

Apostila de Banco de Dados I  
Autor: Gilberto Assis Oliveira  
Ilustrações: Wilmar Ferreira de Freitas  
Faculdade COTEMIG

Parte 1 - Conceitos e definições sobre Banco  
de Dados e Modelagem de Banco de dados

# Índice

Capítulo 1 – Introdução.....	3
1.1 – Banco de Dados X Sistema Gerenciador de Banco de Dados (SGBD).....	6
1.2 – Dado X Informação.....	6
1.3 – Atores do Sistema.....	7
Capítulo 2 – Modelos de Banco de Dados.....	8
2.1 – Introdução.....	8
2.2 – Tipos de Modelos de Dados.....	8
2.3 – Entidades.....	9
2.4 – Atributos.....	10
2.5 – Tabela Filha e Tabela Pai.....	12
2.6 – Relacionamentos.....	13
2.6.1 – Cardinalidade.....	14
2.7 – Relação PAI e FILHA.....	16
2.8 – Modelo Completo.....	16
2.9 – Interpretação do minimundo.....	19
2.10 – A anomalia do NxM.....	23
2.11 – Autorrelacionamento.....	26
2.12 – Múltiplas Relações entre duas entidades.....	27
2.13 – Migrando para o modelo lógico.....	29
Capítulo 3 – Exercícios.....	31
3.1 – Lista 1: Definições, Conceitos e Modelagem de Banco de Dados.....	31
3.2 – Lista 2: Modelagem de Banco de Dados.....	32
3.3 – Lista 3: Exercícios Auxiliares.....	33
3.4 – Lista 4: Exercícios Auxiliares.....	35

# Capítulo 1 – Introdução

Pode-se definir um banco de dados como: **uma coleção logicamente coerente de dados com uma determinada significação intrínseca**. Isso quer dizer que um banco de dados constitui-se em um repositório de dados, ou seja, um lugar destinado a armazenar dados. Mas, isso não é feito de qualquer jeito, um banco de dados deve ser criado de forma coerente com o seu propósito. Ele deve ser capaz de armazenar os dados referentes ao escopo do projeto no qual foi pensado, ser capaz de permitir a seleção, a inclusão, a manutenção e exclusão de registros originados do sistema, tudo de maneira efetiva e segura.

Assim como um guarda-roupa está destinado a guardar peças de roupas, um banco de dados está destinado a guardar dados. Na verdade podemos traçar vários paralelos entre eles (Banco de Dados e o guarda-roupa).

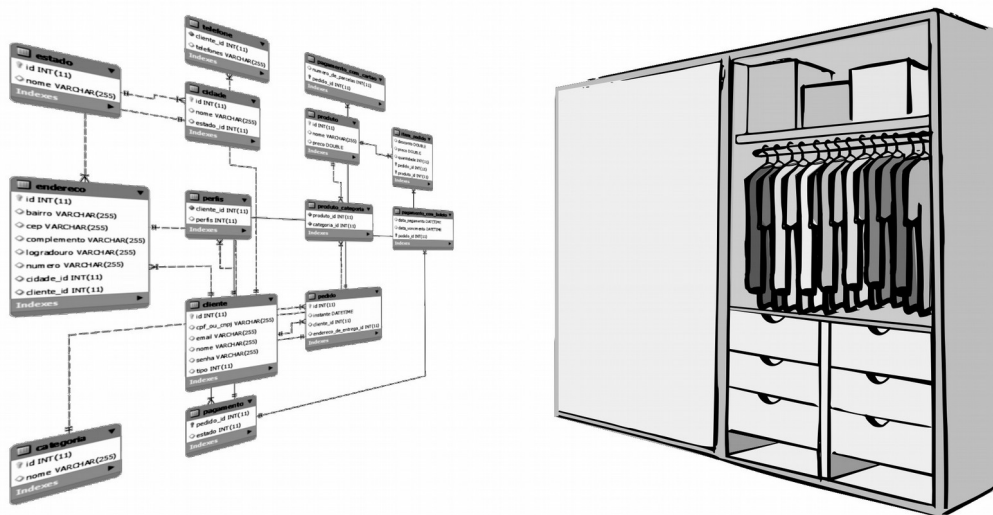


Figura 1: Analogia entre Guarda-roupa e Banco de Dados

Os dois têm o mesmo objetivo que é guardar e proteger os objetos nele armazenados.

Quando guardamos uma peça de roupa no guarda-roupa esperamos que ela ali permaneça preservada e protegida de fatores externos até que uma ação deliberada a retire de lá.

O mesmo acontece com um registro armazenado no banco de dados, uma vez que a ação de inserir um registro no banco de dados foi completada, este registro deverá permanecer íntegro e inalterado até que uma ação explícita de alteração aconteça no banco de dados.

Outro paralelo que pode ser traçado entre os dois é a possibilidade de guardar seus itens (roupas ou registros) de forma organizada com o objetivo de maximizar sua capacidade de armazenamento e diminuir o tempo de busca pelo item procurado (roupa ou registro). Na verdade ambos conseguem cumprir seus objetivos (guardar roupas ou registros) mesmo sem nenhuma preocupação com a forma em que estes itens foram organizados. Vejamos, é possível guardar todas as peças de roupas no guarda-roupa

mesmo que ele não tenha nenhuma divisória (gavetas, cabides, prateleiras), certo? No banco de dados isso também acontece, é possível guardar todos os registros em uma única estrutura de dados (tabela).

Mas afinal, porque então o guarda-roupa tem tantas divisórias?

A resposta é ORGANIZAÇÃO, imagine a necessidade de pegar uma peça de roupa específica e não aleatória ... é fácil perceber que se as roupas foram separadas por tipo e em compartimentos específicos será mais fácil encontrá-la do que se todas as peças de roupas estiverem juntas de forma indiscriminada. **Fácil** aqui significa menor esforço e menor tempo para completar as tarefas de encontrar e retirar a peça de roupa. Vamos para um exemplo prático. Imagine-se procurando um par de meias, é claramente mais fácil de encontrá-las em uma gaveta de meias do que se elas estiverem no meio de um monte de outras peças, todas misturadas em um mesmo compartimento, ou ainda também podemos ter problemas se a meia estiver em uma gaveta incorreta.



Figura 2: Busca de objetos

Da mesma forma, um dado específico é mais facilmente encontrado se ele estiver armazenado em um banco de dados devidamente estruturado por tabelas coerentes e exclusivamente sobre um aspecto do problema que está-se trabalhando. Como dito antes, para o banco de dados tanto faz se o registro estiver armazenado em uma única tabela ou não, entretanto serão consumidos menos recursos computacionais (leitura de disco, memória, tempo de processador, etc.) se este registro estiver em uma tabela específica para o seu armazenamento.

Um outro fato comum entre nossos exemplos é que ambos não criticam o conteúdo armazenado. Você jamais verá um guarda-roupa impedir que você guarde outro item que não roupa dentro dele. Por exemplo, você poderá guardar em seu guarda-roupa uma panela de pressão e mesmo assim não será criticado por isso, além do que a panela estará segura do meio externo, ou seja, o guarda-roupa cumprirá seu objetivo de manter o item guardado protegido. No banco de dados também existe a possibilidade de inserir registros em tabelas estranhas ao seu propósito, mesmo que este registro não esteja coerente com o propósito inicial da tabela, ou seja, nada nos impede

de inserir um atributo sobre livro (nome por exemplo) em uma tabela criada para armazenar dados sobre médicos, basta que o atributo sobre livro caiba na tabela de médicos para a ação ser concluída com sucesso<sup>1</sup>.

Embora os paralelos traçados até aqui sejam válidos, um banco de dados é muito mais do que um guarda-roupa, ou uma agenda de telefones ou livro de receitas. Na verdade é um sistema computacional complexo, seguro, bastante robusto e capaz de armazenar grande volume de informações.

Ainda sobre os bancos de dados podemos destacar a sua importância no desenvolvimento dos sistemas informatizados. Acredito que o banco de dados é a base para quaisquer sistemas que objetiva o registro e a recuperação das informações originadas ou resultantes das ações e/ou eventos ocorridos no próprio sistema. Praticamente toda transação informatizada do mundo fica registrada em um banco de dados, quer seja o registro de uma compra em um site de e-commerce ou dados de uma consulta médica, ou de uma multa de trânsito, ou da produção de um bem ou ainda entre várias outras infinitas possibilidades. Tudo hoje está de alguma forma armazenado dentro de um banco de dados.

Em nosso curso objetivamos que você compreenda o funcionamento do banco de dados e também o que são dados, como organizá-los de forma estruturada e como manipulá-los de forma segura e eficiente.

Para isso nosso curso foi pensado em três partes:

1ª Parte - Organização dos Dados: aqui serão expostos os modelos de dados e as técnicas de levantamento de requisitos, modelagem e refinamento de dados.

2ª Parte - Comandos de manipulação de dados: estudaremos a linguagem SQL (Structure Query Language) para construir o banco de dados e manipular os registros neles contidos.

3ª Parte - Segurança: dedicado a exposição de conceitos e comandos necessários para garantir o acesso restrito aos dados e também técnicas de Backup e Recovery que consistem em basicamente criar cópias de segurança do banco de dados.

Vale destacar que serão apresentados diversos exemplos e **centenas** de exercícios para a devida fixação dos conceitos.

Desejo que você tenha a melhor experiência possível nesse curso e para isso proponho um pacto mútuo de dedicação ao conteúdo teórico e comprometimento com as atividades sugeridas.

Bons estudos,  
Prof. Gilberto Assis Oliveira

---

1 Existem opções de programação (triggers, constraints) que podem diminuir ou impedir dados incoerentes com o propósito da tabela.

## 1.1 – Banco de Dados X Sistema Gerenciador de Banco de Dados (SGBD)

Embora conceitualmente muito próximos são entidades diferentes no nosso escopo de trabalho. Como já definido um banco de dados é a entidade responsável pelo armazenamento dos dados, já o SGBD consiste em sistema que fará a gestão de um ou mais bancos de dados, assim um único SGBD poderá ter sob sua tutela vários bancos de dados. O SGBD tem como premissa garantir a disponibilidade, a segurança e a integridade dos bancos de dados.

Para ser considerado um Sistema Gerenciador de Banco de Dados a ferramenta deve ser capaz de responder a grandes desafios como:

- Eliminar redundância de dados
- Eliminar inconsistências
- Diminuir dificuldades em acessar os dados
- Garantir a integridade dos dados
- Acessos múltiplos e concorrentes
- Garantir a segurança de acesso

Essas são as razões pelo qual não podemos considerar uma agenda de telefones como um banco de dados, a agenda não cumpre com todos esses requisitos.

## 1.2 – Dado X Informação

Aqui também temos sobreposição de definições, também pelo fato de serem muito próximos e comumente relacionados da mesma forma.

- Dado: nada mais é do que uma entrada do sistema, ou seja, corresponde ao registro de algo que se deseja guardar e manter disponível para consultas futuras. Esse registro por si só não tem valor semântico para o usuário do sistema.

- Informação: é tudo aquilo que se obtém a partir da manipulação dos dados armazenados no banco de dados, ou seja, é a resposta com valor semântico para a questão do usuário.

