

# **Apostila**

## **Teoria de Banco de Dados**

**Profª Cristiane Tuji**  
**[cris\\_tuji@yahoo.com.br](mailto:cris_tuji@yahoo.com.br)**

# Teoria de Banco de Dados

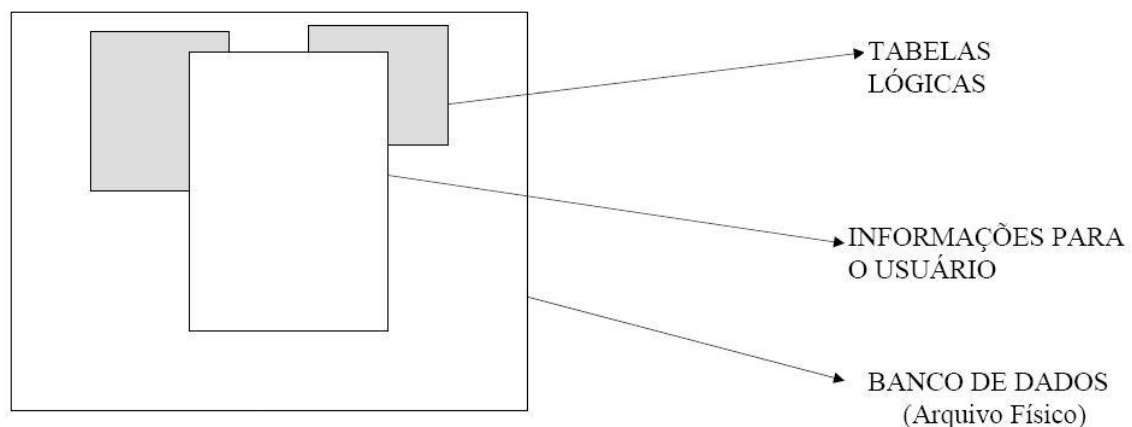
1. CONCEITOS GERAIS DE BANCO DE DADOS
  - 1.1. Conceitos Iniciais
  - 1.2. Representação Física do Banco de Dados
  - 1.3. Visões do Banco de Dados
  - 1.4. Vantagens do Banco de Dados em relação à arquitetura tradicional
    - 1.4.1. Definições
    - 1.4.2. Vantagens do Banco de Dados
2. NORMALIZAÇÃO DE DADOS
  - 2.1. Definição
  - 2.2. Primeira Forma Normal (1FN)
  - 2.3. Segunda Forma Normal (2FN)
  - 2.4. Terceira Forma Normal (3FN)
3. MODELO DE ENTIDADE E RELACIONAMENTO (MER)
  - 3.1. Definição
    - 3.1.1. Entidade
    - 3.1.2. Relacionamento
    - 3.1.3. Atributo
  - 3.2. Representação Gráfica
  - 3.3. Cardinalidade de Relacionamentos
  - 3.4. Atributos do Relacionamento
  - 3.5. Grau do Relacionamento
  - 3.6. Tipos de Relacionamento
  - 3.7. Redundância
4. LINGUAGEM SQL
  - 4.1. Comandos DDL
    - 4.1.1. Create Table
    - 4.1.2. Alter Table
    - 4.1.3. Drop Table
    - 4.1.4. Create Index
    - 4.1.5. Drop Index
  - 4.2. Comandos DML
    - 4.2.1. Insert
    - 4.2.2. Update
    - 4.2.3. Delete
    - 4.2.4. Select

# 1 - CONCEITOS GERAIS DE BANCO DE DADOS

## 1.1 – Conceitos Iniciais

- ▶ **Banco de Dados** - Representa o arquivo físico de dados, armazenado em dispositivos periféricos, onde estão armazenados os dados de diversos sistemas, para consulta e atualização pelo usuário.
- ▶ **Tabelas Lógicas** - Representam as estruturas de armazenamento de dados (arquivos) dos sistemas.
- ▶ **SGBD (Sistema Gerenciador de Banco de Dados)** - É o software responsável pelo gerenciamento (armazenamento e recuperação) dos dados no Banco de Dados.
- ▶ **Dado** - É o valor do campo quando é armazenado no Banco de Dados. Ex. O valor do campo "nome do cliente" para quem está fazendo a entrada de dados.
- ▶ **Conteúdo do campo** - É o valor do campo armazenado no Banco de Dados. Ex. O valor do campo "nome do cliente" sem estar, momentaneamente, sendo utilizado.
- ▶ **Informação** - É o valor que este campo representa para as atividades da empresa. Ex. Resposta a uma consulta. Qual os nomes dos clientes localizados no Rio de Janeiro?
- ▶ **Modelo de Banco de Dados:** Modelo Relacional, Modelo Hierárquico e Modelo em Rede. Representa a estrutura física no qual o armazenamento dos dados foi projetado. O modelo identifica a estrutura interna de recuperação e armazenamento dos dados no qual o SGBD foi projetado.

