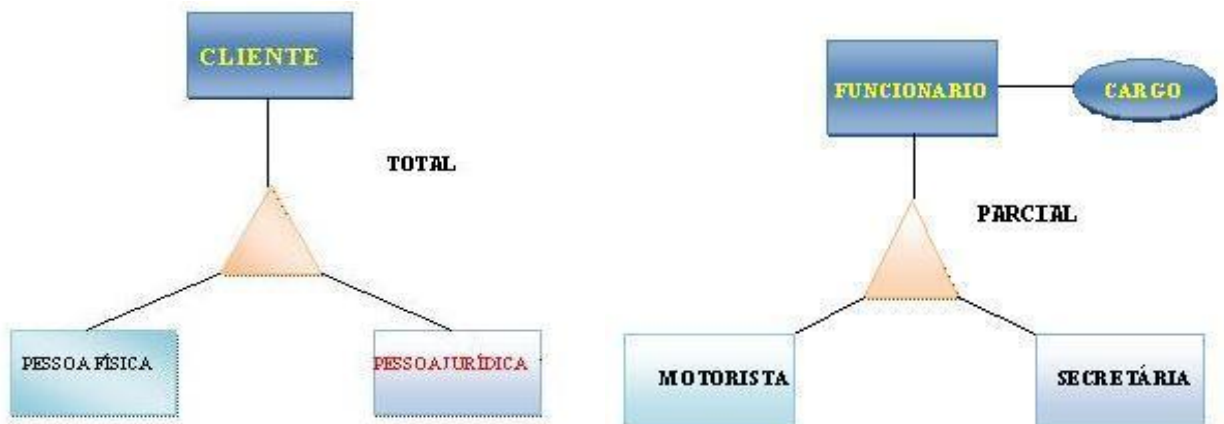
16 Generalização – Especialização

Existem casos em que uma entidade pode ser dividida em categorias, possuindo além dos atributos comuns, alguns específicos para cada categoria. Por exemplo, considerando a entidade MÉDICO, pode-se atestar que todos os médicos são, basicamente, clínicos gerais, mas cada um possui uma especialidade própria como cardiologista, pneumologista, etc. A essa classificação (ou particionamento) dá-se o nome de **especialização**, que nada mais é do que a identificação de uma característica própria de objetos com propriedades similares.



Dentro deste modelo pode-se, ainda, considerar a existência de outros tipos de especialização

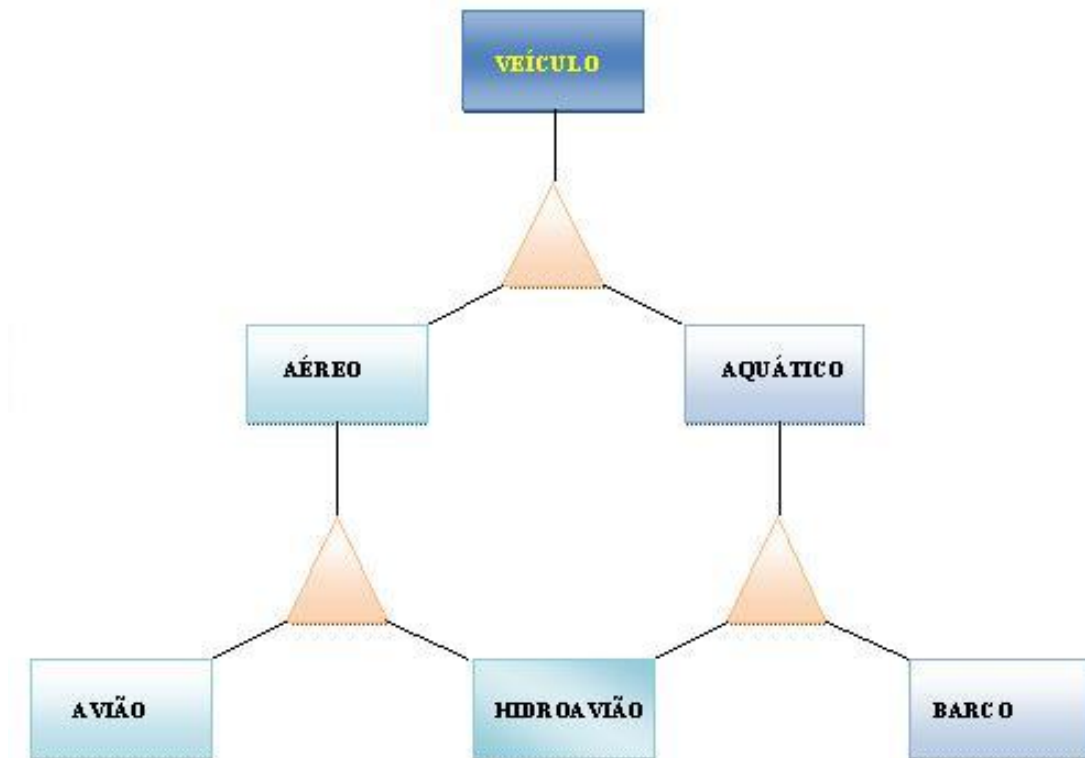
16.1 Total e Parcial



16.2 Não-Exclusiva



16.3 Múltipla



NOTA: Todos os tipos de especializações indicam que a entidade resultante herdar características da entidade origem. Desta maneira, ao resultado apresentado, diz-se que este possui uma HERANÇA, cujo conceito faz parte da ORIENTAÇÃO A OBJETO. Assim sendo, é correto afirmar que, nos itens anteriores, há HERANÇA TOTAL, PARCIAL, NÃO-EXCLUSIVA e MÚLTIPLA.

17 Dependência Existencial e Entidades Fracas

A **dependência existencial** ocorre quando a existência de uma determinada entidade está condicionada à existência da outra entidade relacionada a ela.

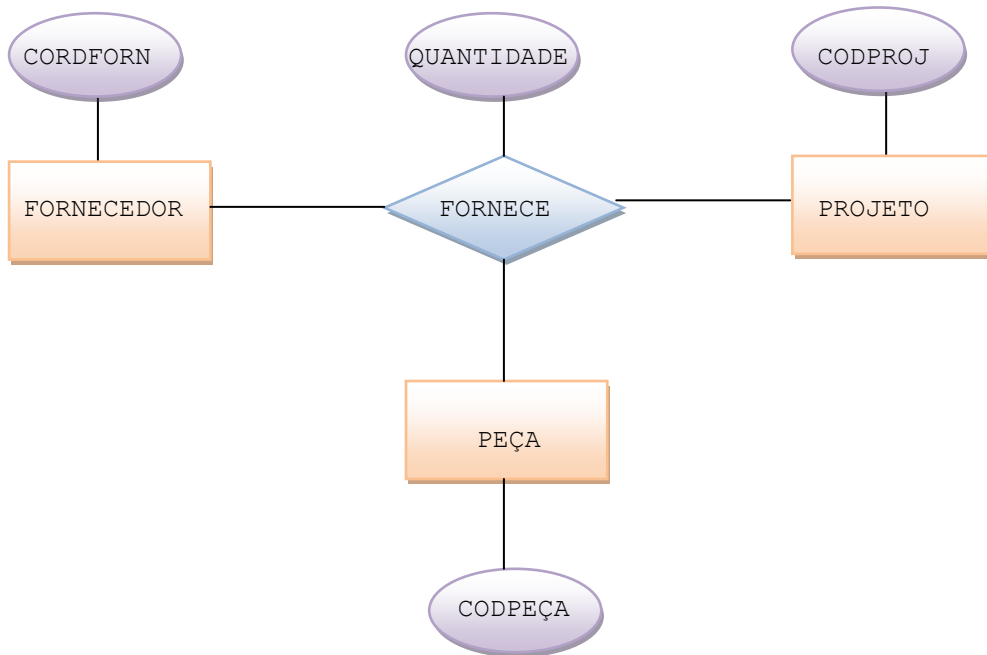


Uma **entidade fraca** não possui uma identidade própria. Assim sendo, sua chave primária é composta pela chave estrangeira proveniente da entidade origem concatenada a um identificador de si mesma (que pode repetir para diferentes instâncias da entidade dona).



