

Escola Técnica Centro Paula Souza

Técnico em Informática

Tecnologias e Linguagens para Banco de Dados I

Docente: Maria Marli Milan Luqueta

Etec Pedro Ferreira Alves

Sumário

1.	CONCEITOS BÁSICOS	3
1.1.	HISTÓRICO	4
1.2.	SISTEMA DE BANCO DE DADOS	4
1.3.	ESTRUTURA.....	4
1.4.	ESQUEMA DE DADOS	5
1.5.	VISÕES	5
1.5.	O QUE UM BD TEM?	6
1.6.	PROFISSIONAIS ENVOLVIDOS COM UM BD	6
2.	SISTEMA GERENCIADOR DE UM BANCO DE DADOS (SGBD)	7
2.1.	ABORDAGEM BANCO DE DADOS X ABORDAGEM PROCESSAMENTO TRADICIONAL DE ARQUIVOS	7
2.2.	PROBLEMAS DOS SISTEMAS DE ARQUIVOS CONVENCIONAIS	7
2.3.	BANCOS DE DADOS X SISTEMA DE ARQUIVOS	8
2.4.	CONCEITOS.....	8
2.5.	CARACTERÍSTICAS DE UM SGBD	8
2.6.	DICIONÁRIO DE DADOS	10
2.7.	LINGUAGENS DE INTERFACES.....	10
2.8.	ESTRUTURA GERAL DE UM SGBD	11
2.9.	CLASSIFICAÇÃO DOS SGBD	12
3.	PROJETO DE BANCO DE DADOS.....	13
3.1.	ETAPAS DO PROJETO DE BD	13
	<i>Estabelecer o objetivo.....</i>	<i>13</i>
	<i>Coleta de dados.....</i>	<i>13</i>
	<i>Projeto conceitual.....</i>	<i>13</i>
	<i>Projeto Lógico.....</i>	<i>14</i>
	<i>Implantação</i>	<i>14</i>
	<i>Teste</i>	<i>14</i>
	<i>Manutenção.....</i>	<i>14</i>
4.	MODELOS DE DADOS	15
5.	MODELAGEM DE DADOS.....ERRO! INDICADOR NÃO DEFINIDO.	
5.1.	OBJETIVO	15
5.2.	TIPOS DE MODELOS.....	16
5.3.	EXEMPLO MODELO	16
5.4.	MAPEAMENTO.....	18
5.5.	NORMALIZAÇÃO.....	18
6.	MODELO ENTIDADE-RELACIONAMENTO (ME-R).....	19
6.1.	ENTIDADES E ATRIBUTOS.....	19
6.2.	RELACIONAMENTOS	21
6.3.	ABSTRAÇÕES DE DADOS	25
7.	DIAGRAMA ENTIDADE-RELACIONAMENTO (DE-R)	28
7.2.	TIPOS DE RELACIONAMENTOS NO DER.....	30
7.3.	ENTIDADE FORTE X ENTIDADE FRACA	31
7.4.	GENERALIZAÇÃO X ESPECIALIZAÇÃO	32
7.5.	AGREGAÇÃO	32
7.6.	MODELANDO UM SISTEMA DE DADOS.....	33
8.	MODELO RELACIONAL.....	35
8.1.	CONCEITOS.....	35

8.2.	MAPEANDO ME-R PARA O MODELO RELACIONAL.....	36
8.3.	PASSOS MAPEAMENTO MER - MR.....	39
9.	NORMALIZAÇÃO.....	44
9.1.	PRIMEIRA FORMA NORMAL:.....	44
9.2.	SEGUNDA FORMA NORMAL:.....	45
9.3.	TERCEIRA FORMA NORMAL:	46
10.	REFERÊNCIA DE CRIAÇÃO E APOIO AO ESTUDO.....	47
11.	ANEXOS.....	48
11.1.	ANEXO 1 – EXERCÍCIOS DE-R.....	48
11.2.	ANEXO 2 – EXERCÍCIOS MR.....	49
11.3.	ANEXO 3 – EXERCÍCIOS NORMALIZAÇÃO.....	51
11.4.	ANEXO 4 – LISTAS DE EXERCÍCIOS.....	52

1. Conceitos Básicos

O conceito de Banco de Dados (BD) está muito presente em nosso dia-a-dia e faz parte de nossa vida. O Banco de Dados desempenha um papel crítico em muitas áreas diferentes onde computadores são utilizados. Há uma grande necessidade em se realizar o armazenamento de informações que não se encontram isoladas umas das outras. Além de uma forma adequada para definir o armazenamento destas informações, os usuários desejam realizar (consultar) um determinado subconjunto de dados, atualizar ou modificar a estrutura dos dados e eliminar informações não mais necessárias.

Uma solução para este problema foi apresentada com o advento da tecnologia em Banco de Dados (ou Base de Dados ou **BD**). Uma definição mais precisa de Banco de Dados não existe formalmente, contudo pode-se dizer que:

- a) Um Banco de Dados é uma coleção logicamente coerente de dados com um determinado significado inerente. Isto significa que um conjunto aleatório de dados não pode ser considerado um Banco de Dados.
- b) Um Banco de Dados é projetado, construído e composto por um conjunto de dados para um propósito específico. Existe um grupo de usuários ou algumas aplicações pré-concebidas onde estes dados serão utilizados.
- c) Um Banco de Dados representa aspectos de uma parte restrita do mundo real, denominado de mini-mundo. Alterações que ocorra no mini-mundo são refletidas no Banco de Dados.

Resumindo, um BD representa uma fonte de onde informações são derivadas, possui um nível de interação com eventos que ocorrem no mundo real, e uma audiência que está interessada em seu conteúdo.

Além do conceito de BD, outros são necessários para se compreender o ambiente de um BD. Um Sistema de Gerenciamento de Banco de Dados (**SGBD**) é uma coleção de programas que permite ao usuário definir, construir e manipular Banco de Dados para as mais diversas aplicações.

DEFINIR um BD envolve a especificação e a descrição detalhada dos tipos de dados a serem armazenados.

CONSTRUIR um BD é o processo de armazenamento dos dados em si em um determinado meio físico que é controlado pelo SGBD.

MANIPULAR um BD inclui uma série de funções para se realizar operações de consulta, atualizações e remoções de dados do BD.

O Banco de Dados e seu software são juntos denominados de Sistema de Banco de Dados (SBD).

O grande objetivo de um sistema de BD é oferecer uma visão “abstrata” dos dados aos usuários e também prover independência de dados às aplicações (estrutura física de armazenamento e à estratégia de acesso).

Sistema são o conjunto de pessoas, máquinas, idéias e atividades que coletam e processa dados de maneira a satisfazer as necessidades de uma organização. Subsistemas são sistemas independentes dentro de um sistema maior. Um sistema é composto de informações e dados. Informação acrescenta algo ao conhecimento da realidade a ser analisada. Dado é a informação a ser registrada, seja no papel e no computador.

1.1. Histórico

- **1945:** Fitas magnéticas para armazenamento de dados
- **1961:** IDS (Integrated Data System), o primeiro SGBD hierárquico.
- **70's:** A tecnologia de BD evolui. Alguns projetos de pesquisa em BD Relacionais: Ingres, System/R, Adabas
- **1970:** O modelo relacional é apresentado por Edgar F. Codd
- **1976:** O modelo Entidade-Relacionamento é apresentado em artigo de P. Chen
- **1980's:** SGBD comerciais se consolidam: DB2, Oracle, Sybase, Informix. Surgem ferramentas para PC's (dBase, Paradox)
- **1985:** Lançado o primeiro padrão SQL.
- **1990's:** SGBD são estendidos para aplicações complexas (espaciais, temporais, multimídia, etc.); Surgem os SGBD Orientados a Objetos (O2, Object Store, Gemstone); Propõem-se extensões de orientação a objetos para SQL; Começa a tendência à distribuição (Ambientes cliente-servidor)
- **1992:** Publicado o padrão SQL2
- **1995:** Discussão "acalorada" no meio acadêmico. Relacional versus Orientado a Objetos => Relacional estendido

1.2. Sistema de Banco de Dados

Um banco de dados pode ser criado e mantido por um conjunto de aplicações desenvolvidas especialmente para esta tarefa ou por um "Sistema Gerenciador de Banco de Dados" (SGBD). Um SGBD, como vimos acima, permite aos usuários criarem e manipularem bancos de dados de propósito geral. O conjunto formado por um banco de dados mais as aplicações que manipulam o mesmo é chamado de "Sistema de Banco de Dados", envolvendo quatro componentes principais:

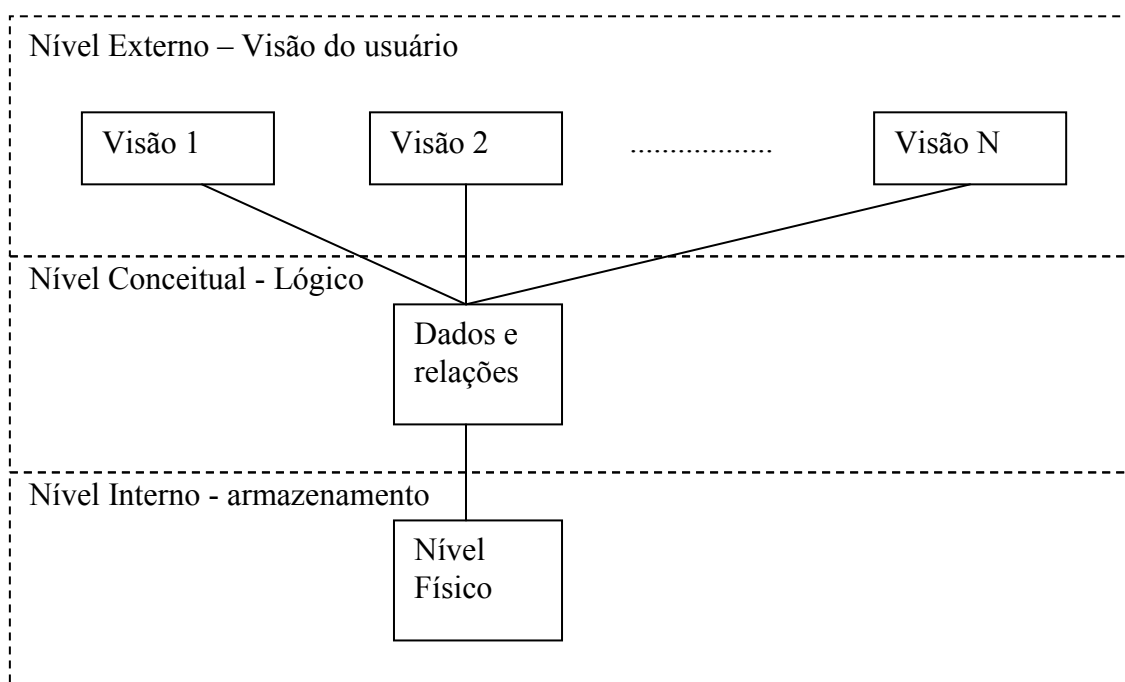
- 1.1. Dados
- 1.2. Hardware
- 1.3. Software
- 1.4. Usuários

1.3. Estrutura

Como vimos, o grande objetivo de um sistema de banco de dados é fornecer uma visão abstrata ao usuário, este conceito está associado à característica de se observar somente os aspectos de interesse, sem se preocupar com maiores detalhes envolvidos

No contexto de abstração de dados, um BD pode ser visto sem se considerar a forma como os dados estão armazenados fisicamente.

O sistema de BD deve prover uma visão abstrata de dados para os usuários. A abstração se dá em três níveis:



- **Nível Interno:** descreve a estrutura de armazenamento físico dos dados do BD, fornecendo um modelo físico dos dados que inclui detalhes sobre os caminhos de acesso aos dados internamente, nível mais baixo;
- **Nível Conceitual:** descreve a estrutura de todo o BD para uma determinada comunidade de usuários, ocultando detalhes sobre a organização física dos dados e apresentando a descrição lógica dos dados e das ligações existentes entre eles. Visão do todo do BD.
- **Nível Externo:** descreve apenas parte do BD. Cada usuário (ou grupo de usuários) tem acesso à parte do BD que lhe é de interesse ou que lhe é permitido, através das visões. Um BD pode ter muitas visões diferentes.

Observe que os três níveis apresentados anteriormente apresentam apenas descrições dos dados. Como os três níveis apresentam descrições diferentes para os mesmos dados, torna-se necessário converter uma representação em outra, ou seja, definir mapeamentos de dados entre os níveis.

1.4. Esquema de dados

O Esquema de Dados é uma parte do BD que contém informações sobre os próprios dados do BD, ou seja, estrutura do BD tanto a nível interno, conceitual e externo.

1.5. Visões

Uma Visão consiste de um subconjunto de dados do BD, ou pode conter dados “virtuais” que são derivados dos existentes no BD, mas que não estão explicitamente armazenados. As Visões existem para satisfazer as necessidades dos usuários na apresentação apenas das informações importantes e ocultando aquelas desnecessárias.

1.5. O que um BD tem?

- Alguma fonte de onde os dados são derivados;
- “Taxa” de interação com eventos do mundo real;
- “Audiência” interessada em seu conteúdo;
- Qualquer quantidade de dados (1, 1000, milhões,...);
- Variação de complexidade - (agenda X Fazenda);
- Criação e manutenção manualmente ou por equipamento(s) - (controle de uso na biblioteca).

1.6. Profissionais envolvidos com um BD

Em um pequeno banco de dados com muitos usuários e com restrições de acesso podem-se indicar alguns perfis de pessoas que interagem com o BD.

▪ **Administrador do banco de Dados (DBA)**

É o supervisor do BD, responsável pela autorização de acesso ao BD, monitorando e coordenando o seu uso. Está envolvido com os aspectos físicos do BD (definição da estrutura de armazenamento, modificação da organização física e esquema, autorização de acesso aos dados, restrições de integridade, etc.).

▪ **Projetista de Banco de Dados**

São responsáveis pela identificação dos dados e a elaboração de estruturas apropriadas para armazená-los, e também compreender os requisitos necessários aos grupos de usuários do BD antes de sua implementação (Responsáveis pelos dados armazenados no BD).

▪ **Analistas**

Determina os requisitos dos usuários e desenvolvem especificações que atendam estes requisitos.

▪ **Programadores de aplicações**

Implementam as especificações na forma de programas elaborando toda a documentação.

▪ **Usuários finais**

Um BD existe para a utilização do usuário final, onde normalmente o seu trabalho requer consultas e atualizações. A maioria dos usuários utilizam programas voltados ao desempenho profissional, do seu dia-a-dia. Em relação às consultas temos:

- Usuários casuais: Consultas diferentes e ocasionais.
- Usuários paramétricos: Consultas predefinidas.
- Usuários sofisticados: Consultas complexas.

2. Sistema Gerenciador de um Banco de Dados (SGBD)

2.1. Abordagem Banco de Dados X Abordagem Processamento Tradicional de Arquivos

Uma característica importante da abordagem Banco de Dados é que o SGBD mantém não somente os dados, mas também a forma como os mesmos são armazenados, contendo uma descrição completa do banco de dados. Estas informações são armazenadas no catálogo do SGBD, o qual contém informações como a estrutura de cada arquivo, o tipo e o formato de armazenamento de cada tipo de dado, restrições, etc. A informação armazenada no catálogo é chamada de “Meta Dados”. No processamento tradicional de arquivos, o programa que irá manipular os dados deve conter este tipo de informação, ficando limitado a manipular as informações que o mesmo conhece. Utilizando a abordagem banco de dados, a aplicação pode manipular diversas bases de dados diferentes.

2.1.1. Separação entre Programas e Dados

No processamento tradicional de arquivos, a estrutura dos dados está incorporada ao programa de acesso. Desta forma, qualquer alteração na estrutura de arquivos implica na alteração no código fonte de todos os programas. Já na abordagem banco de dados, a estrutura é alterada apenas no catálogo, não alterando os programas.

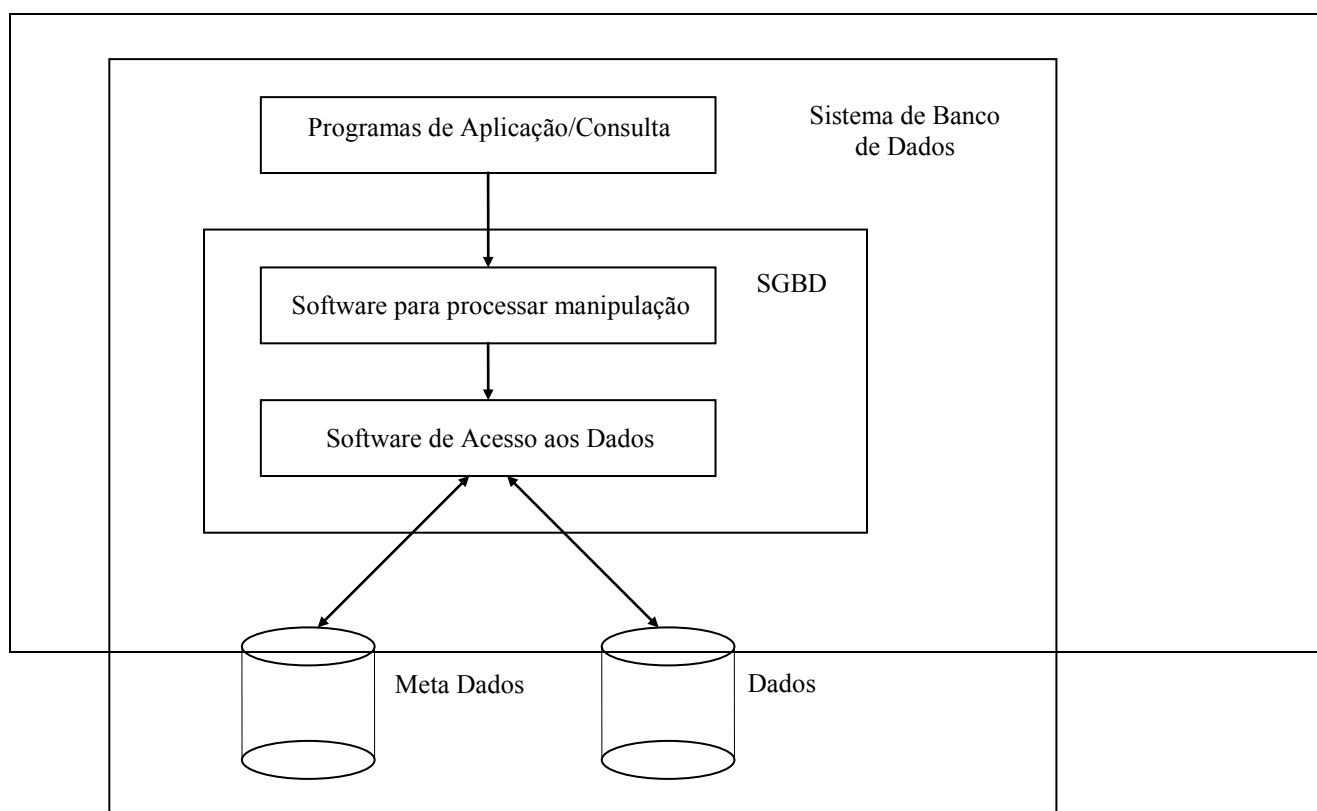


Figura 1. Um ambiente de Sistema de Banco de Dados

2.2. Problemas dos Sistemas de Arquivos convencionais

- Integração de dados.
- Replicação e inconsistência de dados

- Dificuldade de acesso a dados. Dados incompatíveis, estruturas complexas.
- Problemas de segurança.