Vejamos a representação dos registros de venda na tabela abaixo<sup>2</sup>:

DATA	PRODUTO	QUANTIDADE	VALOR TOTAL
01/03	MOUSE	13	286
01/03	TECLADO	1	27
02/03	MONITOR	5	2500
12/03	MONITOR	1	500
12/03	MONITOR	2	1000
12/03	TECLADO	2	54

Tabela 1: Registro de vendas

---

2 Nesse exemplo não estão sendo consideradas as regras de normalização de banco de dados.

Cada valor registrado na tabela representa um dado que por si só não traz significado, observe que se mostrarmos somente o dado `MOUSE` para um usuário, ele não conseguiria dar um sentido a isso (isso é um produto? Qual o valor? É o mais caro?). Agora, se esse valor for resultado de uma ação de manipulação do banco de dados ele terá um significado. Por exemplo, se o usuário deseja saber qual foi o produto mais vendido, o banco de dados deverá pesquisar em todos registros para obter essa informação e então retornar a resposta `MOUSE`. Perceba que agora `MOUSE` deixa de ser apenas um valor e se torna uma informação para o usuário.

De certa forma a informação nada mais é do que o dado trabalhado pelo banco de dados com o objetivo de responder a uma questão do usuário.

Vale destacar que é muito comum que autores, técnicos e usuários tratem dados como informação e vice-versa, na verdade com muito pouco ou nenhum prejuízo de definição, por isso não se chateie ao ouvir usuários mais experientes tratarem dados como informação e informação como dado 8').

## 1.3 – Atores do Sistema

É importante compreender quais atores estão envolvidos no sistema, em especial para definir qual o papel e contribuição de cada um.

- Arquiteto de Dados (AD) - pessoa responsável por idealizar o banco de dados, seu papel é, em conjunto com o cliente, compreender o problema e definir seu escopo, bem como o seu propósito. Esse profissional deve ter não só conhecimento técnico avançado sobre definições do banco de dados, mas também ter a habilidade de extrair do usuário final as informações corretas para o desenvolvimento do projeto do banco de dados. Observe que não basta a este profissional ser um excelente técnico, ele também deve saber conversar, saber ouvir e principalmente saber filtrar as necessidades do usuário.

O produto produzido pelo AD é um modelo de dados contendo todas as informações necessárias para a construção do banco de dados, ou seja, quais tabelas, quais tipos de dados, perspectiva de tamanho inicial e de crescimento do banco de dados, além de possíveis necessidades especiais de hardware e infraestrutura.

- Administrador de Banco de Dados (DBA) - esse é especialista na ferramenta (SGBD) no qual o banco de dados será construído. Seu trabalho consiste em criar um banco de dados de acordo com as especificações passadas pelo Arquiteto de Dados e também garantir sua disponibilidade, eficiência, segurança e planos de recuperação de desastres.

- Desenvolvedor - é aquele responsável por criar o sistema que vai interagir com o banco de dados criado pelo DBA, com o objetivo de atender as necessidades do usuário final. Isto é feito através de telas, formulários de entradas de dados e relatórios.

- Usuário Final - é o principal ator do sistema, pois é em função dele que tudo será criado. De nada valerá um sistema, por maior e mais complexo que seja, se ele não atender as expectativas do usuário final.

# 1ª PARTE – Organização de Dados

## Capítulo 2 – Modelos de Banco de Dados

### 2.1 – Introdução

Por definição um modelo de banco de dados é um conjunto de conceitos e regras rígidas que podem ser usados para descrever os requisitos de um banco de dados necessário para atender um sistema do usuário. Para tal, o modelo de dados deverá obedecer as definições colhidas pelo Arquiteto de Dados e especificadas pelo Usuário durante as várias sessões de entrevistas e levantamentos de requisitos. Espera-se que o modelo descritivo final seja condizente ao cenário e aos objetivos apresentados pelo Usuário.

Observe o seguinte cenário: uma empresa qualquer precisa guardar informações sobre seus funcionários como setor em que trabalha, data de admissão e salário. Além disso é necessário saber quais são os respectivos dependentes com nome e idade.

ADMISSÃO	SETOR	NOME_FUN	SALARIO	NOME_DEP	IDADE_DEP
03/04/2010	Marketing	Mestre dos Magos	2.765	Sheila	16
03/04/2010	Manutenção	Mestre dos Magos	2.765	Bobby	12
14/08/2001	Diretoria	Astrid	33.432	Solução	23
22/11/2011	Suporte	SGT. Calhoun	1.102		
03/04/2010	Marketing	M. dos Magos	2.765	Uni	2

Tabela 2: Registro de Funcionários e Dependentes

Ok! Guardar pequenas quantidades de dados pode ser possível da maneira demonstrada acima, mas e se essa empresa fosse um multinacional com 15 mil funcionários e uns 20 setores? É fácil perceber que concentrar todas essas informações em uma planilha não será tarefa fácil, aliás certamente haverá problemas de inconsistências de informações.

Você é capaz de garantir que na tabela acima Mestre dos Magos e M. dos Magos é a mesma pessoa? Aliás, em qual setor o Mestre dos Magos trabalha? Construir uma solução para esse cenário de forma estruturada, consistente e coerente é a grande missão do modelo de dados.

### 2.2 – Tipos de Modelos de Dados

Para resolver o cenário apresentado acima foram propostos vários modelos de dados que estão diretamente relacionados ao tipo de banco de dados que deverão representar: Hierárquico, Em Rede, Relacional, Orientado a Objetos, NoSQL e outros. Em nossos estudos nos fixaremos no modelo de dados para um banco de dados RELACIONAL, isto porque esse paradigma de banco de dados é o mais usado pelo mercado de trabalho e também suficiente para resolver a quase totalidade dos cenários existentes.



Um modelo de dados pode ser do tipo CONCEITUAL ou LÓGICO.

- **Modelo de Dados Conceitual** - (alto nível) apresentam os dados tal qual o usuário o percebe, ou seja, com as representações de todas as características relatadas pelo usuário mas de maneira simples e sem (ou quase sempre) detalhamento técnico sobre outros aspectos do sistema tal qual rede, disco rígido, sistema operacional entre outros. Neste modelo os dados são representados apenas três tipos básicos de componentes: **entidades**, **atributos** e **relacionamentos** que são suficientes para representar qualquer cenário (problema/minimundo) apresentado pelo usuário.

- **Modelo de Dados Lógicos** - (baixo nível) neste os dados são representados com detalhes sobre onde e como serão armazenados, além dos respectivos relacionamentos. Neste modelo os dados são representados basicamente por **tabelas**, **colunas**, **tipos de dados**, **visões** e **relacionamentos**.

Será a partir de um desses modelos que os técnicos, com a linguagem SQL, construirão o próprio banco de dados usando o SGBD escolhido (Oracle, MS SQLServer, MySql, etc).

A partir desse ponto daremos início ao estudo do **modelo conceitual**.

## 2.3 – Entidades

São simbolizadas como um retângulo no modelo de dados conceitual e são identificadas pelo nome no seu interior.



ENTIDADE

Figura 3: Entidade

As entidades têm como função representar os objetos do cenário indicado pelo usuário, isto é, tudo aquilo que é importante guardar no banco de dados, aliás somente aquilo que é fundamental para atender a necessidade do usuário. Deve-se saber que, embora um banco de dados seja capaz de armazenar volumes gigantescos de dados, sempre devemos ter o cuidado de armazenar somente os dados necessários para atender o cenário indicado, entenda como dados necessários somente aqueles que servirão como valor de respostas as questões do usuário. Por exemplo, guardar informações de filiação do usuário de nada servirá para um sistema de gestão de aluguel de imóveis.

Acumular dados desconexos ao cenário do usuário pode trazer problemas para a gestão do banco de dados, quer seja no desempenho, quer seja nas rotinas de backup (cópias de segurança) e até mesmo na incidência de corrupção de arquivos.

As entidades podem representar os objetos reais do cenário, como exemplos AUTOR, LIVRO ou EMPRÉSTIMO para um sistema de gestão de biblioteca. Você deve saber que não pode haver duas entidades de mesmo nome e/ou com o mesmo objetivo no modelo de dados, ou seja, toda entidade é distinta. Os objetos representados pelas entidades podem ser Concretas como AUTOR e LIVRO ou Abstratas como EMPRÉSTIMO.

Por vezes, reconhecemos entidades distintas dentro do modelo de dados que representam objetos com o mesmo conjunto de características, para esses casos é possível e desejado agrupá-las em uma única entidade mais genérica.

Vejamos, caso a biblioteca faça empréstimos de livros, revistas e periódicos para todos esses objetos as características necessárias para a gestão da biblioteca serão similares, provavelmente nome, número de páginas, autor, edição e idioma. Desse modo, não faz sentido ter uma entidade para cada um desses objetos, pode-se ao invés disso criar a entidade VOLUME que contemple a todos eles. Isso tornará o modelo mais enxuto e mais simples de interpretar.

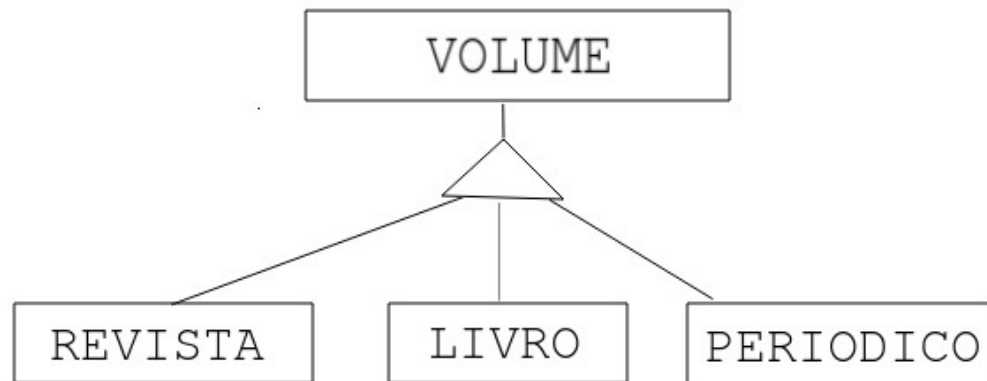


Figura 4: Entidade

Novamente é importante alertar sobre a escolha das entidades que serão criadas no modelo, você sempre deverá ter em mente que só se deve colocar dentro do banco de dados aquilo que será útil para o usuário final. Desse modo, tenha sobriedade na criação desses elementos.

## 2.4 – Atributos

Representam as características da entidade que serão armazenadas no banco de dados, ou seja, assim como a **entidade é sobre o quê** queremos registrar, os **atributos são quais características** serão armazenadas. Voltemos a biblioteca, definimos AUTOR, LIVRO ou EMPRÉSTIMO como entidades para o nosso sistema, certo? Agora, quais informações sobre essas entidades serão pertinentes a nosso sistema?

Sobre o AUTOR temos o NOME, DATA DE NASCIMENTO e CONTATO como características relevantes.

Para LIVRO poderíamos gravar dados sobre seu NOME, DATA DE LANÇAMENTO, EDIÇÃO, NÚMERO DE PÁGINAS e várias outras características úteis para a biblioteca.

Para EMPRÉSTIMO poderíamos registrar a DATA DO EMPRÉSTIMO, IDENTIFICAÇÃO DO USUÁRIO, IDENTIFICAÇÃO DO FUNCIONÁRIO, IDENTIFICAÇÃO DOS LIVROS e DATA DE DEVOLUÇÃO.

Perceba que há uma infinidade de possíveis características de nosso objeto que podem ser registradas nas entidades, entretanto, mais uma vez, você deverá ter o cuidado de registrar apenas o que é útil para o sistema. O acréscimo do atributo SEXO DO USUÁRIO na entidade EMPRÉSTIMO dificilmente será usado como informação pertinente para o sistema, assim esse atributo somente consumiria recursos do próprio banco de dados.

