



Instituto de Ensino Santo Antonio

# Apostila de Modelagem de Dados

Curso: Técnico de Informática

Ano: 2ª Série

Disciplina: A.S.M.D.

Versão 2007

Elaborado por: Prof. Fernando Salles Claro

# Sumário

<b>CAPÍTULO 1 – MODELAGEM DE DADOS .....</b>	<b>1</b>
INTRODUÇÃO.....	1
<i>Sistemas Gerenciadores de Bancos de Dados .....</i>	<i>1</i>
MODELOS DE BANCOS DE DADOS .....	1
<i>Modelo Hierárquico.....</i>	<i>1</i>
<i>Modelo de Rede .....</i>	<i>2</i>
<i>Modelo Relacional.....</i>	<i>3</i>
<b>CAPÍTULO 2 - MODELO ENTIDADE-RELACIONAMENTO .....</b>	<b>5</b>
<i>Modelo Relacional e Modelos Semânticos de Dados .....</i>	<i>5</i>
ENTIDADES.....	5
<i>Concepção de uma entidade .....</i>	<i>7</i>
RELACIONAMENTOS .....	7
<i>Grau e Classe de um relacionamento.....</i>	<i>8</i>
<i>Algumas Extensões ao Modelo ER.....</i>	<i>10</i>
Quando a cardinalidade não é exatamente 1.....	10
Agregação .....	11
Generalização/Especialização .....	12
UTILIZAÇÃO DO MÉTODO ENTIDADE-RELACIONAMENTO.....	13
VANTAGENS DO MÉTODO ENTIDADE – RELACIONAMENTO .....	13
<b>CAPÍTULO 2 – NORMALIZAÇÃO .....</b>	<b>15</b>
INTRODUÇÃO.....	15
PRIMEIRA FORMA NORMAL (1FN): ELIMINAÇÃO DE DOMÍNIOS MULTIVALORADOS.....	15
SEGUNDA FORMA NORMAL (2FN): DEPENDÊNCIA DO ATRIBUTO DETERMINANTE .....	18
TERCEIRA FORMA NORMAL (3FN): ELIMINAÇÃO DA DEPENDÊNCIA FUNCIONAL TRANSITIVA .....	19
FORMA NORMAL DE BOYCE-COOD (FNBC) .....	20
<b>EXERCÍCIOS - MODELAGEM DE BANCO DE DADOS.....</b>	<b>21</b>

## **Capítulo 1 – Modelagem de dados**

### **Introdução**

Bancos de Dados (BD) é uma área da computação que apresentou grande desenvolvimento nas décadas de 70 e 80 e continua em ritmo acelerado de pesquisa e desenvolvimento. Com as bases teóricas da tecnologia relacional lançadas na década de 70 e o lançamento comercial de muitos Sistemas Gerenciadores de Banco de Dados (SGBDs) relacionais na década de 80, o gerenciamento de grandes volumes de dados pode ser feito cada vez mais de forma segura, eficiente e barata.

### **Sistemas Gerenciadores de Bancos de Dados**

SGBD – Sistema Gerenciador de Banco de Dados (ou DBMS – Database Management System) é um sistema utilizado para gerenciar dados que estão armazenados de forma organizada, permitindo incluir, alterar, excluir, consultar e manipular dados.

O tratamento da informação em um banco de dados oferece vantagens:

- A disponibilidade dos dados para exclusão, consulta ou alteração pode ser autorizada pelo Administrador de Banco de Dados (DBA – Data Base Administrator) no projeto, antes das aplicações serem desenvolvidas, permitindo com isso a segurança e a privacidade dos dados;
- A padronização é reforçada. Os tipos de dados mais importantes são atributos que têm seus nomes e tamanhos definidos já no projeto do banco de dados.
- Os dados dizem respeito a um ambiente, empresa ou corporação e estão disponíveis independentemente da aplicação.
- Um projeto de banco de dados coerente reduz ou elimina a redundância, favorecendo a manutenção da integridade dos dados.

A arquitetura de um SGBD é estabelecida a partir de um modelo de dados, que é uma forma de representação que resulta de uma abstração. O projeto de um banco de dados envolve o desenvolvimento de um modelo formal relativo à estruturação dos dados de todo um empreendimento.

Os SGBDs existentes no mercado geralmente adotam o modelo hierárquico, de rede ou relacional. Dentre os três, o mais utilizado é o modelo de rede.

O objetivo de todos os modelos de dados é modelar a realidade tão proximamente quanto possível.

### **Modelos de Bancos de Dados**

#### **Modelo Hierárquico**

Os SGBD's baseados no modelo hierárquico foram os primeiros sistemas a estarem disponíveis comercialmente.

No modelo hierárquico, o usuário percebe o banco de dados como uma estrutura de árvores que envolvem registros e ligações. Cada registro pode possuir um número qualquer de descendentes, mas apenas um ascendente (exceto a raiz, que não possui ascendente). O registro ascendente guarda referências do conjunto de descendentes que possui.

A navegação dentro de um banco de dados hierárquico é feita com comandos do tipo “acessar primeiro” e “acessar próximo” ( get first e get next), admitindo cláusulas e predicados para pesquisa.

A figura abaixo ilustra a representação de alguns registros organizados hierarquicamente para uma locadora de veículos. Como você pode ver, as árvores hierárquicas são plantadas de cabeça para baixo.

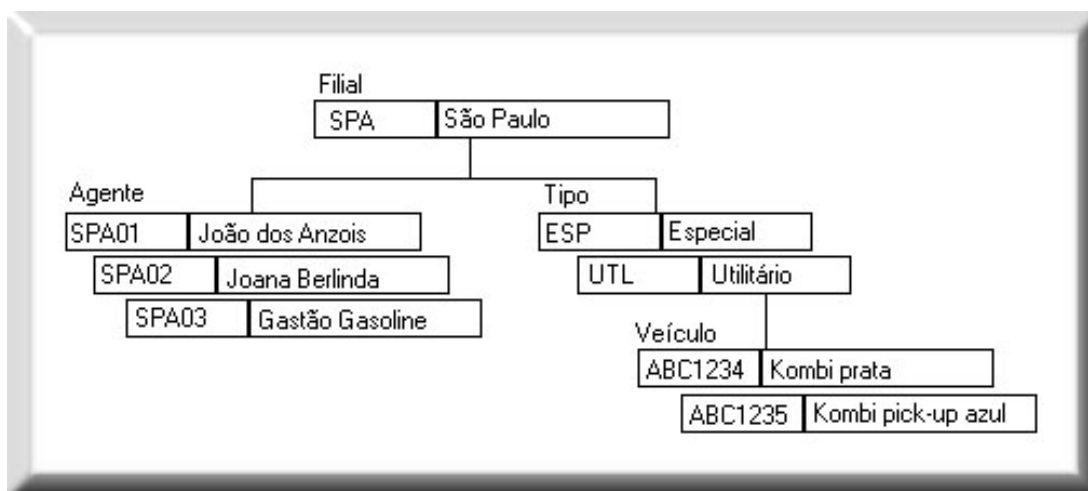


Fig. 1 - Exemplo de um modelo hierárquico

Dada a limitação de que cada registro pode possuir somente um ascendente, seu conteúdo pode ter que ser repetido várias vezes. Se for criada a filial Manaus, por exemplo, que também tem um tipo de veículo esporte, este registro do tipo esporte será diferente do registro já existente para a filial São Paulo.

Esta repetição é uma grande desvantagem: o espaço consumido para armazenagem é desperdiçado com a mesma informação. Além disso, se a atualização de um valor não for feita para todos os registros repetidos, o banco de dados fica inconsistente. Como resultado, a implementação de um banco de dados em um sistema hierárquico pode requerer uma boa quantidade de trabalho para contornar as limitações.

O modelo hierárquico é um caso particular do modelo de rede. É o mais restrito entre os três vistos aqui.