2.3. Bancos de Dados X Sistema de Arquivos

- Auto-contenção
- Separação entre Programas e Dados. Abstração de Dados
- Múltiplas Visões de Dados
- Processamento de Transações, Recuperação.
- Compartilhamento de dados

2.4. Conceitos

- **BD - Banco de Dados:**
 - Coleção de dados, conjunto de informações de uma empresa ou pessoais.
- **SGBD - Sistema de Gerenciamento de Banco de Dados:**
 - Conjunto de dados inter-relacionados e um conjunto de programas para controlar esses dados.
 - Software que manipula todos os acessos ao Banco de Dados.

2.5. Características de um SGBD

2.5.1. Funções do SGBD

O Sistema de Gerenciamento de Banco de Dados tem como principais funções:

- a) **Controle de Redundância:** em um sistema tradicional de controle de arquivos cada usuário normalmente apresenta seus próprios arquivos armazenando o conjunto de dados que é de seu interesse, e nestes casos é comum ocorrer redundância de dados. Esta redundância consiste no armazenamento de uma mesma informação em locais diferentes, o que pode provocar sérios problemas. Alguns destes problemas consiste inicialmente no aumento do esforço computacional para realizar a atualização destes dados e também o aumento do espaço necessário para o armazenamento dos dados. Mas o problema mais sério é que a representação dos dados desta forma pode tornar-se inconsistente, pois duas informações podem aparecer em locais distintos, mas apresentando valores diferentes. Em um sistema de BD as informações só se encontram armazenadas em um único local ou estão existindo duplicação controlada dos dados.
- b) **Compartilhamento dos Dados:** o SGBD deve incluir um software para o controle de concorrência ao acesso dos dados em um ambiente multiusuário, de forma a possibilitar o compartilhamento dos dados, garantindo que se vários usuários tentar realizar operações de atualização sobre um mesmo conjunto de dados, o resultado destas operações possa ser correto. Outro mecanismo que suporta a noção de compartilhamento de dados consiste na facilidade para definir visões de usuário, o qual torna disponível uma porção específica dos dados do BD de acordo com o seu interesse.
- c) **Controle de Acesso:** quando vários usuários compartilham os dados, é comum que alguns não apresentem autorização para acesso a todo o BD. Além disso, alguns

usuários podem apresentar a autorização para apenas leitura dos dados, sendo impedido a este de realizar qualquer modificação no BD. O SGBD deve fornecer serviços que permitam ao DBA definir os níveis de segurança e de autorização para os usuários do BD.

- d) **Controle de Transações:** uma transação é um conjunto de operações sobre o BD que devem ser executados integralmente e sem falhas ou interrupções. O SGBD deve realizar o controle das transações para que sejam executadas com segurança.
- e) **Possibilidade de Múltiplas Interfaces:** vários usuários representam também necessidades diversas no que refere aos tipos de interfaces fornecidas pelo SGBD. Interfaces para consultas de dados, programação, e interfaces baseadas em menus ou em linguagem natural são exemplos de alguns tipos que podem estar disponíveis.
- f) **Reforçar Restrições de Integridade:** a maioria dos aplicativos apresenta serviços que possibilitam garantir a integridade dos dados no BD. A restrição de integridade mais simples consiste na especificação do padrão de formato para os dados ou valores assumidos como “default”.
- g) **Providenciar “Backup” e Recuperação de Dados:** o SGBD deve apresentar facilidades para recuperação de falhas provocadas pelo hardware ou pelo software.
- h) **Independência de Dados:** o SGBD possibilita o desenvolvimento de programas aplicativos que não possuem a descrição real de como os dados (arquivos) estão fisicamente armazenados. Desta forma, alterações nas estruturas dos arquivos do BD não afetam os programas aplicativos. Os três níveis podem ser utilizados para melhor compreender o conceito de independência de dados. Pode-se apresentar dois tipos de independência de dados:

Independência Lógica dos Dados: consiste na capacidade de alterar o esquema conceitual sem provocar modificações nos esquemas externos ou nos programas aplicativos.

Independência Física dos Dados: consiste na capacidade de alterar o esquema interno sem provocar modificações no esquema conceitual.

Apesar de todas as vantagens citadas anteriormente sobre um SGBD, há situações em que se deve ponderar sobre sua utilização, como por exemplo, o alto investimento inicial e possibilidade de compra de um novo hardware; ou a possibilidade de insatisfação no desempenho geral do sistema que pode ser provocado pelas funções de segurança, controle de concorrência, recuperação e manutenção de integridade dos dados.

Para que um SGBD possa fornecer as características de independência de dados, suporte a múltiplas visões de vários usuários, entre outras, torna-se necessário a existência de uma organização no sistema para que isto seja possível.

2.5.2. Sem SGBD:

- **Arquivos físicos:** problemas com espaço físico, dificuldade de acesso, velocidade, replicação.
- **Ambiente de processamento de dados:** arquivos convencionais: problemas de replicação e inconsistência dos dados, dificuldade de acesso (manipulação

de estruturas complexas), isolamento, compartilhamento de dados difícil, pouca segurança, falta de controle de integridade.

2.5.3. Com SGBD:

- **Oferece:** criar, alterar e eliminar estruturas de armazenamento, inserir, consultar, alterar e eliminar dados.
- **Vantagens:** velocidade, facilidade de acesso, redução da redundância, evita-se inconsistência, controle de integridade, controle de concorrência, maior segurança (quedas do sistema e acesso de pessoas não autorizadas).
- **Desvantagens:** hardware adicional, complexidade do sistema e necessidade de pessoal especializado.

Usando-se um SGBD busca-se **compartilhar** informações de forma **segura**, garantir a **integridade** e a **consistência** dos dados, permitir acesso **rápido e eficiente**, oferecer uma forma de afastar do usuário comum a necessidade de compreender as **estruturas físicas** dos dados.

2.6. Dicionário de Dados

Bons SGBD's normalmente apóiam a inclusão de descrições textuais sobre cada um dos arquivos do BD, como também de cada usuário e programas aplicativos que são utilizados. Estas informações são incluídas no Dicionário de Dados para servirem de apoio aos novos projetistas ou programadores que possam ser colocados no grupo de trabalho, e deste modo, facilitar o entendimento dos ambientes que está sendo utilizado.

2.7. Linguagens de Interfaces

Uma vez que o projeto do BD tenha se completado e um determinado SGBD tenha sido escolhido para a sua implementação, a primeira "ordem do dia" consiste em realizar uma especificação dos esquemas conceituais e internos, e os respectivos mapeamentos entre eles. Para estas etapas o SGBD oferece uma série de linguagens. A primeira delas consiste da Linguagem de Definição de Dados (DDL), a qual é utilizada pelos analistas e projetistas do BD para a definição dos esquemas. O SGBD também apresentará um interpretador para a DDL, o qual será responsável pelo processamento dos comandos da DDL, e realiza o armazenamento do esquema definido em estruturas internas do BD.

Uma vez definido e preenchido o BD com os seus dados, estes normalmente sofrerão uma série de operações de acesso às informações nele armazenadas. O SGBD fornece a Linguagem de Manipulação de Dados (DML) para a especificação destas operações. Os comandos da DML podem aparecer embutidos em outra linguagem (geralmente uma linguagem de programação de alto nível), e neste caso esta é denominada de linguagem hospedeira, a DML é denominada de sub-linguagem de dados. De outra forma se DML for utilizada isoladamente de uma forma interativa, possa a ser denominada de linguagem de consulta (ou "query language").

Para interagir com o SGBD além de linguagens descritas anteriormente, o sistema deve apresentar interfaces que sejam amigáveis. Estas interfaces podem ser de diversas formas:

- a) **Interfaces baseadas em Menus:** permite ao usuário selecionar o trabalho a ser realizado através de uma lista de opções apresentada na forma de Menus. Neste tipo de interface uma consulta pode ser construída através da escolha apropriada de um conjunto de opções;
- b) **Interfaces Gráficas:** apresenta ao usuário o esquema de Base em uma forma de diagrama, e a criação de consultas é realizada através da manipulação deste diagrama. Normalmente se apresenta associada a um conjunto de menus;
- c) **Interfaces Baseadas em Formulários:** o sistema apresenta ao usuário um formulário, através do qual pode ser preenchida certa entrada. Através destes formulários pode ser realizada operação de consultas, inserções a remoções de dados;
- d) **Interface em Linguagem Natural:** aceita comandos ou requisições escritas em uma linguagem natural ou em alguma outra linguagem de consulta, sendo capaz de "compreender" o que está sendo pedido;
- e) **Interface para Usuários Comuns:** normalmente estes usuários, como o de um terminal bancário, podem realizar apenas um conjunto limitado de operações e de forma repetitiva. Sendo assim, é oferecida uma interface que seja fácil de compreender e com um número reduzido de informações a serem manipuladas a poucas opções disponíveis.

No controle de acesso e transações dos dados utiliza-se uma Linguagem de Controle de Dados, que inclusive possibilita estabelecer os diversos níveis de segurança de cada usuário.

2.8. Estrutura geral de um SGBD

- **Gerenciador de arquivos:** gerencia a alocação de espaço de armazenamento em disco e estrutura de dados usadas para representar informação em disco.
- **Gerenciador de BD:** faz a interface dos dados armazenados no BD e os programas de aplicação e consultas submetidas ao sistema. É responsável por:
 - Interação com o gerenciador de arquivos (dados em disco): gerenciador traduz LMD em comandos de baixo nível do sistema de arquivos
 - Garantia de integridade: os valores dos dados armazenados devem satisfazer certas restrições de integridade. O gerenciador pode verificar se alterações resultam em violação dessas restrições.
 - Garantia de segurança: permissões de acesso são controladas pelo gerenciador de banco de dados
 - Recuperação e backup: recuperação de falhas mecânicas ou elétricas, incluindo quebra de discos, falhas de alimentação e erros de software
 - Controle de concorrência: controlar o acesso ao BD por diversos usuários. A consistência dos dados deve ser preservada.
- **Processador de consultas:** traduz uma linguagem de consulta para instruções de baixo nível que o gerenciador de BD entenda.
- **Pré-compilador DML (Data Manipulation Language):** converte comandos DML dos programas de aplicação para rotinas em linguagem de máquina.
 - **DML ou LMD - Linguagem de Manipulação de Dados:** usada para inserir, consultar, alterar e eliminar os dados das tabelas.

- **Linguagem de Consulta:** parte da LMD responsável apenas pela consulta de dados. Exemplo: SQL
- **Compilador DDL (Data Definition Language):** converte comandos DDL para tabelas armazenadas no dicionário de dados.
 - **DDL ou LDD - Linguagem de Definição de Dados:** usada para criar e alterar estruturas das tabelas
- **Arquivos de dados:** armazena os dados propriamente ditos.
- **Dicionário de dados:** armazena as informações sobre a estrutura de dados
- **Índices:** facilitam a localização e o acesso aos dados

2.9. Classificação dos SGBD

- **Modelo de dados**
 - Rede, hierárquico, Relacional, Orientado a objetos
- **Número de usuários**
 - Monousuário, multiusuário
- **Localização do BD**
 - Centralizado
 - Distribuído
- **Generalidade**
 - Propósito geral
 - Propósito específico

3. Projeto de Banco de Dados

Na informatização de um mini-mundo com a utilização de um BD, deve-se primeiramente preocupar-se com a escolha do SGBD que será utilizado. De nada servirá um Projeto de BD bem realizado se o gerenciador oferecer restrições inaceitáveis como, por exemplo, baixa capacidade de armazenamento. As máquinas utilizadas e a arquitetura do Sistema de Informações (Centralizada, Distribuída ou Cliente/Servidor) também são elementos de igual importância.

O Projeto de Banco de Dados é um dos temas mais discutidos e estudados nesta área. A razão é forte, um projeto incorreto pode resultar em um SBD que não atende as necessidades dos seus usuários, pedindo freqüentes e custosos reparos ou mesmo um completo reprojeito. O tempo gasto em estudos e na aplicação de princípios de um bom projeto de BD é sempre um tempo bem empregado.

A maior habilidade exigida para os especialistas em Projeto de BD é saber recolher informações a partir da vivência com o mini-mundo, é também extrair dos usuários por meio de entrevistas e reuniões o que desejam do BD e é estruturar todas estas informações de modo a se tornarem concretamente um BD. A grande valorização deste tipo de profissional ocorre devido à dificuldade em se encontrar pessoas experientes tanto na área técnica quanto no tratamento com os usuários, que na verdade são os clientes.

3.1. Etapas do Projeto de BD

Um bom projeto de BD compreende resumidamente os seguintes passos:

1	Estabelecer o objetivo Anotar o objetivo do BD;
2	Coleta de dados Coletar todos os tipos de informações que serão precisos armazenar. Pensar sobre os relatórios e/ou formulários que possam ser desejados. Para esta fase é necessário conhecer profundamente o mini-mundo a isto pode requerer algum tempo a ser utilizado com reuniões e entrevistas com os futuros usuários. A experiência (sabedoria) nesta etapa é fundamental. Não deve ser realizada por uma única pessoa, mas por um grupo de projetistas;
3	Análise e especificação O grupo de projetistas deve ponderar sobre todas as informações obtidas e gerar documentos que especifiquem as necessidades dos usuários. Caso perceba-se a falta de alguma informação deve retornar a fase anterior (coleta);
4	Projeto conceitual Através de uma ferramenta conceitual (modelo de dados a nível conceitual) um grupo de especialistas repassa a especificação textual em um diagrama ou outras forma de linguagem que descreve com clareza as necessidades dos usuários, isto é chamado modelagem. O objetivo desta fase é melhor organizar as idéias passadas pela fase anterior (Análise e Especificação) e perceber alguma falta de informações neste caso é necessário retornar;

5	Projeto Lógico Através de uma ferramenta lógica (modelo de dados a nível lógico) um grupo de especialistas repassa a especificação conceitual em lógica através de um diagrama ou outra linguagem, isto é chamado mapeamento. Com esta fase obtêm-se a definição lógica dos arquivos de dados. Especificação lógica depende do SGBD que será utilizado, pois as estruturas dos arquivos devem ser compatíveis com o gerenciador utilizado;
6	Implantação Tendo a especificação dos arquivos pode-se implantá-los, através da linguagem DDL, para formarem um BD;
7	Teste Os testes são realizados medindo a satisfação dos usuários em termos de velocidade e o padrão com que os dados são apresentados;
8	Manutenção A insatisfação de um usuário ou mesmo modificação no mini-mundo (expansão da empresa, etc.) pode exigir pequenas alterações na estrutura do BD. Mudanças drásticas como, por exemplo, modificações no organograma de uma empresa podem exigir um trabalho maior na reestruturação do BD.

A similaridade do processo do Projeto de BD com o Projeto de Software não é coincidência. Na verdade pode-se dizer que ambos são complementares, uma vez que no Projeto de Software é necessária a criação dos arquivos de dados (BD) e no Projeto de BD é necessária a criação dos programas aplicativos (software) que irão operar o Banco de Dados.

4. Modelos de Dados

Um Modelo de Dados é um conjunto de conceitos utilizados para descrever a estrutura de um BD a nível conceitual, lógico ou físico. Os modelos apresentam um conjunto de operações para a especificação e manipulações dos dados do BD.

Atualmente o Modelo de Dados Conceitual mais difundido e utilizado para a modelagem de dados é o MER criado por Peter Chen. O resultado da modelagem utilizando o MER é um diagrama (DER) bastante simples, porém de grande utilidade.

Tendo exatamente definido qual o problema a ser resolvido, ou seja, tendo determinado as fronteiras que delimitam e restringem o mini-mundo a ser modelado e realizado a especificação, utiliza-se um roteiro a ser seguido para se determinar uma primeira versão do DER.

Depois de criada a primeira versão do DER deve-se apresentar aos usuários para que seja verificado se o diagrama está correto e completo. Sucessivas apresentações do DER devem ser realizadas enquanto forem detectadas falhas na representação. À primeira vista pode-se pressupor que esta rotina de trabalho trará atrasos na construção do Banco de Dados, porém os especialistas em engenharia (software, mecânica e etc) sabem da extrema importância da fase de projeto. Erros ocorridos nesta fase acarretam graves atrasos e aumento no custo de realização do produto. Por esse motivo, o MER se tornou uma ferramenta de modelagem entre as mais difundidas, estimadas e utilizadas no mercado de informática.

4.1. Objetivo

Em suma, o objetivo da modelagem de dados é coletar os dados do mundo real e transformar em informações que podem ser armazenadas em um SGBD.

Há uma série de modelos que podem ser utilizados para a transformação dos dados do mundo real em informações que facilmente possam ser organizadas em um computador.

Em nossas aulas, serão analisados dois modelos: o Modelo de Entidade-Relacionamento (ME-R) e o Modelo Relacional (os mais usados). O Modelo de Entidade-Relacionamento é um modelo conceitual (não preocupado com a implementação, mas sim com a organização dos dados necessários para o sistema) e o Modelo Relacional é um modelo operacional, onde a preocupação é organizar os dados em estruturas que serão diretamente implementadas em um sistema de banco de dados que implemente esse modelo (os dados são organizados em tabelas).