## 1.2 - Representação Física do Banco de Dados



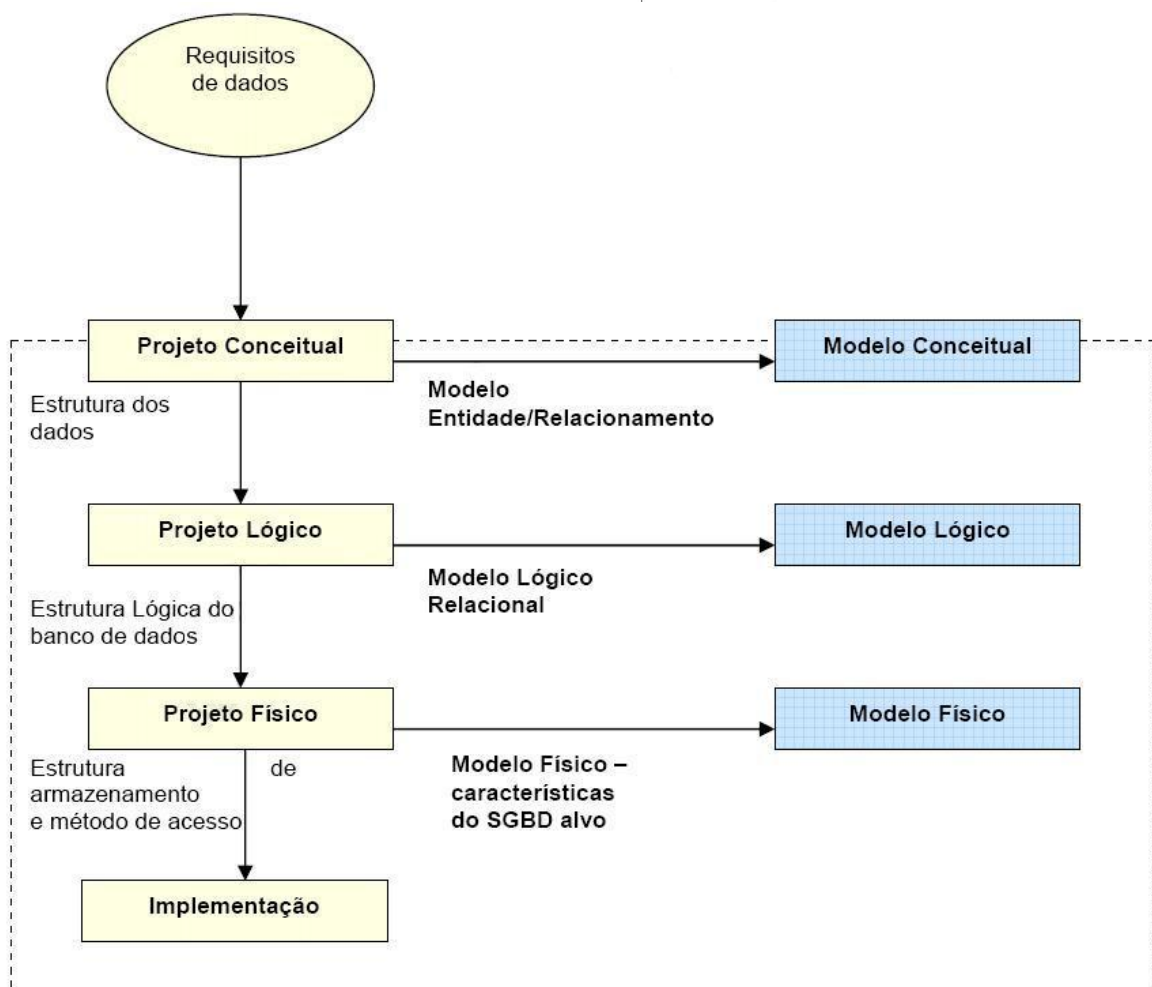
## 1.3 - Visões do Banco de Dados

- Visão Interna** - É aquela vista pelo responsável pela manutenção e desenvolvimento do SGBD. Existe a preocupação com a forma de recuperação e manipulação dos dados dentro do Banco de Dados.
- Visão Conceitual** - É aquela vista pelo analista de desenvolvimento e pelo administrador das bases de dados. Existe a preocupação na definição de normas e procedimentos para manipulação dos dados, para garantir a sua

segurança e confiabilidade, o desenvolvimento de sistemas e programas aplicativos e a definição no banco de dados de novos arquivos e campos. Na visão conceitual, existem 2 (duas) linguagens de operação que são:

- Linguagem de definição dos dados (DDL)** - Linguagem que define as aplicações, arquivos e campos que irão compor o banco de dados (comandos de criação e atualização da estrutura dos campos dos arquivos).
- Linguagem de manipulação dos dados (DML)** - Linguagem que define os comandos de manipulação e operação dos dados (comandos de consulta e atualização dos dados dos arquivos).
- Visão Externa** - É aquela vista pelo usuário que opera os sistemas aplicativos, através de interfaces desenvolvidas pelo analista (programas), buscando o atendimento de suas necessidades.

UTILIZAÇÃO DAS APLICAÇÕES DESENVOLVIDAS	VISÃO EXTERNA
DESENVOLVIMENTO DE APLICAÇÕES UTILIZANDO RECURSOS DO SGBD	VISÃO CONCEITUAL
DESENVOLVIMENTO DO SGBD	VISÃO INTERNA



#### 1.4 - Vantagens do Banco de Dados em relação à arquitetura tradicional

##### 1.4.1 - Definições

- ▶ **Sistema Tradicional** - São aqueles em que os dados do sistema estão armazenados fisicamente separados um do outro. O acesso é feito pelos programas de aplicação, associando o nome externo dos arquivos e definindo todo o registro independente da utilização dos campos.
- ▶ **Sistema de Banco de Dados** - É aquele em que os dados são definidos para o SGBD, através da DDL (linguagem de definição de dados). Fisicamente estão armazenados em um único local, sendo o acesso realizado apenas através do SGBD. Nos programas de aplicação, é necessário apenas definir os campos que serão utilizados pelo programa.

##### 1.4.2 - Vantagens do Banco de Dados

- ▶ **Redução ou Eliminação de Redundâncias** - Possibilita a eliminação de dados privativos de cada sistema. Os dados, que eventualmente são comuns a mais de um sistema, são compartilhados por eles, permitindo o acesso a uma única informação sendo consultada por vários sistemas.
- ▶ **Eliminação de Inconsistências** - Através do armazenamento da informação em um único local com acesso descentralizado e, sendo compartilhada à vários sistemas, os usuários estarão utilizando uma informação confiável. A inconsistência ocorre quando um mesmo campo tem valores diferentes em sistemas diferentes. Exemplo, o estado civil de uma pessoa é solteiro em um sistema e casado em outro. Isto ocorre porque esta pessoa atualizou o campo em um sistema e não o atualizou em outro. Quando o dado é armazenado em um único local e compartilhado pelos sistemas, este problema não ocorre.
- ▶ **Compartilhamento dos Dados** - Permite a utilização simultânea e segura de um dado, por mais de uma aplicação ou usuário, independente da operação que esteja sendo realizada. Deve ser observada apenas o processo de atualização concorrente, para não gerar erros de processamento (atualizar simultaneamente o mesmo campo do mesmo registro). Os aplicativos são por natureza multiusuário.
- ▶ **Restrições de Segurança** - Define para cada usuário o nível de acesso a ele concedido (leitura, leitura e gravação ou sem acesso) ao arquivo e/ou campo. Este recurso impede que pessoas não autorizadas utilizem ou atualizem um determinado arquivo ou campo.
- ▶ **Padronização dos Dados** - Permite que os campos armazenados na base de dados sejam padronizados segundo um determinado formato de armazenamento (padronização de tabela, conteúdo de campos, etc) e ao nome de variáveis seguindo critérios padrões preestabelecido pela empresa. Ex. Para o campo "Sexo" somente será permitido armazenamento dos conteúdos "M" ou "F".
- ▶ **Independência dos Dados** - Representa a forma física de armazenamento dos dados no Banco de Dados e a recuperação das informações pelos programas de aplicação. Esta recuperação deverá

ser totalmente independente da maneira com que os dados estão fisicamente armazenados. Quando um programa retira ou inclui dados o SGBD compacta os para que haja um menor consumo de espaço no disco. Este conhecimento do formato de armazenamento do campo é totalmente transparente para o usuário. A independência dos dados permite os seguintes recursos:

- Os programas de aplicação definem apenas os campos que serão utilizados independente da estrutura interna dos arquivos.
  - Quando há inclusão de novos campos no arquivo, será feita manutenção apenas nos programas que utilizam esses campos, não sendo necessário mexer nos demais programas. Obs: Nos sistemas tradicionais este tipo de operação requer a alteração no lay-out de todos os programas do sistema que utilizam o arquivo.
- **Manutenção da Integridade** - Consiste em impedir que um determinado código ou chave em uma tabela não tenha correspondência em outra tabela. Ex. um código de uma determinada disciplina na tabela "Histórico Escolar" sem a sua descrição na tabela "Disciplina".

## 2 - NORMALIZAÇÃO DE DADOS

### 2.1 - Definição

Consiste em definir o formato lógico adequado para as estruturas de dados identificados no projeto lógico do sistema, com o objetivo de minimizar o espaço utilizado pelos dados e garantir a integridade e confiabilidade das informações.

A normalização é feita, através da análise dos dados que compõem as estruturas utilizando o conceito chamado "Formas Normais (FN)". As FN são conjuntos de restrições nos quais os dados devem satisfazê-las. Exemplo: pode-se dizer que a estrutura está na primeira forma normal (1FN), se os dados que a compõem satisfizerem as restrições definidas para esta etapa.

A normalização completa dos dados é feita, seguindo as restrições das quatro formas normais existentes, sendo que a passagem de uma FN para outra é feita tendo como base o resultado obtido na etapa anterior, ou seja, na FN anterior. Para realizar a normalização dos dados, é primordial que seja definido um campo chave para a estrutura, campo este que permite identificar os demais campos da estrutura. Formas Normais existentes:

### 2.2 - Primeira Forma Normal (1FN)

Consiste em retirar da estrutura os elementos repetitivos, ou seja, aqueles dados que podem compor uma estrutura de vetor. Podemos afirmar que uma estrutura está normalizada na 1FN, se não possuir elementos repetitivos. Exemplo:

Estrutura original:

Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Cód. do Cliente, Nome do cliente, Endereço do cliente, CGC do cliente, Relação das mercadorias vendidas (onde para cada mercadoria temos: Código da Mercadoria, Descrição da Mercadoria, Quantidade vendida, Preço de venda e Total da venda desta mercadoria) e Total Geral da Nota)

Analisando a estrutura acima, observamos que existem várias mercadorias em uma única Nota Fiscal, sendo, portanto elementos repetitivos que deverão ser retirados.

Estrutura na primeira forma normal (1FN):

- ▶ Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente, Nome Cliente, Endereço do cliente, CGC do cliente e Total Geral da Nota)
- ▶ Arquivo de Vendas (Num. NF, Código da Mercadoria, Descrição da Mercadoria, Quantidade vendida, Preço de venda e Total da venda desta mercadoria)

Obs. Os campos sublinhados identificam as chaves das estruturas.

Como resultado desta etapa ocorre um desdobramento dos dados em duas estruturas, a saber:

- ▶ Primeira estrutura (Arquivo de Notas Fiscais): Dados que compõem a estrutura original, excluindo os elementos repetitivos.
- ▶ Segunda estrutura (Arquivo de Vendas): Dados que compõem os elementos repetitivos da estrutura original, tendo como chave o campo chave da estrutura original (Num. NF) e o campo chave da estrutura de repetição (Código da Mercadoria).

**2.3 - Segunda Forma Normal (2FN)**

Consiste em retirar das estruturas que possuem chaves compostas (campo chave sendo formado por mais de um campo), os elementos que são funcionalmente dependente de parte da chave.