### Modelo de Rede

No modelo de rede, a visão do usuário é a de um grafo ou uma malha de ligações um-para-muitos entre os registros. Um tipo de registro pode estar envolvido em mais de um relacionamento, pode ter vários ascendentes e vários descendentes

O grupo CODASYL (que desenvolveu o Cobol), através do Database Task Group (DBTG) desenvolveu um modelo de rede que é o mais difundido e aceito. O modelo DBTG criou a figura do grupo ou conjunto, uma representação para os relacionamentos. Cada conjunto tem um tipo de registro **proprietário**, que só pode participar de uma ocorrência do conjunto, e um tipo de registro **membro**, que pode participar em qualquer quantidade de ocorrências

A figura a seguir apresenta registros ligados de acordo com uma estrutura de grafo, própria do modelo rede:

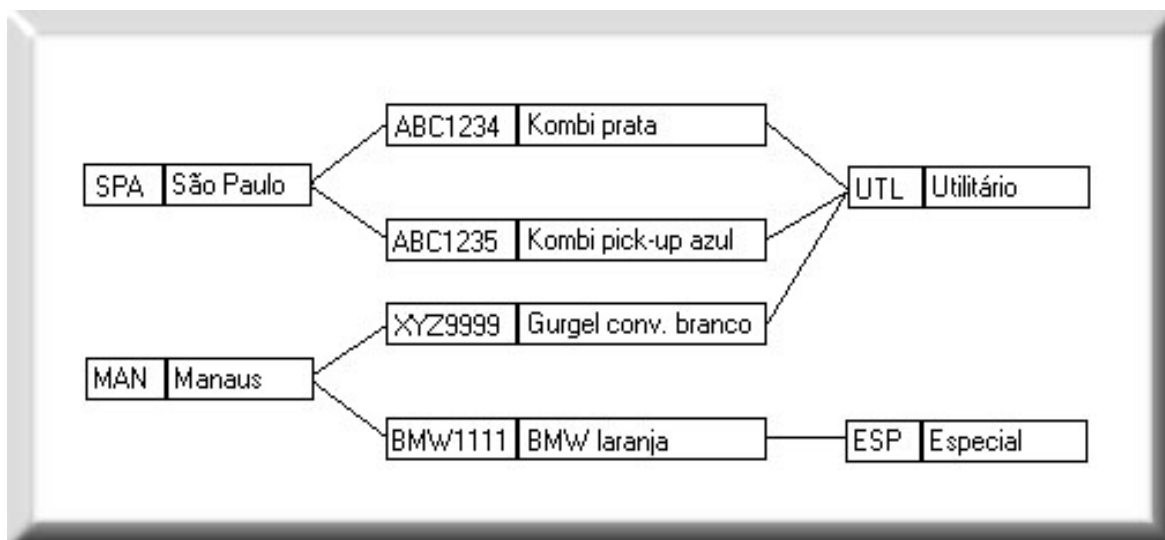


Fig. 2 - Exemplo de um modelo rede

Na figura acima, há um relacionamento filial-veículo, onde o veículo de placas ABC1234 é proprietário e a filial de código SPA é um registro membro do conjunto.

### Modelo Relacional

Um banco de dados relacional é visto pelo usuário como um conjunto de tabelas. Uma tabela ou relação é formada por linhas conhecidas como *t-uplas* (lê-se “tuplas”) e colunas. Cada coluna tem um conjunto de valores possíveis chamado **domínio**. Em linguagem de computação, tabela ou relação equivale a arquivo, *t-upla* equivale a registro e coluna equivale a campo. A representação por um conjunto de tabelas estabelece a diferença em relação ao modelo rede, que é um conjunto de registros e relacionamentos através de ligações.

Portanto, definimos um **sistema relacional** como aquele onde os dados são notados pelos usuários como tabelas e as operações aplicáveis ao sistema geram tabelas partindo da primeira.

Na figura abaixo, podemos observar um exemplo de modelo relacional:



Fig. 3 - Exemplo de um modelo relacional

A teoria sobre o modelo relacional apresenta dois formalismos para a manipulação de banco de dados: a **álgebra relacional** e o **cálculo relacional**, que serviram de base para a construção de todas as ferramentas relacionais hoje disponíveis.

A álgebra relacional é formada por operações cujos nomes vêm da teoria de conjuntos, e outras operações relacionais que foram especialmente criadas:

- **União** : Produz uma tabela resultante da união de outras tabelas operadas;
- **Interseção** : Cria uma tabela, resultado da interseção de outras tabelas

operadas;

- **Diferença** : Cria uma tabela contendo t-uplas que pertencem a primeira tabela operada, mas não à segunda;
- **Produto Cartesiano** : Gera todas as combinações possíveis entre as tuplas de duas tabelas.
- **Projeção** : Cria uma tabela contendo alguns atributos específicos da tabela operada;
- **Seleção ou Restrição** : Serve para extrair t-uplas de uma certa tabela;
- **Junção** : Gera uma tabela que é a combinação das tabelas operadas segundo critérios impostos sobre atributos de uma e outra tabela;
- **Divisão** : Opera duas tabelas, criando uma terceira com os atributos da primeira tabela, cujos atributos que os acompanham existem também na segunda tabela.

O cálculo relacional tem poder equivalente à álgebra e dá como resultado relações (tabelas), da mesma forma. A diferença está na forma de expressão: a álgebra especifica as operações a fazer sobre as tabelas, enquanto o cálculo relacional apenas nomeia a informação desejada e um certo predicado ou critério para os valores a serem pesquisados, sem dizer quais as operações a fazer.

Dentre os sistemas relacionais, os **System/R** e o **Ingres** foram os primeiros a serem comercializados. Dentre os sistemas relacionais atuais mais utilizados no Brasil, temos:

- MS-Access
- DB2
- Informix
- Oracle
- Progress
- Sybase
- TSGDB
- Unify
- Zim
- MS-SQL Server
- PostgreSQL
- MySQL
- Interbase / Firebird
- entre outros...

## Capítulo 2 - Modelo Entidade-Relacionamento

### Modelo Relacional e Modelos Semânticos de Dados

Um banco de dados é um acervo de informações armazenadas segundo certos critérios de organização.

O modelo relacional de dados é uma maneira de conceber a organização de um banco de dados como uma coleção de tabelas e associações entre outras tabelas. Por exemplo, veja a figura abaixo: cada tabela é armazenada em um arquivo, cada linha é um registro e tem um conteúdo único na tabela. Cada coluna é um campo, com um nome que representa o tipo de dados contido na coluna.

Tabela DEPTO

Id_depto	Nome_depto
4	Compras
9	Tele vendas
3	Manutenção

Tabela FUNCIONÁRIO

Id_func	Nome_func	Data_nasc	Sexo	Id_depto
39	Maria de Bem	10/01/61	F	4
14	João dos Anzóis	22/11/55	M	3
76	Johnson Hooks	11/03/77	M	9
90	Carmem Bizet	02/02/67	F	4
91	Antenor Ópera	12/12/59	M	4

Tabela 1 – Exemplo de Tabelas

No exemplo, as tabelas têm uma coluna em comum: **id\_depto** e um atributo que cumpre o papel em **DEPTO**, de identificador de cada registro e, em **FUNCIONÁRIOS**, de apontador que permite chegar ao nome do departamento.

Projetar tabelas diretamente a partir da simples observação da realidade do ambiente nem sempre é fácil e pode não ser o modo mais eficiente de projetar.

Os modelos semânticos de dados surgiram com o objetivo de agregar **sentido** à descrição de um banco de dados. A abordagem semântica mais conhecida e adotada é a do modelo Entidade-Relacionamento (ER), publicado em artigo de Peter Pin-Shan Chen, baseado na teoria relacional e teoria dos conjuntos.

O ER é um modelo de dados que se propôs a englobar algumas propostas de modelagem anteriores, como o modelo relacional e o de rede, e dar uma visão mais abstrata e de mais alto nível a um ambiente de banco de dados.

### Entidades

**Entidade** é um conjunto de objetos de mesma natureza, com as mesmas características ou atributos, abrigados sob um nome genérico. Em um banco de dados de uma universidade, por

exemplo, são entidade: *Aluno*, *Professor*, *Curso*, *Departamento*, *Disciplina*, etc., pois cada uma destas entidades representa uma coleção de objetos com os mesmos tipos de atributos.