4.2. Tipos de modelos

Existem modelos para diferentes níveis de abstração de representação de dados. Os tipos de modelos existentes na literatura se classificam em:

• Seres, objetos, fatos	Mundo Real	Representa organização e suas alterações
• Informações informais	Modelo Descritivo	Descrição de Estruturas e Procedimentos
• Informações formais	Modelo Conceitual	Estruturas de Informações e Definições de Manipulação
• Dados	Modelo Operacional	Estruturas Externas de Dados
• Cadeias de bits	Modelo Físico	Estruturas Internas de Arquivos

O reflexo das necessidades evolutivas do mercado pode ser cronologicamente visualizado através dos diversos Modelos de Dados desenvolvidos, desde os Modelos Baseados em Registros passando pelos Semânticos até os recentes Modelos Orientados a Objetos. Abaixo a descrição das categorias existentes:

Modelos Baseados em Registros (Operacional)

São modelos que fornecem conceitos de modelagem a nível lógico utilizando-se da representação de registros. Comercialmente, os modelos mais encontrados são o Rede, o Hierárquico e o Relacional;

Modelos Semânticos (Conceitual)

São modelos conceituais que apresentam uma capacidade maior de representação fiel de um problema, pela simples razão de possuir conceitos mais elaborados. Exemplos de modelos desta categoria é o Modelo Entidade-Relacionamento (MER) e o Modelo Funcional;

Modelos Orientados a Objetos (Conceitual)

São modelos conceituais que procuram representar a estrutura de Bancos de Dados através do paradigma de "Orientação a Objetos". Este tipo de modelo esta em grande destaque mundial (inclusive no Brasil) nas pesquisas atuais. Alguns dos modelos em pesquisa são: O2 (França); PCTE (Comunidade Européia); Orion (EUA) e o Modelo de Representação de Objetos - MRO (Brasil - USP/São Carlos).

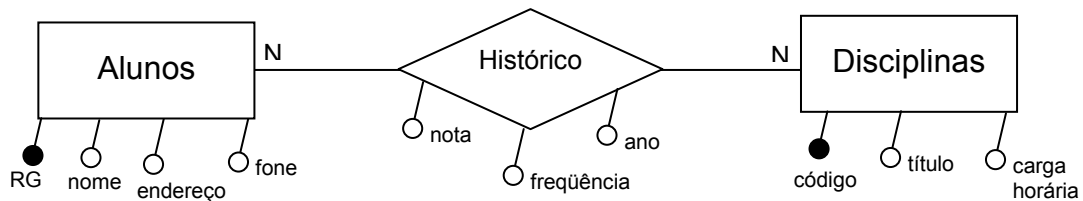
4.3. Exemplo de Modelo na prática

Sistema para o controle de dados dos alunos de uma escola.

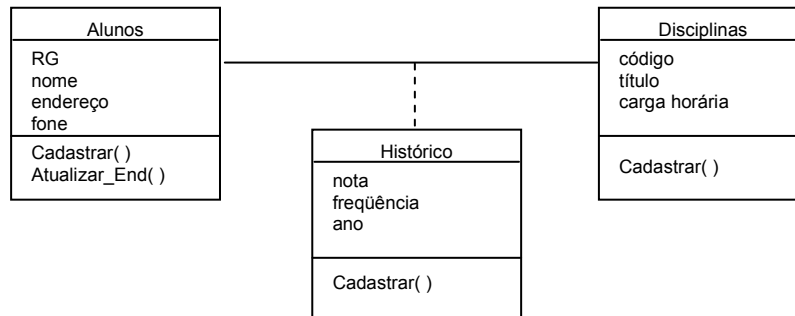
→ **Modelo descritivo:** A escola mantém o cadastro de seus alunos por RG, nome, endereço e fone. Sobre as disciplinas que são oferecidas a escola mantém o código, título e carga horária. O histórico de um aluno associa os alunos às disciplinas que eles cursaram em cada ano e mantém a nota e frequência que eles obtiveram.

→ Modelo Conceitual:

- **ME-R – Modelo Entidade-Relacionamento:** objetos são representados como entidades e as associações entre eles como relacionamentos.



- **Modelo Orientado a Objetos:** dados do mundo real são representados como objetos, que possuem atributos e métodos. Objetos organizados em classes ligados por associações. Exemplo usando modelagem UML (Linguagem de Modelagem Unificada):

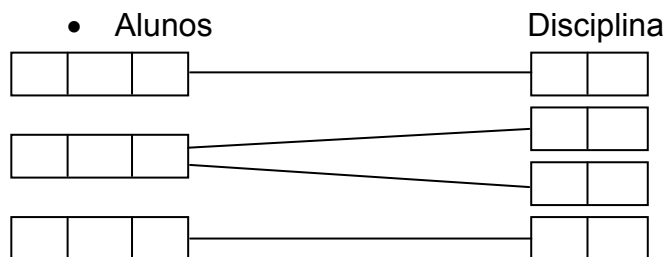


→ Modelo Operacional: descrição de alto nível da implementação

- **Modelo de Rede:** dados representados por uma coleção de registros e os relacionamentos são representados por ligações (ponteiros).

♦ Implementações:

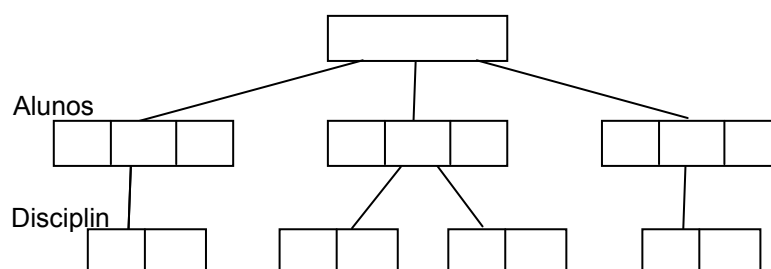
- IDMS Cullinet/ DMS 1100 Sperry/ TOTAL Cincom Systems



- **Modelo Hierárquico:** dados e relacionamentos também representados por registros e ligações, organizados como árvores.

♦ Implementações:

- IMS IBM/ System 2000 Intel



- **Modelo Relacional:** dados e relacionamentos representados por uma coleção de tabelas. Ocupa mais de 90 % do mercado atual.
 - ♦ Implementações:
 - SQL/DS e DB2 – IBM, Oracle - Oracle Corp., Access - Microsoft.

Alunos				Histórico					Disciplinas		
RG	nome	endereço	fone	RG	código	ano	nota	frequência	código	título	carga horária

- Características:
 - Cada tabela tem seu nome diferente das demais na mesma base de dados
 - Cada coluna tem seu nome diferente das demais na mesma tabela
 - Colunas contêm os atributos (todos do mesmo domínio)
 - Linhas contêm informações de 1 entidade
 - Cada célula pode conter no máximo 1 item de dado
 - Ordem das linhas é irrelevante
 - Ordem das colunas é irrelevante
 - Nunca temos duas linhas iguais
 - Chave primária

4.4. Mapeamento

O Modelo Entidade Relacionamento é responsável por realizar uma representação dos dados de uma determinada aplicação a um nível mais conceitual, um pouco distante da forma como os seus elementos serão efetivamente implementados. Os modelos de registros, dentre eles Modelo Relacional (mais usado), fornece uma representação dos dados de forma mais próxima como estes se encontrarão quando forem definidos os arquivos para o BD.

A passagem do diagrama do MER para a representação do Modelo Relacional é denominada Mapeamento do MER para o Relacional e é uma das técnicas mais utilizadas por ser uma ligação entre os modelos de dados mais utilizados.

Existem estabelecidos outros mapeamentos como

- MER para o Modelo Rede e;
- MER para o Modelo Hierárquico,

Contudo não abordaremos neste curso por serem pouco utilizados.

4.5. Normalização

O processo de Normalização é uma etapa complementar ao Mapeamento e consiste da aplicação de uma série de regras (Formas Normais). Estas regras irão provocar algumas modificações na especificação lógica. A Normalização permite ao projetista definir quanto da consistência é garantida pela estrutura dos arquivos e quanto deve ser responsabilidade dos aplicativos e/ou do SGBD. Deve ser realizada alguma ponderação: normalizar demais diminui a eficiência dos aplicativos e de menos abre flancos para inconsistências.

5. Modelo Entidade-Relacionamento (ME-R)

O Modelo Entidade-Relacionamento (MER) foi originalmente criado pelo norte-americano Peter Pin-Shan Chen enquanto trabalhava no Massachusetts Institute of Technology. O MER foi colocado para conhecimento público durante a conferência internacional VI_DB no ano de 1975. Segundo seu criador, o MER é uma poderosa ferramenta de apoio ao projeto conceitual de Bancos de Dados propiciando uma visão mais natural e consistente do mundo real através do uso de conceitos simples, porém de grande representatividade.

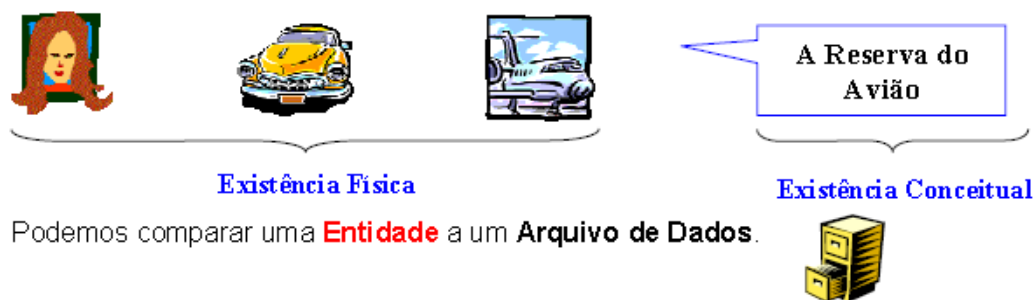
O ME-R é um modelo utilizado para descrição dos dados do mundo real a nível conceitual, isto é, em um nível em que o interesse é selecionar as informações importantes para serem armazenadas na base de dados, mas não há preocupação de como essas informações serão armazenadas nem de qual base de dados será utilizada.

No ME-R os dados são representados por Entidades, atributos e Relacionamentos.

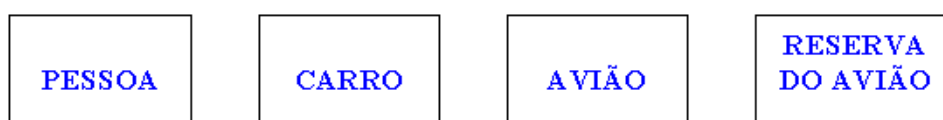
5.1. Entidades e Atributos

Um Banco de Dados normalmente armazena as informações em grupos ou conjuntos, isto porque para as aplicações mais usuais existem muitas semelhanças entre os elementos que são armazenados.

- **Entidades:** São conjunto de elementos, evento, ser, fato ou objeto (concreto ou abstrato) do mundo real que existe e é distinguível dos outros elementos.
- Por exemplo: Clientes, Frota de carros, Frotas de avião, Reserva do voo.



A representação de uma Entidade no modelo Entidade-Relacionamento é feita por um retângulo, com o nome desta entidade em seu interior, como mostra a figura abaixo:



- **Conjunto de Entidades:** grupo de entidades de um mesmo tipo, isto é, com os mesmos atributos e sujeitas aos mesmos relacionamentos. Exemplo: alunos, professores, livros - contas, clientes e funcionários.
- **Atributos:** São as informações que serão usadas para obter os dados dentro de uma determinada entidade. É cada elemento pertencente a um conjunto. Todo objeto para ser entidade possui propriedades que são descritas por atributos e

valores. Estes atributos e seus valores, juntos, descrevem as instâncias de uma entidade. Os conjuntos de entidades não precisam ser independentes, ou seja, uma entidade pode pertencer a mais de um conjunto de entidades simultaneamente. Exemplo: um funcionário de um banco pode ser também cliente do banco, nesse caso o funcionário estaria fazendo parte dos dois conjuntos de entidades (funcionários e clientes) simultaneamente.

Uma entidade é representada e distinguida das demais por um conjunto de atributos.

Exemplos:

Aluno – RG,
Nome,
Endereço,
Fone,
...

Conta – nº da conta,
Saldo,
...

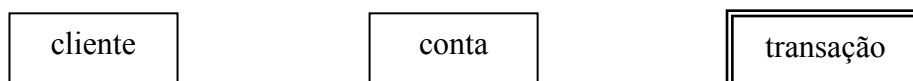
Livro – nº de tombo,
Título,
Autores *,
Editora,
Edição,
Ano de publicação,
...

O * ao lado do atributo autores representa que esse atributo é multivalorado. Um atributo é dito multivalorado quando nele pode-se armazenar um ou mais valores distintos ao mesmo tempo, no exemplo um mesmo livro pode ter um ou mais autores.

Pode ocorrer que alguma Entidade “A” possua uma dependência existencial com outra Entidade “B” e neste caso cada Atributo da Entidade “B” exista somente porque está associado à Entidade “A”. Este tipo de Entidade é denominada Entidade Fraca.

Exemplo:

A existência de um grupo de transações numa conta bancária somente será verdadeira se o cliente já tiver realizado a abertura da conta. Logo, em relação às entidades Cliente e Conta, a entidade Transação constitui-se uma Entidade Fraca.

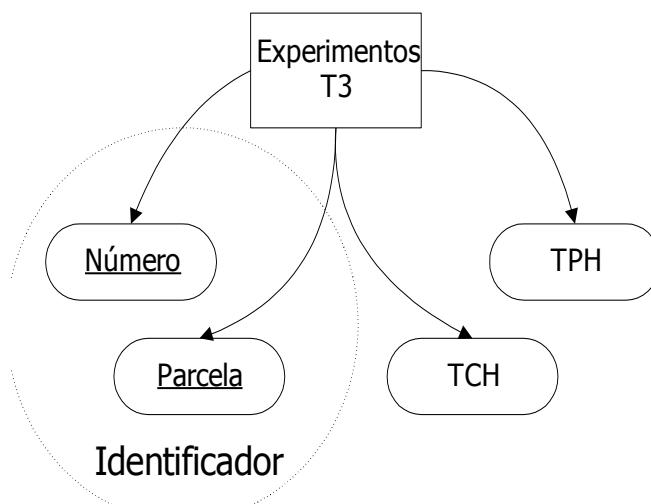


Produtos é um Tipo de Entidade Fraca, pois cada produto existente neste conjunto (e armazenado no Banco de Dados) existe somente porque um cliente (Entidade de clientes) o comprou.



Os Atributos são normalmente associados a Entidades. Entre estes Atributos deve existir um ou mais onde suas instâncias são diferentes para cada Entidade. O menor conjunto de Atributos para distinguir cada Entidade é chamado de **Identificador**.

Por exemplo:



O Domínio é o conjunto de valores válidos para um atributo, a definição do tipo de dados que esse atributo representa.

Exemplos:

Nome: texto [30]

Nota: numérico

≥ 0 e ≤ 10

Idade: numérico ≥ 0

5.2. Relacionamentos

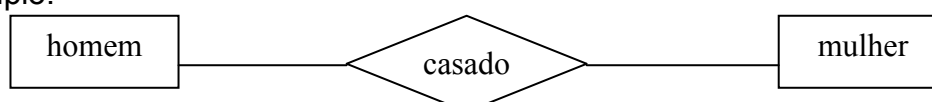
Nenhuma informação armazenada no Banco de Dados existe isoladamente. Todos os elementos pertencentes ao mundo real (restrito) modelado de alguma forma estão associados a outros elementos. Normalmente estas associações representam ações físicas ou alguma forma de dependência entre os elementos envolvidos.

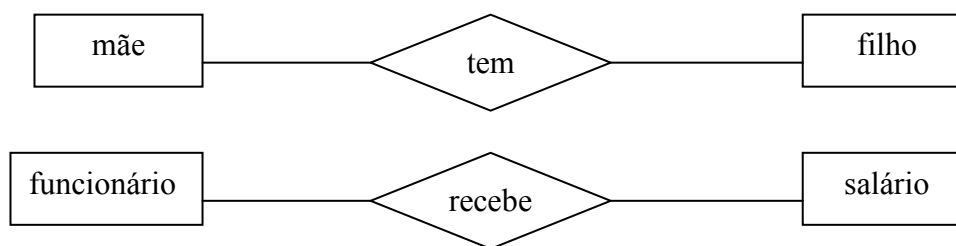
5.2.1. Definição

Relacionamento: Associação entre duas ou mais entidades, ou seja, é um fato, um acontecimento que liga dois objetos, duas entidades existentes no mundo real.

Considerando que estamos nos orientando para aplicações que serão desenvolvidas e administradas por um Sistema de Gerenciamento de Banco de Dados, poderíamos estender o conceito, principalmente para ambientes relacionais, como sendo relacionamento o fato que efetua a junção de duas ou mais tabelas de dados.

Por exemplo:





Os Relacionamentos podem possuir atributos próprios.

Exemplo:

O empréstimo de um livro deve conter a **data da retirada** e a **data de devolução**.

O Papel é a função que uma entidade executa em um relacionamento.

Exemplos:

Leitor **empresta** um livro

Livro **é emprestado** por um leitor

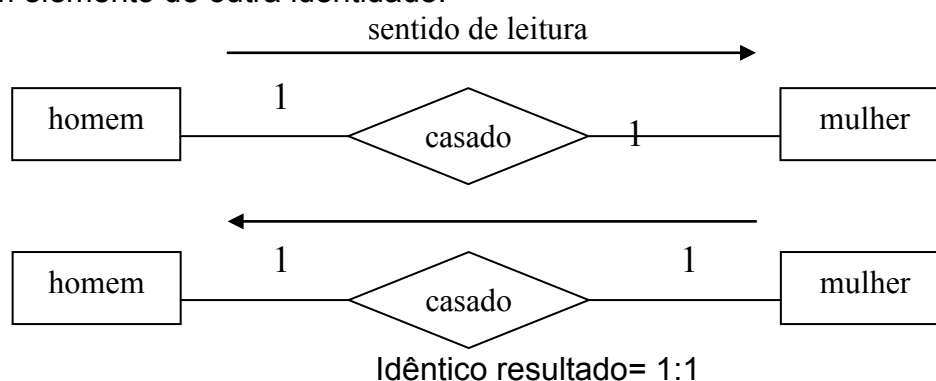
5.2.2. Grau do relacionamento ou Cardinalidade

Quando temos um relacionamento entre duas entidades, o número de ocorrências de uma entidade que está associado com ocorrências de outra entidade determina o **Grau do Relacionamento** ou **Cardinalidade**.