Como já foi definido, os atributos representam as características da entidade, saiba que essas características são agrupadas em três tipos específicos e com objetivos distintos dentro do modelo de dados.

- COMUM - esse é o tipo de atributo usado apenas para representar as características da entidade e não tem função dentro da estrutura do modelo de dados.

- CHAVE PRIMÁRIA (PK) - é um atributo fundamental para o modelo de dados, pois tem a função de garantir que um determinado registro seja único dentro da entidade. Isso é importante para criarmos as relações entre as entidades, já que em toda relação é necessário que os atributos que se relacionam tenham dependência exclusiva com seu par.

Para ser chave primária o campo não pode conter valores nulos nem repetidos, com isto deve-se escolher características do objeto que pertençam somente a ele como CPF para pessoa ou PLACA para veículos. O mesmo raciocínio se aplica ao contrário, pois campos que permitem valores repetidos não são candidatos a serem chave primária, por exemplo NOME para pessoa não é uma boa indicação devido à possibilidade de termos homônimos no banco, e nem MODELO para veículo pela mesma razão.

- CHAVE ESTRANGEIRA (FK) - outro atributo que contém função dentro do modelo de dados, neste caso ele representa o atributo na tabela filha que está relacionado com o atributo chave primária na tabela pai.<sup>3</sup>

Toda chave estrangeira representará um único valor da chave primária da tabela pai, daí a necessidade da PK ser exclusiva. Entretanto, o contrário não é verdade, pois uma chave primária poderá se relacionar com nenhuma ou várias chaves estrangeiras da tabela filha. Isso pode ser observado mais adiante na explicação da Tabela 3.

Antes de irmos para os exemplos práticos vamos demonstrar como estes atributos são representados no modelo de dados.

COMUM - representado por uma haste contendo um círculo de linha contínua e sem preenchimento.

CHAVE PRIMÁRIA (PK) - representado por uma haste com um círculo totalmente preenchido.

CHAVE ESTRANGEIRA (FK) - representado por uma haste contendo um círculo de linha tracejada e sem preenchimento.

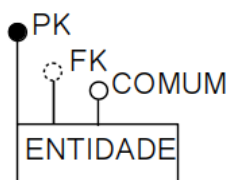


Figura 5: Entidade e seus atributos

Não há limite de atributos para uma entidade, a não ser o bom senso de restringir esses atributos somente a aqueles necessários ao cenário estudado. Você deve estar ciente, no entanto, que o campo CHAVE PRIMÁRIA é único, não existe entidade com

---

3 O conceito de relacionamento entre tabelas e a definição de PAI e FILHO serão apresentadas mais adiante.

mais de uma chave primária, entretanto uma chave primária poderá ser composta de vários campos. Essa definição é muito importante e será estudada em um capítulo próprio que tratará da normalização do modelo de dados.

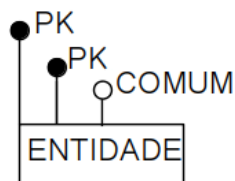


Figura 6: Chave primária composta

## 2.5 – Tabela Filha e Tabela Pai

Antes de continuarmos com o assunto relacionamento, se faz necessário definir o que é tabela PAI e o que é tabela FILHA. Em primeiro lugar deve-se saber que esses conceitos são exclusivos as tabelas que tenham relacionamento com alguma outra.

- TABELA PAI: essa é a entidade que contém o dado original da relação e que será passado para a outra tabela como referência de dependência e determinação.
- TABELA FILHA: entidade que receberá o valor de referência da tabela original e que servirá como valor de ligação único e inequívoco entre elas (entidades).

Vejamos as seguintes relações:

### a) PESSOA e AUTOMÓVEL

Para esta relação a entidade PESSOA é a PAI, isto porque, todo AUTOMÓVEL estará ligado a uma PESSOA, ou seja, todo automóvel terá um proprietário (PESSOA) e deverá conter alguma característica ou dado que referencie esse proprietário de modo exclusivo. Perceba que o contrário não é verdade pois nem toda PESSOA estará ligada a um AUTOMÓVEL.

PESSOA		AUTOMOVEL		
COD_PESSOA	NOME_PESSOA	PLACA	ANO	COD_PESSOA
1	Nelson Piquet	CDX-9900	2001	2
2	Ayrton Senna	GWE-0032	2001	1
3	Alain Prost	HBA-1330	2005	1
		OPW-9877	2017	1

sem carro

Tabela 3: Registros de PESSOA e AUTOMOVEL

## b) RECEITA e PACIENTE

Aqui ocorre o mesmo fenômeno: Toda RECEITA pertence a um PACIENTE, por isso a necessidade de existir um dado em RECEITA que o identifique. Nesse caso PACIENTE é PAI e RECEITA é FILHO.

PACIENTE		RECEITA		
COD_PAC	NOME_PAC	COD_R	DATA_R	COD_PAC
1	Goku	1	22/01/2019	3
2	Vegeta	2	22/01/2019	1
3	Freeza	3	23/01/2019	2

Tabela 4: Registros de RECEITA e PACIENTE

## 2.6 – Relacionamentos

Já foi dito anteriormente que as entidades representam um único objeto do cenário estudado. Ora, então fica óbvio que um modelo de dados conterá várias entidades representando cada qual um aspecto do cenário e que necessariamente deverão interligarem uma a outra de forma a criar relações entre suas informações.

Por exemplo, se a entidade EMPRÉSTIMO trata exclusivamente dos dados de um empréstimo, como saber quem foi o usuário que realizou esta ação?

- Simplesmente relacionando o atributo que identifica unicamente esse usuário (PK) com um atributo na tabela empréstimo que referenciará ao mesmo dado da tabela cliente.

Observe as tabelas abaixo:

CLIENTE		EMPRESTIMO		
COD_CLIENTE	NOME_CLIENTE	COD_EMP	COD_CLIENTE	VALOR_EMP
1	Zé Colméia	1	3	5.000
2	Manda Chuva	2	1	20.000
3	Babalu	3	1	12.000

Tabela 5: Registro de Clientes e Empréstimos

Analisando as tabelas é possível associar o empréstimo de código 1 e valor de 5.000 com o cliente Babalu, isto porque existe um valor comum às duas tabelas, ou seja, um valor responsável pela relação entre as mesmas. Neste caso o campo comum entre as tabelas CLIENTE e EMPRESTIMO é o COD\_CLIENTE, este campo é uma chave primária (PK) na tabela CLIENTE porque cada empréstimo pertencerá a um único cliente e desse modo quando nos referenciarmos a um valor código de cliente na tabela CLIENTE esse valor deverá representar um único cliente (valor único e inequívoco) e chave estrangeira (FK) na tabela EMPRESTIMO porque embora esteja presente nessa tabela esse valor representa um registro que está em outra tabela, no caso CLIENTE.

Recapitulando de forma mais direta:

- COD\_CLIENTE é PK na tabela CLIENTE e FK na tabela EMPRESTIMO, para relacionar as tabelas de modo que todo empréstimo pertença exclusivamente a um único cliente.

A regra de referência de que um campo CHAVE ESTRANGEIRA (FK) está relacionado a um único campo CHAVE PRIMÁRIA (PK) é fundamental para o desenvolvimento de todo o modelo de dados e também para a eliminação das possíveis redundâncias. Se assim não o fosse, não seria possível afirmar qual cliente realizou o empréstimo.

### 2.6.1 – Cardinalidade

As cardinalidades fazem parte da notação do relacionamento entre entidades e tem como objetivo indicar qual o grau da relação, toda relação tem um par de cardinalidades para indicar como essa relação acontece. É importante saber que as cardinalidades de uma relação não interferem em outra relação, de tal modo que ao analisar uma relação não é necessário olhar as demais.

Recapitulando de forma mais direta:

Toda relação tem um par de cardinalidades que indicam como a relação acontece e toda cardinalidade diz respeito exclusivamente a uma única relação sem interferência as demais relações existentes.

A representação da cardinalidade se dá na forma de (x, y) onde x representa o menor grau da relação e y o grau máximo, sempre nessa ordem.

As relações podem conter os seguintes valores:

(0, 1) - lê-se no mínimo zero e no máximo 1 e quer dizer que a relação é de nenhuma ocorrência ou no máximo uma ocorrência. Um exemplo seria na relação entre CLIENTE e EMPRESTIMO, e isso quer dizer que o sistema permite que um cliente contraia nenhum empréstimo ou um único empréstimo.

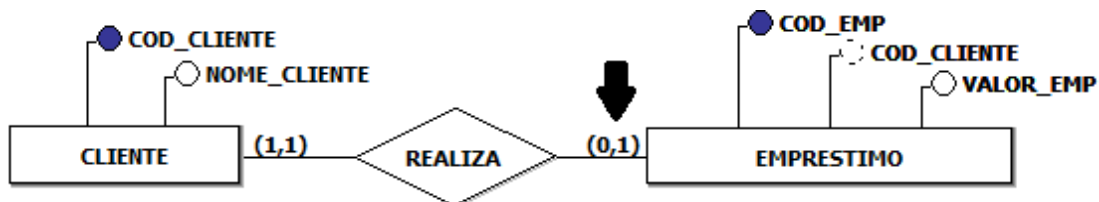


Figura 7: Cardinalidade (0,1) do CLIENTE

Observe que estamos analisando a relação de CLIENTE para EMPRESTIMO e não o contrário, ou seja, quais as possíveis relações entre um cliente e os empréstimos.

(1, 1) - lê-se no mínimo um e no máximo 1 e quer dizer que a relação é de no mínimo uma ocorrência e no máximo uma ocorrência. Ainda considerando o mesmo cenário teríamos que um empréstimo deverá estar vinculado a pelo menos um cliente e no máximo um cliente. Veja que isto tem sentido, pois todo empréstimo pertence a um cliente e somente a ele. Agora estamos analisando a relação entre EMPRESTIMO e CLIENTE.

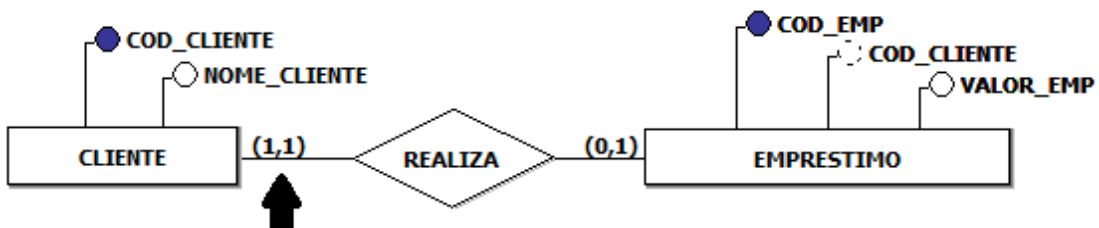


Figura 8: Cardinalidade (1,1) do EMPRESTIMO

(0, N) - lê-se no mínimo zero e no máximo N e quer dizer que a relação é de nenhuma ocorrência ou no máximo várias ocorrências. Considerando o mesmo cenário de CLIENTE e EMPRESTIMO essa relação indicaria que agora o cliente poderia contrair nenhum ou vários empréstimos.