Chamamos *ocorrência* de uma entidade à um objeto que pertence entidade. No exemplo acima, *Ciências da Computação* poderia ser uma ocorrência da entidade *Curso* e *Bancos de Dados* uma ocorrência de *Disciplina*.

Cada **ocorrência** de entidade apresentar-se-á como uma coleção de elementos de dados ou *atributos*, por exemplo:

Entidade	Atributos
ALUNO	Nome Número de matrícula Endereço
PROFESSOR	Nome Código Data de admissão Salário

Tabela 2 – Entidades e seus atributos

**Atributo determinante** é aquele que identifica uma ocorrência da entidade. Por exemplo: *Número de matrícula*, na entidade *ALUNO*.

Observe, no exemplo acima, que os atributos *Nome* aparecem tanto para a entidade *ALUNO* quanto para *PROFESSOR*, mas são coisas diferentes, com significados diferentes.

Nos **diagramas de entidades e relacionamento** as entidades são representadas por retângulos com seu nome inscrito. Os elementos de dados, quando representados, apresentam-se ligados a entidade por um traço de detalhe.

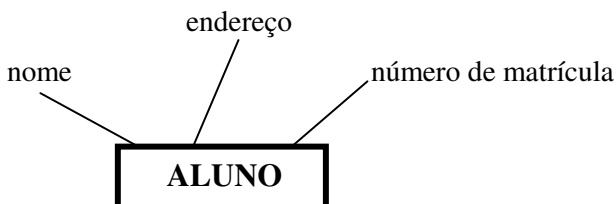


Fig. 4 - Exemplo de uma entidade com seus atributos

A figura 4 mostra uma representação de uma entidade e seus atributos.

O modelo original de Chen especifica um tipo especial de entidade cujas *ocorrências têm sua existência dependente da existência de ocorrências de outra entidade*. É a **entidade fraca**, representada no diagrama ER como um retângulo de linhas duplas. Na figura 5, *Dependente* é uma entidade fraca, pois sua existência só tem sentido em função da *entidade regular Funcionário*.



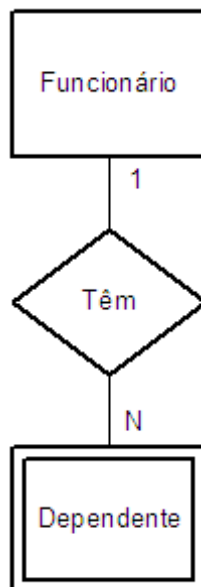


Fig. 5 – Representação de uma entidade fraca

Na entidade fraca o que interessa é a identidade da entidade regular. O termo *entidade regular* é apenas para diferenciá-la da entidade fraca. Quando não houver entidade fraca em jogo, é chamado apenas de entidade.

### Concepção de uma entidade

A definição sobre o que deve ou não ser entidade depende do ambiente para o qual se projeta o banco de dados.

Vamos examinar, por exemplo, *chassi*. Para uma fábrica de chassis, certamente haverá uma entidade *Chassi*, com elementos de dados dando as características de cada ocorrência ou modelo de chassi. Agora, no banco de dados de uma montadora de veículos, *chassi* pode ser uma ocorrência da entidade *Parte* (Item, Peça ou que nome possua), relacionando-se com outras ocorrências da entidade para compor produtos ou subprodutos. Por fim, para o banco de dados do Cadastro Nacional de Veículos Roubados, *chassi* poderá ser um elemento de dados da entidade *Veículo* correspondente ao número de série do chassi (neste caso, a única informação sobre chassi que interessa).

Portanto, conceber uma entidade é uma tarefa cuja decisão é relativa, depende do âmbito do projeto. Uma boa garantia de que uma entidade foi bem projetada é ter a capacidade de lhe atribuir um nome que represente o que é cada uma de suas ocorrências e identificar o atributo determinante que lhe confere identidade.

### Relacionamentos

**Relacionamento** é uma associação entre entidades. Pode haver relacionamento envolvendo uma (auto-relacionamento) ou mais entidades. Nos diagramas ER, estes últimos são representados por losangos com o nome rotulado, como mostra a figura 6:

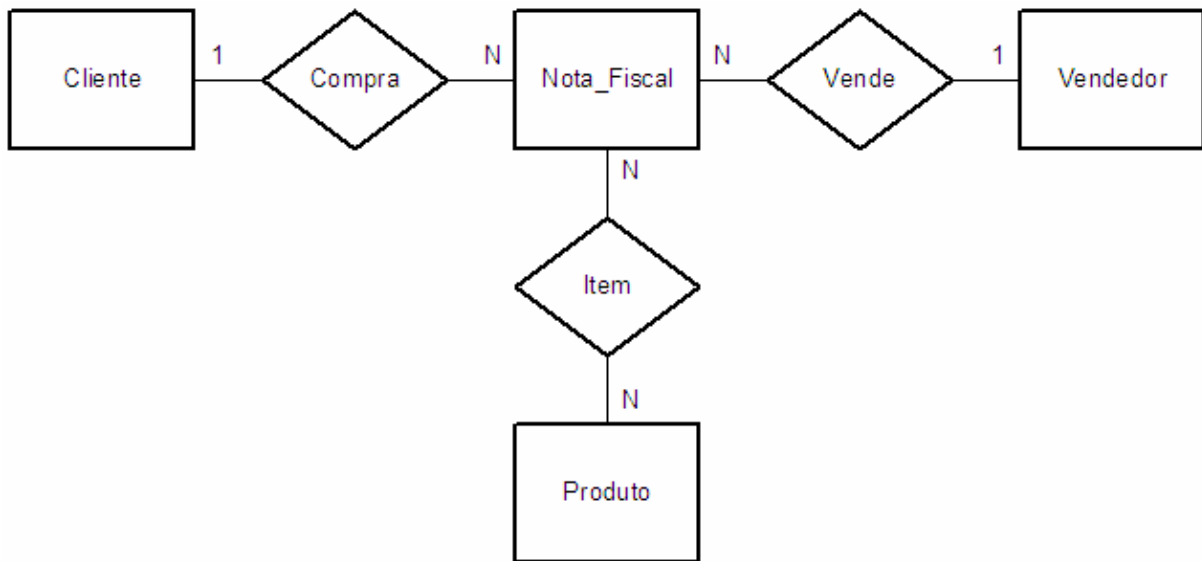


Fig. 6 – Exemplo de um DER

Entre o mesmo conjunto de entidades pode haver mais de um relacionamento. Por exemplo, as entidades *Funcionário* e *Departamento*, na figura 7, estão associadas por dois relacionamentos: *Gerência* e *Lotação*.

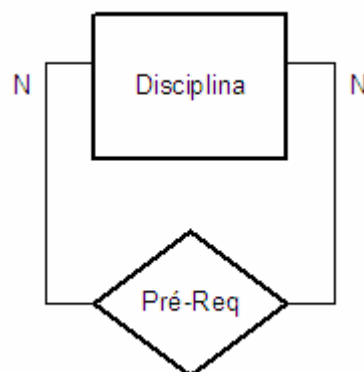
Alguns relacionamentos possuem atributos próprios que não pertencem isoladamente a nenhuma das entidades envolvidas. Por exemplo, o relacionamento *Gerência*, na figura 9, entre *Departamento* e *Funcionário*, poderá requerer a *data de passe* e *volume de vendas no início da gestão* como atributos próprios do relacionamento.

### Grau e Classe de um relacionamento

Os relacionamentos são classificados segundo **grau** e **classe**.

O **grau** indica o número de entidades que participam da associação.