É importante salientar que lemos o diagrama somente em um sentido, e isto está incorreto dentro do conceito de relacionamento, pois os mesmos não são unidirecionais. Muitos dos erros na construção do modelo de dados ocorrem por serem realizados exames apressados sobre a cardinalidade dos relacionamentos, efetuando-se a análise somente no sentido de um interesse específico, sem que se verifique a cardinalidade no sentido inverso. Logo, o grau de um relacionamento é definido pela interpretação da cardinalidade dos dois sentidos. O mundo real apresenta-se com três possibilidades de relacionarmos os dados, ou seja, três graus de relacionamento, que são:

Relacionamento Um-para-Um (1:1)

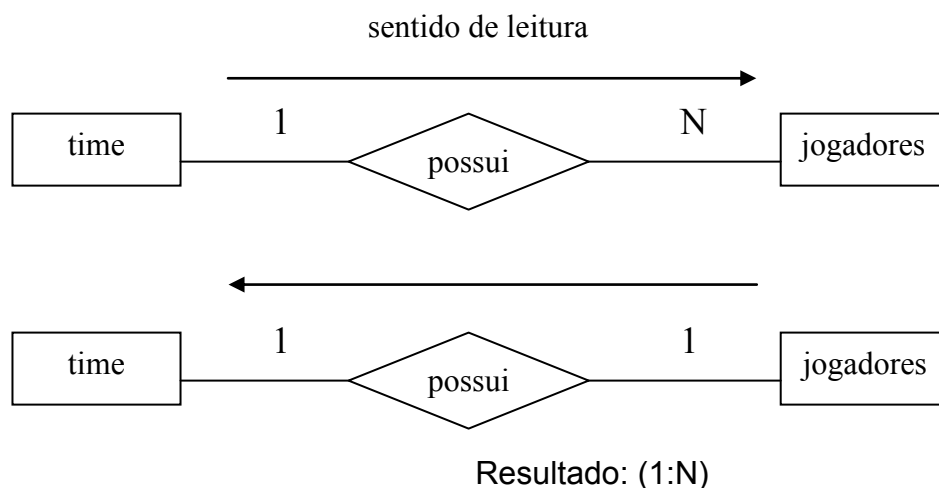
Neste grau de relacionamento, cada elemento de uma entidade relaciona-se com um e somente um elemento de outra identidade.



Relacionamento de Um-para-Muitos (1:N)

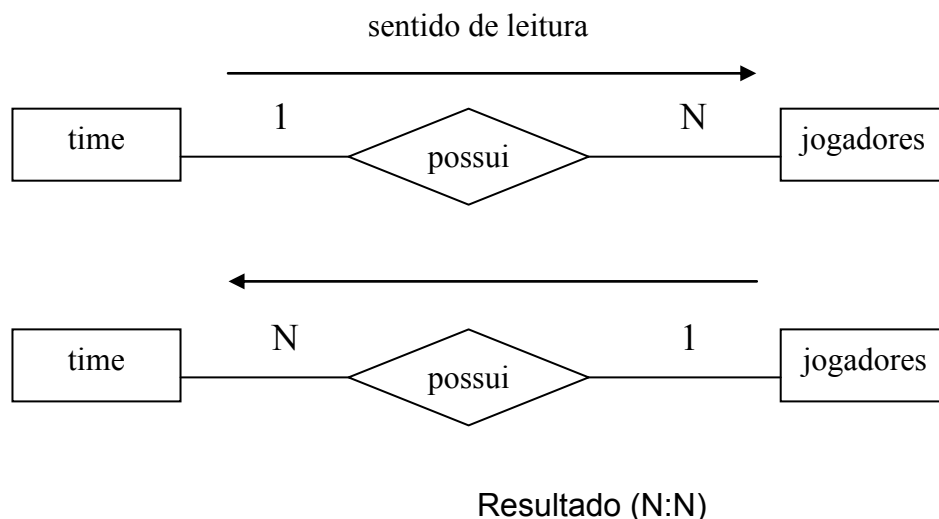
Este grau de relacionamento é o mais comum no mundo real, sendo o que denominamos de relacionamento básico entre entidades, entretanto possui características específicas, quanto ao sentido de leitura dos fatos e sua interpretação.

Um elemento da entidade 1 relaciona-se com muitos elementos da entidade 2, mas cada elemento da entidade 2 somente pode estar relacionado a um elemento da entidade 1. Ou seja, o grau de cardinalidade determinante sempre é o maior grau obtido da interpretação dos fatos.



Relacionamento de Muitos-para-Muitos (N:N)

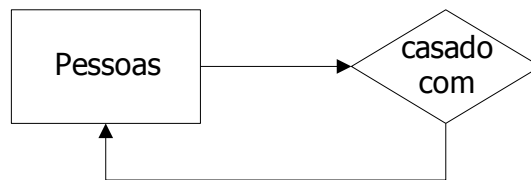
Identifica-se esta cardinalidade pelo fato de que em ambos os sentidos de leitura encontramos um grau Um-para-Muitos, o que caracteriza ser então um contexto geral de Muitos-para-Muitos.



5.2.3. Categoria do relacionamento

A quantidade de Entidades envolvida em um Relacionamento pode ser determinada por sua semântica. Desta forma podem-se categorizar os relacionamentos em:

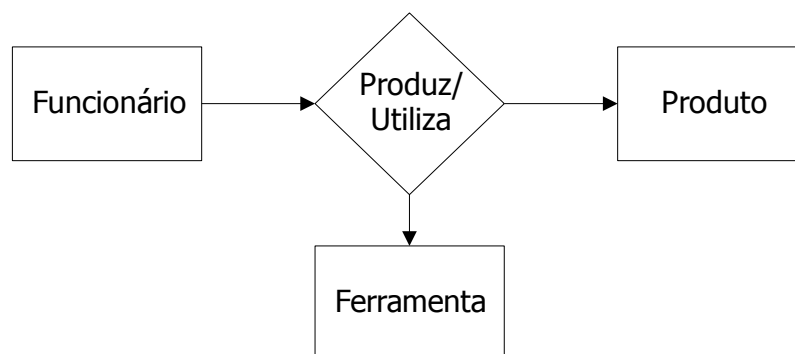
Unário: é um relacionamento que envolve uma única entidade.



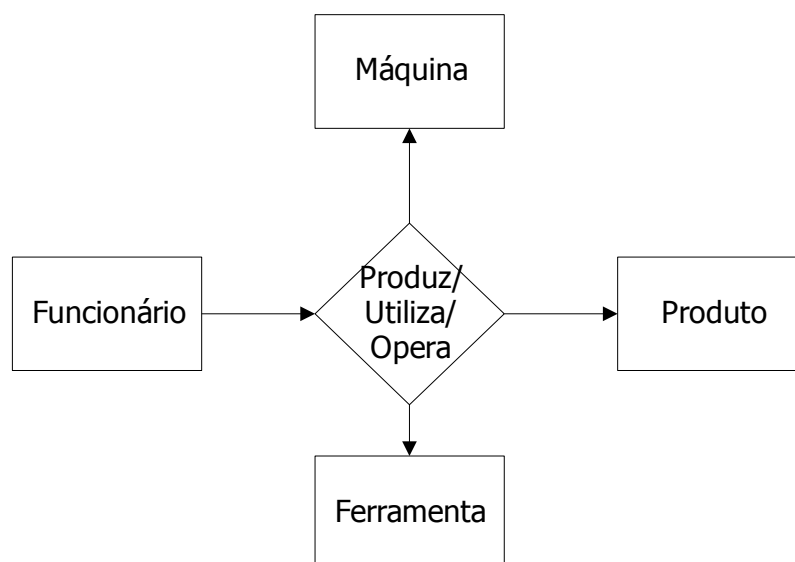
Binário: é um relacionamento que envolve duas entidades.



Ternário: é um relacionamento que envolve três entidades.



Quaternário: é um relacionamento que envolve quatro entidades.



Os relacionamentos também podem possuir atributos para caracterizá-los. Como os relacionamentos são normalmente ações (realizadas ou a se realizar) ou dependências entre as Entidades envolvidas, então cada atributo que caracteriza estas associações normalmente representa algum dos seguintes aspectos:

Quantificável: quantidade de elementos que a associação utiliza (ou utilizou) ou realiza (ou realizou).

Temporal: data (início ou fim) da associação, intervalo de tempo da associação.

Explicativo: descrição de como ou porque foi realizada (ou se realiza) a associação.

Espacial: descrição da localização da associação

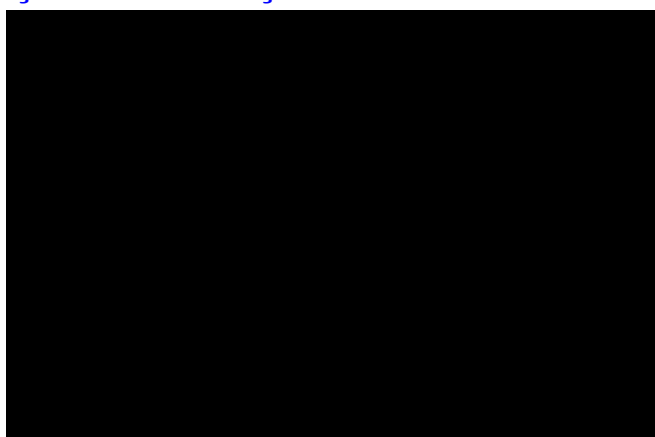
Referencial: indicação do agente da associação

5.3. Abstrações de Dados

Existem algumas construções semânticas que são naturais dos seres humanos e são usadas para facilitar o entendimento do mundo. Para tentar representar com fidelidade estas construções foram criadas as diversas Abstrações de Dados.

Abstração é o ato de separar mentalmente elementos de uma totalidade complexa, seja material, representação ou fato real.

5.3.1. Especialização / Generalização

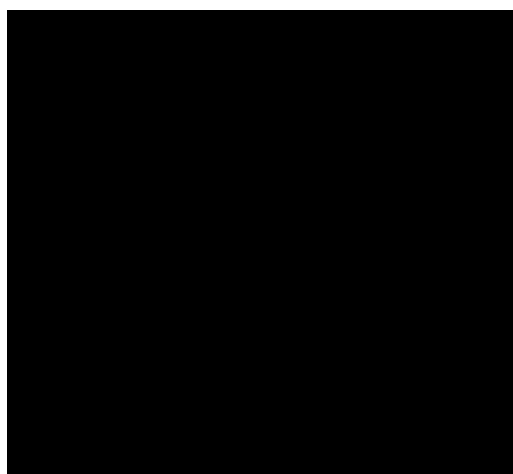


Especialização é uma abstração que permite subdividir um grande conjunto de elementos em conjuntos menores de acordo com padrões que estabelecidos pelas necessidades da aplicação.

Generalização é a abstração inversa da Especialização, ou seja, agrupam-se conjuntos de elementos com alguma semelhança semântica em um conjunto maior.

As características semelhantes que existem entre os Homens e as Mulheres são representadas no conjunto Pessoas, como por exemplo, o nome, a idade e a profissão. Todo Homem (ou Mulher) possui estas características, pois é uma Pessoa. Esta migração de características é denominada Herança.

Além das características herdadas de Pessoas, o conjunto de Homens possui suas características próprias, por exemplo, altura, o mesmo ocorrendo com as Mulheres com a característica peso.



Sucessivas operações de Generalização (ou Especialização) descrevem uma hierarquia de conjuntos que pode:

- a) Não ser uma árvore, uma vez que conjunto Filho pode ter mais de um conjunto Pai, o que pode estabelecer Múltiplas Heranças, quando então pode existir a possibilidade de conflito entre as características herdadas. Exemplo: Um conjunto de animais Anfíbios são especializações de Terrestres e também de Aquáticos.

Terrestres e Aquáticos possuem a característica Velocidade. Nos animais Terrestres representa a velocidade de locomoção na terra e nos animais Aquáticos é a velocidade de deslocamento na água. Se Anfíbios herdam Velocidade de ambos os grupos existe o conflito.

- b) Ser constituída por conjuntos não mutuamente exclusivos, ou seja, dois ou mais conjuntos podem conter um mesmo elemento. Exemplo: um réptil que anda e também nada.

O tamanho da Hierarquia de Generalização é determinado pelo interesse da aplicação, ou seja, subdivide-se em conjunto até que a representação esteja de acordo com as necessidades.

5.3.2. Decomposição / Composição

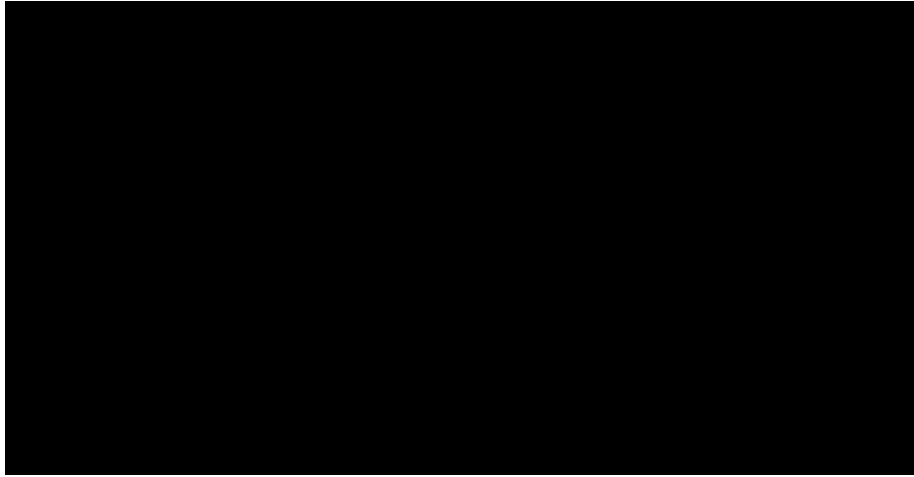
Decomposição é uma abstração que permite criar conjuntos de elementos que representam partes de um conjunto de elementos complexos.

Composição é a abstração inversa da Decomposição, no DER é o atributo composto.



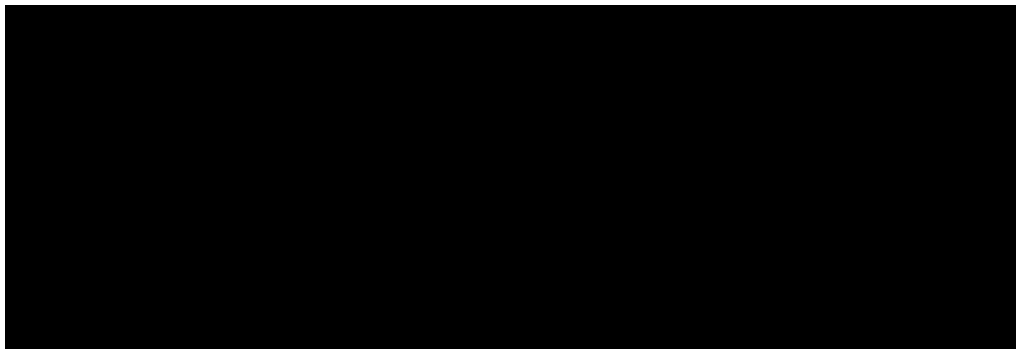
5.3.3. Derivação (Sumarização)

Alguns dados armazenados no BD podem ser resultado de algum tipo de cálculo envolvendo outros dados (no DER é o atributo derivado).



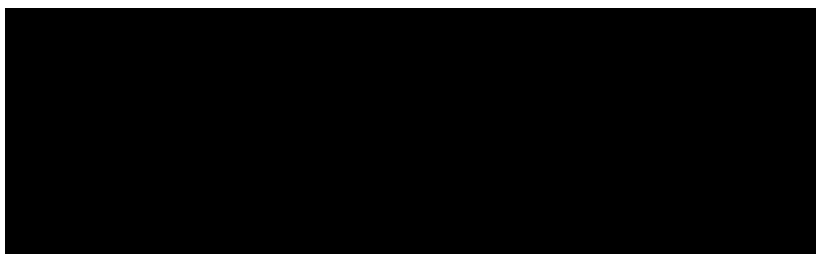
5.3.4. Agregação

O resultado de uma associação entre Entidades pode criar outra Entidade, ou melhor, um Agregado. Um Agregado pode se relacionar com outras Entidades e possuir características próprias.