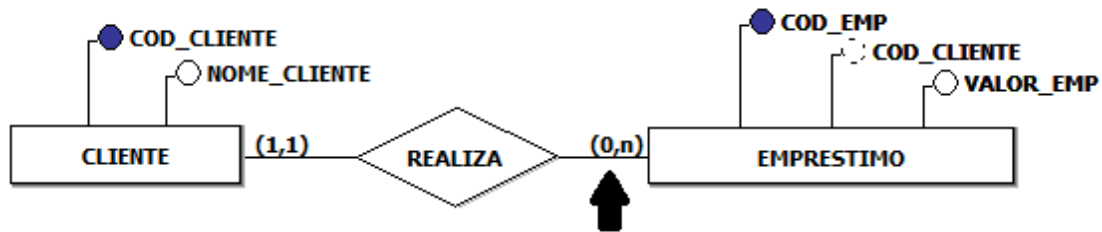


Figura 9: Cardinalidade (0,n) do CLIENTE

(1, N) - lê-se no mínimo um e no máximo N e quer dizer que a relação é de no mínimo uma e no máximo várias ocorrências. Neste caso, o cliente OBRIGATORIAMENTE deve conter pelo menos um empréstimo e no máximo vários deles. Repare que para essa situação todo cliente cadastrado terá pelo menos um empréstimo.

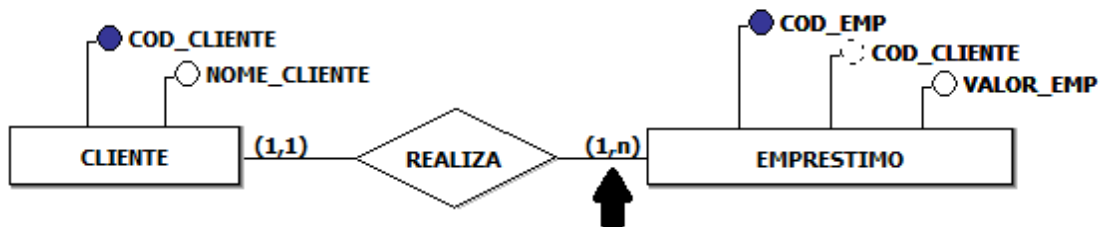


Figura 10: Cardinalidade (1, n) do CLIENTE

Recapitulando de forma mais direta:

A análise da cardinalidade se faz observando como uma entidade se relaciona com a outra, ou seja, quais as possíveis interações entre a entidade referenciada com o seu par na relação. Quando partimos da entidade CLIENTE por exemplo analisamos todas as possíveis relações com a entidade EMPRESTIMO e não ao contrário. Para analisar a relação oposta (EMPRESTIMO para CLIENTE) seria necessário verificar suas próprias ocorrências.

Assim como tudo no modelo de dados, **a análise da relação e respectivas cardinalidades devem obedecer sucintamente as regras de negócio** do cenário observado.

## 2.7 – Relação PAI e FILHA

Um conceito importante no modelo dados é a definição do papel da entidade dentro de uma relação. Em toda relação entre entidades promove-se o papel de PAI e FILHA o que na prática define quem passará o atributo de ligação entre as entidades, nesse caso a entidade PAI fornecerá um valor de ligação para a entidade FILHO. Vejamos:

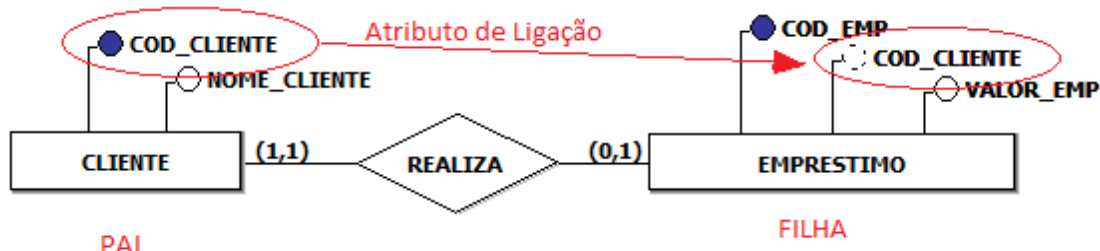


Figura 11: Modelo de Dados para Controle de Empréstimos

Na relação mostrada na Figura 11 a entidade CLIENTE é a entidade PAI, isto porquê é ela quem passa o atributo de ligação COD\_CLIENTE entre as entidades, isto significa que todo registro presente na entidade EMPRESTIMO estará vinculado obrigatoriamente a um registro da entidade CLIENTE, indicado pela cardinalidade (1,1), e desse modo EMPRESTIMO será a entidade FILHA.

Recapitulando de forma mais direta:

A entidade CLIENTE é pai porque a entidade EMPRESTIMO é dependente dela, pois para que haja um empréstimo registrado é necessário que exista antes um cliente registrado. Isso pode ser confirmado pela cardinalidade (1,1) na relação entre EMPRESTIMO e CLIENTE.

## 2.8 – Modelo Completo

A partir desse ponto vamos analisar um modelo de dados completo, ou seja, com todas as representações (entidades, atributos e relações) presentes.

Exemplo 1: Controle de Vendas

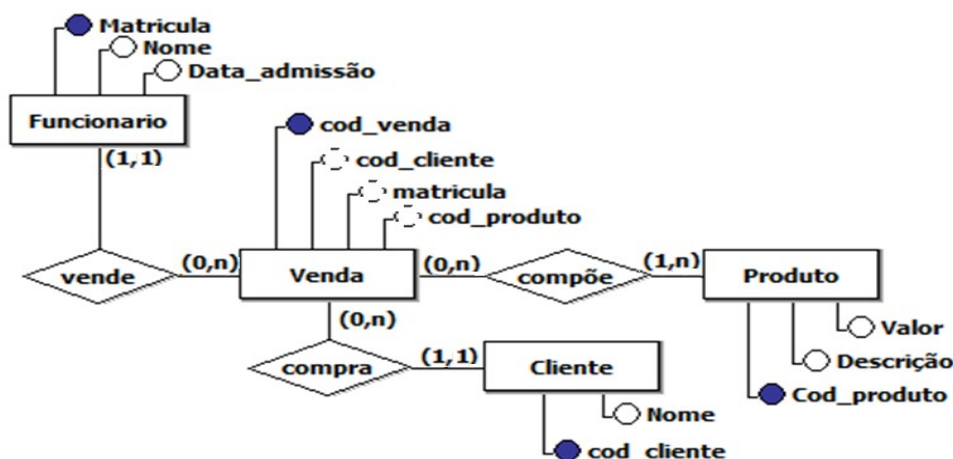


Figura 12: Modelo de Dados para Controle de Vendas



O modelo de dados acima tem como pretensão registrar as vendas de produtos de um estabelecimento qualquer. O objetivo inicial é saber quem vendeu, quem comprou e o que comprou.

Vamos em primeiro lugar identificar quais entidades existem no modelo. No nosso exemplo temos 4 entidades: FUNCIONARIO, VENDA, CLIENTE e PRODUTO.

A entidade FUNCIONARIO contém 3 atributos, a chave primária MATRICULA e os atributos comuns NOME e DATA\_ADMISSAO.

Já a entidade CLIENTE contém apenas dois atributos COD\_CLIENTE que é chave primária e o campo comum NOME.

A entidade PRODUTO contém 3 atributos, a chave primária COD\_PRODUTO e os campos comuns DESCRICAO e VALOR.

E a entidade VENDA contém 4 atributos sendo COD\_VENDA sua chave primária e os demais atributos do tipo chave estrangeira COD\_CLIENTE (indica qual cliente é dono da venda), MATRICULA (indica qual funcionário realizou a venda) e COD\_PRODUTO (quais produtos foram vendidos).

Nesse modelo é possível observar 3 relacionamentos:

- VENDE entre FUNCIONARIO e VENDA
- COMPRA entre VENDA e CLIENTE
- COMPÕE entre PRODUTO e VENDA

Ao analisar cada um deles chegamos as seguintes conclusões:

- Um funcionário pode ter realizado nenhuma venda ou várias delas, isto porque ele se relaciona com a entidade VENDA com a cardinalidade (0,n). Já a venda só pode ter sido realizada por um único funcionário, como indicado na cardinalidade (1,1). Deve-se perceber que nesta relação a entidade FUNCIONARIO é PAI e por isso a sua PK é usada para relacioná-lo a VENDA.

- Um cliente pode ter comprado nenhuma vez ou várias vezes como indicado em sua relação com a entidade VENDA (0,n). Entretanto, a cardinalidade (1,1) indica que toda venda estará vinculada a um único cliente. Aqui a entidade CLIENTE é a PAI porque toda venda depende de um cliente.

- Na relação entre PRODUTO e VENDA temos que todo produto<sup>4</sup> foi vendido uma única vez ou várias vezes e naturalmente toda venda tem pelo menos um produto vinculado ou vários deles. Cabe aqui uma observação: embora seja possível indicar no modelo de dados que duas entidades se relacionam em seu grau máximo com N ocorrência para ambas, na prática isso não é viável para o banco de dados pois fere a regra da 1FN (primeira forma normal)<sup>5</sup>. Para resolver essa anomalia é necessário acrescentar ao modelo uma nova entidade contendo informações de ambas entidades e consequentemente desfazendo a relação NxM.

Não se preocupe, essa resolução será detalhada em aula.

---

4 Considera-se nesse modelo que a entidade produto representa um item de produto e não um produto específico.

5 A 1FN diz que não pode haver campos multivalorados nas entidades. As Formas Normais serão estudadas mais adiante.

## Exemplo 2: Obrigações de um funcionário

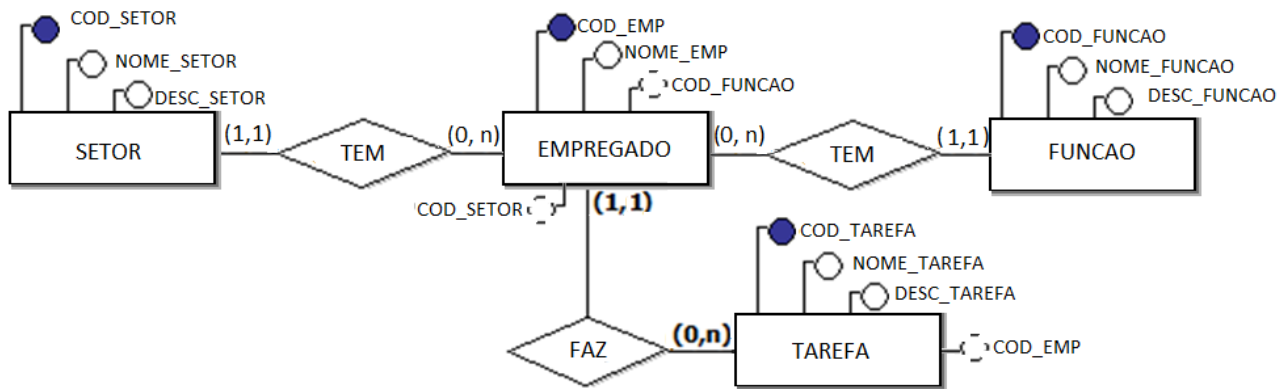


Figura 13: Modelo de Dados para Obrigações de Funcionário

Nesse modelo de dados queremos registrar quais são os empregados, bem como o setor em que trabalha, sua função e quais tarefas ele deve realizar.

Repare que o modelo contém 4 entidades: SETOR, EMPREGADO, FUNCAO e TAREFA.