#### GRAU 1:



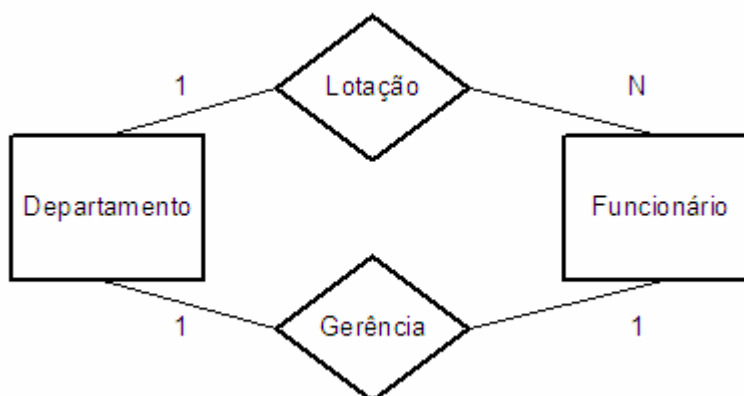
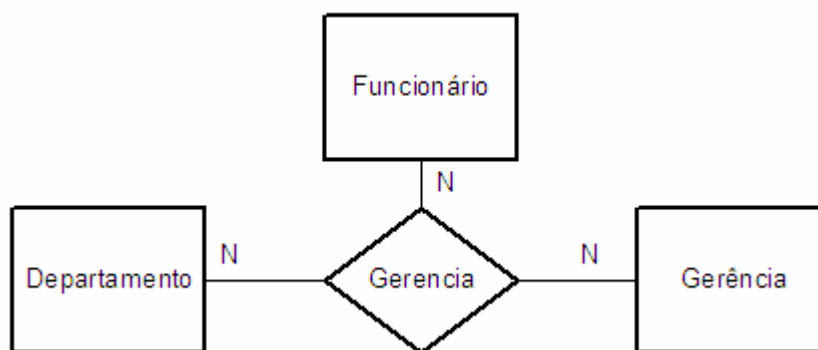
**GRAU 2:****GRAU 3:**

Fig. 7 – Relacionamentos de graus diversos

A **classe** de um relacionamento dá conta de quantas ocorrências de cada entidade podem estar envolvidas no relacionamento. É representada no diagrama ER sobre a linha e chamada entre a entidade e o relacionamento.

Assim, no exemplo da figura 7, *Lotação* é um relacionamento 1:N, porque cada *Departamento* pode ter N (vários) *Funcionários* lotados, mas um *Funcionário* está lotado em apenas 1 *Departamento*, *CompCC* (composição de centro de custos) é um relacionamento 1:N porque um centro de custos pode ser componente de 1 centro de custos e pode ser composto por N (vários) centros de custo.

Usaremos o exemplo de relacionamento de grau 3 da figura 7 para enunciar a maneira de estabelecer a classe de um relacionamento. O relacionamento *Estoque*, que seria mais apropriadamente chamado *Item de Estoque*, é uma construção bastante singular, existente apenas em algumas lojas de departamentos com várias filiais. Cada filial tem a liberdade de distribuir os produtos nos departamentos como melhor aprouver à sua gerência. O relacionamento *Estoque* tem a função de representar os itens de estoque, com os atributos: *quantidade em estoque* e *preço*.

Para estabelecer o indicador 1 ou N junto a cada uma das entidades participantes, devemos fazer a seguinte pergunta: “Quantas ocorrências desta entidade podem estar associadas às demais no relacionamento em questão?”.

Usando o exemplo da figura 7:

- Quantos produtos podem estar associados a um departamento e a uma filial através do relacionamento *Estoque*? Vários (N).
- Quantos departamentos podem estar associados a um produto e a uma filial através do relacionamento *Estoque*? Apenas um(1).
- Quantas filiais podem estar associados a um departamento e a um produto através do relacionamento *Estoque*? Várias (N)

Podemos ainda exemplificar a pergunta com ocorrências, conforme a figura 7.

<u>Filial</u>	<u>Departamento</u>	<u>Produto</u>
Itajaí	Esportes	1 ou N?
Florianópolis	Cama-mesa-banho	1 ou N?
Itajaí	1 ou N?	Toalha mod. 1234
Itajaí	1 ou N?	Bota Sete Léguas 41 preta
1 ou N?	Calçados	Bota Sete Léguas 41 preta
1 ou N?	Esportes	Parafina 250 g

Tabela 3 - Estratégia para determinação da classe de um relacionamento

### Algumas Extensões ao Modelo ER

O modelo ER vem recebendo, desde a publicação do artigo de Chen em 1976, inúmeras contribuições e extensões. Novos modelos vêm sendo propostos, também para alcançar o objetivo de qualquer modelo de dados modelar a realidade tão proximamente quanto possível.

#### *Quando a cardinalidade não é exatamente 1*

O modelo ER original prevê o estabelecimento da classe de um relacionamento com os símbolos 1 ou N (vários). Porém, em algumas situações, isto não é suficiente para expressar a realidade.

Veja o relacionamento da figura 8, de um ambiente de banco de dados de uma imobiliária que administra contratos de aluguel. Existe a entidade *Imóvel*, associada a *Inquilino* através do relacionamento *Aluga*. Um inquilino pode alugar quantos imóveis quiser, e um imóvel pode ser alugado por um inquilino.

Ou, o imóvel pode estar desocupado, momentaneamente. O projeto do relacionamento como 1 N obriga a que cada imóvel tenha um inquilino. Este fato tem implicações na implementação do banco de dados. Vamos permitir, então, que a classe do relacionamento seja expressa como 0.1:N, quando o lado 1 não for obrigatório.



Fig. 8 – Relacionamento 0,1:N

### Agregação

Uma limitação do modelo ER é que os relacionamentos acontecem entre entidades. Acontece, porém, que às vezes é preciso construir um relacionamento envolvendo algum relacionamento na associação. Aí precisamos utilizar um novo conceito o de *agregação*.

Vejamos o exemplo de um projeto de banco de dados em evolução, conforme a figura 9, associamos *Cliente* a *Vendedor* através do relacionamento *Venda*. Uma venda, no entanto, costuma estar associada a vários produtos. Seria necessário, então, associar a *Venda*, através de um relacionamento *Item* de venda, uma série de ocorrências da entidade *Produto*.

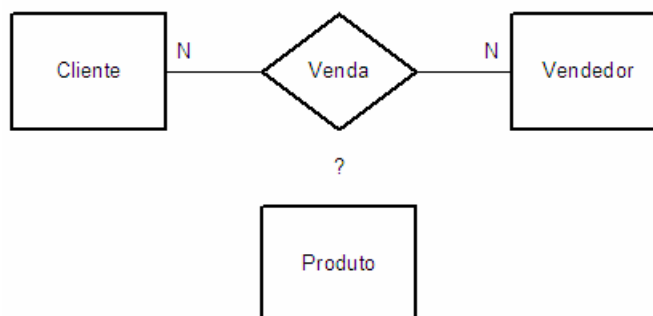


Fig. 9 – Como associar uma entidade ? em relacionamento ?

A solução é usar agregação, que é a maneira de dar a um relacionamento o caráter de entidade.

A figura 10 mostra a representação clássica para agregação, uma representação alternativa e a representação usando apenas os elementos do modelo ER original.