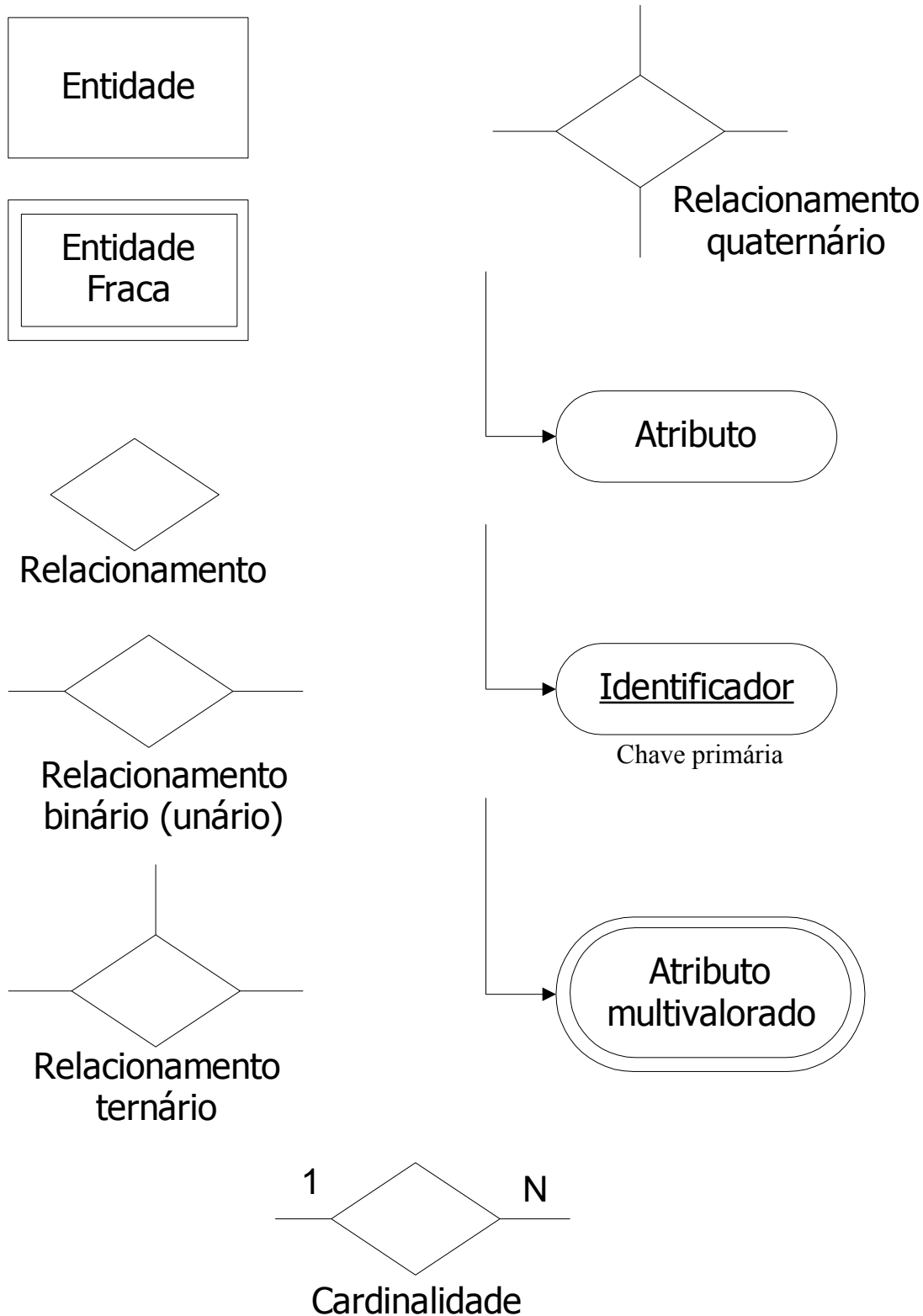
5.3.5. Divisão

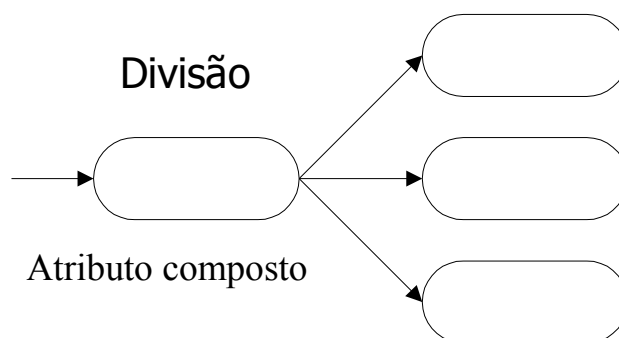
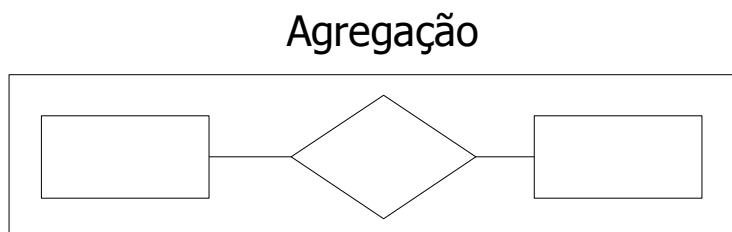
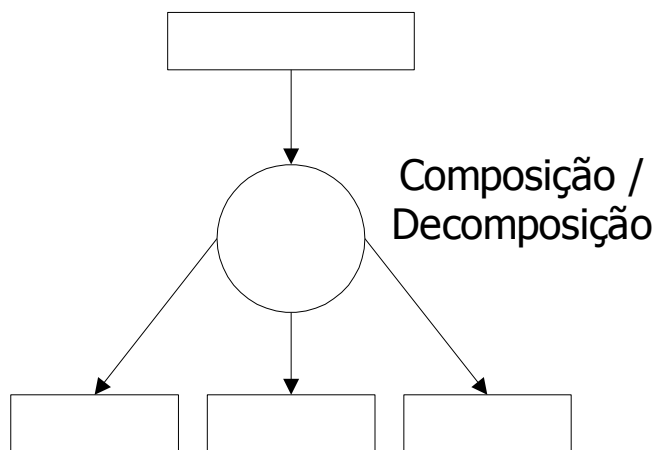
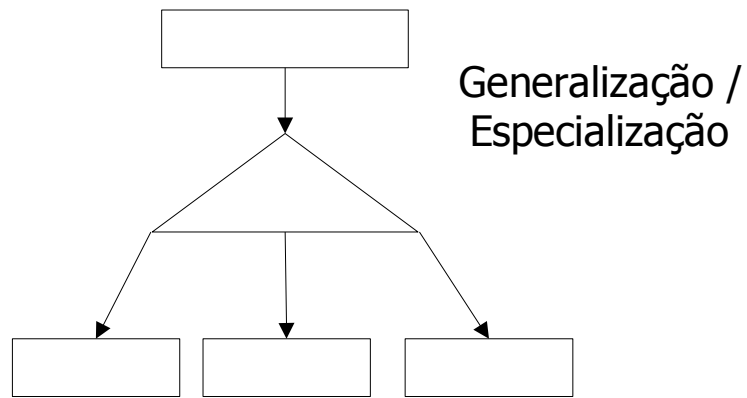
As características de uma Entidade ou Relacionamento podem apresentar-se complexas e assim talvez criar dificuldades para o seu completo entendimento. Através desta Abstração pode-se representar partes de Atributos complexos (no DER é o atributo composto)



6. Diagrama Entidade-Relacionamento (DE-R)

A representação gráfica através do Diagrama Entidade-Relacionamento foi criada por Peter Chen para facilitar a visualização dos resultados de uma modelagem. Como o DER apresenta símbolos muito simples podem ser utilizados para a depuração da modelagem junto ao cliente.

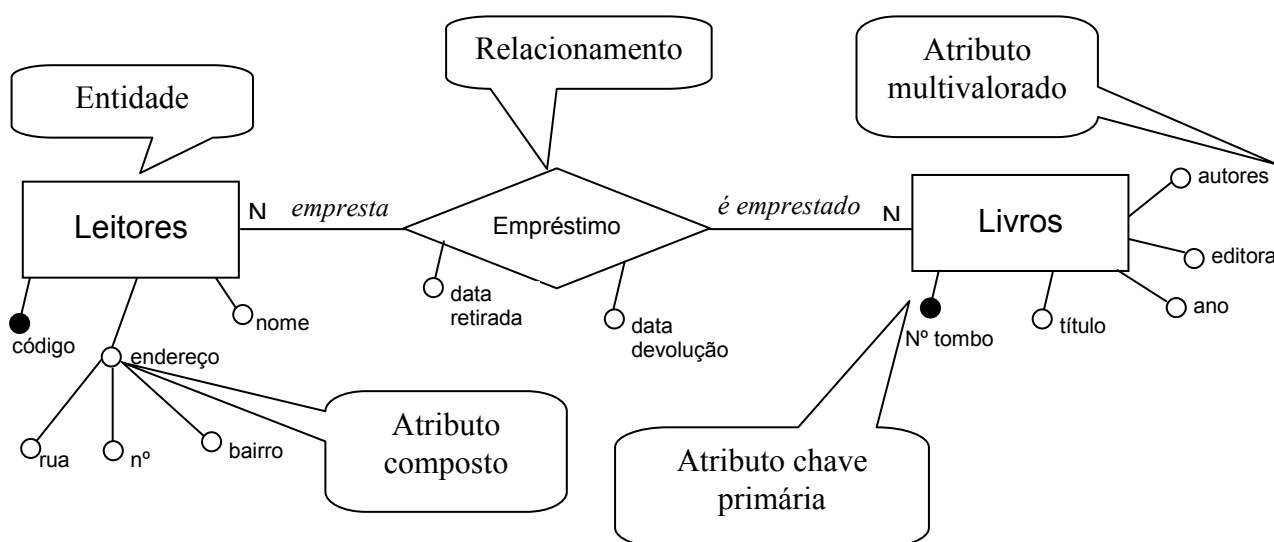




- **Linhas:** liga conjunto de entidades a conjunto de relacionamentos e atributos a conjuntos de entidades / conjunto de relacionamentos.

6.1.1. Exemplo DER

- Um DER para representar um sistema de BD que controla o empréstimo de livros de uma biblioteca. Os leitores possuem um cadastro contendo seu código, nome, endereço, com rua, número e bairro. Os leitores podem emprestar vários livros que são cadastrados por número do tombo, título, ano, editora e autores, os livros podem ter vários autores. Ao se emprestar o livro ao leitor é registrada a data de retirada e a data de devolução. No diagrama essa descrição narrativa será representada assim:

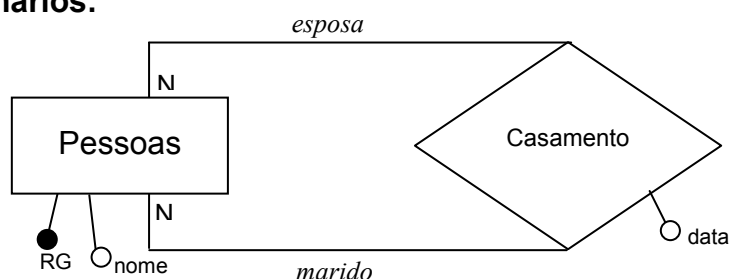


Obs. Chave Primária é o identificador da entidade, ou seja, é o Conjunto mínimo de atributos (um ou mais) que permitem identificar unicamente uma entidade em um conjunto de entidades.

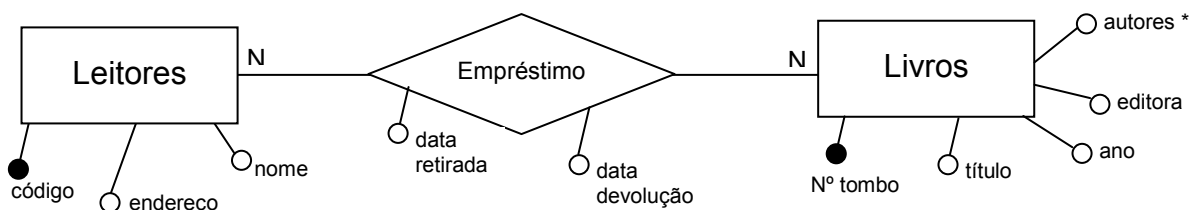
6.2. Tipos de Relacionamentos representados no DER

6.2.1. Exemplos

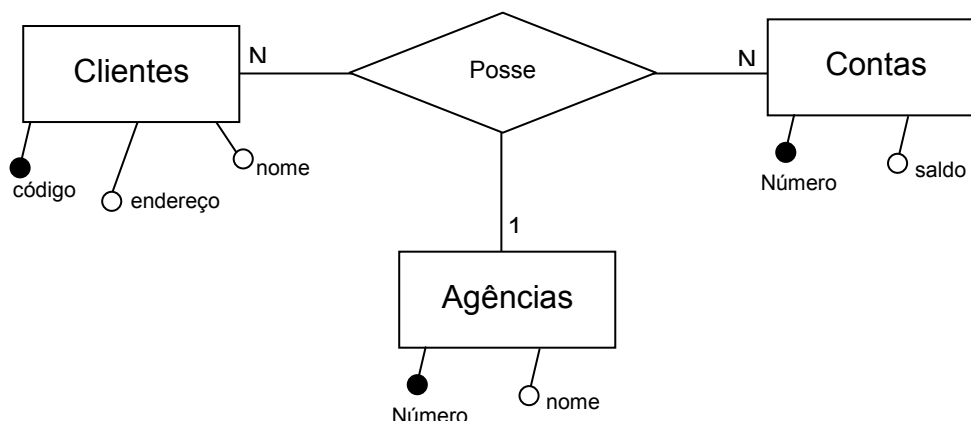
- **Relacionamentos unários:**



▪ Relacionamentos binários:



▪ Relacionamentos múltiplos:



6.3. Entidade Forte X Entidade Fraca no DER

- **Entidades Fortes:** entidades que possuem Chave Primária e não dependem de outra entidade para existir.

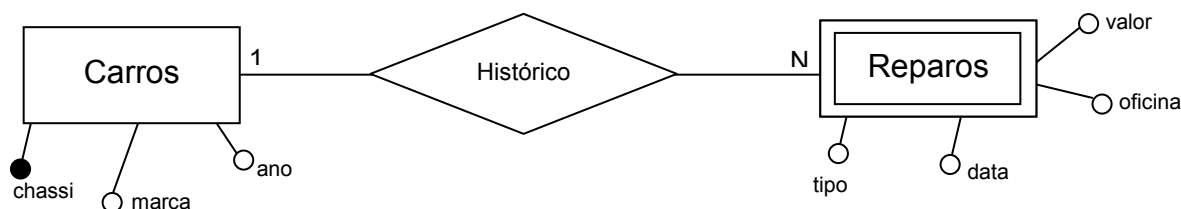
Exemplos:

- Carros, Alunos, Clientes, Contas.

- **Entidades Fracas:** entidades que não possuem atributos suficientes para formar uma chave primária depende de outra entidade para existir.

Exemplo:

- **Reparo de um carro:** tipo, data, valor, oficina



Como a Entidade Fraca não possui chave primária própria, sua chave primária é formada pela composição da chave primária da Entidade Forte da qual ela depende mais um discriminador. Discriminador é a composição de um ou mais atributos da entidade fraca que são únicos quando associados à entidade forte.

Exemplo:

- **Chave primária dos reparos:** chassi + tipo + data.

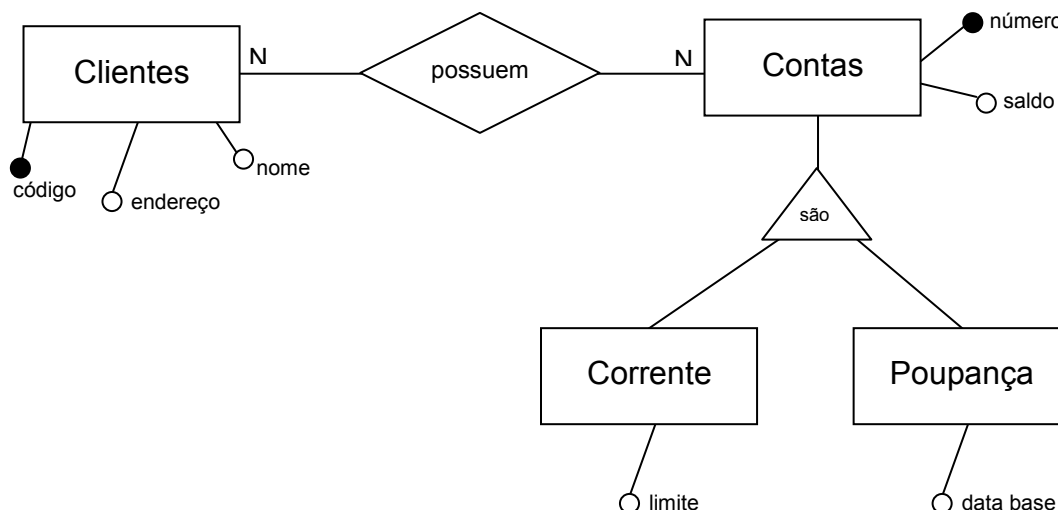
Dependência existencial: as entidades fracas dependem existencialmente de entidades fortes, por exemplo, não faz sentido manter o histórico de reparos de um carro,

se o carro em questão não existe mais, portanto os reparos dependem existencialmente dos carros.

6.4. Generalização X Especialização no DER

- **Generalização:** é o resultado da união de 2 ou mais conjunto de entidades de um nível mais baixo (mais detalhado), gerando um único conjunto de entidades em um nível mais alto (mais geral). Enfatiza-se as semelhanças entre entidades de conjuntos distintos.
- **Especialização:** é o resultado da separação de um ou mais subconjunto de entidades, a partir de um conjunto de entidades no nível mais alto. Enfatiza-se as diferenças entre entidades do mesmo conjunto.

Exemplo:

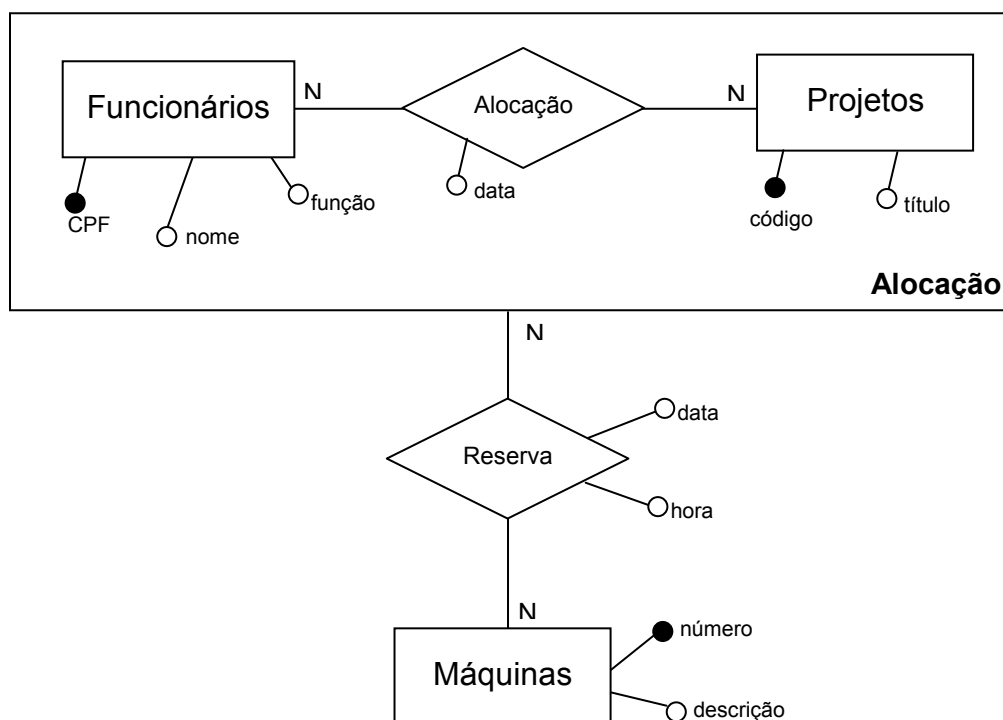


Obs.: Os atributos comuns são associados às entidades do nível + alto. Entidades do nível + baixo herdam os atributos das entidades superiores e, além disso, possuem seus atributos próprios.

6.5. Agregação no DER

- **Agregação:** abstração de dados onde relacionamentos são tratados como conjunto de entidades em nível mais alto.

Agregações são usadas quando desejamos associar um relacionamento a um conjunto de entidades através de outro relacionamento (não é permitido ligar dois relacionamentos)

Exemplo:

6.6. Modelando um Sistema de Dados - Etapas

Tendo realizado entrevistas com o cliente e/ou usuário(s) para determinar suas necessidades de informação e definido qual o problema a ser resolvido, ou seja, tendo determinado as fronteiras que delimitam e restringem o mini-mundo a ser modelado, então deve-se seguir roteiro para se determinar uma primeira versão do DER, que pode ser descrito pelos seguintes passos:

1. Determinar o Universo.
2. Determinar as entidades envolvidas.
3. Determinar os atributos das entidades.
4. Determinar os identificadores de entidades (chave primária).
5. Determinar os relacionamentos entre elas.
6. Determinar os atributos dos relacionamentos.
7. Determinar as restrições de cardinalidade.
8. Determinar as abstrações de dados.

Obs.: Nunca ligar duas entidades sem um relacionamento no meio.

Nunca ligar dois relacionamentos diretamente, quando necessário é preciso criar uma agregação.

6.6.1. Cuidados na criação do DER

Alguns cuidados devem ser tomados durante a criação do DER:

- Um atributo não pode ter outros atributos associados, de modo que se forem encontrados (em sua aplicação) significa que não se trata de um atributo e sim de uma entidade.

- Uma entidade que não possui pelo menos um atributo além do identificador ou está com sua especificação incompleta ou não se trata de uma entidade mais de um atributo.
- Um relacionamento é uma associação entre entidades. A completa e perfeita representação de uma associação somente está correta se todas as entidades necessárias para a existência do relacionamento estão interligadas.