As entidades são compostas pelos atributos indicados abaixo:

- SETOR: a chave primária COD\_SETOR e os campos comuns NOME\_SETOR e DESC\_SETOR.
- FUNCAO: a chave primária COD\_FUNCAO e os campos comuns NOME\_FUNCAO e DESC\_FUNCAO.
- TAREFA: a chave primária COD\_TAREFA, a chave estrangeira COD\_EMP e os campos comuns NOME\_TAREFA e DESC\_TAREFA.
- EMPREGADO: a chave primária COD\_EMP, o campo comum NOME\_EMP e as chaves estrangeiras COD\_FUNCAO e COD\_SETOR.

Ao interpretar o modelo de dados é possível chegar a conclusão de que:  
SETOR e EMPREGADO

- Todo SETOR pode conter nenhum ou vários empregados (0,n);
- Todo EMPREGADO pertence a um único SETOR (1,1) e como todo empregado depende de um SETOR então nessa relação SETOR será o PAI;

FUNCAO e EMPREGADO

- Toda FUNCAO pode conter nenhum ou vários empregados vinculados (0,n);
- Todo EMPREGADO exerce uma única FUNCAO (1,1) e como todo empregado tem uma FUNÇÃO então nessa relação FUNÇÃO será o PAI;

TAREFA e EMPREGADO

- Um EMPREGADO pode não exercer nenhuma TAREFA ou várias delas (0,n);
- Todo TAREFA é executada exclusivamente por um único EMPREGADO (1,1) e como toda tarefa depende de um EMPREGADO então nessa relação EMPREGADO será o PAI;

## 2.9 – Interpretação do minimundo

Na verdade cada pessoa cria sua própria estratégia de interação com o usuário na busca das regras de negócio para a construção do modelo de dados. Algumas pessoas se valem de formulários com perguntas diretas, outras através de entrevistas e até mesmo através de engenharia reversa.

Em nosso curso realizaremos a coleta dos requisitos do modelo de dados através da descrição dos minimundos, das exposições das regras de negócio e de narrativas simuladas.

Minimundo 1: Durante o levantamento de requisitos de uma editora descobriu-se que um livro é escrito por um único autor e trata de um único assunto específico. Elabore o modelo de dados para esse cenário.

Resolução: é necessário realizar a leitura de forma a compreender o cenário, neste caso o objetivo é apenas registrar os livros, qual autor o escreveu e sobre qual assunto se trata o livro. Aqui podemos observar que interessam somente 3 entidades no qual queremos registrar dados: LIVRO, AUTOR e ASSUNTO, qualquer outra entidade criada nesse cenário será um excesso, ou seja, registros armazenados no banco que não serão usados pelo usuário final.

Sobre as entidades encontradas podemos guardar as seguintes características.

LIVRO: COD\_LIVRO (pk), NOME\_LIVRO, NUM\_PAGINAS, COD\_AUTOR (fk) e COD\_ASSUNTO (fk)

AUTOR: COD\_AUTOR (pk), NOME\_AUTOR

ASSUNTO: COD\_ASSUNTO (pk), NOME\_ASSUNTO, DESC\_ASSUNTO

Repare que **toda entidade tem uma PK** (chave primária) que serão usadas para as devidas relações. Tenha isso como uma verdade! Veja também que a entidade LIVRO é a entidade que recebe as chaves estrangeiras (FK), por isso ela é a filha nas duas relações, uma com a entidade AUTOR e outra com a entidade ASSUNTO.

Ótimo! Agora é só desenhar... para isso colocamos a entidade principal, ou seja, aquela que contém maior número de relações no centro do modelo de dados e em seguida as que se ligam a entidade principal, depois as que se ligam a essas últimas e assim por diante até que todas as entidades estejam presentes no modelo e com as devidas relações.

Resumindo de forma mais direta:

Após colocar a entidade principal no centro do modelo, escolha uma entidade que se relacione com ela, verifique quem é a entidade PAI, passe a chave estrangeira para entidade FILHA e indique os relacionamentos. Depois é só repetir esses passos para as demais entidades.

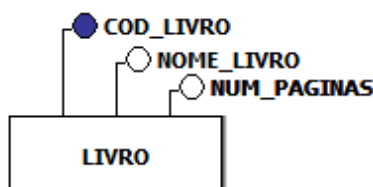


Figura 14: Entidade principal com seus atributos

Perceba que neste momento estamos focados apenas nos atributos relativos a entidade LIVRO, não precisamos nos preocupar ainda com os atributos chaves estrangeiras porque ainda não existe relação.

A figura 15 mostra a inclusão da entidade AUTOR e seus respectivos atributos.



Figura 15: Inclusão da entidade AUTOR e seus atributos.

Mais uma vez, por não existir a relação, indicaremos apenas os atributos da entidade analisada, que neste caso é AUTOR.

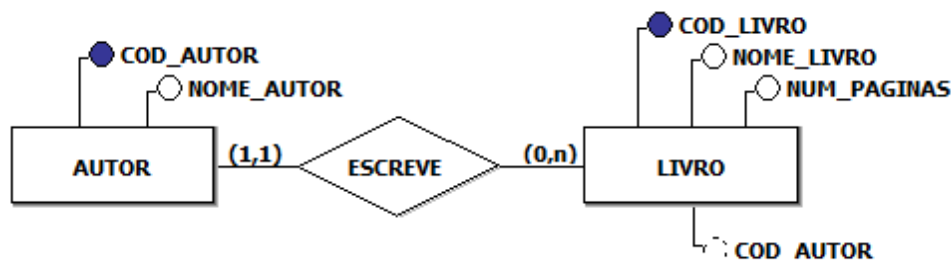


Figura 16: Inclusão da relação ESCREVE e indicação da FK COD\_AUTOR.

Para desenhar o modelo como indicado na Figura 16 foi necessário em primeiro lugar perceber que as duas entidades se relacionam e isso foi possível graças ao cenário proposto no exercício, **“um livro é escrito por um único autor”**. Como o livro depende de um autor para ser escrito (não há livros sem autores) temos que a entidade AUTOR é pai de LIVRO e por isso surge o atributo chave estrangeira COD\_AUTOR na entidade LIVRO.

A cardinalidade indicada deriva do seguinte raciocínio, primeiro, a partir da entidade PAI realizamos 3 perguntas:

- Um autor pode ter nenhuma relação com livro?  
SIM, porque um autor<sup>6</sup> pode nunca ter escrito um livro.
- Um autor pode ter uma única relação com livro?  
SIM, porque um autor pode ter escrito um único livro até hoje.
- Um autor pode ter várias relações com livro?  
SIM, o autor pode ter escrito mais de um livro até hoje.

Observando as perguntas e respostas acima chegamos a conclusão de que a menor relação possível entre AUTOR e LIVRO é de 0 (zero) relações, pois um autor

---

6 Para ser considerado autor basta estar vinculado a entidade AUTOR e não necessariamente ter escrito um livro.

pode não ter escrito nenhum livro até hoje. Já o maior grau de relação será de **n** ocorrências, o que quer dizer mais de uma ocorrência. Na cardinalidade sempre indicamos o MENOR e o MAIOR grau de relação, neste caso, indicaremos **(0,n)**. Repare que estamos analisando a relação de AUTOR para LIVRO, então a cardinalidade estará do lado de LIVRO indicando que o sentido da relação (de autor para livro).

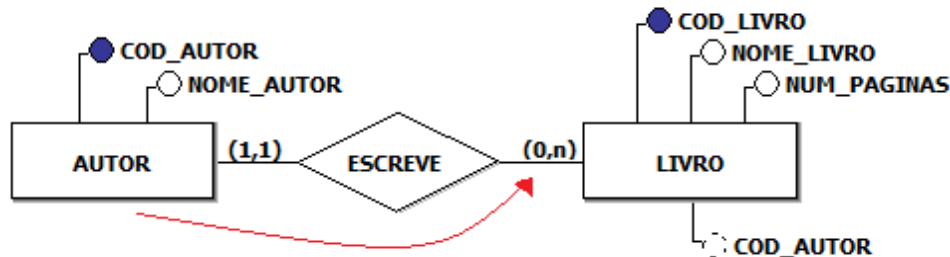


Figura 17: Indicação da cardinalidade entre AUTOR e LIVRO.

Tenha cuidado! A indicação da cardinalidade no lado incorreto da relação incorre em erro.

Agora precisamos analisar a interação contrária, ou seja, de LIVRO com AUTOR (nesse sentido). Novamente faremos as 3 perguntas, mas com foco em LIVRO:

- Um livro pode ter nenhuma relação com autor?  
NÃO, porque para haver livro é necessário que um autor o tenha escrito.

- Um livro pode ter uma única relação com livro?  
SIM, porque o livro tem que ter sido escrito por um único autor. Tal qual mostrado no enunciado **“um livro é escrito por um único autor”**.

- Um livro pode ter várias relações com autor?  
NÃO, o enunciado diz que **“um livro é escrito por um único autor”**.

Feito esse estudo, chegamos a conclusão de que os livros são escritos por no mínimo 1 autor e no máximo 1 autor resultando na relação (1,1). Não se esqueça que essa cardinalidade reflete a relação entre LIVRO com AUTOR logo, deverá ser indicada nesse sentido da relação.

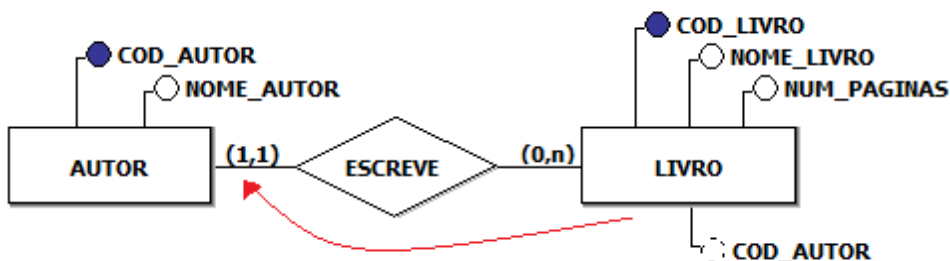


Figura 18: Indicação da cardinalidade entre LIVRO e AUTOR.

Chegou a hora de incluir no modelo de dados a próxima entidade ASSUNTO. Novamente, ainda não precisamos nos preocupar com a chave estrangeira, pois ainda não há relação formada, nesse momento certifique-se que todos os atributos da entidade estudada estejam corretos.

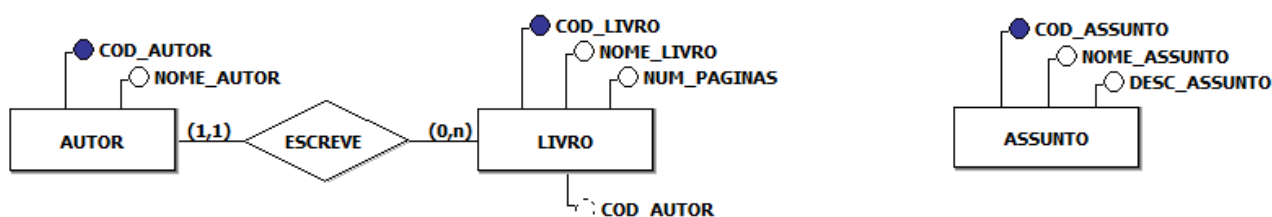


Figura 19: Inclusão da entidade ASSUNTO e seus atributos.

Ao estudar a relação entre as entidades LIVRO e ASSUNTO você perceberá que a segunda é a PAI da relação, pois todo livro **“trata de um único assunto específico”** e isso denota dependência de livro com assunto, o que quer dizer que todo livro deve estar vinculado a um assunto. A consequência dessa observação é a necessidade de indicar a chave primária COD\_ASSUNTO de ASSUNTO como chave estrangeira na entidade LIVRO. Daqui para frente vamos nos ocupar em estabelecer como essas entidades se relacionam, ou seja, indicar as cardinalidades da relação.