Podemos afirmar que uma estrutura está na 2FN, se ela estiver na 1FN e não possuir campos que são funcionalmente dependentes de parte da chave. Exemplo:

Estrutura na primeira forma normal (1FN):

- ▶ Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente, Nome do cliente, Endereço do cliente, CGC do cliente e Total Geral da Nota)
- ▶ Arquivo de Vendas (Num. NF, Código da Mercadoria, Descrição da Mercadoria, Quantidade vendida, Preço de venda e Total da venda desta mercadoria)

Estrutura na segunda forma normal (2FN):

- ▶ Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente, Nome do cliente, Endereço do cliente, CGC do cliente e Total Geral da Nota)
- ▶ Arquivo de Vendas (Num. NF, Código da Mercadoria, Quantidade vendida e Total da venda desta mercadoria)
- ▶ Arquivo de Mercadorias (Código da Mercadoria, Descrição da Mercadoria, Preço de venda)

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

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

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

#### **2.4 - Terceira Forma Normal (3FN)**

Consiste em retirar das estruturas os campos que são funcionalmente dependentes de outros campos que não são chaves. Podemos afirmar que uma estrutura está na 3FN, se ela estiver na 2FN e não possuir campos dependentes de outros campos não chaves. Exemplo:

##### Estrutura na segunda forma normal (2FN):

- ▶ Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente, Nome do cliente, Endereço do cliente, CGC do cliente e Total Geral da Nota)
- ▶ Arquivo de Vendas (Num. NF, Código da Mercadoria, Quantidade vendida e Total da venda desta mercadoria)
- ▶ Arquivo de Mercadorias (Código da Mercadoria, Descrição da Mercadoria, Preço de venda)

##### Estrutura na terceira forma normal (3FN):

- ▶ Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente e Total Geral da Nota)
- ▶ Arquivo de Vendas (Num. NF, Código da Mercadoria, Quantidade vendida e Total da venda desta mercadoria)
- ▶ Arquivo de Mercadorias (Código da Mercadoria, Descrição da Mercadoria, Preço de venda)
- ▶ Arquivo de Clientes (Código do Cliente, Nome do cliente, Endereço do cliente e CGC do cliente)

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

As estruturas alteradas foram pelos motivos, a saber:

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

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



### 3 - MODELO DE ENTIDADE E RELACIONAMENTO (MER)

#### 3.1 – Definição

Consiste em mapear o mundo real do sistema em um modelo gráfico que irá representar o modelo e o relacionamento existente entre os dados.

##### 3.1.1 Entidade:

Identifica o objeto de interesse do sistema e tem "vida" própria, ou seja, a representação abstrata de um objeto do mundo real sobre o qual desejamos guardar informações.

Exemplo: Clientes, Fornecedores, Alunos, Funcionários, Departamentos, etc.

Não são entidades:

- ▶ Entidade com apenas 1 elemento;
- ▶ Operações do sistema;
- ▶ Saídas do sistema;
- ▶ Pessoas que realizam trabalhos (usuários do sistema);
- ▶ Cargos de direção

#### Construção do Modelo Conceitual

Esta fase destina-se à conceituação dos objetos, suas principais características e relacionamentos, independente de limitações impostas por tecnologias, técnicas de implementação ou dispositivos físicos. A atenção é voltada para o conceito, buscando identificá-lo de forma a representar fielmente a realidade observada.

A base para a identificação e definição dos conceitos é representada pela frase, quase poética, apresentada abaixo:

***“O mundo está cheio de coisas que possuem características próprias e se relacionam entre si”***

Dentro do enunciado acima, vamos identificar os seguintes pontos de interesse:

- ▶ O mundo está cheio de Coisas: Define que dentro do universo observado, identificamos objetos individualizados e que sendo relevantes dentro do universo em questão, devem ser agrupados como Entidades.
- ▶ Que possuem características próprias: Define que para os objetos identificados notamos características que os identificam e os qualificam. Tais características recebem o nome de Atributos.
- ▶ E que se relacionam entre si: Define que entre os objetos, podem existir interrelacionamentos particulares, estabelecidos em função de suas próprias características ou por restrições do universo observado.

## Comprovando as Entidades encontradas

Devemos analisar as entidades encontradas aplicando-se alguns testes para rejeitar aquelas que não são entidades:

- ▶ **Teste da Uniformidade:**  
Este teste baseia-se na definição de uma entidade. Cada instância deve possuir o mesmo conjunto de características e estar sujeita às mesmas normas. Este teste rejeita um objeto Licença, que represente licenças de veículos, licenças para cães e licenças de pesca, por exemplo.
- ▶ **Teste do mais-que-um-nome:**  
Cada entidade possui, obrigatoriamente, atributos que a caracterizam. Se a entidade candidata não pode ser descrita por atributos - não tem outra característica a não ser seu nome - provavelmente é um atributo de alguma outra “coisa” (entidade ou relacionamento multivalorado).
- ▶ **Teste do OU:**  
Se os critérios de inclusão afirmados na descrição da entidade usam a palavra OU de uma maneira significativa, provavelmente não é uma entidade, mas um punhado de idéias conglomeradas no seu lugar.