Depois de criada a primeira versão do DER deve-se apresentar ao cliente para que seja verificado se o diagrama está correto. Sucessivas apresentações do DER devem ser realizadas enquanto forem detectadas falhas na representação.

À primeira vista pode-se pressupor que esta rotina de trabalho trará atrasos na construção do banco de dados, porém os especialistas em engenharia (software, mecânica e etc.) sabem da extrema importância da fase de projeto. Erros ocorridos nesta fase acarretam graves atrasos e aumento no custo de realização do produto.

A modelagem através do MER é exatamente uma etapa do Projeto de Banco de Dados em que se realiza o projeto conceitual do Banco de Dados, uma etapa muito importante. Por este motivo, o MER se tornou uma ferramenta de modelagem entre as mais difundidas, estimadas e utilizadas no mercado de informática.

7. Modelo Relacional

7.1. Conceitos

- Banco de Dados representado por um conjunto de tabelas também chamadas de relações (relacionam um conjunto de valores).
- Cada linha de uma tabela também é chamada de uma tupla.
- **Esquema de um BD Relacional:** definição do conjunto de tabelas e seus atributos que irão compor a base de dados (estrutura do BD relacional)
- **Instância de um BD Relacional:** conjunto de dados armazenados no BD em um determinado momento.

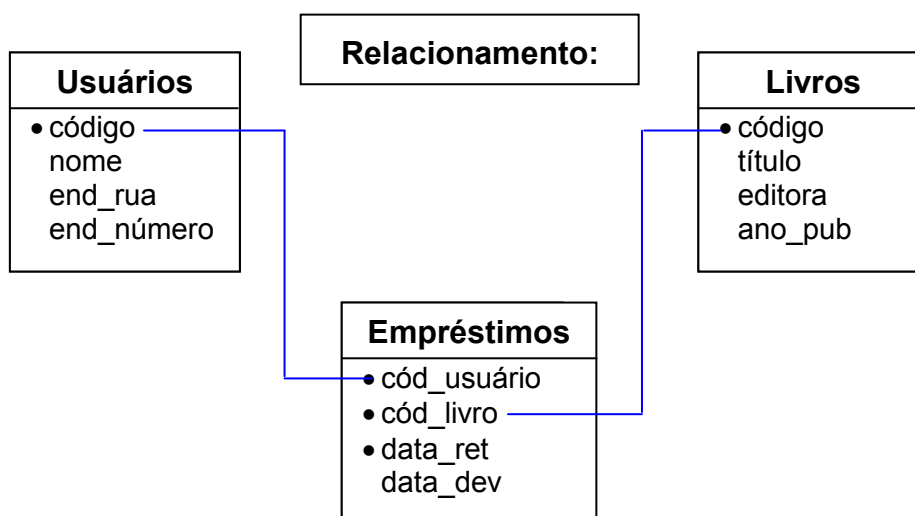
Exemplos:

Esquema:

Usuários (código inteiro [chave primária],
nome caracteres [30],
end_ rua caracteres [20],
end_número inteiro)

Livros (código inteiro [chave primária],
título caracteres[20],
editora caracteres[10],
ano_pub inteiro)

Empréstimos (cod_usuário inteiro [chave primária],
cod_livro inteiro [chave primária],
data_ret data [chave primária],
data_dev data)



Instância:

Usuários

código	nome	end_rua	end_número

Empréstimos

cód_usuario	cód_livro	data_ret	data_dev

Livros

código	título	editora	ano_pub

7.2. Mapeando ME-R para o Modelo Relacional

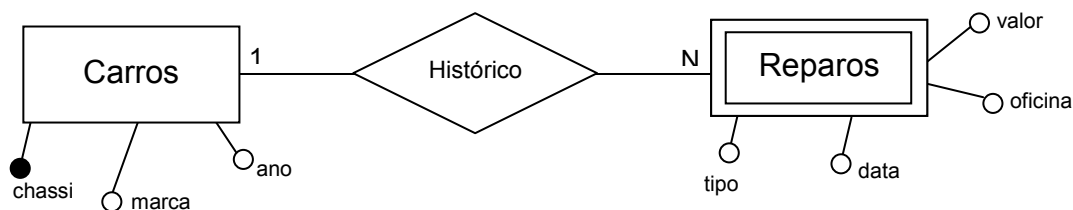
Um Diagrama de Entidade relacionamento pode ser representado por um conjunto de tabelas (esquema do Modelo Relacional)

- **Conjunto de Entidades:** cada conjunto de entidades do DE-R gera uma tabela no Modelo Relacional.
 - **Entidades Fortes:** tabela com seus atributos próprios



Contas (número, saldo)

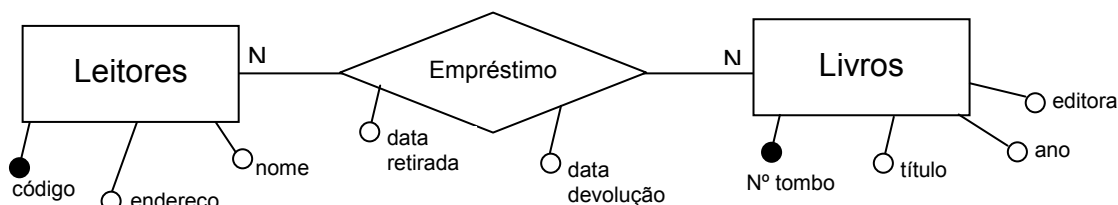
- **Entidades fracas:** tabela com chave primária da entidade forte da qual ela depende mais seus atributos próprios



Carros (chassi, marca, ano)

Reparos (chassi, tipo, data, oficina, valor)

- **Conjunto de Relacionamentos:** podem ou não gerar tabelas:
 - **Relacionamentos Múltiplos ou Relacionamentos Binários de n para n:** geram tabelas com as chaves primárias das entidades envolvidas mais os atributos próprios do relacionamento.



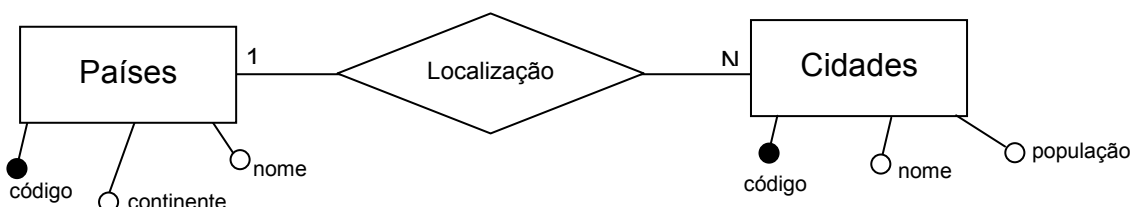
Leitores (código, nome, endereço)

Livros (N° tombo, título, ano, editora)

Empréstimos (código, N° tombo, data retirada, data devolução)

- **Relacionamentos Binários de 1 para n:** não geram tabelas. Para associar as tuplas das tabelas no Modelo Relacional deve-se transpor a chave de um conjunto de entidades para o outro (a chave da entidade do lado com 1 é transposta para a entidade do lado n). A chave transposta nesse caso não compõe a chave primária da entidade que a recebeu (é uma chave estrangeira).

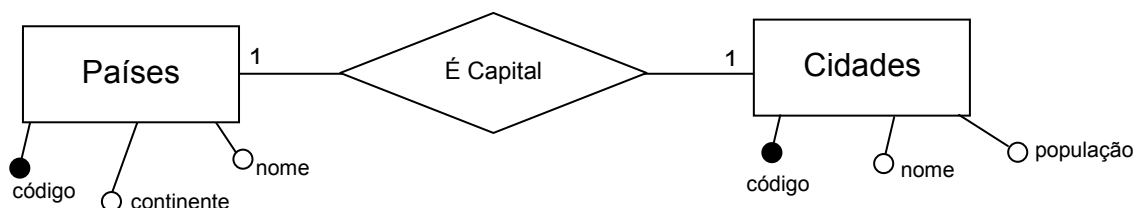
Obs.: no caso da transposição de chave entre a entidade forte e a entidade fraca que dela depende, a chave transposta compõe a chave primária da entidade fraca.



Países (código, nome, continente)

Cidades (código, nome, população, código_país)

- **Relacionamentos de 1 para 1:** para decidir como será feita a transposição de chaves, deve-se observar a participação total com relação ao relacionamento.

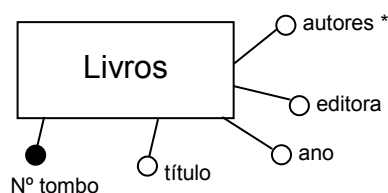


Todo país tem uma capital, mas nem toda cidade é capital de um país, portanto a tabela de países é que deve receber a código da cidade que é capital.

Países (código, nome, continente, código_cidade)

Cidades (código, nome, população)

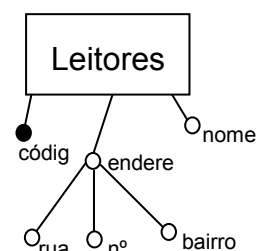
- **Atributos Multivalorados:** novas tabelas devem ser criadas para armazená-los juntamente com a chave primária da entidade que os possui.



Livos (N° tombo, título, ano, editora)

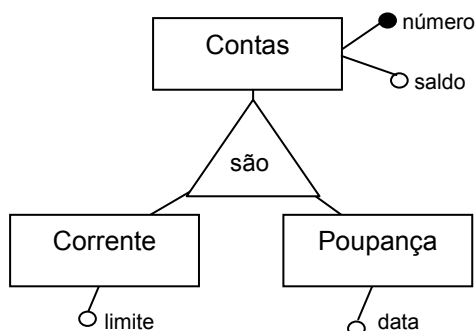
Autores (N° tombo, autor)

- **Sub-Atributos:** além dos atributos comuns, são gerados somente os sub-atributos do seu atributo.



Leitores (código, endereço_rua, endereço_nº, endereço_bairro, nome)

- **Generalização e Especialização:** há duas maneiras de definir as tabelas para uma generalização ou especialização:
 - Define-se uma tabela para o conjunto de entidades do nível mais alto (com os atributos comuns) e uma tabela para cada entidade do nível mais baixo (com seus atributos próprios + a chave primária da entidade do nível mais alto).



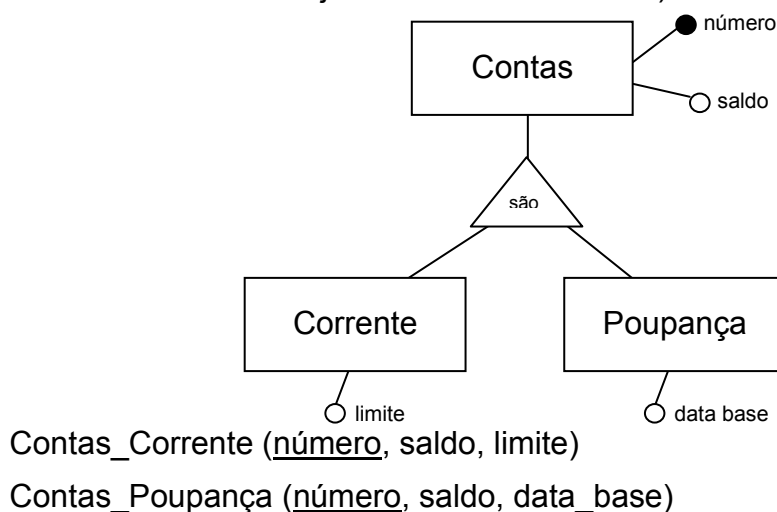
Contas (número, saldo)

Corrente (número, limite)

Poupança (número, data_base)

- Define-se tabelas apenas para os conjuntos de entidades do nível mais baixo (com seus atributos próprios, mais todos os atributos herdados do conjunto de entidades do nível mais alto).

Obs.: Essa opção só é permitida se a generalização / especialização for mutuamente exclusiva (uma entidade não pertence a mais do que um dos subconjuntos) e total (todas as entidades do nível mais alto pertencem a um dos subconjuntos do nível de baixo).



- **Agregação:** como a agregação é a representação de um conjunto de relacionamentos como se fosse um conjunto de entidades, o mapeamento para tabelas é semelhante ao mapeamento aplicado em entidades e relacionamentos normais.

Obs.: Faça primeiro o mapeamento das entidades e relacionamentos que estão agregados.

7.3. Passos Mapeamento MER – MR

Passo 1: Mapear Conjuntos de Entidades Regulares

Mapear todos os conjuntos de entidades regulares (que não são fracas).

Para cada conjunto de entidade E no esquema ER cria-se uma relação R que inclui todos os atributos de E. Caso exista atributo composto, inclua todos os atributos elementares que compõem o atributo composto. Escolha um dos atributos chave de E como chave primária para a relação R.

Primeiro Passo:

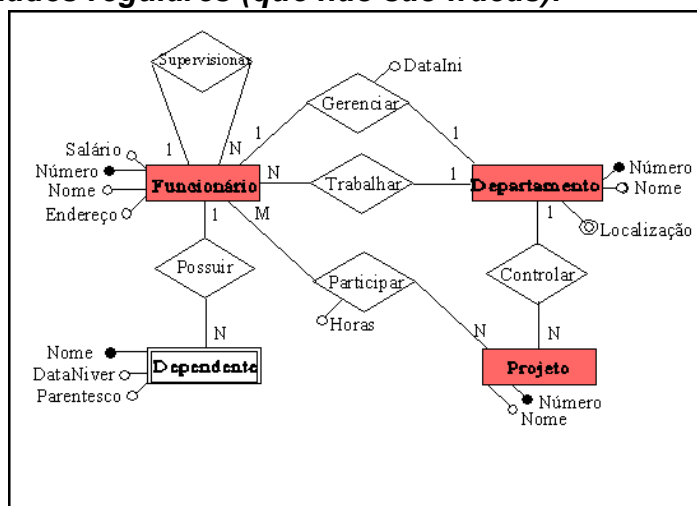
Funcionário = {FNúmero, FNome, Endereço, Salário}

Departamento = {DNúmero, DNome}

Projeto = {PNúmero, PNome}

OBSERVAÇÕES:

1) O atributo "Localização" não foi representado na relação "Departamento" pois é um atributo multi-valorado e será tratado no passo 7.



2) O conjunto de entidade "Dependente", é um conjunto de entidade fraco e será tratado no seguinte, o passo 2.

Passo 2: Mapear Conjuntos de Entidades Fracas

Para cada conjunto de entidade fraca F no esquema ER cria-se uma relação R formada por todos os atributos do conjunto de entidade fraca, mais os atributos que são chave das relações que mapeiam os conjuntos de entidades envolvidos pelo conjunto de relacionamento total que torna esse conjunto de entidade fraca.

A chave da relação que mapeia o conjunto de entidade fraca é a chave desse conjunto de entidade concatenada com as chaves das relações que mapeiam os demais conjuntos de entidades envolvidos no conjunto de relacionamento total que torna esse conjunto de entidade fraca.

É importante notar que um conjunto de entidade fraca só pode ser mapeado quando os demais conjuntos de entidades envolvidos no conjunto de relacionamento total, que torna esse conjunto de entidade fraca, já estiverem mapeados.

Primeiro Passo:

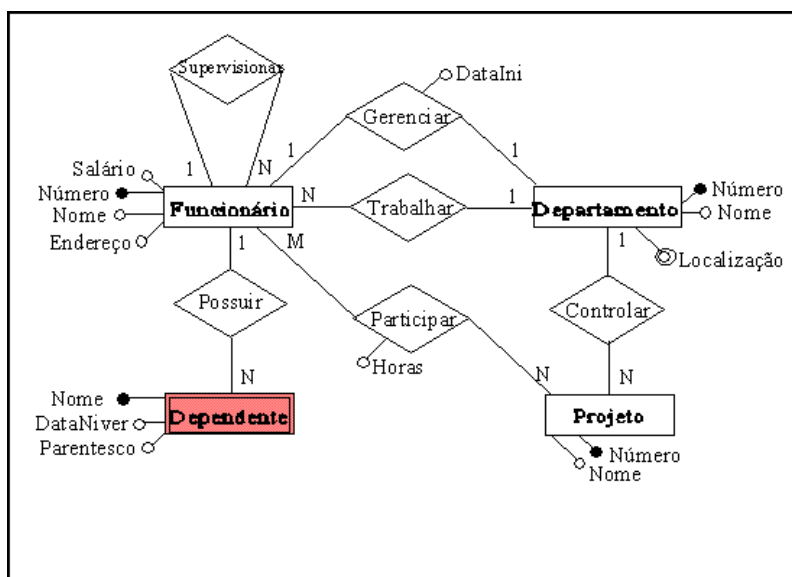
Funcionário = {FNúmero, FNome, Endereço, Salário}

Departamento = {DNúmero, DNome}

Projeto = {PNúmero, Pnome}

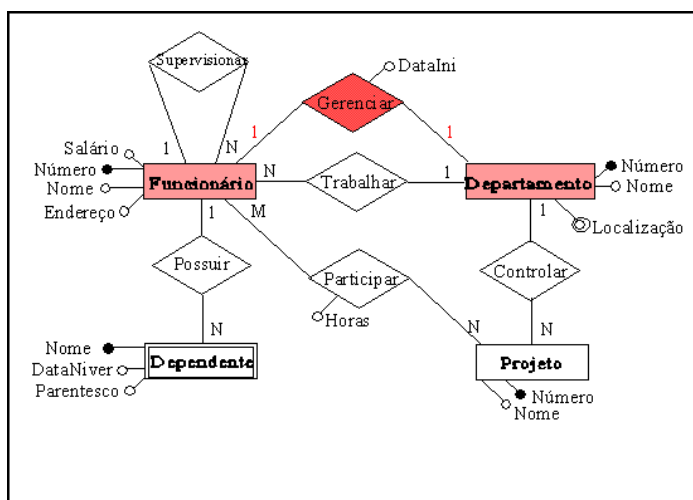
Segundo Passo:

Dependente = {DDependNome, FNúmero, DataNiver, Parentesco}



Terceiro Passo: Mapear Conjuntos de Relacionamentos Binário 1:1

O mapeamento de conjuntos de relacionamentos (CR) nem sempre resultam em uma nova relação, como é o caso dos conjuntos de relacionamento binário 1:1. Deve-se identificar os conjuntos de entidades S e T que participam do relacionamento. Um dos conjuntos de entidades deve ser escolhido, por exemplo S, e acrescenta-se a ele os atributos do conjunto de relacionamento e ainda os atributos chave da relação que mapeia o outro conjunto de entidade. Deve-se ressaltar que os atributos chave do conjunto de entidade T são incluídos como atributos não chave no conjunto de entidade S.