#### Representação Clássica

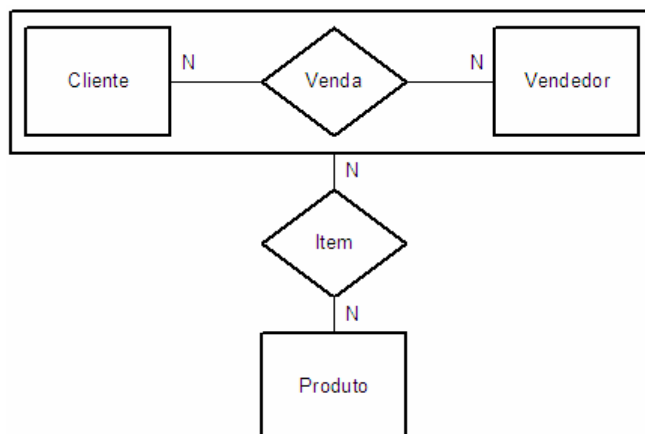


Fig. 10 – Exemplo de agregação – representação clássica

### Representação Alternativa

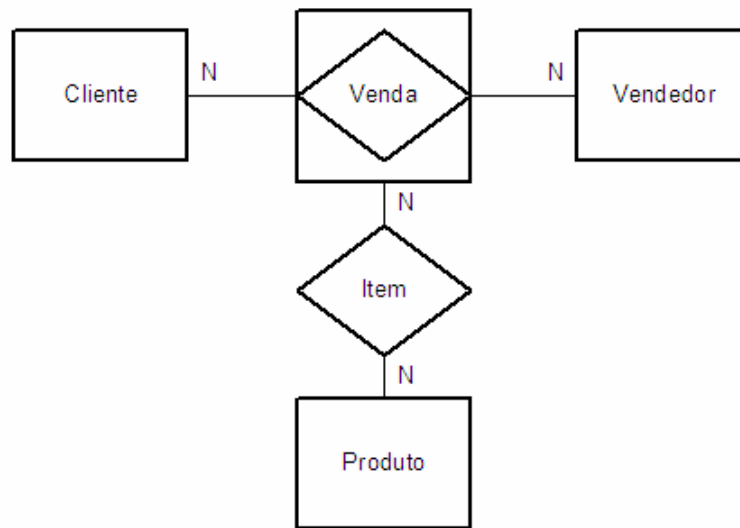


Fig. 11 – Exemplo de agregação – representação alternativa

### Redução ao modelo ER original

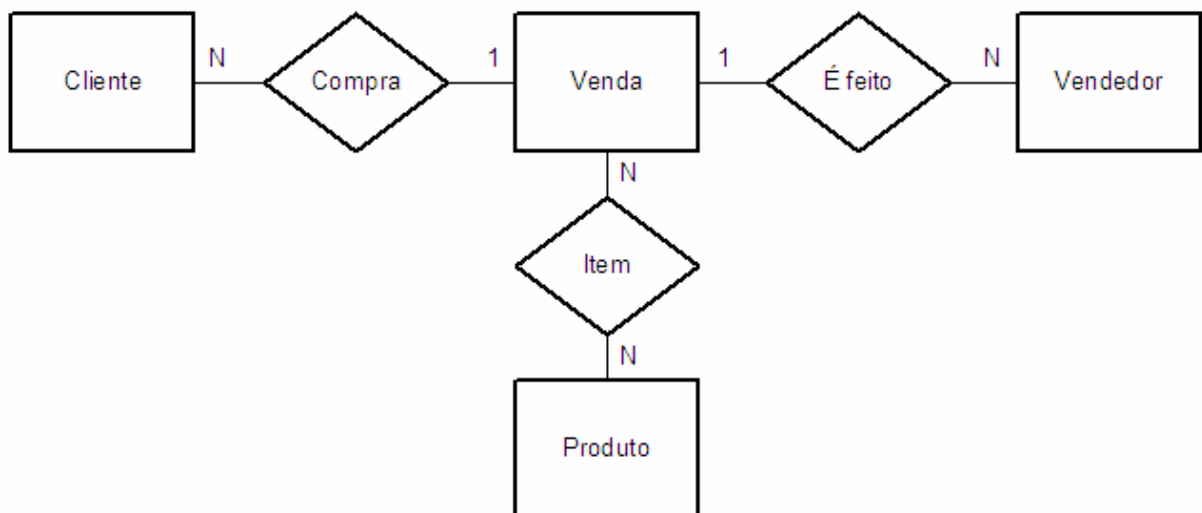


Fig. 12 –Três formas de representar logicamente uma entidade resultante de agregação

A entidade *Venda* na figura 12, é conhecida como *entidade de ligação* pois funciona como um relacionamento que associa *Cliente* a *Vendedor*, e como uma entidade associada a *Produto*.

### Generalização/Especialização

Relacionamentos de generalização-especialização ocorrem entre entidades com atributos globais e entidades especializadas a partir destas, com atributos específicos. Este relacionamento, também conhecido como “is-a” (“é-um”), e pode ser representado na notação mostrada na figura 13. Na mesma figura, apresentamos a maneira de representar este tipo de associação usando

apenas entidades e relacionamentos.

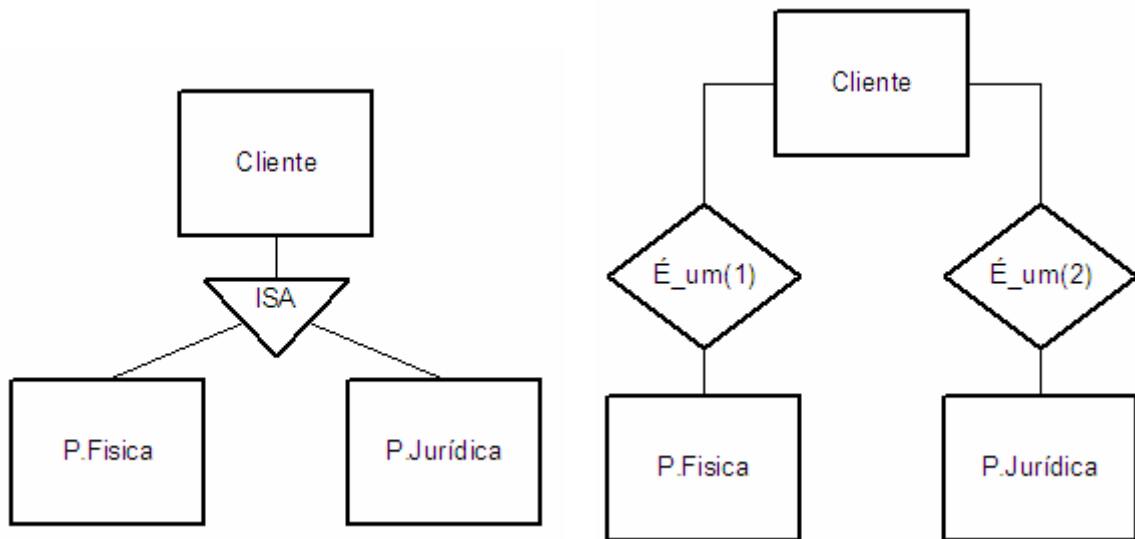


Fig. 13 –Relacionamento de generalização-especialização

### Utilização do método Entidade-Relacionamento

O método entidade-relacionamento é um método para projeto lógico de banco de dados. A idéia chave no modelo E - R é acrescentar um estágio intermediário ao projeto lógico de banco de dados. O projetista do banco de dados primeiro identifica as entidades e relacionamentos que são de interesse para a empresa ou instituição usando a técnica diagramática Entidade - Relacionamento. Nesse estágio, o projetista deve examinar os dados do ponto de vista da empresa como um todo, e não com a visão de programador de aplicação específica. Por exemplo, vamos chamar a descrição da visão de uma empresa quanto aos dados de “esquema da empresa”. O esquema da empresa deve ser uma representação pura do mundo real e deve também ser independente de consideração sobre armazenamento e eficiência. O projetista primeiro projeta o esquema da empresa e então o traduz a um esquema do usuário para seu sistema de banco de dados.

### Vantagens do método Entidade – Relacionamento

Como já dissemos acima, o método entidade – relacionamento para projeto lógico de banco de dados é formado por duas partes principais:

- Definição do esquema da empresa usando o diagrama Entidade – Relacionamento;
- Tradução do esquema da empresa em um esquema do usuário.

As vantagens de se utilizar esse método são:

- A divisão da funcionalidade e trabalho em duas etapas torna o processo do projeto de banco de dados mais simples e mais bem organizado;
- esquema da empresa fica mais fácil de ser projetado do que o esquema do usuário, porque não precisa ser restrito pelas características do sistema de banco de dados e é independente de considerações sobre armazenamento e eficiência;

O esquema da empresa é mais estável do que o esquema do usuário;

- esquema da empresa expresso pelo diagrama de Entidade – Relacionamento é mais facilmente compreendido por pessoas não ligadas ao processamento eletrônico de dados.

Etapas no projeto lógico de banco de dados segundo o modelo ER. O método E – R consiste nas seguintes etapas:

- Identificar tipos de entidades;
- Identificar tipos de relacionamentos;
- Desenhar um diagrama E – R com tipos de entidade e relacionamentos;
- Identificar tipos de valor e atributos;
- Traduzir o diagrama E – R em um diagrama de estrutura de dados;
- Projetar formatos de registros.



## Capítulo 2 – Normalização

### Introdução

Antes de sabermos o que é normalização, devemos saber o conceito de formas normais, pois cada uma das regras das formas normais apresenta um critério segundo o qual a tabela tem ou não um projeto adequado. **Formas normais** são várias regras expressando critérios práticos de simplificação de tabelas. E finalmente, o processo de adequação de tabelas às regras práticas chama-se *normalização*.