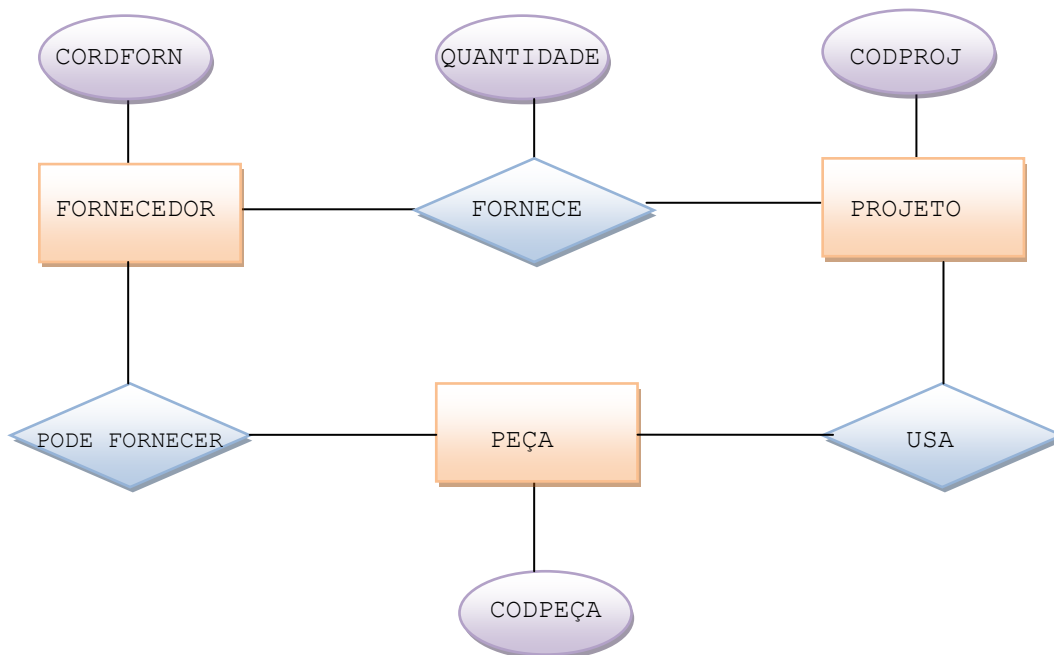
18 Relacionamentos de Grau Superior a 2

PARTE 1

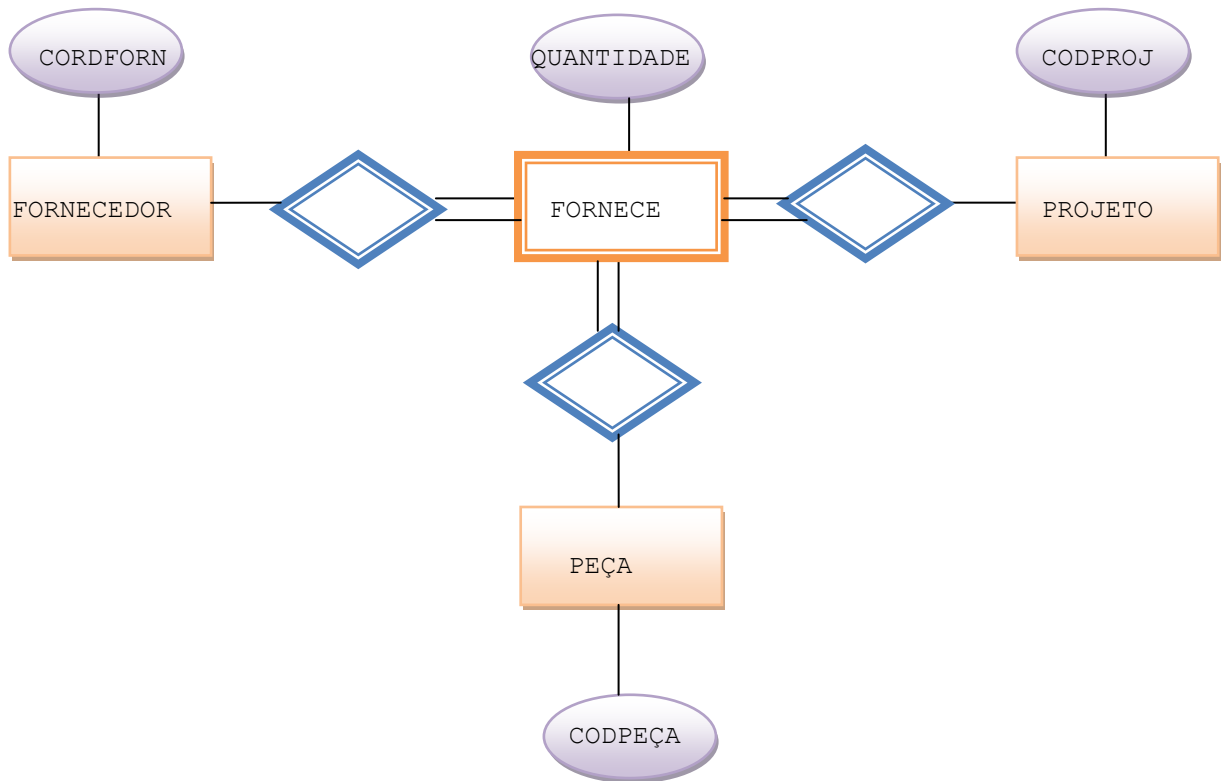


Relacionamentos de Grau Superior a 2 (cont)

PARTE 2



PARTE 3



UM POUCO MAIS SOBRE CHAVES

Uma chave é chamada de primária (**primary key – pk**) quando um atributo dado é único e obrigatório em uma tabela. Por exemplo, tomando-se como base o CODFORN da tabela FORNECEDOR acima é fácil observar que se trata de um atributo que representa um único fornecedor, tal como o CGC. No caso de uma tabela PESSOA poderia ser o CPF, o RG ou a Carteira de Habilitação. Nota-se, então, que a escolha de uma chave deve significar um atributo que não possua elementos que impossibilitem, ou minimizem erros de digitação. Portanto, não se deve escolher como chave atributos que contenham tamanhos variáveis ou com formatos alfanuméricos tal como nome, endereço ou descrição, por exemplo.

Quando há a existência de um relacionamento de uma entidade forte com uma fraca, **obrigatoriamente** a chave da tabela forte é **incluída** na fraca. Essa chave na tabela fraca é chamada de chave estrangeira (**foreign key – fk**) como ocorre nos relacionamentos um-para-muitos (1:N). Quando se trata, porém, de um relacionamento muitos-para-muitos (N:N) é necessária a criação de uma tabela associativa que, neste caso, será a entidade fraca. Assim, nesta nova entidade serão colocadas, como chaves estrangeiras (**fk**), as chaves primárias (**pk**) das tabelas de origem.

A colocação de uma chave estrangeira é sempre um indicativo de que uma entidade é dependente de outra por uma razão qualquer. Os motivos que determinam a necessidade de criação de uma chave (quer seja pk ou fk) estão, na maioria das vezes, associados a regras de negócio identificadas no levantamento dos requisitos. Em outros casos são para resolver problemas de performance que são identificados após a implementação das bases de dados ou dos sistemas. Portanto, as chaves sempre estão associadas a melhorias de acesso a base de dados.

19 Exercícios

19.1: Modele o relacionamento entre um Lote e um Produto considerando que o Produto refere-se a materiais de farmácia e pode ser subdividido em, pelo menos, duas entidades. Crie os atributos que considerar pertinente assim como as cardinalidades apropriadas.

19.2: Especialize a entidade Veículo em, pelo menos, seis entidades relacionadas.

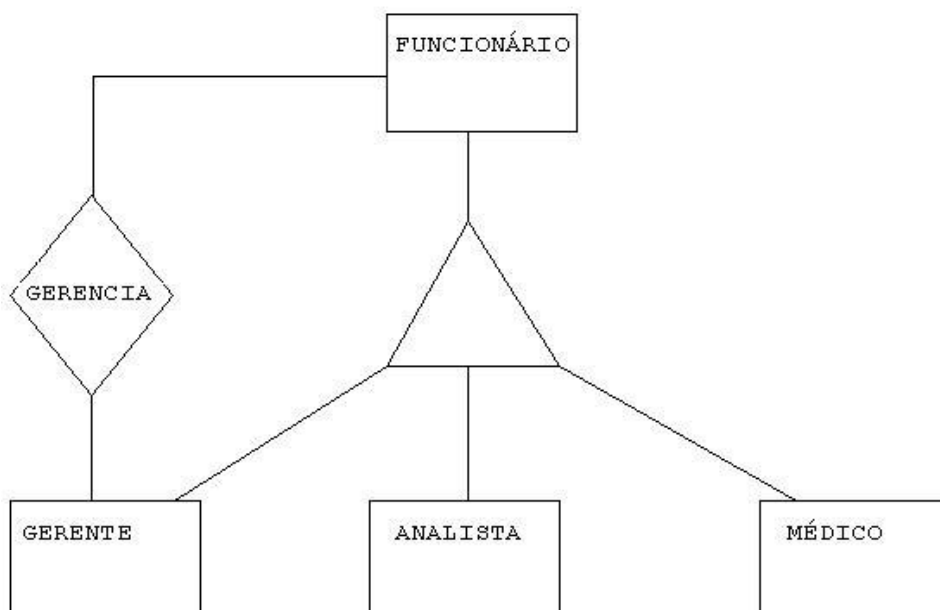
19.3: Modele o relacionamento entre Empregado e Departamento considerando que o empregado pode ser dividido em Gerente, Secretária e Engenheiro. Para cada relacionamento do DER defina atributos quando possível. Defina os atributos identificadores ou chaves

19.4: Considere agora que a secretária, do exercício 3, use um Aplicativo que pode ser subdividido em três itens. O Engenheiro participa de um Projeto, e ambos utilizam a entidade Máquina

19.5: Do exercício 4 considere que um Fornecedor se relaciona com um Lote e com um Fabricante. Este, por sua vez, está relacionado com o Produto. Efetue a alteração no modelo para que ele atenda aos novos requisitos.