Partindo da entidade pai ASSUNTO faremos as 3 perguntas:

- Um assunto pode ter nenhuma relação com livro?  
SIM, porque pode não haver nenhum livro sobre um determinado assunto.
- Um assunto pode ter uma única relação com livro?  
SIM, porque um assunto pode ser tema um único livro escrito até hoje.
- Um assunto pode ter várias relações com livro?  
SIM, o assunto pode ser tema de vários livros escritos.

Observando as perguntas e respostas acima chegamos a conclusão de que a menor relação possível entre ASSUNTO e LIVRO é de **0** (zero) relações, pois um assunto pode não ter sido tema de nenhum livro até hoje. Já o maior grau de relação será de **n** ocorrências, o que quer dizer que um assunto foi tema de vários livros. Não se esqueça que a ordem da cardinalidade tem significado, o menor grau da relação vem primeiro e depois o maior deles resultando em (0,n), além de observar o sentido da relação que vai indicar o local da cardinalidade estudada.

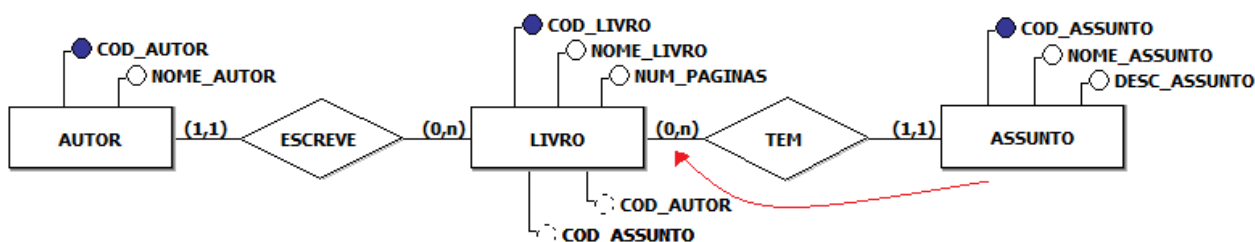


Figura 20: Indicação da cardinalidade entre ASSUNTO e LIVRO.

Novamente faremos as 3 perguntas, mas agora no sentido inverso da relação, de LIVRO para ASSUNTO.

- Um livro pode ter nenhuma relação com assunto?  
NÃO, porque todo livro é sobre um assunto.

- Um livro pode ter uma única relação com assunto?  
SIM, porque o livro trata um único assunto. Tal qual mostrado no enunciado **“trata de um único assunto específico”**.

- Um livro pode ter várias relações com assunto?  
NÃO, o enunciado diz que **“trata de um único assunto específico”**.

O estudo dessa relação resulta que um livro é obrigatoriamente sobre no mínimo 1 assunto e no máximo 1 assunto, resultando na relação (1,1).

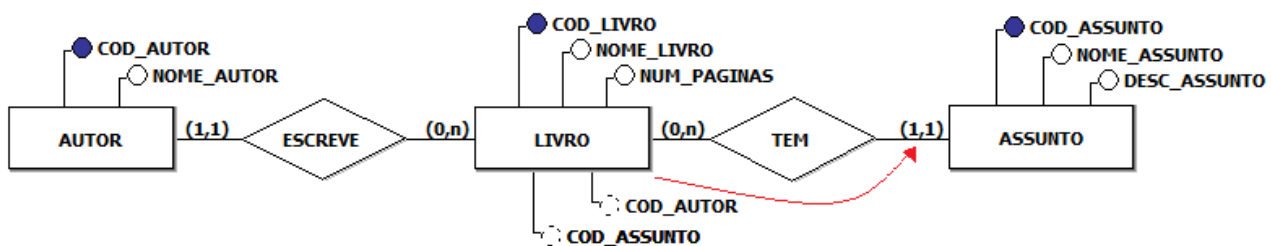


Figura 21: Indicação da cardinalidade entre LIVRO e ASSUNTO.

Dessa forma concluímos o modelo de dados para o cenário apresentado no minimundo 1. Perceba que esse modelo de dados é capaz de atender a qualquer anseio do usuário do sistema, veja só, é possível saber: quais livros estão registrados? Qual autor escreveu um determinado livro? Quantas páginas têm o maior livro? Qual a média de páginas por livro? Quais livros de um assunto qualquer? Para qual assunto um determinado autor já escreveu? E vários outros questionamentos.

## 2.10 – A anomalia do NxM

Para contextualizar esse problema vamos alterar o minimundo 1 para o novo minimundo 2, que prevê a possibilidade de que um livro possa ser escrito por mais de um autor e que é necessário conhecer quantos capítulos cada autor escreveu em cada um dos livros. Essa alteração afeta diretamente a cardinalidade entre LIVRO e AUTOR (nesse sentido) transformando-a em (1,n), pois agora o livro deve ter sido escrito por pelo menos 1 autor e no máximo n autores. Observe essa alteração no modelo de dados.

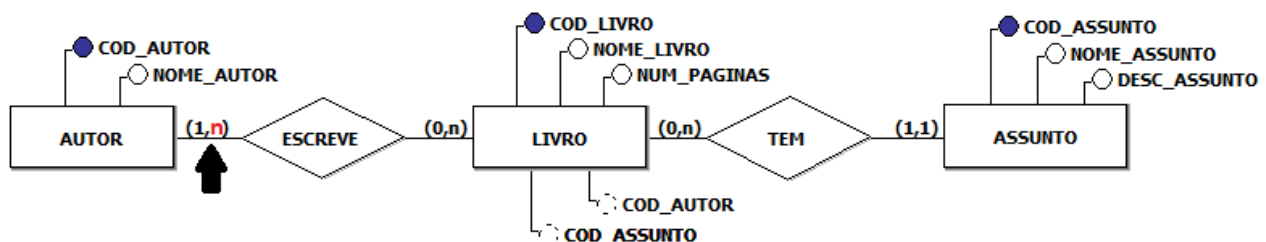


Figura 22: Indicação da cardinalidade entre LIVRO e AUTOR.

- E qual o problema com essa alteração?

Basicamente, é porque ela gera uma anomalia no modelo de dados que consiste nos campos multivalorados. Isso quer dizer que para atender a essa cardinalidade seria necessário que o atributo chave estrangeira COD\_AUTOR na entidade LIVRO recebesse mais de um valor (os códigos dos autores que escreveram o livro) ao mesmo tempo e isso não é possível em um banco de dados. **Cada célula da tabela de um banco de dados só é capaz de receber um único valor.**

No nosso exemplo um LIVRO poderia ter sido escrito por um ou vários autores como indicado na cardinalidade (1,n) entre LIVRO e AUTOR. Logo, teríamos um problema: se não é possível armazenar mais de um valor no mesmo atributo, onde vamos armazenar os autores que escreveram o livro?

A resposta está na criação de outra entidade que servirá apenas como associação entre as tabelas LIVRO e AUTOR. Para isso, a relação entre AUTOR e LIVRO será desfeita e sua nova relação passará pela nova entidade. A nova entidade vai conter os atributos chave primária das duas outras entidades (LIVRO e AUTOR), além dos atributos (QDE\_CAPITULOS) dependentes dessa relação. Dependência nesse caso significa que para retornar um único valor é necessário a combinação dos atributos que compõem a chave primária.

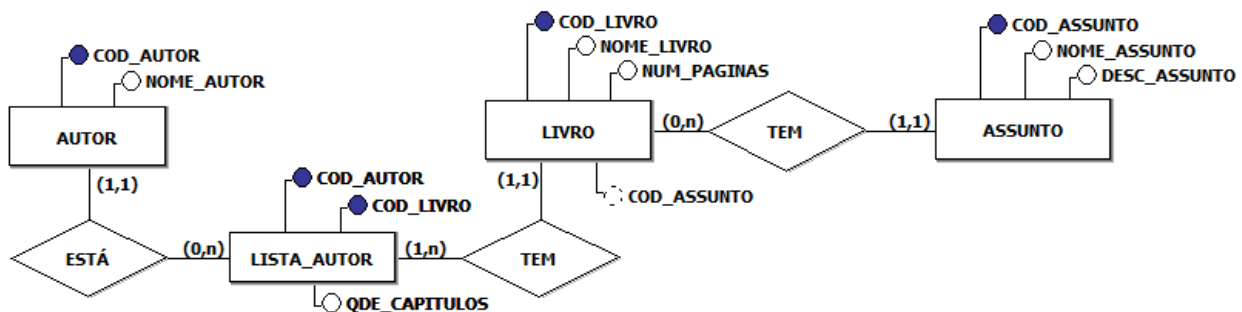


Figura 23: Resolvendo a anomalia NxM.

Observe que a chave estrangeira COD\_AUTOR deixa de existir na entidade LIVRO, isso é devido a exclusão da relação entre LIVRO e AUTOR. Vale aqui um alerta: **Só existe chave estrangeira entre as entidades que se relacionam.**

De modo mais direto: não é possível passar atributo para outras entidades que não se relacionam com a entidade PAI.

Você pode ter algumas perguntas nesse momento...

- Por que o COD\_AUTOR e COD\_LIVRO na entidade livro estão sinalizadas como chave primária (PK)? elas não vêm de outras entidades?

A resposta para essa primeira pergunta é sim! Esses atributos TAMBÉM são chaves estrangeiras, respectivamente COD\_AUTOR da entidade AUTOR e COD\_LIVRO da entidade LIVRO, entretanto, na entidade LISTA\_AUTOR elas terão outro papel mais importante que é definir a chave primária e por isso são sinalizadas como tal.



- Por que duas chaves primárias na entidade LISTA\_AUTOR?

Na verdade **nenhuma entidade terá mais do que uma chave primária!** O que está acontecendo nesse caso é uma **chave primária composta** por dois atributos, já que um atributo sozinho não é suficiente para determinar a unicidade dos registros. E isso sempre acontecerá! Toda vez que um só atributo não for suficiente para identificar um único registro teremos que escolher outros atributos para compor a chave primária até que estes sejam suficientes para retornar um único registro.

Vamos analisar isso mais de perto com auxílio da simulação de registros da entidade LISTA\_AUTOR transcrita na tabela abaixo:

COD_AUTOR	COD_LIVRO	QDE_CAPITULOS
13	1	18
2	2	3
13	2	10
8	2	5
2	3	5
11	3	8

Tabela 6: Registro de LISTA\_AUTOR

Nesse exemplo você deve ser capaz de perceber que existem 3 livros registrados como indicado pelos valores distintos de COD\_LIVRO: 1, 2 e 3 representam livros diferentes no banco de dados. O mesmo vale para autores que nesse caso são 4 autores distintos (2, 8, 11 e o 13) certo? Agora você deve se lembrar do conceito da chave primária que é a unicidade na identificação de registros, ou seja, **toda vez que nos referirmos a um valor de chave primária ele deverá retornar uma única linha da tabela** (registro). Para o nosso exemplo, se for perguntado a QDE\_CAPITULOS do autor de COD\_AUTOR 13 serão retornados duas linhas da tabela:

13	1	18
13	2	10

Tabela 7: Registros vinculados ao COD\_AUTOR 13

Isso quer dizer que, somente o campo COD\_AUTOR não é suficiente para determinar um registro na entidade LISTA\_AUTOR. O mesmo vale para COD\_LIVRO, se queremos saber qual a quantidade de capítulos do livro COD\_LIVRO 2 serão três os registros:

2	2	3
13	2	10
8	2	5

Tabela 8: Registros vinculados ao COD\_LIVRO 2

Desse modo, somente COD\_AUTOR ou somente COD\_LIVRO não são suficientes para garantir a unicidade de um registro, e como já foi dito, toda tabela deve ter um determinante (chave primária). Então, para garantir a unicidade, associaremos mais de um atributo à chave primária até que estes sejam suficientes para a unicidade do registro gerando assim a chave primária composta. Para retornar um único registro de QDE\_CAPITULOS seria necessário informar o valor para cada um dos atributos da chave primária composta por COD\_AUTOR e COD\_LIVRO. Logo, qual a quantidade de capítulos escritos pelo autor COD\_AUTOR 13 no livro de COD\_LIVRO 2?