A normalização tem a função de analisar tabelas e organizá-las de forma que a sua estrutura seja simples, relacional e estável, a fim de que o gerenciamento possa ser também simples, eficiente e seguro. Os objetivos são evitar a perda e a repetição da informação e atingir uma forma de representação adequada para o que se deseja armazenar.

Para adequar uma tabela a uma forma normal, deve-se redesenhar seu formato. Também pode-se usar normalização para projetar, partindo de um documento existente, considerando-o como sendo uma tabela única, e aplicando as regras. A normalização não faz parte do modelo ER, mas pode auxiliar enormemente o controle da qualidade de um projeto lógico.

Quando uma tabela não atende ao critério de uma forma normal, sua estrutura é redesenhada através da *projeção* de alguns atributos e construção de nova(s) tabela(s), onde algum atributo é incluído na(s) nova(s) tabela(s) para que se possa refazer o conteúdo da tabela original através da *junção* das tabelas resultantes. O uso da normalização como auxiliar no projeto lógico segundo o modelo ER implica em redesenho do diagrama ER, ou seja, nova concepção lógica.

A aplicação das regras de normalização requer o conhecimento de alguns conceitos, que serão vistos conforme a necessidade de utilização. Eles são:

Conceito Básico	Visto em
Domínio multivalorado	Primeira forma normal (1FN)
Atributo determinante ou Chave	Primeira forma normal (1FN)
Determinante	Segunda forma normal (2FN)
Dependência funcional	Segunda forma normal (2FN)
Dependência funcional transitiva	Terceira forma normal (3FN)
Chave candidata	Forma normal de Boyce-Codd (FNBC)
Fato multivalor	Quarta forma normal (4FN)

### Primeira Forma Normal (1FN): Eliminação de domínios multivalorados

#### Enunciado:

*“Uma tabela só estará na 1FN se nenhum dos seus atributos tem domínio multivalorado”*

Para entendermos melhor, imaginaremos uma nota fiscal, onde teríamos nome do cliente, endereço do cliente, número da nota, nome do vendedor, data, código do produto, descrição do

produto, quantidade, preço unitário, total. Para isso, teríamos que ter atributos para cada coisa armazenada na nota fiscal.

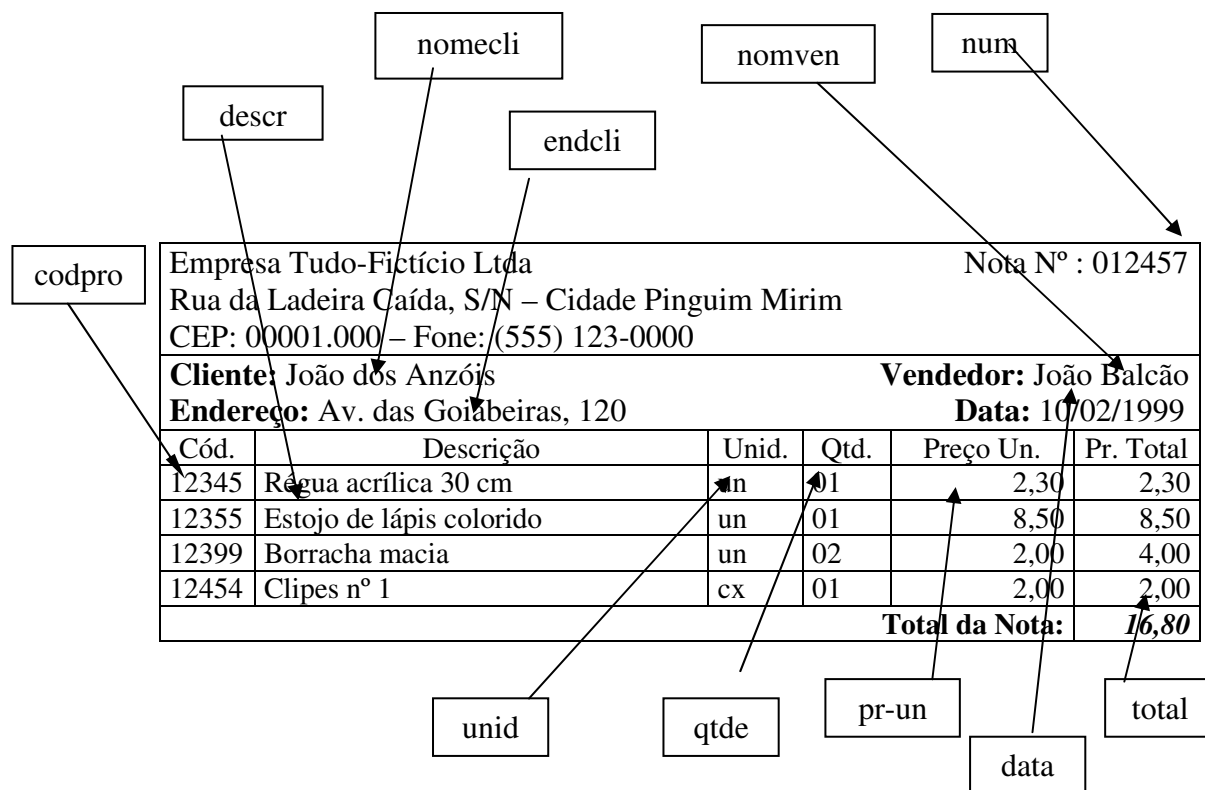


Fig. 14 – Modelo de uma nota Fiscal

O total da nota é um elemento de dados redundantes, pois pode ser calculado a partir das quantidades e preços de cada item vendido. Idealmente, não é necessário armazená-lo.

A idéia é armazenar todos os atributos em um tipo de registro, ou seja, em uma única tabela. Chamaremos **domínio** ao conjunto de valores possíveis para um atributo. Assim, são **domínios simples**: *num*, *data*, *total*, *codven*, *codcli*, *nomecli*, *endcli*, pois seus elementos são atômicos, assumem um e só um valor. São **domínios multivalorados** ou múltiplos: *codpro*, *descr*, *un*, *pr\_un* e *qtde*, pois podem assumir muitos valores em cada registro da tabela (em cada nota fiscal).

Portanto, temos que projetar os atributos *codpro*, *descr*, *un*, *pr-un* e *qtde* para fora da tabela original, se quisermos torná-la uma tabela 1FN.

Cinco atributos violaram a condição 1FN, então, a normalização fará surgir uma a cinco novas tabelas, além da tabela original desmembrada. No caso, podemos colocar todos os cinco atributos projetados em uma única tabela, pois eles representam um item da nota fiscal. Para cada nota, haverá exatamente o mesmo numero de atributos *codpro*, *descr*, *un*, *pr-un* e *qtde*.

Para preservar a informação dada pela tabela original precisamos dizer de qual nota fiscal ela veio. Portanto, o atributo a ser incluindo na nova tabela é *num*.

A escolha de *num* para integrar a tabela de itens não é casual, pois *num* é o **atributo determinante** ou **chave** da Nota\_Fiscal, o atributo cujo valor identifica uma ocorrência da tabela, pois nunca haverá duas notas fiscais com o mesmo numero (*num*).

**Nota\_Fiscal**

Atributo	Significado
num	Número da nota
data	Data de emissão
total	Valor total da nota
codven (*)	Código do vendedor
nomven	Nome do vendedor
codcli (*)	Código do cliente
nomcli	Nome do cliente
endcli	Endereço do cliente
codpro (*)	Código do produto
descr	Descrição do produto
un	Unidade de medida na qual o produto é vendido
pr-un	Preço unitário de venda do produto
qtde	Quantidade vendida

*tabela 4 – Levantamento inicial baseado na nota fiscal*

Mostraremos, agora, o resultado da normalização:

Analisando, onde sublinhamos a chave, vemos que contem apenas elementos atômicos e portanto, esta na 1FN.

O conteúdo da tabela original sempre poderá ser recuperado através da junção, usando o ponteiro que associa a duas tabelas.