19.6: Uma empresa desenvolve projetos de grande porte. Esta empresa está organizada em departamentos, sendo que cada projeto é **sempre** coordenado por um departamento. Os departamentos possuem empregados que podem ser chefes. Embora um empregado pertença sempre a um departamento, ele pode ser alocado a projetos de outros departamentos. Os funcionários possuem nome, data de nascimento e cpf. Os responsáveis pelo projeto são os chefes de departamento ao qual o projeto está alocado. Todo projeto possui uma área (engenharia, urbanismo, etc.) e período **definido** de tempo.

19.7: De acordo com o DER abaixo um Analista ou Médico não podem ser gerentes. Por quê? Qual a alteração necessária para tornar isso possível?



19.8: Uma organização que atua no ramo de vendas de materiais de construção deseja montar um banco de dados para emissão de faturas em suas lojas, gerenciando a comissão de cada empregado. Sabe-se que:

- a) A empresa possui diversas lojas;
- b) Um empregado pertence sempre a uma loja;
- c) Uma nota fiscal é composta de dados genéricos (número da nota fiscal, nome do cliente, data de emissão, valor total da Nota fiscal, nome do empregado responsável pela venda) e dados do detalhe da venda (nome do material vendido, quantidade deste material, valor unitário, valor total do item de material vendido).

.

20 Normalização³

O processo de **normalização** pode ser visto como o processo no qual são eliminados esquemas de relações (tabelas) não satisfatórios, decompondo-os, através da separação de seus atributos em esquemas de relações menos complexas, mas que satisfaçam as propriedades desejadas.

Por que Normalizar ?

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

O processo de normalização como foi proposto inicialmente por Codd conduz um esquema de relação através de uma bateria de testes para certificar se o mesmo está na **1ª, 2ª e 3ª Formas Normais**.

Para que se compreenda melhor suponha que haja uma entidade **Funcionários** que armazene as informações dos funcionários de uma empresa e que o resultado físico final seja a tabela mostrada abaixo.

	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

Ao se observar a tabela vê-se que ela sofre das seguintes anomalias:

- **Anomalia de Exclusão** – Se o funcionário de código igual a 3 for excluído, o Setor será excluído também.
- **Anomalia de Alteração** – Se o nome do Setor Suporte mudar para Apoio será obrigatório a alteração, deste nome, em todos os registros da tabela.
- **Anomalia de Inclusão** – Se um novo funcionário for contratado para o Setor Suporte será obrigatória a alteração da quantidade de funcionários no campo – **QuantidadeFuncionarios** – em todas as ocorrências que houver o setor de nome SUPORTE.

Para poder resolver o dilema acima é necessário **NORMALIZAR** a entidade. Para isto aplica-se as **formas normais** como será estudado a seguir:

³ Os conceitos de Normalização e Formas Normais foram baseados na literatura disponibilizada no endereço <http://leitejr.files.wordpress.com/2008/09/modelagem.pdf>



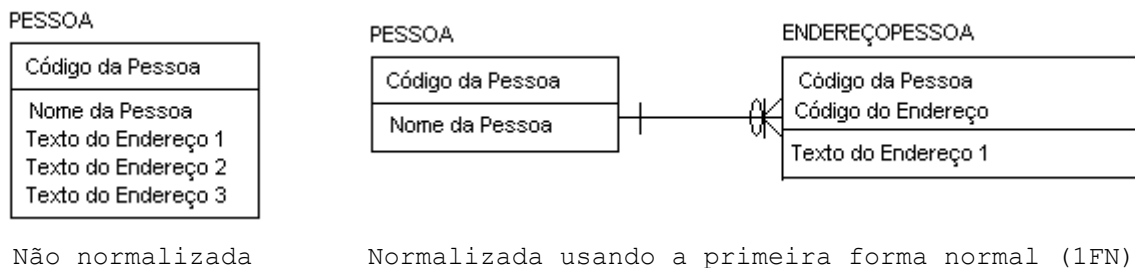
20.1 Primeira Forma Normal (1FN)

Uma relação está na 1FN se todos os domínios básicos contiverem valores únicos (não contiver grupos repetitivos). Para atingir esta forma normal é necessário eliminar as redundâncias da base (tautologia) e os grupos de repetição. Isso significa dizer que todos os atributos de uma tabela devem ser **atômicos** (indivisíveis), ou seja, não são permitidos atributos multivalorados, atributos compostos ou, ainda, atributos multivalorados compostos.

- Identificar qual elemento é, potencialmente, 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.

Ex.



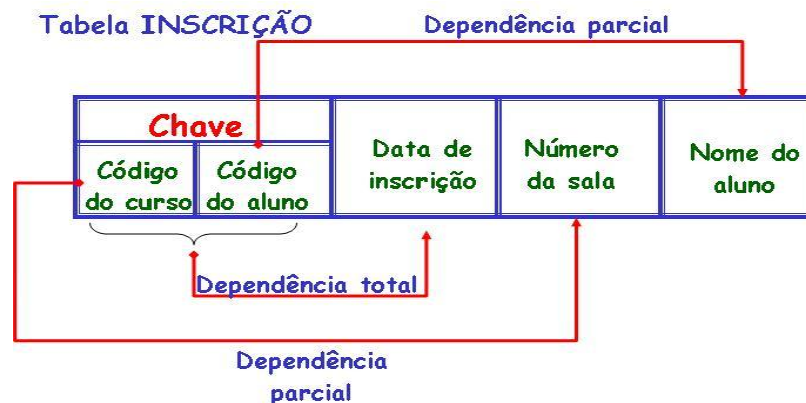
20.2 Segunda Forma Normal (2FN)

Uma relação R está na 2FN se, e somente se, ela estiver na 1FN e todos os atributos não-chave forem totalmente dependentes da chave primária.

- Aplica-se conceito da dependência funcional para 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 dependência funcional $X \rightarrow Y$ é **total** se a remoção de um atributo A do componente X, implica que a dependência funcional deixa de existir.

A dependência funcional $X \rightarrow Y$ é **parcial** se existir um atributo A no componente X que pode ser removido e a dependência não deixa de existir.



Considerando-se, agora, 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)

O resultado após a aplicação da segunda forma normal (2FN) será:

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 nota-se um desdobramento do **Arquivo de Vendas** em duas estruturas – **Vendas** e **Mercadoria** (o arquivo de Notas Fiscais, não foi alterado, por não possuir chave composta):

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

Segunda 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.

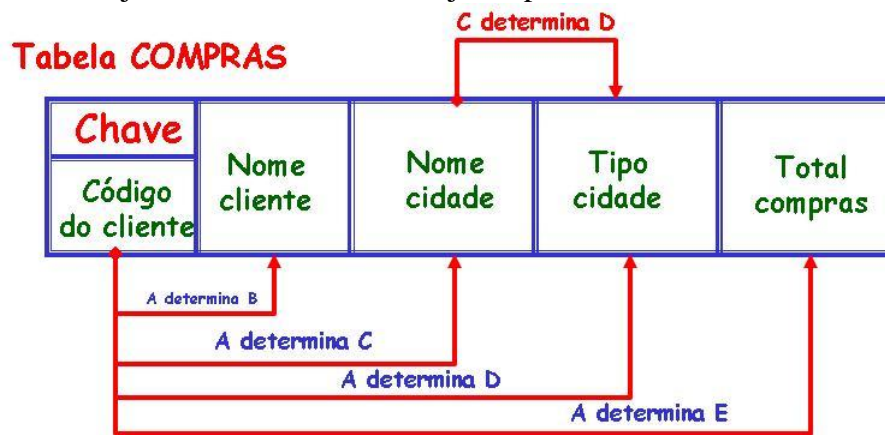


20.3 Terceira Forma Normal (3FN).

Uma relação R está na 3FN se estiver na 2FN e se não houver dependência transitiva entre atributos não chave. Em outras palavras, se cada atributo for funcionalmente dependente apenas dos atributos componentes da chave primária ou se todos os seus atributos não chave forem independentes entre si.