Primeiro Passo:

Funcionário = {FNúmero, FNome, Endereço, Salário}

Departamento = {DNúmero, Dnome, FNúmero, DataIni}

Projeto = {PNúmero, Pnome}

Segundo Passo:

Dependente = {DependNome, FNúmero, DataNiver, Parentesco}

Terceiro Passo:

Os atributos **FNúmero** e **DataIni** são adicionados à relação "Departamento", definida no primeiro passo. Note que o atributo **FNúmero** é adicionado como um atributo não chave na relação "Departamento".

Quarto Passo: Mapear Conjuntos de Relacionamento Binário Regular 1:N

Os conjuntos de relacionamento binário regular (não fraco) 1:N também não são representados como novas relações. Primeiro identifica-se o conjunto de entidade que participa da relação com cardinalidade N que será chamada de S e o outro conjunto de entidade chamada de T. Os atributos do conjunto de relacionamento são acrescentados no conjunto de entidade S, ou seja, o conjunto de entidade com cardinalidade N. Os atributos chave da relação que mapeia o conjunto de entidade que participa com cardinalidade 1, representado por T, são também acrescentados no conjunto de entidade S como atributos não chave.

Primeiro Passo:

Funcionário = {FNúmero, FNome, Endereço, Salário, SuperNúmero, DNum}

Departamento = {DNúmero, Dnome, FNúmero, DataIni}

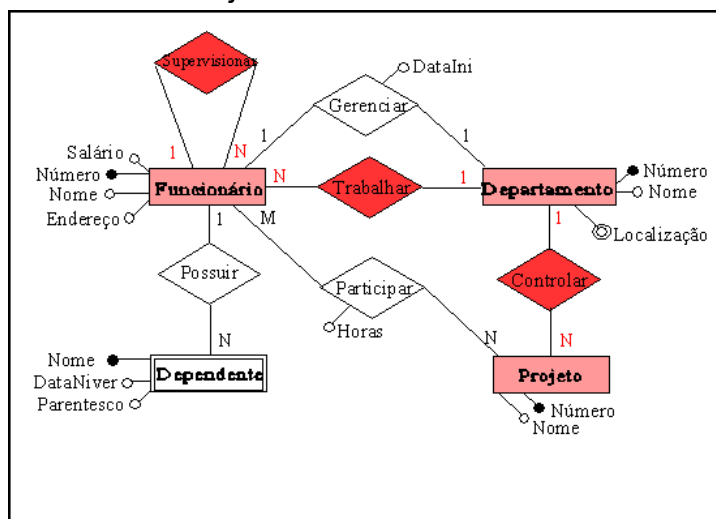
Projeto = {PNúmero, Pnome, DNro}

Segundo Passo:

Dependente = {DependNome, FNúmero, DataNiver, Parentesco}

Terceiro Passo:

- Os atributos FNúmero e DataIni são adicionados à relação "Departamento", definida no primeiro passo. Note que o atributo FNúmero é adicionado como um atributo não chave na relação "Departamento".



Quarto Passo:

- Neste passo são mapeados os relacionamentos: supervisionar, trabalhar e controlar.
- Avaliando o relacionamento **supervisionar**, nota-se que este relacionamento não possui atributos, logo, apenas o atributo **SuperNúmero** foi adicionado a relação "Funcionário", definida no primeiro passo.
- O mesmo acontece com o relacionamento **trabalhar**; apenas o atributo **DNum** foi adicionado a relação "Funcionário".
- No relacionamento **controlar**, que também não possui atributos, foi adicionado o atributo **DNro** na relação Projeto que já havia sido definida no primeiro passo.

Quinto Passo: Mapear Relacionamento Binário M:N

Para cada relacionamento binário M:N cria-se uma nova relação. Os atributos da relação são os atributos do conjunto de relacionamento juntamente com os atributos chave das relações que mapeiam os conjuntos de entidades envolvidos. A chave da relação é a concatenação dos atributos chave das relações que mapeiam os conjuntos de entidades envolvidos.

Primeiro Passo:

Funcionário = {FNúmero, FNome, Endereço, Salário, SuperNúmero, DNum}

Departamento = {DNúmero, Dnome, FNúmero, DataIni}

Projeto = {PNúmero, Pnome, DNro}

Segundo Passo:

Dependente = {DependNome, FNúmero, DataNiver, Parentesco}

Terceiro Passo:

- Os atributos FNúmero e DataIni são adicionados à relação "Departamento", definida no primeiro passo. Note que o atributo FNúmero é adicionado como um atributo não chave na relação "Departamento".

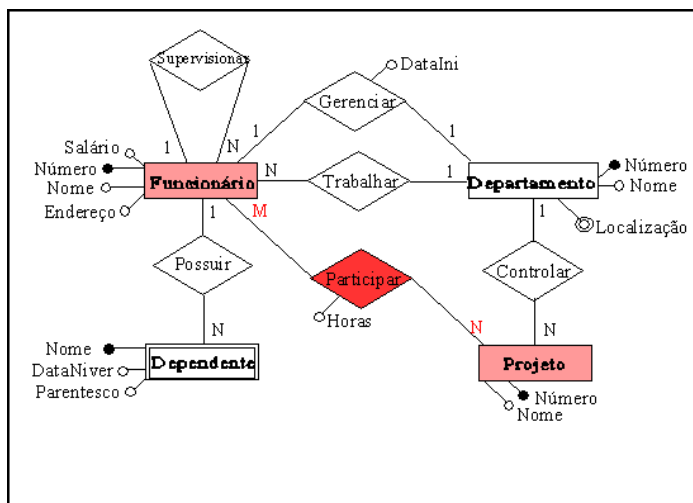
Quarto Passo:

- Neste passo são mapeados os relacionamentos: supervisionar, trabalhar e controlar.
- Avaliando o relacionamento **supervisionar**, nota-se que este relacionamento não possui atributos, logo, apenas o atributo **SuperNúmero** foi adicionado a relação "Funcionário", definida no primeiro passo.
- O mesmo acontece com o relacionamento **trabalhar**; apenas o atributo **DNum** foi adicionado a relação "Funcionário".
- No relacionamento **controlar**, que também não possui atributos, foi adicionado o atributo **DNro** na relação Projeto que já havia sido definida no primeiro passo.

Quinto Passo:

- O relacionamento **participar** que envolve os conjuntos de entidades "Funcionário" e "Projeto", é avaliado neste passo.
- Cria-se uma nova relação:

Participar = {FNum, PNum, horas}

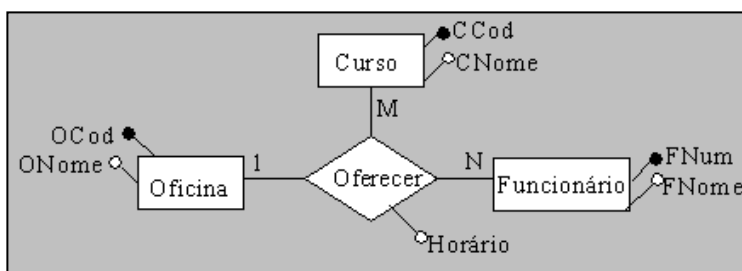


Sexto Passo: Mapear Conjuntos de Relacionamentos n-ário, n>2

Para conjuntos de relacionamentos n-ário, n>2 sempre considera-se que possuam cardinalidade **vários:vários:vários**.

Para cada conjunto de relacionamento (CR) será criada uma nova relação cujos atributos próprios são os do CR (se existirem) e cuja chave é formada pelos atributos chave das relações que mapeiam os conjuntos de entidades (CE's) envolvidos.

Os CR's de ordem maior que três são tratados da mesma maneira que os ternários. Seu mapeamento cria uma relação para cada CR e esta relação possui: os atributos do CR como atributos próprios e, como chave, os atributos concatenados de cada relação que mapeia os CE's envolvidos.



Como a modelagem utilizada para ilustrar os passos anteriores não tem relacionamento n-ário, com $n > 2$, escolheu-se um exemplo particular que será apresentado ao lado:

Mapeamento do conjunto de relacionamento ternário:

Oferecer = {OCod, CCod, FNum, Horário}

Passo 7: Mapear Atributos Multivalorados

Existem duas maneiras de mapear atributos multivalorados.

A primeira maneira não leva-se em conta conhecimento adicional sobre o atributo que está sendo mapeado. Para cada atributo multi-valorado cria-se uma nova relação que tem como chave os atributos chave da relação a qual pertencia juntamente com o atributo multi-valorado tomado como um atributo mono-valorado.

A segunda forma de mapear atributos multivalorados leva-se em conta o conhecimento adicional sobre o atributo que está sendo mapeado. Em alguns casos é possível determinar a quantidade de ocorrências de valores nos atributos. Quando isso acontece e essa quantidade é pequena, pode-se instanciar essa quantidade de atributos como mono-valorados na mesma relação que mapeia o conjunto de entidade ou conjunto de relacionamento ao qual o atributo multi-valorado está associado. Esta segunda forma de mapeamento impede a automação do procedimento de mapeamento do ME-RX para o modelo Relacional.

Primeiro Passo:

Funcionário = {FNúmero, FNome, Endereço, Salário, SuperNúmero, DNum}
Departamento = {DNúmero, Dnome, FNúmero, DataIni}

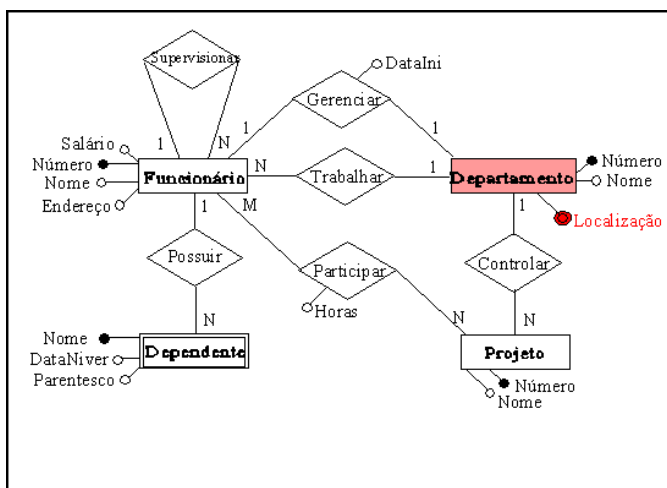
Projeto = {PNúmero, Pnome, DNro}

Segundo Passo:

Dependente = {DependNome, FNúmero, DataNiver, Parentesco}

Terceiro Passo:

- Os atributos FNúmero e DataIni são adicionados à relação "Departamento", definida no primeiro passo. Note que o atributo FNúmero é adicionado como um atributo não chave na relação "Departamento".



Quarto Passo:

- Neste passo são mapeados os relacionamentos: supervisionar, trabalhar e controlar.
- Avaliando o relacionamento **supervisionar**, nota-se que este relacionamento não possui atributos, logo, apenas o atributo SuperNúmero foi adicionado a relação "Funcionário", definida no primeiro passo.
- O mesmo acontece com o relacionamento **trabalhar**; apenas o atributo DNum foi adicionado a relação "Funcionário".
- No relacionamento **controlar**, que também não possui atributos, foi adicionado o atributo DNro na relação Projeto que já havia sido definida no primeiro passo.

Quinto Passo:

- O relacionamento **participar** que envolve os conjuntos de entidades "Funcionário" e "Projeto", é avaliado neste passo.
- Cria-se uma nova relação:

Participar = {FNum, PNum, horas}

Sexto Passo:

- Nesta modelagem não há relacionamento n-ário, $n > 2$

Sétimo Passo:

LocalDep = {DNúmero, Localização}

8. Normalização

O objetivo da normalização é evitar os problemas provocados por falhas no Projeto do Banco de Dados, bem como eliminar a **"mistura de assuntos"** e as correspondentes repetições desnecessárias de dados. Uma Regra de Ouro que devemos observar quando do Projeto de um Banco de Dados baseado no Modelo Relacional de dados é a de **"não Misturar assuntos em uma mesma Tabela"**. Por exemplo, na Tabela Clientes devemos colocar somente campos relacionados com o assunto Clientes. Não devemos misturar campos relacionados com outros assuntos, tais como Pedidos, Produtos, etc. Essa "Mistura de Assuntos" em uma mesma tabela acaba por gerar repetição desnecessária dos dados bem como inconsistência dos dados.

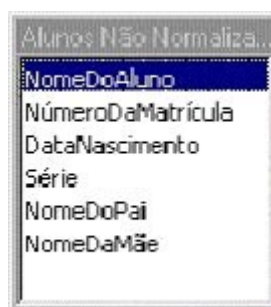
O Processo de Normalização aplica uma série de Regras sobre as Tabelas de um Banco de Dados, para verificar se estas estão corretamente projetadas. Embora existam 5 formas normais (ou regras de Normalização), na prática usamos um conjunto de 3 Formas Normais.

Normalmente após a aplicação das Regras de Normalização, algumas tabelas acabam sendo divididas em duas ou mais tabelas, o que no final gera um número maior de tabelas do que o originalmente existente. Este processo causa a simplificação dos atributos de uma tabela, colaborando significativamente para a estabilidade do modelo de dados, reduzindo-se consideravelmente as necessidades de manutenção. Vamos entender o Processo de Normalização na Prática, através de exemplos com tabelas criadas no Access.

8.1. Primeira Forma Normal:

"Uma Tabela está na Primeira Forma Normal quando seus atributos não contêm grupos de Repetição".

Por isso dissemos que uma Tabela que possui Grupos de Repetição não está na Primeira Forma Normal. Considere a estrutura da Tabela Indicada na Próxima Figura:



NomeDoAluno
NúmeroDaMatrícula
DataNascimento
Série
NomeDoPai
NomeDaMãe

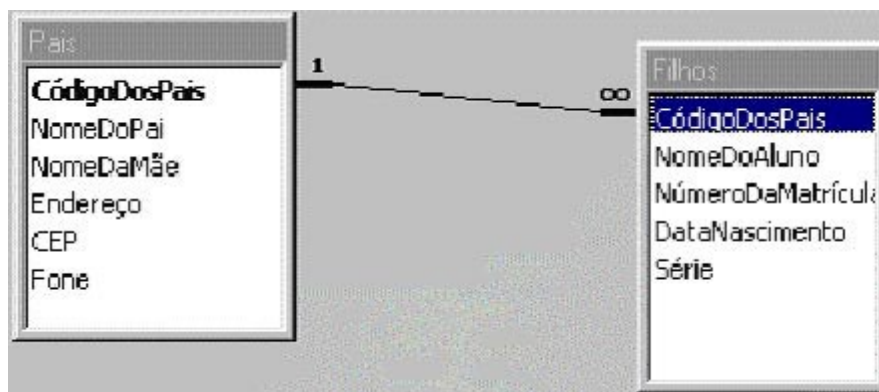
Tabela que não está na Primeira Forma Normal.

Uma tabela com esta estrutura apresentaria diversos problemas. Por exemplo, se um casal tiver mais de um filho, teremos que digitar o Nome do Pai e da Mãe diversas vezes, tantas quantos forem os filhos. Isso forma um Grupo de Repetição. Além do mais pode ser que por erro de digitação o Nome dos Pais não seja digitado exatamente igual todas as vezes, o que pode acarretar problemas na hora de fazer pesquisas ou emitir relatórios.

Este problema ocorre porque **"Misturamos Assuntos"** em uma mesma tabela. Colocamos as informações dos Pais e dos Filhos em uma mesma tabela. A resolução para este problema é simples: Criamos uma tabela separada para a Informação dos Pais e

Relacionamos a tabela Pais com a Tabela Filhos através de um relacionamento do tipo Um para Vários, ou seja, um casal de Pais pode ter Vários Filhos.

Observe na figura abaixo as duas tabelas: Pais e Filhos, já normalizadas.



Informações sobre Pais e Filhos em Tabelas Separadas.

As duas tabelas Resultantes da Aplicação da Primeira Forma Normal: Pais e Filhos estão na Primeira Forma Normal, a Tabela Original, a qual misturava informações de Pais e Filhos, não estava na Primeira forma Normal.

8.2. Segunda Forma Normal:

Ocorre quando a chave Primária é composta por mais de um campo. Neste caso, devemos observar se todos os campos que não fazem parte da chave dependem de todos os campos que compõem a chave. Se algum campo depender somente de parte da chave composta, então este campo deve pertencer a outra tabela. Observe o Exemplo Indicado na Tabela da Figura abaixo:

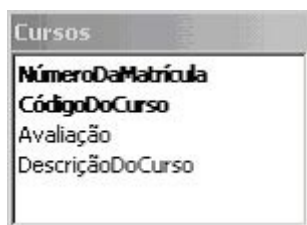


Tabela com uma Chave Primária Composta. Não está Na Segunda Forma Normal.

A Chave Primária Composta é formada pela combinação dos Campos "NúmeroDaMatrícula" e "CódigoDoCurso". O Campo Avaliação depende tanto do CódigoDoCurso quanto do NúmeroDaMatrícula, porém o campo DescriçãoDoCurso, depende apenas do CódigoDoCurso, ou seja, dado o código do curso é possível localizar a respectiva descrição, independentemente do NúmeroDaMatrícula. Com isso temos um campo que não faz parte da Chave Primária e depende apenas de um dos campos que compõem a chave Primária Composta, por isso que dizemos que esta tabela não está na Segunda Forma Normal.

A Resolução para este problema também é simples: "Dividimos a Tabela que não está na Segunda Forma Normal em duas outras tabelas, conforme indicado pela figura abaixo, sendo que as duas tabelas resultantes estão na Segunda Forma Normal.



Informações sobre Avaliações e Cursos em Tabelas Separadas.

OBS -> A Distinção entre a Segunda e a Terceira forma normal, que veremos logo em seguida, muitas vezes é confusa. A Segunda Forma normal está ligada a ocorrência de Chaves Primárias compostas.

8.3. Terceira Forma Normal:

Na definição dos campos de uma entidade podem ocorrer casos em que um campo não seja dependente diretamente da chave primária ou de parte dela, mas sim dependente de outro campo da tabela, campo este que não a Chave Primária. Quando isto ocorre, dizemos que a tabela não está na Terceira Forma Normal, conforme indicado pela tabela da figura abaixo:

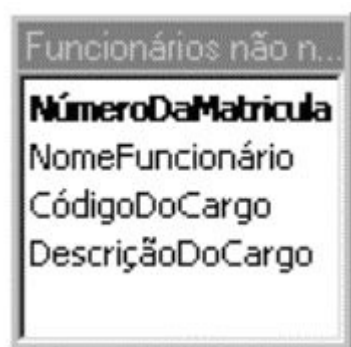
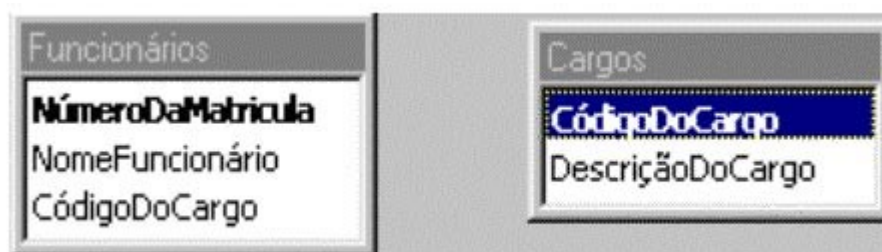


Tabela com um Campo dependente de Outro campo que não a Chave Primária. Não está na Terceira Forma Normal.