## Documentação da Entidade

Uma entidade deve ser definida claramente para que, associando-se seu nome, sua representação e sua definição, possamos ter o completo entendimento do conceito que esta procura transmitir.

Caso existam deficiências em qualquer um desses aspectos (na identificação, representação ou definição), é possível que estejamos inserindo em nosso modelo uma anomalia chamada ambigüidade.

A Identificação de entidades é uma tarefa simples. Inicia-se perguntando:

### ***Quais são as coisas neste problema?***

A maioria das “coisas” provavelmente cairá em uma das cinco categorias relacionadas a seguir:

- ▶ **Coisas Tangíveis**
- ▶ **Funções exercidas por elementos**
- ▶ **Eventos ou Ocorrências**
- ▶ **Interações**
- ▶ **Especificações**

Observe que estas categorias não são um sistema de classificação de entidades. São apenas um conjunto de idéias para que a identificação das mesmas seja uma tarefa mais fácil e direta.

- ▶ **Coisas Tangíveis:** Como o próprio nome indica, são as coisas que podem ser tocadas. Portanto englobam todos os elementos que possuem existência concreta, são manipuláveis e nos fornecem a visão de existirem fisicamente.  
Exemplo: Empregado, Cliente, Contrato, Fornecedor, Localidade;
- ▶ **Funções desempenhadas por pessoas ou organizações:** Por funções entende-se todo o tipo de papel, atribuição, classificação, capacitação, ou outra característica qualquer que, para uma dada entidade, especifique sua atuação no ambiente no qual está inserida.

Nesses casos, muito provavelmente, tal entidade poderia ser caracterizada como uma Coisa Tangível, porém sua identificação torna-se mais clara se pensarmos em termos do seu papel em relação ao ambiente.

Exemplo:

<b>Entidade (Coisa Tangível)</b>	<b>Entidade (Função, Papel)</b>
Empregado	Motorista
Localidade	Município
Especialista	Médico Cirurgião

**OBSERVAÇÃO:** A identificação de papéis associados as entidades, pode trazer uma maior semântica ao modelo de dados, introduzindo outros conceitos tais como especialização/generalização.

#### Eventos ou ocorrência:

Algumas entidades só conseguem ser percebidas ou caracterizadas, durante a ocorrência de uma certa ação. São fatos que acontecem em um determinado período.

Exemplo: Vão Comercial, Acidente, Chamada de Serviço;

#### Interações:

Os chamados objetos-interação são resultantes da associação de objetos em função de um processo executado. Tais entidades podem ser substituídas por relacionamentos multivalorados e associações.

Exemplo:

<b>Entidade-Interação</b>	<b>Entidades Participantes</b>	<b>Substituição Possível</b>
Compra de imóvel	Comprador Imóvel Proprietário	<ul style="list-style-type: none"> <li>▶ relacionamento é: comprado por</li> <li>▶ evento é: aquisição</li> <li>▶ contrato de coisa</li> </ul>
Venda	Fornecedor Produto Cliente	relacionamento é vendido por evento venda coisa tangível

#### Especificações:

Estas entidades representam características de outras entidades, em geral são entidades que aparecem quando do processo de normalização.

Exemplo:

<b>Entidade</b>	<b>Entidade especificações</b>
Equipamento	Cor Marca Modelo

#### Instância de Entidade:

São os elementos da entidade.

Exemplo: Cliente 10, Funcionário João, Aluno Pedro, etc.

### **3.1.2 Relacionamento**

Ao observarmos as entidades e reconhecê-las, estaremos, quase que imediatamente, reconhecendo as relações existentes entre elas. Muitas vezes a observação de um relacionamento será o ponto de partida para a

identificação das entidades que dele participam. O inverso também é verdadeiro, através das estruturas das entidades podemos identificar relacionamentos existentes.

Através dos relacionamentos estaremos demonstrando como uma entidade se comporta em relação às demais, qual o grau de dependência entre elas, dentre outros fatores. Para facilitar a identificação de relacionamentos, classificá-los em dois grupos.

- ▶ Relacionamento entre instâncias de entidades distintas: De acordo com a realidade observada você identificará os relacionamentos envolvidos.

Exemplo: Entidades: Pessoa e Veículo.