- Aplica-se conceito da dependência transitiva para identificar todos os atributos que são funcionalmente dependentes de outros atributos não chave;
- Removê-los e criar uma nova entidade com os mesmos.

A dependência funcional $X \rightarrow Y$ em uma tabela T é transitiva se existir um conjunto de atributos Z que não é um subconjunto de chaves de T e cujas dependências $X \rightarrow Z$, $Z \rightarrow Y$, são válidas.



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)

Pode-se perceber que ocorreu um desdobramento da tabela de **Notas Fiscais**, por ser a única que possuía campos que não eram dependentes da chave principal (Num. NF). Os elementos Nome e Endereço não se alteram independente do que ocorra com a Nota Fiscal.



21 Exercícios

21.1. Considerando a existência de três tabelas (FUNCIONÁRIO, DEPENDENTE, CURSO) normalize-as de acordo com o que foi desenvolvido

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 (RN):

RN1. Um funcionário pode ter mais de um dependente
RN2. Um funcionário pode fazer mais de um curso

21.2 Imagine que seja necessário desenvolver um sistema para cadastro de clientes e foi recebida, do cliente, uma lista com os dados que deverão compor o sistema, com base nela normalize a estrutura de dados de acordo com as formas normais estudadas. A título de treinamento considere o tratamento das informações como um pequeno dicionário de dados, em que serão descritos os atributos pertinentes aos elementos.

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

Nome do dado	Descrição	Tipo	Tamanho	Obrigatório
Nome	Nome do cliente pessoa física	Char	45	Sim
Nome do Pai				
Nome da Mãe				
Endereço				
Telefone1				
Telefone2				
Número do Fax				
Número do Celular				
Telefone do trabalho				
Data de Nascimento				
Naturalidade				
Nacionalidade				
Endereço 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 identidade				

21.3 Considere o desenvolvimento de um modelo lógico que representa a situação a seguir:

O Departamento de Vendas da Indústria Beleza Ltda, após estudos de mercado, verificou que para atingir seus objetivos seria necessário adquirir uma 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. Deve ser construído um sistema para administração do novo modelo de vendas adotado 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

21.4 De acordo com as regras estudadas 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**
- Código do Programa
- Nome do Programa
- Tempo Estimado
- Nível de Dificuldade (1, 2 ou 3)
- Descrição da Dificuldade (Fácil, Médio, Difícil)

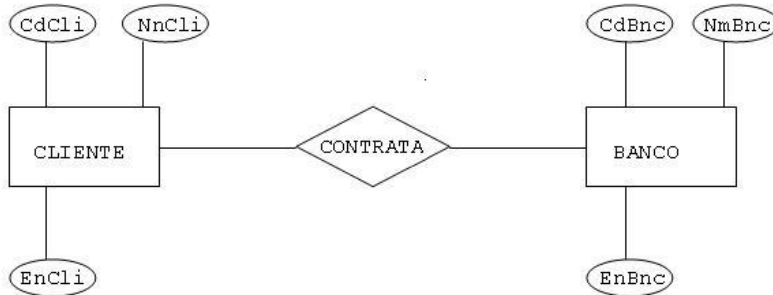
Regras do negócio:

- 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;

22 Soluções

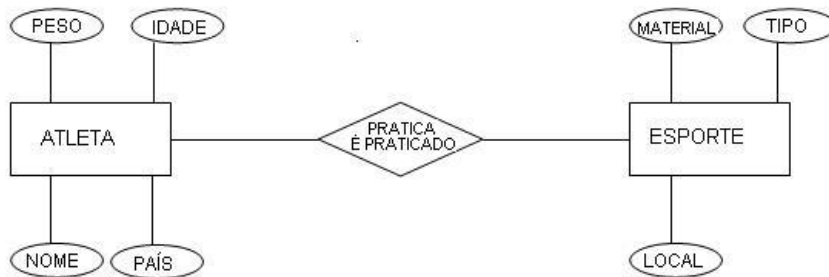
As respostas apresentadas representam apenas uma das soluções possíveis. Em se tratando de modelagem de dados o caso certo é o que atende a uma especificação dentro dos padrões estabelecidos. Em muitos casos é necessário desnormalizar um modelo para melhorar o acesso a uma base e, com isso, melhorar a performance do processo de busca.

Uma solução do 12.1:



As soluções 12.1, 12.2 e 12.5 representam um modelo que não contempla os conceitos de normalização, entretanto, como treinamento inicial, é importante entender como se constrói um relacionamento e de que maneira pode-se, então, melhorar o particionamento dos dados

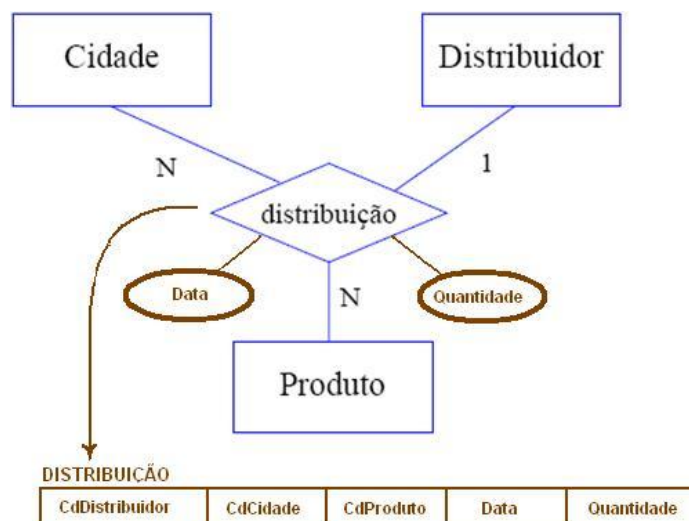
Uma solução do 12.2:



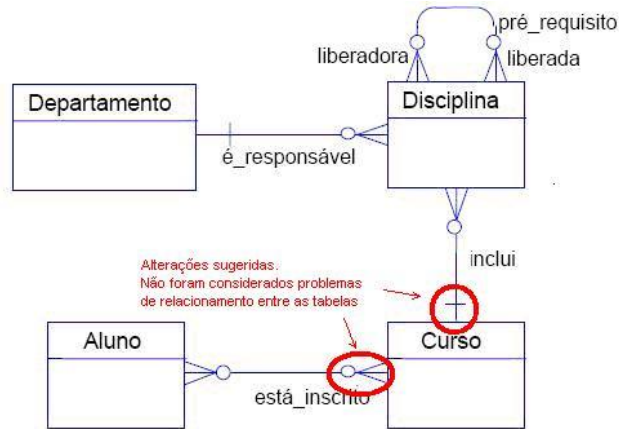
Uma solução do 12.3:

TABELA CLIENTE (CdCli (pk), NnCli, EnCli)
TABELA BANCO(CdBnc (pk), NmBnc, EnBnc)

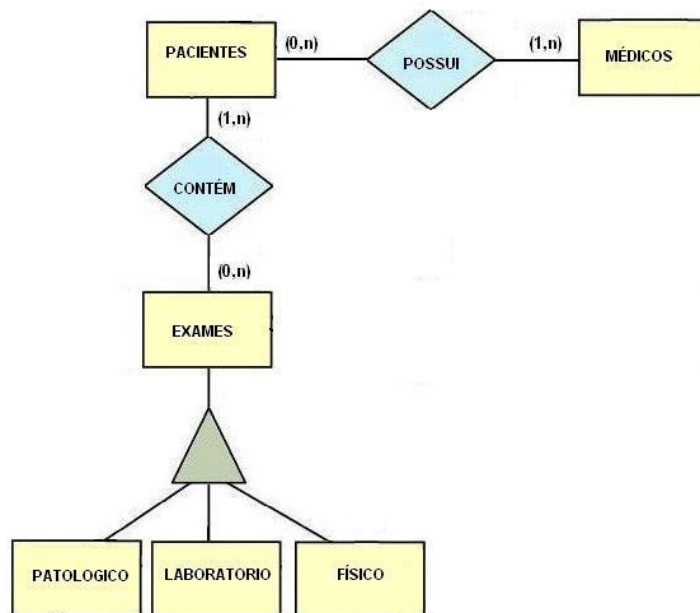
Uma solução do 12.4:



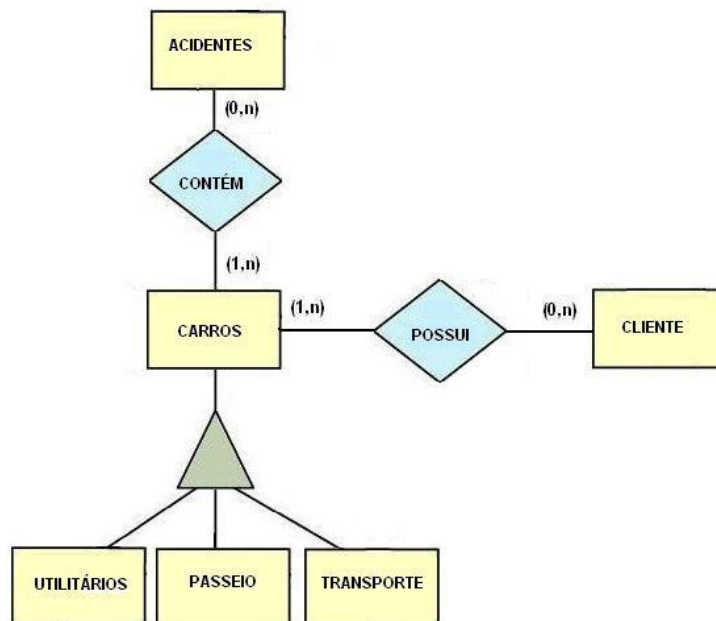
Uma solução do 12.5:



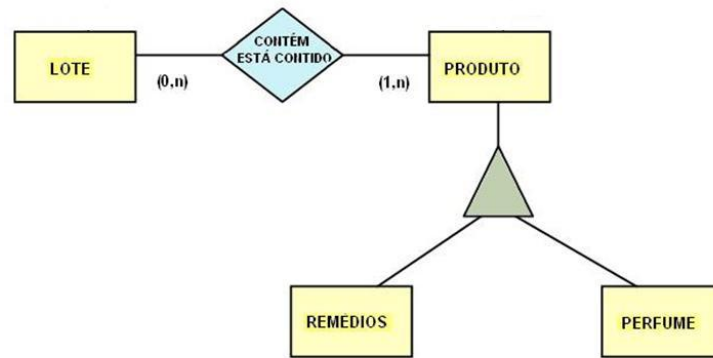
Uma solução do 12.6:



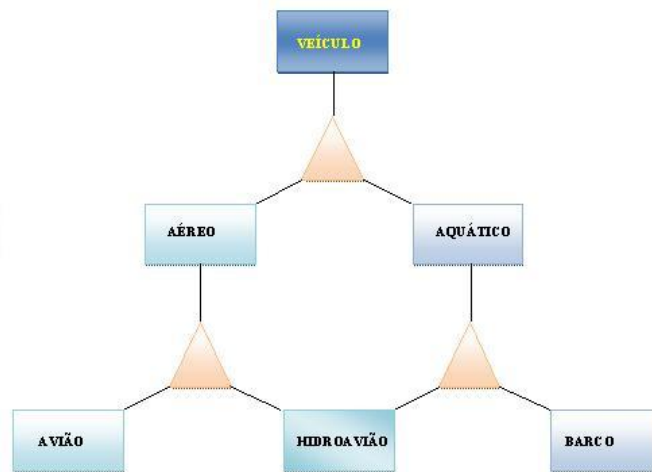
Uma solução do 12.7:



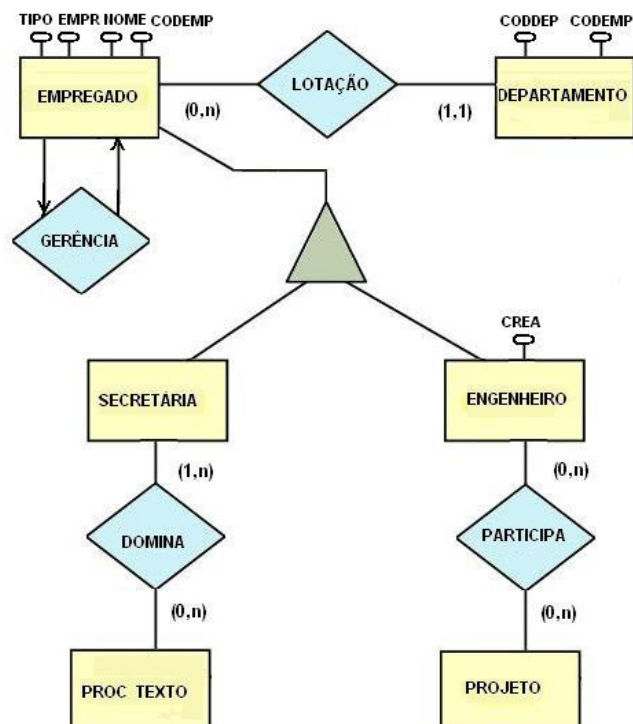
Uma solução do 19.1:



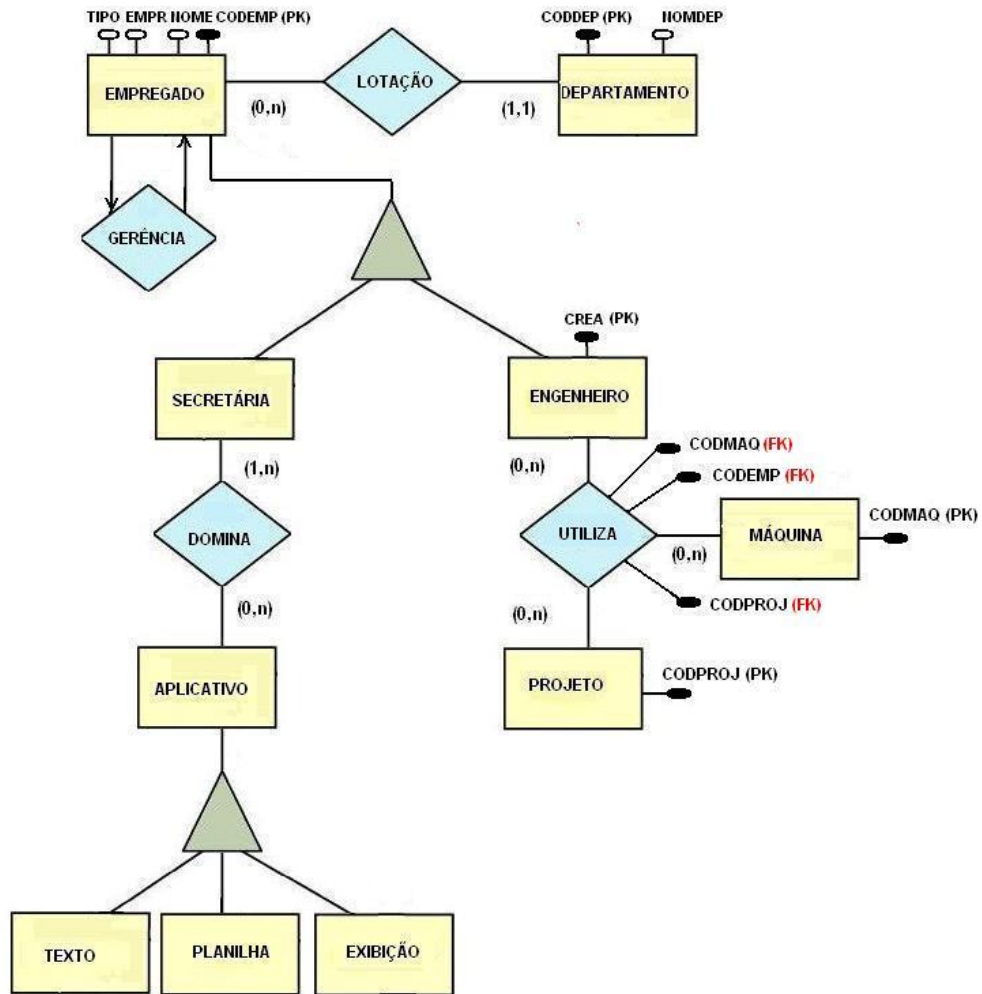
Uma solução do 19.2:



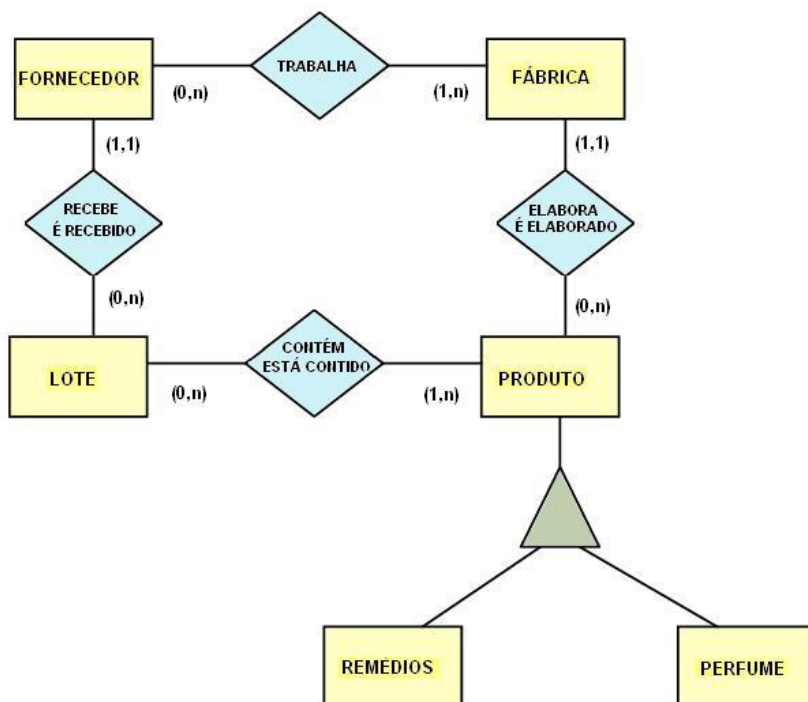
Uma solução do 19.3:



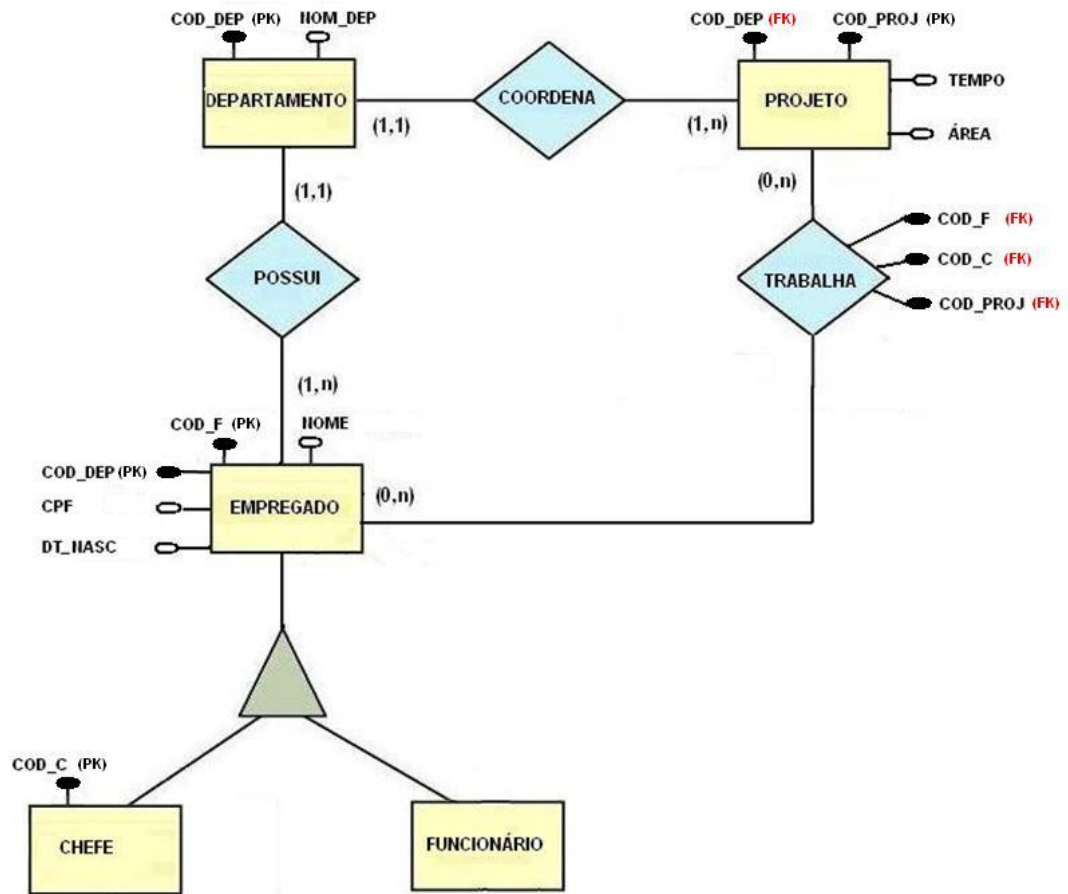
Uma solução do 19.4:



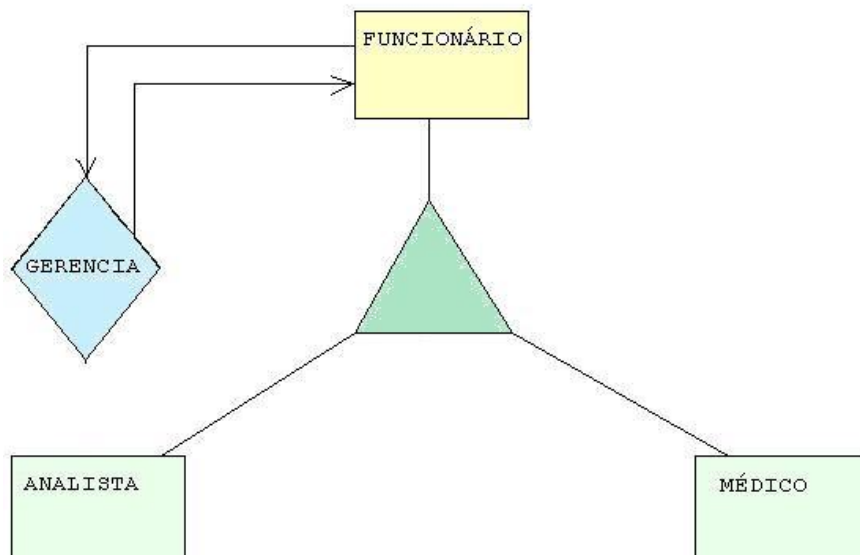
Uma solução do 19.5:



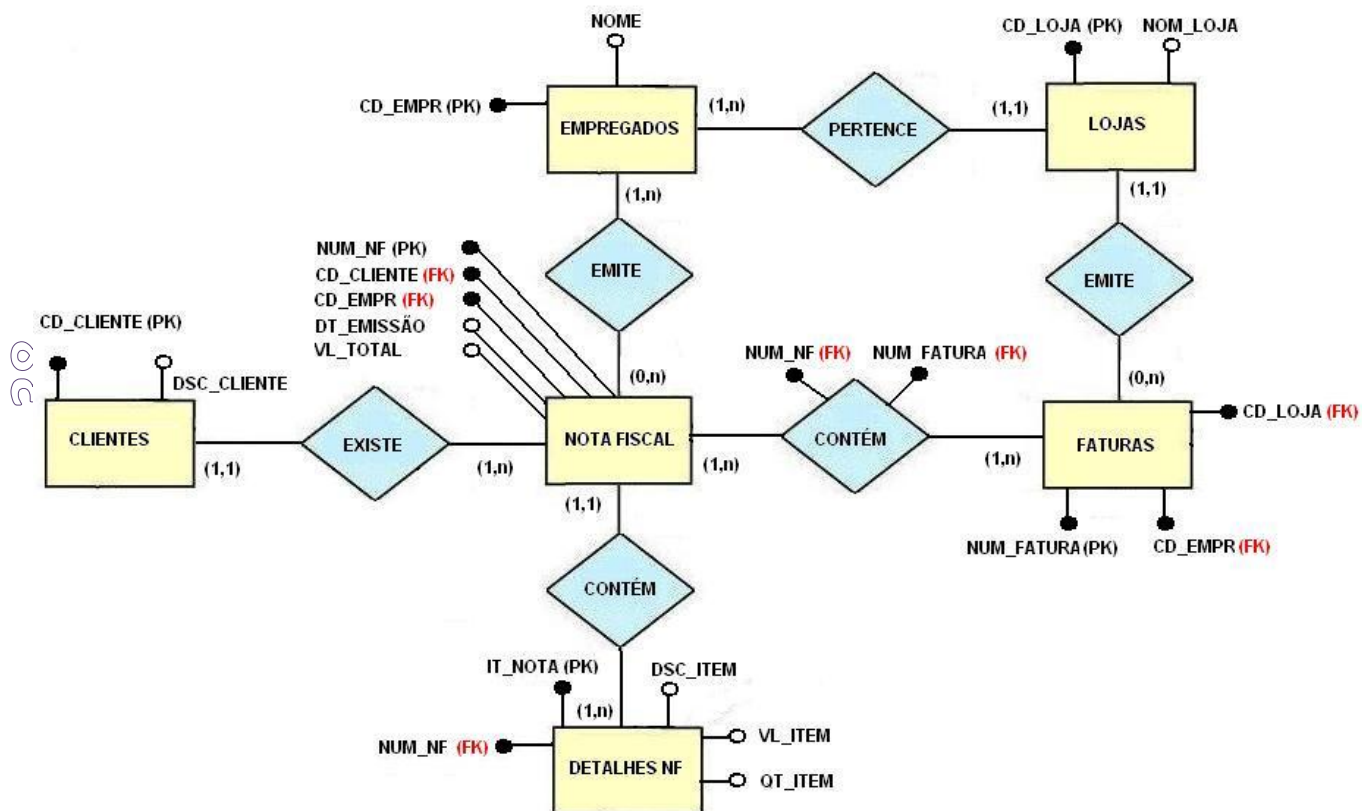
Uma solução do 19.6:



Uma solução do 19.7:

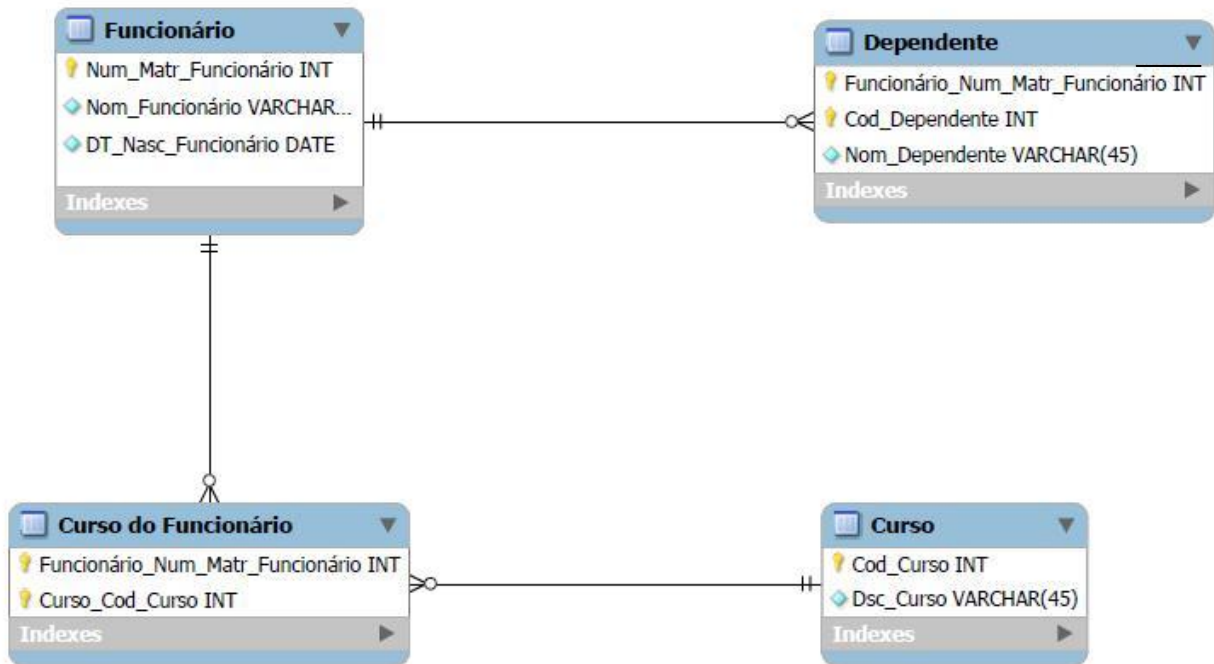


Uma solução do 19.8:



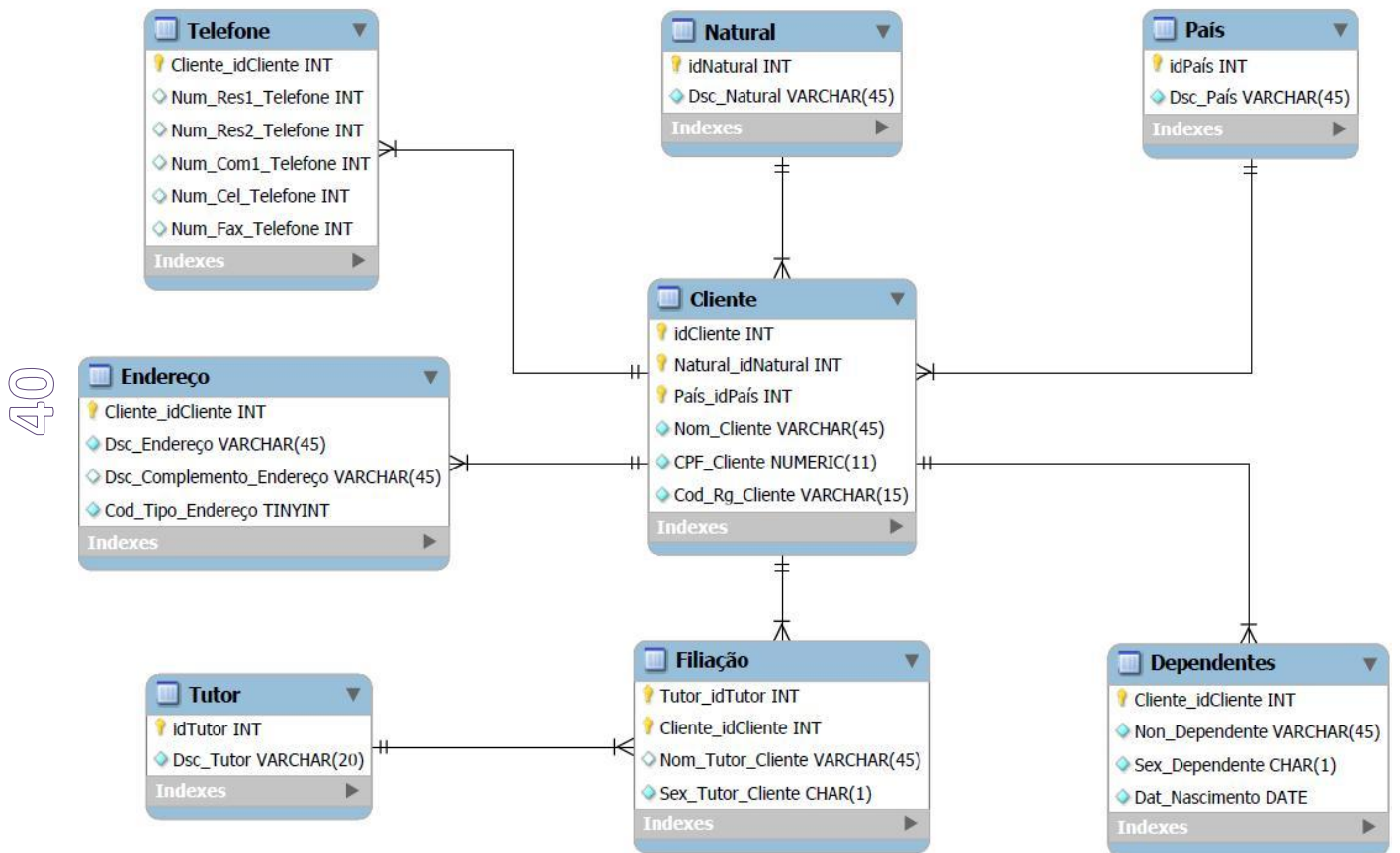
O modelos abaixo foram desenvolvidos no *software* gratuito MySQL Workbench

Uma solução do 21.1:



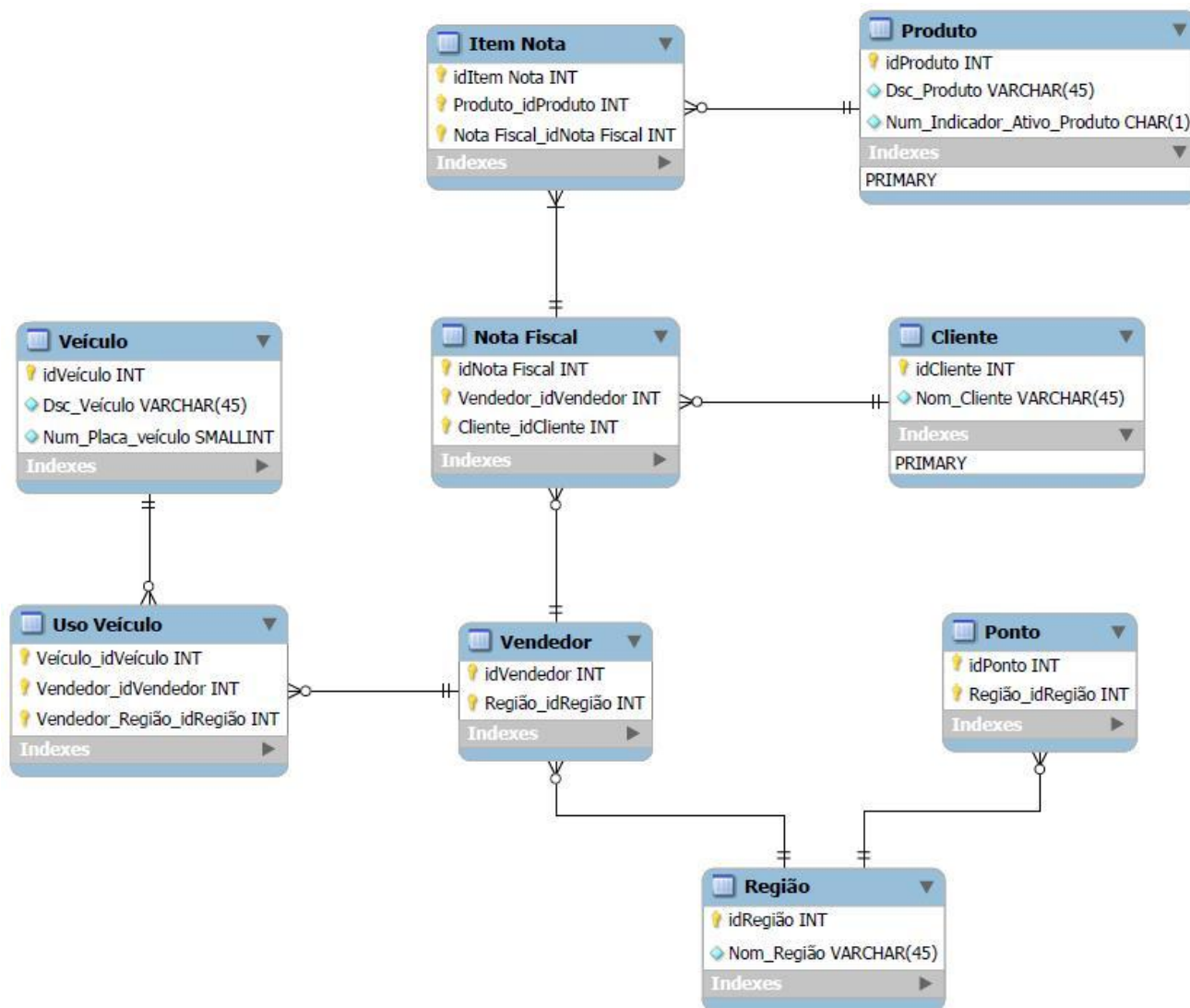
Como foi explicado no tópico 12.3, toda vez que for identificado um relacionamento n:n sempre será necessária a criação de uma tabela extra para representar o relacionamento. Neste caso um funcionário pode se inscrever em muitos (n) cursos e, por sua vez um curso pode possuir muitos (n) funcionários. Assim, foi criada tabela associativa (ou junção) **Curso do Funcionário** para evitar a redundância de dados nas tabelas origens.

Uma solução 21.2:



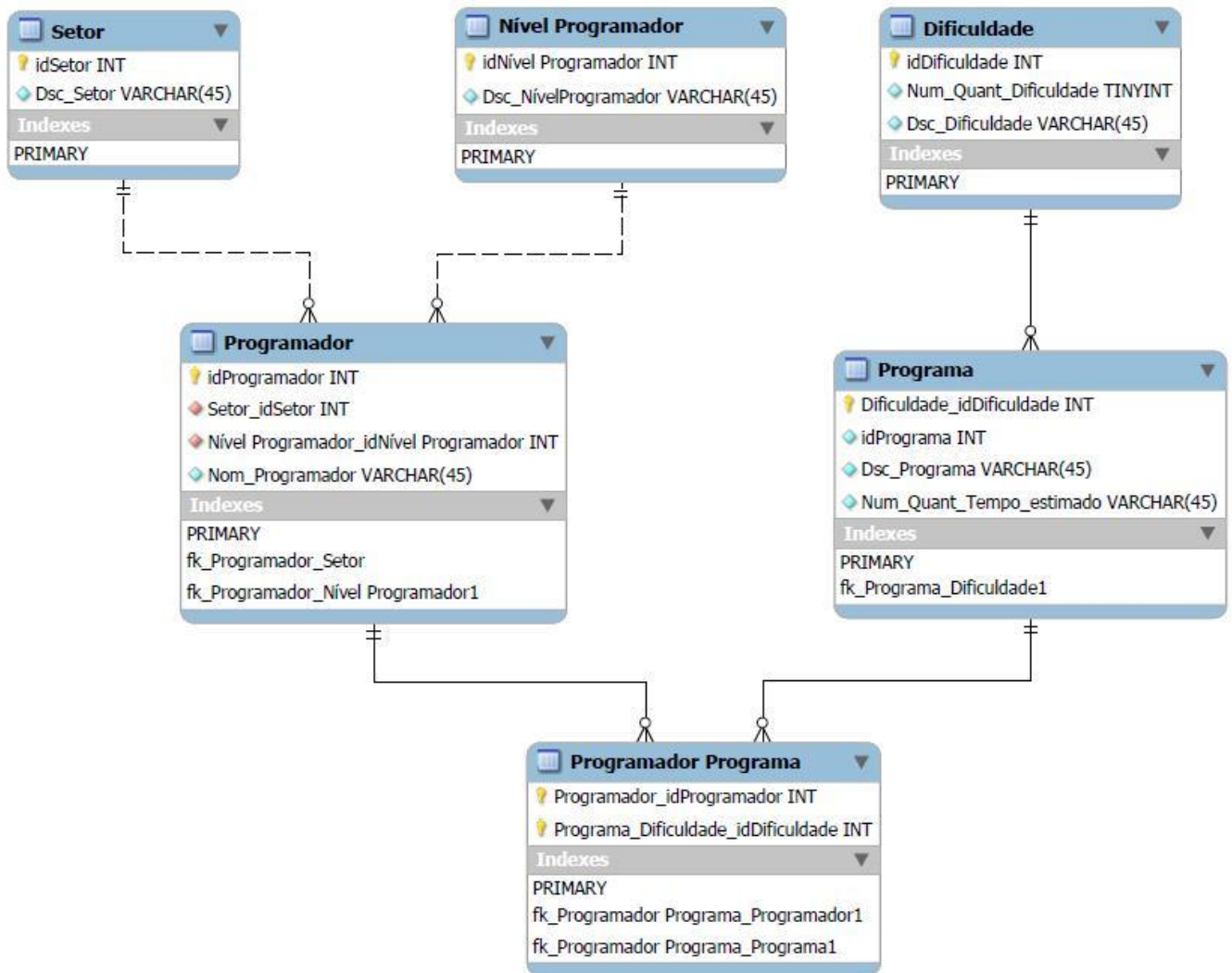
Atualmente já é possível, legalmente, um indivíduo possuir pais de mesmo sexo. Por esta razão, a **Filiação** é uma tabela associativa (que representa a paternidade/maternidade) de **Tutor** com a tabela **Cliente**. Talvez fosse interessante criar uma tabela que identificasse o tipo do dependente, o tipo do telefone e o tipo do endereço. No entanto a divisão dessas tabelas pode influenciar em um esforço extra considerando os valores que podem ser assumidos por elas.

Uma solução 21.3:



A tabela **Uso Veículo** representa uma junção dos veículos que podem ser utilizados pelos vendedores. É importante notar que sua existência só acontecerá se as tabelas pai existirem.

Uma solução 21.4:



A tabela **Programador Programa** é associativa pelas mesmas razões apresentadas nas soluções anteriores.

É possível que sejam encontradas soluções melhores que as apresentadas anteriormente. Cabe lembrar que modelagem de dados é como a construção de conhecimento que vai se edificando com o tempo. Raramente se constrói uma modelagem que não sofra alterações, ou que não contenha equívocos, e isso faz parte do processo de desenvolvimento de um modelo de bases de dados. Exercitar é o que vai trazer experiência e mais habilidade no tratamento dos dados.

No próximo manual será tratada a instalação, a modelagem e o manuseio de uma base de dados MySQL através da ferramenta CASE MySQL Workbench e MySQL Query Browser.

A instalação do MySQL está na apostila “Free Acess: O MySQL na prática” disponível no endereço <http://artigocientifico.tebas.kinghost.net/pesquisadores/?mnu=2&smnu=5&id=29536>

Alvaro Caetano Pimentel Sobrinho é
Doutorando em Ciência da Informação UFRJ/IBICT,
Mestre em Tecnologia da Informação UNESA,
Formado em Telecomunicações UNESA,
Administrador de bases de dados Oracle, Db2 e Idms e
Docente em Sistemas de Informação