Observe que o Campo **DescriçãoDoCargo** depende apenas do Campo **CódigoDoCargo**, o qual não faz parte da Chave Primária. Por isso dizemos que esta tabela não está na terceira forma normal. A Solução deste problema também é simples. Novamente basta dividir a tabela em duas outras, conforme indicado pela figura a seguir. As duas tabelas resultantes estão na Terceira Forma Normal.



Tabelas Resultantes que estão na Terceira Forma Normal.

Com isso podemos concluir que como resultado do Processo de Normalização, iremos obter um número maior de tabelas, porém sem problemas de redundância e inconsistência dos dados.

9. Referência de Criação e Apoio ao Estudo

Projeto de Banco de Dados – Uma Visão Prática

Felipe Machado; Maurício Abreu – Editora Érica

Sistema de Banco de Dados

Abraham Silberschatz; Henry F. Korth; S. Sudarshan – Makron Books

Uma Introdução a Sistemas de Bancos de Dados

C. J. Date – Editora Edgard Blücher Ltda. (6a edição)

“Modelagem de Negócio”

Fulan José Davi – Editora Makron Books, 1997

www.juliobattisti.com.br/.../modelorelacional_p4.asp

<http://gbdi.icmc.sc.usp.br/documentacao/apostilas/me-r/>

Elaboração: Docente Maria Marli Milan Luqueta

Ver. 02-09

10. Anexos

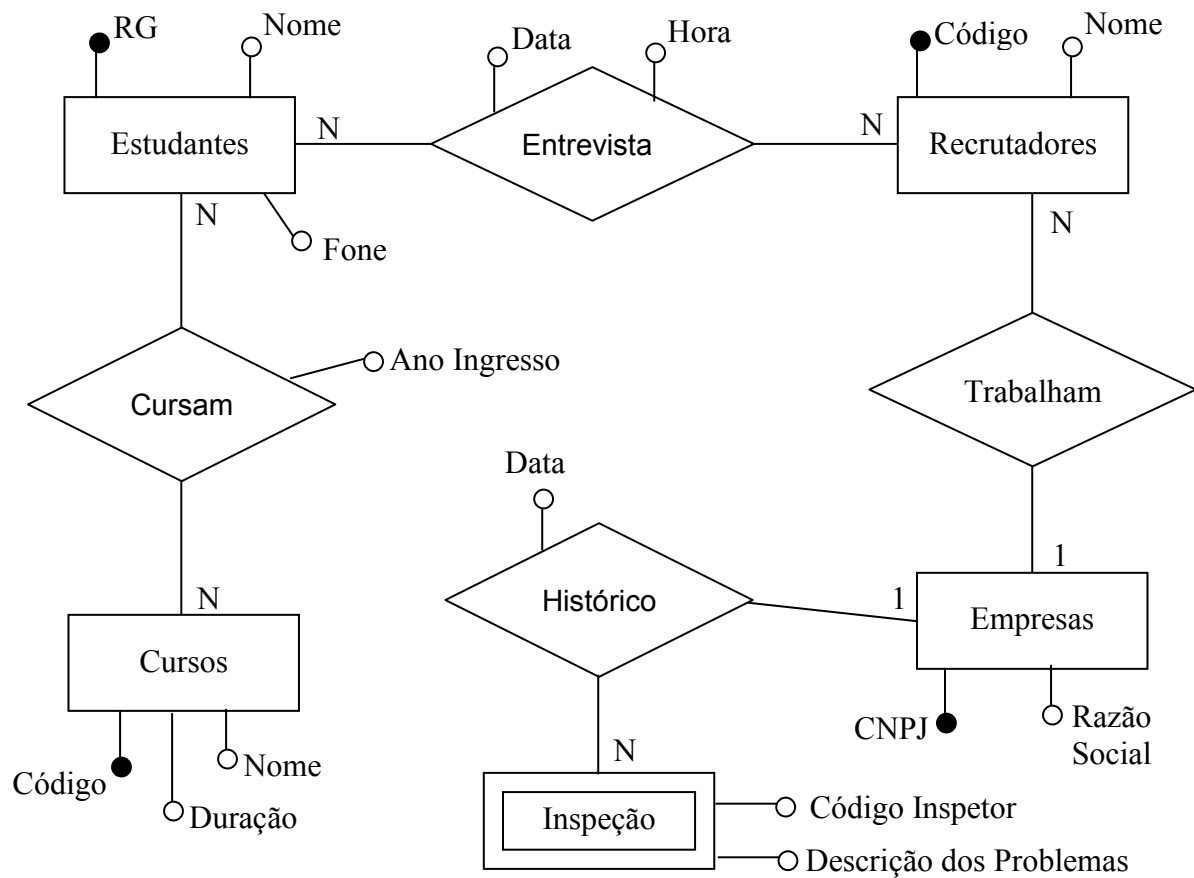
10.1. Anexo 1 – Exercícios DE-R

1. Faça o DE-R para uma locadora que mantém um cadastro de todos os seus clientes por código, nome, data nascimento, residência, telefone, RG. Esses clientes podem locar várias fitas. No momento da locação são anotados os seguintes dados data da retirada, data de entrega e o número do comprovante de alocação. As fitas são cadastradas por código do item, título e preço. A locação deve ser autorizada pelos funcionários que são cadastrados por nome e n.º do funcionário.
2. Faça um DE-R para uma empresa que é dividida em departamentos que são cadastrados por um número, nome e localização. Os funcionários da empresa são contratados para trabalhar nos departamentos, portanto um contrato associa cada funcionário ao departamento onde ele trabalha e possui a data de contratação. Os funcionários possuem um código, nome, endereço, cidade e fone. Dois grupos de funcionários são destacados: engenheiros e motoristas. Os engenheiros possuem com atributos adicionais o CREA e a formação; e os motoristas o número da carteira de habilitação e a data de vencimento da mesma. Os motoristas dirigem os carros da empresa entre determinadas datas/horas (data/hora de saída e data/hora de devolução). Os carros são cadastrados por chassi, marca, modelo, cor e ano. A cada carro é associado um histórico de reparos com a data, tipo do reparo, custo, garantia e oficina. Os engenheiros são alocados para trabalhar em determinados projetos em períodos determinados (data entrada e de saída do projeto). Os projetos são cadastrados por código, título, descrição e custo.
3. Faça um DER para o controle de pragas das fazendas de uma cooperativa agrícola. A cooperativa é composta por uma série de fazendas cadastradas por código, nome, tamanho e localização. As fazendas são associadas aos seus proprietários que possuem um código, nome, endereço e fone. Além disso, as fazendas são associadas aos produtos que nelas são cultivados em determinadas épocas (mês/ano). Os produtos possuem um código, nome, descrição e tempo de vida. Um produto cultivado em uma fazenda pode sofrer o ataque de uma praga, e isso será detectado em uma determinada data. As pragas são cadastradas por um código, nome popular, nome científico e tempo de vida. Os ataques de pragas serão combatidos por um defensivo em certa data e com certo número de aplicações. Os defensivos possuem um código, nome e podem ser de 2 categorias: naturais (biológicos) ou químicos. Os defensivos químicos possuem um volume, prazo de validade, prazo de contaminação, descrição dos componentes e efeitos colaterais; e os biológicos possuem o nome científico do agente biológico.
4. Faça um DE-R para resolver o problema de uma imobiliária que deseja manter o controle de condomínios pelo quais são responsáveis. Para isso, temos as seguintes informações: Um condomínio é formado por diversos apartamentos. Cada apartamento possui uma única garagem. Existem os síndicos dois responsáveis por cada condomínio. Deseja-se manter os dados dos proprietários de cada apartamento (um apartamento pode ser de propriedade de mais de uma pessoa). Pode haver casos de um proprietário possuir mais de um apartamento. A imobiliária deve também manter o cadastro dos herdeiros de cada proprietário. Os dados a serem cadastrados são: *Condomínio* = Código, Nome e Endereço. *Apartamento* = Número, Tipo. *Proprietário* = RG, Nome, Telefone, E-Mail. *Herdeiro*: RG, Nome, Telefone. *Garagem* = Número, Tipo. *Síndico* = Matrícula, Nome, Endereço, Telefone

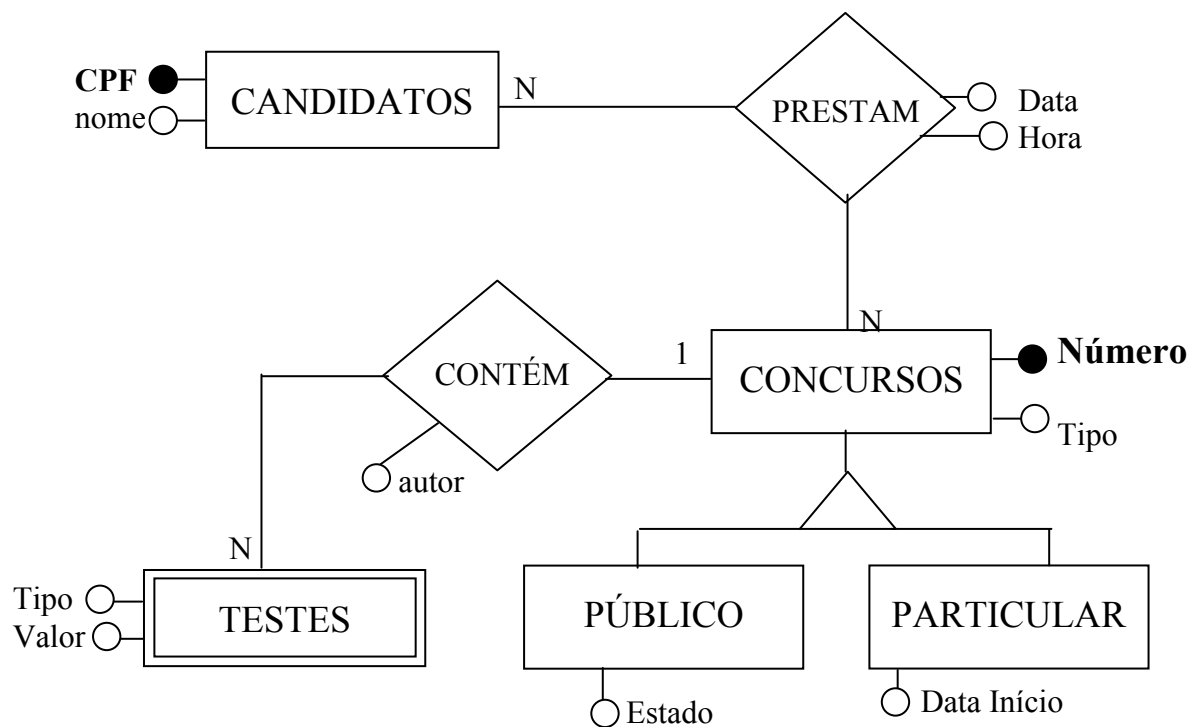
10.2. Anexo 2 – Exercícios MR

- Passe para o Modelo Relacional as situações a seguir:

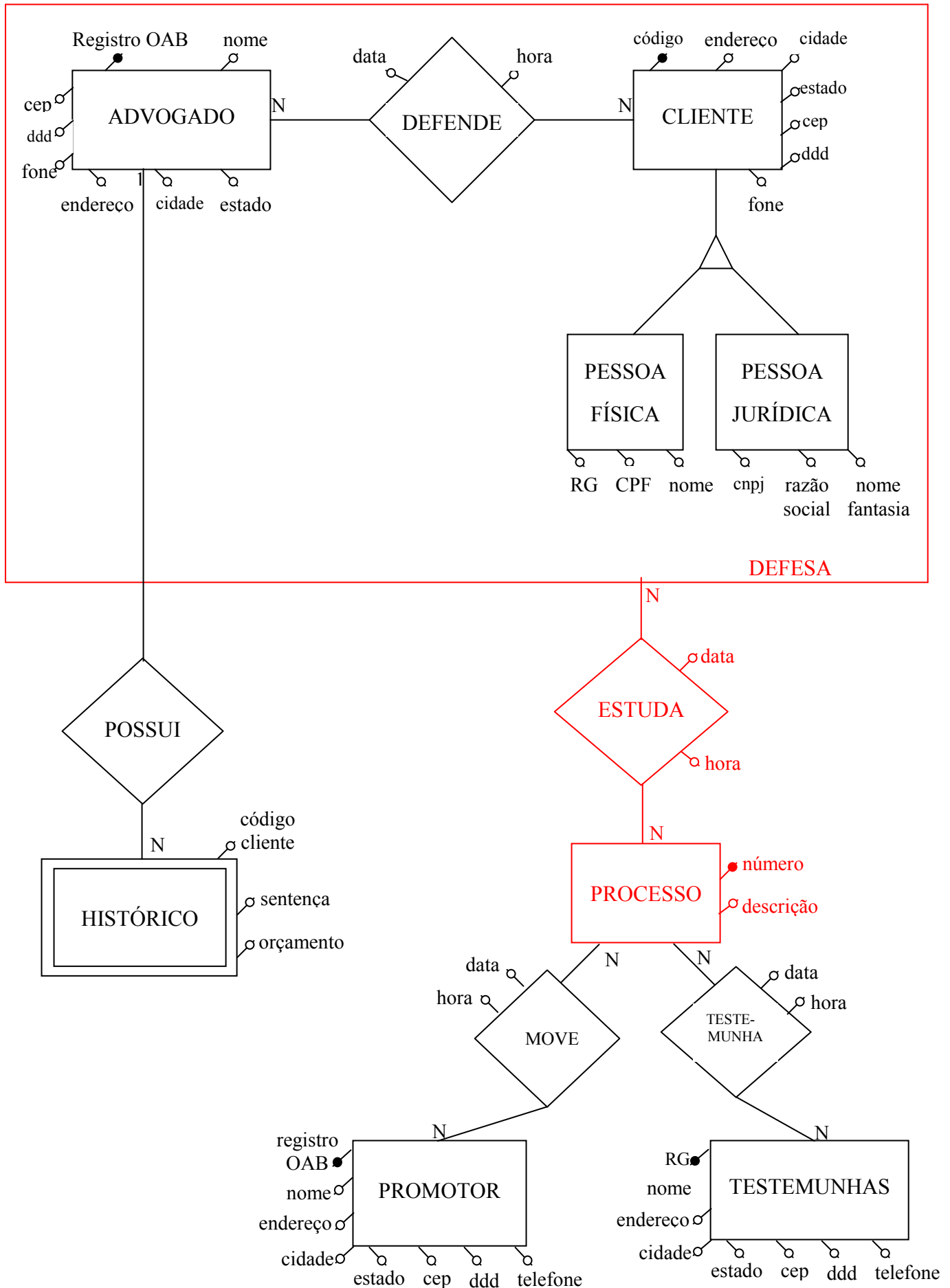
1.



2.



3.



10.3. Anexo 3 – Exercícios Normalização

1º) Normalizar o seguinte conjunto de atributos:

Aluno (cod_aluno, nome, naturalidade, ano_nascimento, cod_curso, curso, diretor, ano_inscrição, cod_disciplina, disciplina, nota).

2º) Normalizar o seguinte conjunto de atributos:

Encomenda (num_encomenda, num_cliente, cliente, endereço_cliente, data_encomenda, cod_produto, produto, quantidade).

3º) Normalizar o seguinte conjunto de atributos:

Apólice (num_ap, data, valor, estado, matricula, cod_cliente, nome, localidade, cod_tipo, tipo, num_pag, data_pag, valor_pag, marca_viatura)

estado = {ativo, inativo}

tipo = {terceiros, todos os riscos,...}

10.4. Anexo 4 – Listas de Exercícios

Lista de Exercícios (a ser entregue)

- 1) Defina Banco de Dados? Qual é o seu objetivo?
- 2) Quanto à estrutura, quais são os três níveis de abstração existentes e o que esses níveis descrevem?
- 3) Quais são os problemas dos arquivos convencionais?
- 4) O que é um SGBD? Quais são as suas funções?
- 5) Qual é a classificação dos SGBD?
- 6) Quais são as etapas de um projeto de BD?
- 7) Qual o objetivo da modelagem de dados? Quais os tipos de modelos existentes?
- 8) Para que serve o mapeamento de dados?
- 9) O que é ME-R?
- 10) O que é Entidade? Dê exemplos?
- 11) Defina atributo? Dê exemplos?
- 12) Para que serve os relacionamentos no ME-R? Quais são os graus (cardinalidades) existentes?
- 13) O que é DE-R? Quais são os símbolos utilizados e o que eles representam?
- 14) Quais são as etapas para se modelar um sistema?
- 15) Faça um diagrama entidade-relacionamento para uma Transportadora, que possui uma frota de caminhões cadastrados por chassi, modelo, cor e capacidade de carga. Os caminhões são dirigidos por motoristas da transportadora. (obs.: um motorista dirige apenas um caminhão). Os caminhões prestam serviços a várias empresas que são cadastradas por código, nome, endereço. As empresas possuem várias linhas de produtos a serem transportados. Os produtos são identificados por um código, tipo, descrição e destino. A transportadora possui diversos motoristas com nº de registro, nº da carteira de motorista e data vencimento. Depois de pronto mapeie para o modelo relacional.
- 16) Faça um DER para o controle de pragas das fazendas de uma cooperativa agrícola.
A cooperativa é composta por uma série de fazendas cadastradas por código, nome, tamanho e localização. As fazendas são associadas aos seus proprietários que possuem um código, nome, endereço e fone. Além disso, as fazendas são associadas aos produtos que nelas são cultivados em determinadas épocas (mês/ano). Os produtos possuem um código, nome, descrição e tempo de vida.
Um produto cultivado em uma fazenda pode sofrer o ataque de uma praga, e isso será detectado em uma determinada data. As pragas são cadastradas por um código, nome popular, nome científico e tempo de vida.
Os ataques de pragas serão combatidos por um defensivo em uma certa data e com um certo número de aplicações. Os defensivos possuem um código, nome e podem ser de 2 categorias: naturais (biológicos) ou químicos. Os defensivos químicos possuem um volume, prazo de validade, prazo de contaminação, descrição dos componentes e efeitos colaterais; e os biológicos possuem o nome científico do agente biológico. Depois de pronto mapeie para o modelo relacional.