Entre tais entidades podemos encontrar os relacionamentos:

- Pessoa adquire veículo.
- Pessoa aluga veículo.
- Pessoa utiliza veículo.
- Pessoa vende veículo.

Cada relacionamento terá características distintas e serão corretamente identificados se inseridos num determinado contexto.

- ▶ Relacionamentos entre instâncias da mesma entidade: é também conhecido como autorelacionamento ou relacionamento recursivo.

Exemplo:

Vigilante substitui vigilante.

Disciplina é pré requisito de outra disciplina.

### **3.1.3 Atributo**

Informações que desejamos guardar sobre a instância de entidade.

Exemplo: Nome do aluno, Número da turma, Endereço do fornecedor, Sexo do funcionário, etc.

Um atributo é um elemento de dado específico, que caracteriza, qualifica e/ou descreve uma entidade ou as instâncias de um relacionamento.

Nesta atividade são identificados os principais atributos de cada entidade encontrada. Para tanto podemos fazer as seguintes referências:

- ▶ Às características das instâncias da entidade: O que cada instância deverá conter para ser entendida?
- ▶ À descrição da entidade: O que deverá ser informado para uma determinada entidade a fim de que ela seja completamente entendida?

#### Identificando Atributos

O grande objetivo dessa tarefa é obter um conjunto de atributos, que sejam:

- ▶ Completos: devem abranger todas as informações pertinentes à entidade que está sendo definida.
- ▶ Totalmente fatorados: cada atributo deve captar um aspecto separado da abstração da entidade.
- ▶ Mutuamente independentes: os atributos devem assumir seus valores tornando-se independentes uns dos outros.

Os atributos identificados devem estar dentro da seguinte classificação:

- ▶ Descritivos: possuir características inerentes à entidade.
- ▶ Nominativos: possuir nomes e rótulos.
- ▶ Referenciais: permitir a ligação de uma instância de uma entidade a uma instância de outra entidade (são as chaves estrangeiras).

### Atributos Identificadores Únicos (Chaves Primárias)

São os atributos que identificam cada uma das instâncias da entidade de forma única. Este identificador pode ser um atributo único ou ser formado por um conjunto de atributos. Estes atributos devem obedecer às seguintes restrições:

- ▶ O valor do atributo ou do conjunto é sempre único.
- ▶ O atributo ou conjunto de atributos não pode conter valores nulos.

### Documentação dos atributos

A documentação dos atributos complementa a documentação da própria entidade.

Definir e documentar os atributos de forma precisa possibilita o controle dos dados da empresa.

As seguintes regras aplicam-se à definição dos atributos ou elementos de dados ou simplesmente dados:

- ▶ Os dados devem ter nomes precisos e significativos, para que identifiquem de forma única os fatos que representam.
- ▶ O significado dos dados deve ser definido através de afirmações explícitas das políticas do negócio.
- ▶ O tipo de dado e a estrutura sintática devem ser esclarecidos.
- ▶ O conteúdo de dado, ou conjunto de valores permitidos para cada elemento de dado, deve ser especificado.
- ▶ As regras de dependência entre os elementos de dados devem ser explicitadas.

### Nome e significado do atributo

O maior cuidado que se deve ter na documentação de atributos, é com a sua simplificação.

Pelo fato do analista de sistemas estar intimamente envolvido com a concepção do modelo, o seu entendimento acerca do problema é bastante claro, o que induz a uma simplificação na hora de documentar.

Entretanto, é necessário lembrar que a documentação será útil para toda a organização, incluindo profissionais que não estão envolvidos com os detalhes do modelo. Portanto, a documentação deve abordar toda e qualquer informação que seja de valia para o processo de compreensão e unificação de conceitos.

Exemplo de Atributo: Tipo Atleta

Documentação simplificada (não recomendada):

É o tipo de atleta existente na empresa.

Note que algumas questões, não podem ser respondidas com essa definição. Por exemplo:

- ▶ Qual a finalidade em diferenciar os atletas?
- ▶ Como são enquadrados os atletas em tipos?
- ▶ O que significa um tipo de atleta?
- ▶ Qual o papel do tipo de atleta no ambiente? E no sistema?

Desta forma, uma documentação do atributo Tipo Atleta deveria ser algo como: Define, dentre o grupo geral dos atletas da empresa, quais são os que participam como titulares ou como reservas em respectivas equipes. Todo atleta é, inicialmente, cadastrado sem um enquadramento específico e, após análise é feito o enquadramento pela comissão técnica, como titular ou reserva. Essa diferenciação visa orientar o processo de escalação das equipes para eventos externos a empresa, pois somente os titulares têm suas horas de ausência abonadas.