Estamos fazendo o caminho inverso do projeto lógico, partindo de um projeto de registro e alterando a concepção do modelo lógico. A alteração feita, partindo a tabela não-normalizada *Nota fiscal* em duas, gerou as tabelas que tem a representação equivalente em termos de entidade e relacionamento.

A 1FN serve para evitar que se tenha que reservar espaços para armazenar dados múltiplos, sendo que o espaço pode ser desperdiçado em um registro e ser insuficiente em outro.

Usamos a 1FN projetando-se os atributos com domínio multivalorado para fora da tabela, levando um atributo (que geralmente é a chave da tabela original) como elo para refazer a ligação e recuperar o conteúdo da tabela original.

Depois de normalizada a tabela *Nota\_Fiscal*, vamos obter as seguintes novas estruturas:

<b>Nota_Fiscal</b>	<b>Item</b>
<u>Num</u>	<u>num</u>
Data	<u>codpro</u>
Total	descr
Codven	un
Codcli	pr_un
Nomcli	qtd
Endcli	

**Segunda forma normal (2FN): dependência do atributo determinante****Enunciado:**

*“Para uma tabela estar na 2FN ela deve estar na 1FN e seus atributos dependerem funcionalmente da totalidade da chave ou atributo determinante.”*

Assim, aplicando-se o enunciado da 2FN nas duas tabelas (Nota\_Fiscal e Item):

**Nota\_fiscal**chave: **num**

<u>Atributo</u>	<u>Depende da totalidade da chave?</u>
data	Sim
total	Sim
codven	sim
nomven	sim
codcli	sim
nomcli	Sim
endcli	Sim

**Item**chave: **num+codpro**

<u>Atributo</u>	<u>Depende da totalidade da chave?</u>
descr	não, apenas do codpro
un	não, apenas do codpro
pr_un	Sim
qtd	Sim

O **atributo determinante** ou **chave** é tal que, conhecendo o seu valor, identifica-se um registro da tabela. **Determinante** é um atributo que denuncia o valor de outro atributo, o **determinante funcional**. Por exemplo, na tabela *Nota fiscal*, *num* é determinante de *data*, *total*, *codven*, e etc. , pois o conhecimento do atributo *num* implica em conhecer também os demais atributos. Além de determinante desses atributos, *num* é a chave da tabela. Na mesma figura, se conhecermos *codpro*, determinaremos *descr* e *un* na tabela *item*. Se conhecermos *num* e *codpro*, determinaremos *qtde* e *pr\_un*, ou seja, o preço e a quantidade vendida de um item de nota fiscal são dependentes funcionais do número da nota e do código do produto.

A 2FN aplica-se a tabela onde a chave (atributo determinante) é composta por mais de um atributo. Portanto, a tabela *Nota\_Fiscal* já está na 2FN.

Com isso teremos as tabelas:

Nota fiscal	Item	Produto
<u>num</u>	<u>num</u>	<u>codpro</u>
data	<u>codpro</u>	descr
total	pr_un	un
codven	qtde	
Nomven		
codcli		
Nomcli		
Endcli		

A 2FN serve para evitar que se mantenham informações sobre um conjunto que tem interseção com o conjunto representado na tabela, mas tem existência independente.

Usamos a 2FN projetando-se os atributos que dependem funcionalmente de parte da chave para fora da tabela, levando a parte da chave que os determina como elo para refazer a ligação e recuperar o conteúdo da tabela original.

### Terceira forma normal (3FN): eliminação da dependência funcional transitiva

#### Enunciado:

“Uma tabela estará na 3FN quando estiver na 2FN e não houver dependência funcional transitiva entre seus atributos.”.

Caracterizamos a dependência funcional na definição da 2FN. **Dependência funcional transitiva** é a situação onde um atributo depende de outro e este segundo depende de um terceiro. É o caso da tabela Nota: *nomven* depende funcionalmente de *codven*, e este depende de *num*. Também *nomcli* e *endcli* dependem transitivamente de *codcli* e *num*.

Nota_Fiscal	Cliente	Item	Vendedor	Produto
<u>num</u>	<u>Codcli</u>	<u>num</u>	<u>codven</u>	<u>codpro</u>
data	Nomcli	<u>codpro</u>	nomven	descr
total	Endcli	pr_un		un
codven		qtde		
codcli				

Para construir tabelas 3FN, devemos eliminar de *nota* as dependências transitivas, mantendo apenas dependências funcionais diretas. Logo, *nomven*, *nomcli* e *endcli* saem da tabela, levando junto seu determinante direto para que seja possível recompor a tabela original.

A 3FN serve para separar subconjuntos insertos em um superconjunto e evitar redundância nas informações.

Usamos a 3FN projetando-se os atributos que dependem funcionalmente da chave para fora da tabela, levando o seu determinante direto como elo para refazer a ligação e recuperar o conteúdo da tabela original.

## Forma normal de Boyce-Cood (FNBC)

### Enunciado:

*“Uma tabela está na FNBC quando todo atributo determinante existente na tabela é chave candidata”.*

Todo atributo (ou concatenação de atributos) cujo valor é único em uma tabela constitui uma **chave candidata**. Pode ser que haja situações em que uma tabela possua mais de um atributo com valor único, como por exemplo, se resolvêssemos colocar também o número do CPF e do RG do cliente, a tabela Cliente terá três chaves candidatas. Na tabela acima, *num* é chave candidata de *Nota\_Fiscal*, *codven* de *Vendedor*, *codcli* de *Cliente*, *codpro* de *Produto* e *num+codpro* de *Item*. Se verificarmos as tabelas resultantes da 3FN, não encontraremos outros determinantes além das chaves candidatas. Portanto, todas as tabelas estão de acordo com a FNBC.

Não se admira, pois a forma de Boyce-Cood foi enunciada como um aperfeiçoamento da 3FN. Vemos que a FNBC garante que estão satisfeitas a 1FN, 2FN e 3FN.

## Exercícios - Modelagem de banco de dados

- 1) Explique o que é uma entidade fraca. Dê um exemplo (gráfico).
- 2) Escreva como são as visões que um usuário pode ter diante dos 3 modelos de dados
- 3) Em que situação aplicam-se os Sistemas Gerenciadores de Bancos de Dados
- 4) Quem é o responsável pela Administração de S.G.B.D. ? Quais são as suas principais responsabilidades?
- 5) Qual a relação aos termos utilizados em Modelagem de Bancos de Dados com os das utilizadas em Programação?
- 6) Como chamamos o local onde os dados são armazenados ? Como fazemos a sua representação gráfica?
- 7) Defina:
  - a) Atributo Determinante
  - b) ISA
  - c) Entidade Regular
  - d) Ocorrência
  - e) Banco de Dados
  - f) Modelo Relacional de Dados
- 8) Explique como devemos proceder quando identificamos que o relacionamento pode ter classe 0. Demonstre graficamente.
- 9) Cite 4 operações relacionais que podemos utilizar em um modelo relacional. Explique-os.
- 10) Fazendo-se uma relação entre modelos de dados e ambientes computacionais, escreva qual(is) são as possibilidades encontradas hoje em dia?
- 11) Imagine que você terá que montar um DER para um consultório médico. Fazendo os levantamentos de dados, você identificou em entidade Médico. Sabendo-se que um Médico possui somente uma especialidade e que nesta clínica podem existir várias especialidades, como você resolveria este problema? Demonstre graficamente.
- 12) Qual a relação aos termos utilizados em Modelagem de Bancos de Dados com os das utilizadas em Programação?
- 13) Dentro de um DER, como fazemos para representar um entidade ?
- 14) Defina:
  - a) Generalização/Especialização
  - b) Grau 3 de um relacionamento
- 15) Uma Empresa de Consultoria, foi contratada para elaborar um banco de dados para uma Vídeo Locadora. Para isso foram realizadas uma série de entrevistas, donde os principais pontos que se destacaram foram:
  - a) Necessidade de controlar os seus clientes. Sendo que um cliente poderia ter até 3 dependentes autorizados por ele.
  - b) Necessidade de controlar as fitas alugadas e disponíveis.
  - c) Necessidade de controlar os recebimentos dos alugueis.
  - d) Necessidade de controlar e classificar as fitas por gênero, origem (País), diretor(es), ator(es), dublado e/ou legendado, etc.