13	2	10
----	---	----

Tabela 9: Registros vinculados ao COD\_AUTOR 13 e COD\_LIVRO 2

Para essa combinação somente haverá uma única linha (registro) sempre, o que resultará em um valor suficiente para garantir a unicidade dos registros.

## 2.11 – Autorrelacionamento

Esse tipo de relação se dá quando uma entidade se relaciona com ela mesma usando seus próprios atributos como pontos de ligação. Imagine um cenário no qual se deseja registrar as características do empregado: **código**, **nome**, **salário**, **admissão** e o **chefe** desse empregado. Pois bem, como mostrado no modelo abaixo já conseguimos expressar os atributos COD\_EMP, NOME\_EMP, SALARIO\_EMP e ADMISSAO\_EMP na entidade EMPREGADO, uma vez que todas essas características são do próprio empregado.

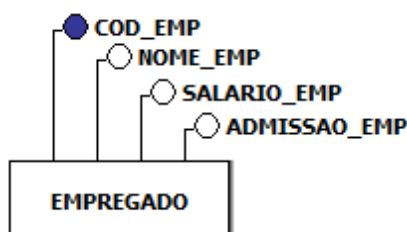


Figura 24: Entidade EMPREGADO.

Mas, e quanto ao chefe? Como faríamos a sua representação? A ideia de criar uma entidade para armazenar CHEFE com os atributos COD\_CHEFE e NOME\_CHEFE está equivocada, pois CHEFE é um EMPREGADO como outro qualquer e desse modo deve ser registrado nesta mesma entidade. Imagina que confusão seria termos um mesmo registro armazenado em duas entidades diferentes? E mais, e se o empregado deixasse de ser chefe como faríamos?

Na verdade existe uma boa saída para esse tipo de dependência: o chamado autorrelacionamento que consiste em relacionar um atributo de uma entidade com outro atributo da própria entidade.

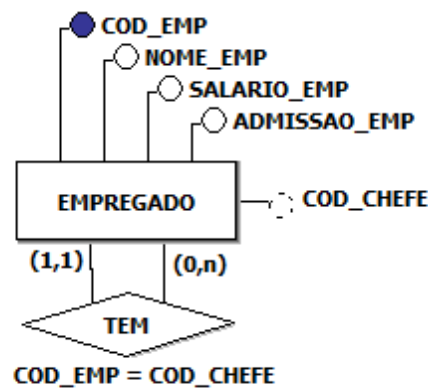


Figura 25: Autorrelacionamento de EMPREGADO.

O modelo apresentado acima mostra o autorrelacionamento da entidade EMPREGADO acontecendo pelos atributos COD\_EMP com COD\_CHEFE, onde o primeiro é a CHAVE PRIMÁRIA da relação e o segundo é a CHAVE ESTRANGEIRA. Nesse relacionamento um empregado chefe pode conter nenhum ou vários empregados subordinados (0,n) e todo empregado subordinado terá um único chefe (1,1). O atributo COD\_CHEFE nulo (sem valor) indica os funcionários que não tem chefe.

## 2.12 – Múltiplas Relações entre duas entidades

Até agora temos tratado situações no qual as entidades se relacionam com outra somente uma vez, como mostrado no exemplo abaixo:

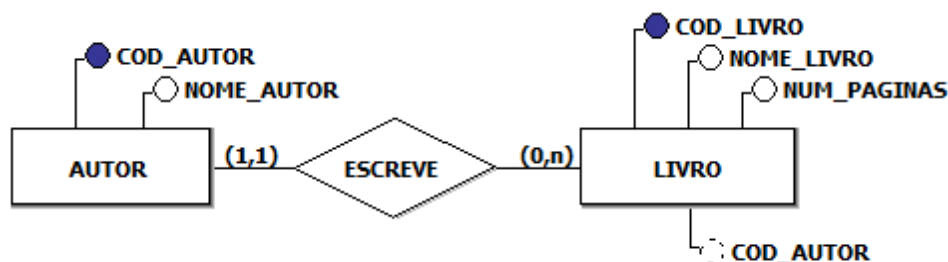


Figura 26: Relacionamento Simples.

Mas, existem casos em que são necessários mais de uma relação entre duas entidades como forma de resolver as interações exigidas pelo cenário. Um bom exemplo disso seria um cenário em que se deseja controlar o envio de requisições entre funcionários. Melhor explicando, controlar quais requisições são enviadas por um funcionário originador para outro de destino. Como a ação se dá entre funcionários, não se justifica criar uma entidade ORIGINADOR e outra DESTINATARIO com atributos de funcionários para resolver esse cenário, na verdade isso geraria um problema de redundância de dados. A boa prática diz que, basta criar uma dupla relação entre as entidades FUNCIONARIO e REQUISICAO para resolver esse problema. Observe no modelo abaixo que a entidade REQUISICAO tem dois campos chaves estrangeiras relacionadas ao atributo COD\_FUNC da entidade FUNCIONARIO, repare também que o nome dos atributos chaves estrangeiras (COD\_ORIGINADOR e COD\_DESTINATARIO) refletem o papel desse atributo no cenário proposto. Sim, como você percebeu **não há obrigatoriedade de manter o nome do campo chave primária no campo chave**

**estrangeira**, normalmente se faz isso para facilitar a compreensão do modelo, entretanto no modelo abaixo os nomes COD\_ORIGINADOR e COD\_DESTINATARIO são mais legíveis ao usuário.

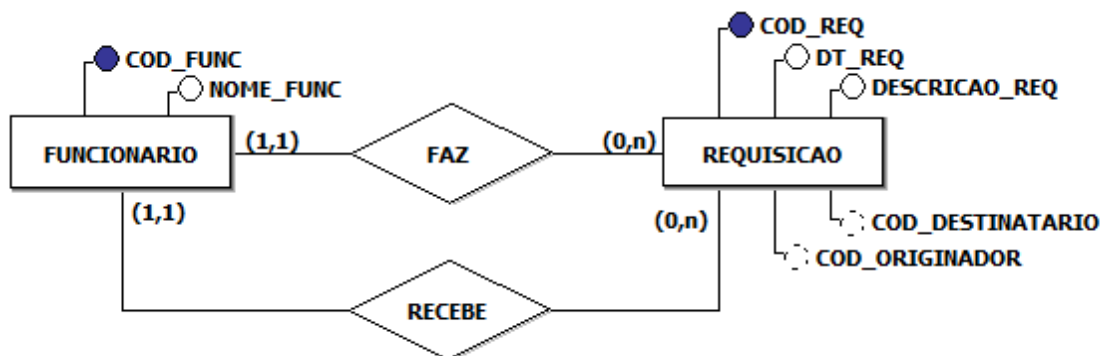


Figura 27: Relacionamento Composto.

Para melhor ilustrar esse modelo vamos recorrer as tabelas que transcrevem os possíveis registros para FUNCIONARIO e REQUISICAO.

FUNCIONARIO	
COD_FUNC	NOME_FUNC
1	Tony Stark
2	Selina Kyle
3	Diana Prince

Tabela 10: Registros da tabela FUNCIONARIO

REQUISICAO				
COD_REQ	DT_REQ	DESCRICAO_REQ	COD_ORIGINADOR	COD_DESTINATARIO
1	01/02	Troca de Mouse	2	3
2	02/02	Avaliar rede interna	3	1
3	02/02	Pouco espaço no HD	3	1
4	02/02	Meu mouse não funciona	1	3
5	03/02	Analisar o backup	3	1

Tabela 11: Registros da tabela REQUISICAO

Quando analisamos as tabelas fica fácil perceber quem originou a requisição e para quem ela foi destinada. Por exemplo, a requisição de COD\_REQ 1 foi encaminhada pela Selina Kyle para a Diana Prince. Já a requisição 5 foi encaminhada pela Diana Prince para o Tony Stark.

## 2.13 – Migrando para o modelo lógico

Sempre que aprendemos os conceitos e fundamentos do modelo conceitual, na verdade estamos também nos preparando para o modelo lógico. Isto porque existe um paralelo entre eles, de tal modo que, todo modelo conceitual gera um modelo lógico idêntico em suas características e representações. O que muda é apenas a forma de expressar essas representações.

Tabela de associação entre os elementos do modelo conceitual com o modelo lógico.

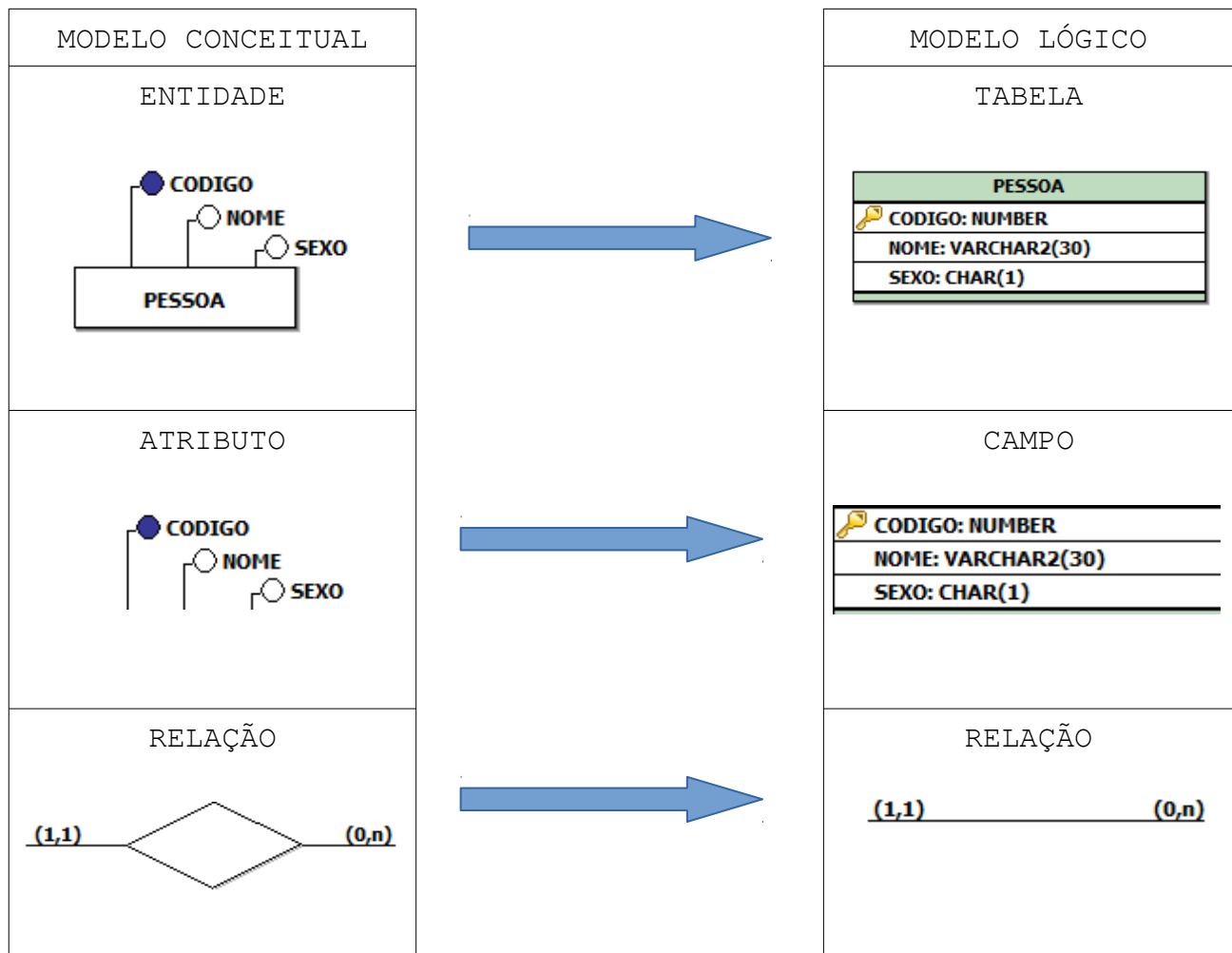


Tabela 12: Associação entre os Modelos

A transposição do modelo conceitual apresentado na Figura 26 para o modelo lógico ficaria assim:

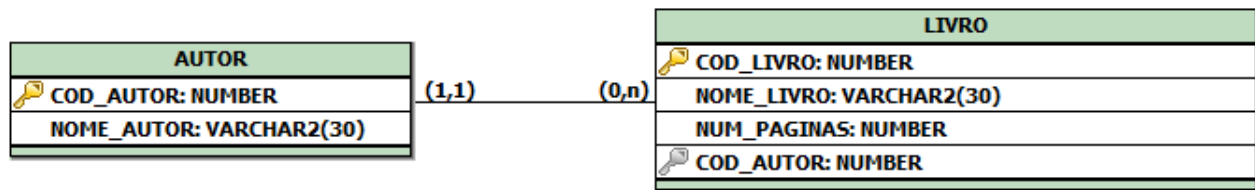


Figura 28: Modelo Lógico.

Observe que as pequenas mudanças nas representações não são suficientes para dificultar sua leitura. Por exemplo, a representação das chaves primárias e chaves estrangeiras e também a presença do tipo de dado<sup>7</sup> que o campo armazenará.

---

<sup>7</sup> Tipo de dado refere-se ao valor que será armazenado naquele campo. Por exemplo, o tipo NUMBER só permite valores numéricos.

## Capítulo 3 – Exercícios

### 3.1 – Lista 1: Definições, Conceitos e Modelagem de Banco de Dados

1) Discorra sobre os seguintes termos:

- a) Dados
- b) Informações
- c) Registros
- d) Entidades
- e) Atributos

2) O que você entende por redundância de dados? Quais possíveis problemas elas podem causar?

3) Como você conceituaria MINIMUDO?

4) Qual a importância da abstração dos dados para os usuários do banco de dados?

5) Qual o papel do SGBD em relação ao banco de dados?

6) Qual a diferença entre DADO e INFORMAÇÃO?

Considere a tabela de clientes para responder as questões 7, 8, 9 e 10.

cod_cliente	nome_cliente	edereco_cliente	tel_cliente	nome_consultor	tel_consultor	
1	Pato Donald	Rua Itajubá, 50 Bairro Floresta	31345111635	Homer Simpson	3135462254	
2	Zé Colméia	Av. Carlos Prates, 889 - Sion	33874569852	Homer Simpson	3199999999	
3	Narizinho	R. Icaivera, 44, Apto 193 B. Serra		Tia Nastácia	3189565241	
4	Kitana	Rua do Reino, 2 bairro Serra	35541324565	Homer Sipson	3135465224	
5	Sub-Zero	R. do Picolé, 88 Bairro Sion	33569863422	Homer Simpson		

7) Quantos registros existem na tabela? E quantos campos?

8) Que problema teríamos para gerar um relatório de clientes por bairro?

9) Você percebeu alguma redundância na tabela? Quais?

10) Quais problemas você consegue perceber nessa tabela?

11) Reflita sobre como melhorar a distribuição de dados nesse sistema.  
A criação de outras tabelas é uma boa opção?

12) Quais são as condições para um campo ser chave primária?

## 3.2 – Lista 2: Modelagem de Banco de Dados

1) Identifique as entidades, os principais atributos, os relacionamentos e a cardinalidade de cada cenário abaixo:

- a) Médico e Receita
- b) Produto e Supermercado
- c) Setores e Funcionários
- d) Estúdio, Filme e Gênero
- e) Veículo e Montadora

2) Construa o MER para os seguintes cenários:

- a) Um livro pode ter vários autores e estes podem ter escrito vários livros.
- b) Um projeto arquitetônico é produzido por vários funcionários, inclusive de setores distintos, e que por sua vez já participaram de vários projetos.

3) A organização de uma exposição de cães (ExpoDOG - 2012) deseja manter informações sobre a mesma. Para tal, sabe-se que todo cão exposto tem nome, idade, raça e dono. Sabe-se que um cão pertence a um único dono, mas este (dono) por ter vários cães na exposição.

4) Modifique o Modelo de Dados anterior para que ele contemple também a possibilidade de manter os dados de várias exposições.

5) Uma biblioteca deseja manter o controle de empréstimos de seus volumes aos seus respectivos associados. Para tanto é necessário obedecer as seguintes regras de negócios:

- Um associado pode pegar até 3 volumes.
- O empréstimo deve ser registrado com a data de empréstimo e a data de devolução.
- Na biblioteca os livros são classificados por faixa etária, dessa maneira os livros somente serão emprestados para os associados que tenham idade coerente.
- Periodicamente são observados quais autores, livros e gêneros são mais requisitados (populares) como referência para aquisição de novos livros.

Construa o MER para esse cenário.



### 3.3 – Lista 3: Exercícios Auxiliares

#### NÍVEL 1 - CONCEITOS E DEFINIÇÕES

1 - O que são e para que servem:

- a) Entidades
- b) Atributos
- c) Relacionamentos
- d) Cardinalidade

2 - O que você entende por:

- a) Entidade Pai e Entidade Filho

3 - Crie 3 exemplos de Entidades Pai e respectiva Entidade Filho

4 - Marque (P) para PAI e (F) para Filho nas relações abaixo e justifique sua resposta.

- a) ( ) Funcionário                      ( ) Loja
- b) ( ) Dentista                        ( ) Consulta
- c) ( ) Titular                          ( ) Dependente
- d) ( ) Folha de Cheque              ( ) Conta-Corrente
- e) ( ) Cliente                         ( ) Nota Fiscal

#### NÍVEL 2 - FÁCIL

1 - Relacione as entidades e coloque os atributos:

- a) Prova e Disciplina
- b) Conta-Corrente e Cliente
- c) Moradores e Residência
- d) Conta de Luz e Cliente
- e) Carteira de Motorista e Categoria

2 - Considere o cenário: Uma galeria de arte pretende manter informações a respeito de seus quadros. Para isso ela deseja saber o nome do artista, o nome do quadro e o ano de sua criação. Sabe-se que cada quadro é pintado por um único artista e esse mesmo artista pode ter criado várias obras.

Sabe-se também que cada obra pertence a um estilo de arte (Barroco, Pontilhismo, Cubismo, etc).

3 - Um estudante deseja criar um programa de agenda de telefones para armazenar informações de seus contatos. Ele deseja saber apenas quais contatos e respectivos telefones (Celulares (Vivo, Oi, Claro, etc), Residencial, Comercial).

### NÍVEL 3 - MÉDIO

1 - Um jornal (Ex.: Estado de Minas) é composto por várias matérias/reportagens que são escritas por um ou mais jornalistas e pertencem a uma única seção (Esporte, Lazer, Economia, etc). Naturalmente cada matéria tem um título e uma quantidade de palavras.

2 - Considere o exercício anterior e acrescente a possibilidade da reportagem conter várias fotos e essas fotos pertencerem a um fotógrafo.

3 - Um estudante de TI deseja manter seus compromissos (Data de provas, Data de entrega de trabalhos, Data de consultas médicas, etc) em um sistema de banco de dados. Durante o processo ele percebeu que cada compromisso acontece em uma única data. Entretanto, um compromisso pode ser cíclico, ou seja, acontecer mais de uma vez em datas diferentes.

### NÍVEL 4 - UM POUCO MAIS DIFÍCIL :-)

1 - Construa um modelo de Banco de Dados para os cenários abaixo:

a) Sistema de E-mail.

b) Sistema de Condomínios (Moradores, contas a pagar, condomínios pagos, condomínios em aberto, etc)

c) Operadora de Cartões de Crédito.

### 3.4 – Lista 4: Exercícios Auxiliares

1 - Em uma determinada empresa, Cada funcionário trabalha em um único departamento com um ou vários colegas. Sabe-se que todo funcionário tem um único chefe.

2 - A empresa Ualdisnei contém vários colaboradores em seu quadro, cada qual trabalhando em seu departamento e exercendo sua função. Sabe-se que todo departamento terá um único diretor, pelo menos um gerente e vários colaboradores.

3 - Construa um Modelo de dados para armazenar dados sobre o corpo humano, ou seja, deseja-se saber quais órgãos e suas respectivas funções.

4 – A BatEmpresa produz vários produtos que serão usados no combate ao crime. Bruce Wayne, vulgo Batman e CEO da referida empresa deseja saber detalhes da produção de seus produtos. Por exemplo, é de seu interesse saber quais produtos são produzidos, com qual propósito ele foi produzido, quais peças foram usadas na confecção do produto, qual o custo de produção de cada produto e quais empregados participaram da sua produção. Cada produto é composto por pelo menos uma peça (Ex. parafuso, corda...) e uma peça pode fazer parte da montagem de vários produtos (Ex. Batcinto, Batlaço ...). Todos os funcionários participam da linha de montagem de qualquer um dos produtos. Para construir o modelo de dados para esse cenário você deve se lembrar de que os preços das peças variam ao longo do tempo.

5.a – A empresa Space Gosth Linhas Aéreas deseja controlar os seus voos de modo que saiba os aeroportos de origem e de destino de cada um. Cada voo tem: data e horário de saída e de chegada, um código de identificação do voo (trecho) e um plano de voo registrado. Todo voo é composto por uma tripulação (conjunto de funcionários responsáveis pelo voo), sabe-se que um funcionário pode participar de várias tripulações ao longo de sua carreira. Por fim é importante para empresa saber qual aeronave participou de cada um dos voos.

5.b – Altere a questão anterior de modo que o modelo forneça informações dos funcionários para que sempre que possível, a tripulação seja composta por integrantes residentes na cidade de origem do voo.

5.c – Altere a questão anterior de tal modo que o modelo contemple informações dos passageiros. Sabe-se que o voo só é concretizado se houver pelo menos 30 passageiros e no máximo 120 passageiros registrados, caso o mínimo não seja atingido o voo será cancelado e os passageiros remanejados para outro voo. Um passageiro pode ter participado de vários voos ao longo de sua vida. São consideradas informações obrigatórias de passageiro: CPF, Nome, Data de nascimento e sexo.

5.d – Altere o modelo anterior para informar a empresa quais voos foram cancelados. Isso será importante para planejamento gerencial dos voos.