**NOTAS:** O diretor de um filme pode ser ator em outro. Podem existir 1 ou mais fitas de um mesmo filme.

Baseado nestas informações, determine as entidades e os respectivos atributos, necessários para este banco de dados.

- 16) Tomando-se como base os conceitos das três formas normais (1FN, 2FN e 3FN), explique como devemos proceder para aplicarmos estes conceitos em um modelo ER.
- 17) Em que situações, aplicam-se: atributo determinante, chave candidata e chave estrangeira?
- 18) Como identificamos uma entidade fraca em um DER?
- 19) Explique classe e grau de um relacionamento.
- 20) Em generalização/especialização, cita-se atributos locais e atributos globais. Explique o que significam estes tipos de atributos.
- 21) Cite quais são as etapas que devemos seguir para projetar um dado banco de dados? Fale sobre 2 delas.
- 22) Analise o documento abaixo e:
  - a) Aplique a normalização até 3FN.
  - b) Monte o DER.

Empresa XYZ – Demonstrativo de Pagamentos					
Departamento: <i>Informática</i>			Data Ref.: <i>Nov/98</i>		
Funcionário: <i>José Manda Chuva</i>			Matrícula: <i>00120</i>		
Conta Corrente: <i>87986-X</i>			Banco: <i>Pouca Grana</i>		
Proventos			Descontos		
Cód.	Descrição	Valor	Cod.	Descrição	Valor
01	Salário Normal	570,00	51	Faltas não Justificadas	10,00
11	Hora Extras	47,00	61	I.R.	57,00
21	Salário Família	6,23	71	INPS	20,00
			81	Farmácia	12,00
Total Proventos...		623,23	Total Descontos...		99,00
Valor Total a Pagar...					524,23

- 23) Analise o documento abaixo, e responda o que se pede.
  - a) Faça a normalização de 1FN até 3FN;
  - b) Monte o DER, não esquecendo de destacar: ENTIDADES, RELACIONAMENTOS, CARDINALIDADES E ENTIDADES FRACAS (SE EXISTIREM).

EMPRESA SECOS E MOLHADOS S/C LTDA		DEMONSTRATIVO DE PAGAMENTOS – MÊS 05/2001	
FUNCIONÁRIO: 00120 – JOSÉ DA SILVA		FUNÇÃO: ESCRITURÁRIO	
Código	Descrição do Evento	Tipo Lançam.	Valor
001	SALÁRIO NORMAL	Prov	1.200,00
010	HORAS-EXTRAS	Prov	68,00
015	SALÁRIO FAMÍLIA	Prov	35,70
200	INPS	Desc	-120,00
210	IMPOSTO DE RENDA	Desc	-112,00
215	FARMÁCIA	Desc	-46,00
SALÁRIO LÍQUIDO →			1.025,70
Tot. Proventos: R\$ 1.303,70		Tot. Descontos: R\$ 278,00	FGTS: R\$ 102,57
		Base de Calc.: R\$ 1.268,00	



24) Em um estudo realizado na biblioteca de nossa Escola, deseja-se criar um banco de dados que vise controlar todos os livros que compõem o acervo. Para tanto, é desejado manter os seguintes controles.

- Cadastro de alunos
- Cadastro de livros
- Cadastro de livros emprestados e disponíveis.

OBS: Para realização deste banco de dados, considere os seguintes fatos:

- Numa biblioteca, podem existir 1 ou mais exemplares de um mesmo livro. Sendo que cada exemplar tem seu código.
- Um aluno pode emprestar quantos livros desejar.

Abaixo, temos algumas perguntas que poderíamos fazer ao nosso banco de dados.

- Quantos exemplares temos de um determinado título de um livro ?
- Quais são os livros escritos por um determinado autor ?
- Quais livros estão emprestados para um determinado aluno ?
- Qual é a data prevista de devolução de um determinado exemplar ?
- Quais são os livros de uma determinada área ? (Por exemplo: Livros de Matemática para o 2º Grau).

Baseado nestas informações, crie as tabelas e os respectivos atributos. Não se esqueça de grifar os atributos chaves de cada tabela.

25) Analise o documento abaixo e:

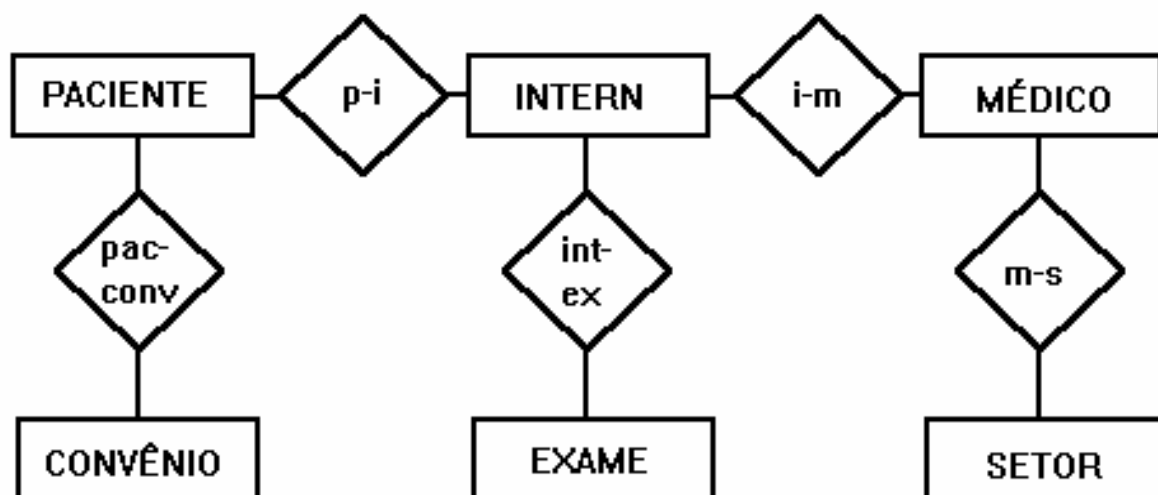
- Aplice a normalização até 3FN
- Monte o DER da normalização realizada no item anterior.

Extrato Bancário		
Banco Pouca Grana		Conta Corrente: 0001-X
Cliente: José Exemplo e/ou Maria Exemplo	Tipo Conta: Conjunta	Especial: R\$ 1.000,00
Cliente desde: 05/02/1990	Saldo Médio do Último Período: R\$ 325,00	
Data	Histórico de Lançamento	Valor Lançamento ( R\$ )
10/10/98	Saldo Anterior	10,00
10/10/98	Pagamento Cheque 123456	8,00 -
12/10/98	Lançamento de Salário	1.200,00
15/10/98	Pagamento Água mês 10/98	12,00 -
20/10/98	CPMF	0,25 -
23/10/98	Resgate Aplicação	52,00
26/10/98	Saldo Atual	1.241,75
Caro Cliente lembre-se, todo mês será sorteada uma balinha para os que mantiverem maior saldo médio.		
Telefones para Teleconsulta: (555) 555-5555		

26) Analise o DER abaixo e:

- Defina a cardinalidade de cada relacionamento (no próprio diagrama);

b) Defina o grau de cada relacionamento.



27) Analise o documento abaixo e realize a normalização até 3FN.

Horário de Aula						
Curso: <b>Processamento de Dados</b> Série: <b>3º</b> Turma: <b>X</b> Período: <b>Mat</b>						
Dia/Sem	2ª-Feira	3ª-Feira	4ª-Feira	5ª-Feira	6ª-Feira	Sábado
1ª Aula	Português	Inglês	Português		Linguagem	Laboratório
2ª Aula	Matemática	Inglês	Matemática	ISO	Téc. Prog.	Laboratório
3ª Aula	Geografia	Biologia	Biologia	ISO	Téc. Prog.	Laboratório
4ª Aula	Português	Português	Português	Estatística	Ed. Física	
5ª Aula	Física	História	T.S.P.D.	Estatística	Ed. Física	
6ª Aula	Física	História	T.S.P.D.	Linguagem	Matemática	

28) Refaça o exercício 27 de tal forma que sejam considerados uma tabela de professores que lecionam determinada matéria em um determinado dia/aula de uma classe.