Outro ponto a ser documentado refere-se aos domínios e consistências associadas ao atributo.

### 3.1.4 Domínio do Atributo

Domínio é o conjunto de valores que um atributo pode assumir. Este conjunto de valores, deve ser definido e criado nos casos em que se tenha mais de um atributo (coluna) com as mesmas características, tais como tipo (caractere, numérico, etc), conteúdo default, regras de validação etc.

Exemplo:

Conjunto de valores do atributo Sexo do funcionário: M ou F;

Conjunto de valores do atributo Nome do aluno: 40 caracteres alfanuméricos;

Conjunto de valores do atributo salário: inteiro maior que 5000;

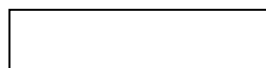
- Domínios de Atributos Descritivos: O domínio de um atributo descritivo deve ser especificado

de uma das seguintes formas:

- Enumeração: definimos uma lista explícita de todas as possibilidades.
  - Citação: citamos um documento que atua como fonte potencial das possibilidades.
  - Norma de Aceitação: Formulamos uma norma para determinar se um valor declarado pode ser aceito como valor legal do atributo.
  - Alcance: Formulamos as unidades e a faixa aceitável dos valores para o atributo.
- Domínios de Atributos Nominativos: o domínio de um atributo nominativo pode ser especificado conforme descrito acima. Em geral utiliza-se a enumeração e a citação.
  - Domínios de Atributos Referenciais: O domínio de um atributo referencial é necessariamente igual (ou compatível) ao domínio de um identificador de outra entidade do modelo.

### 3.2 - Representação Gráfica

- Entidade



- Relacionamento



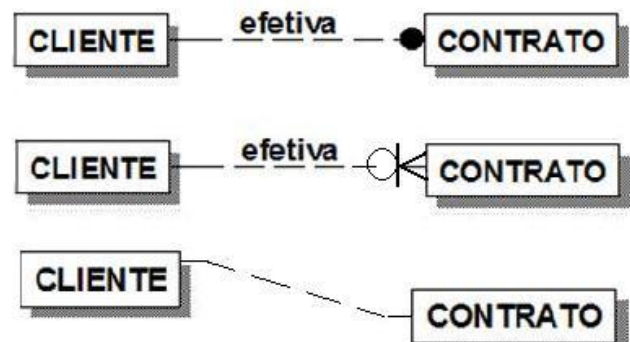
- Atributo



Notações: IDEF1X

Information Engineering

Dimensional Modeling

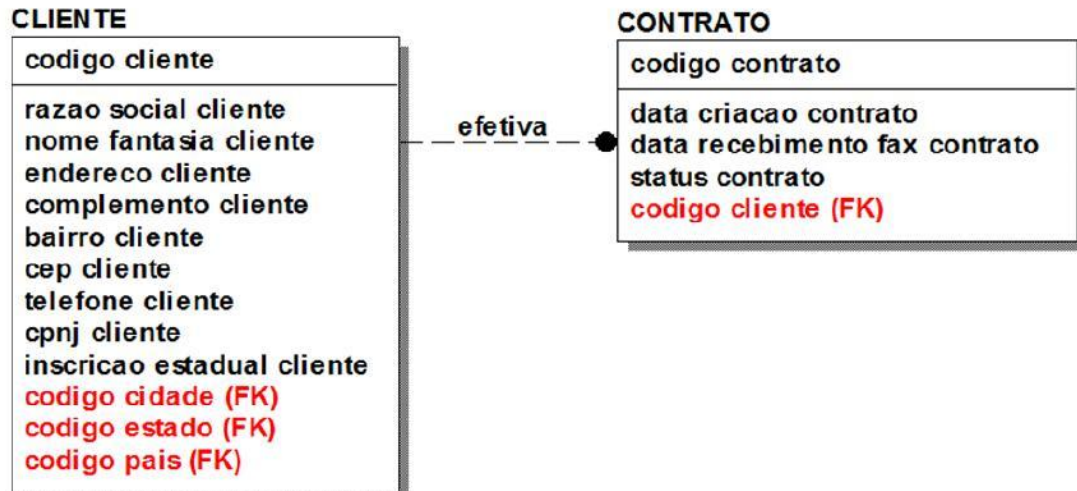




## IDEF1X

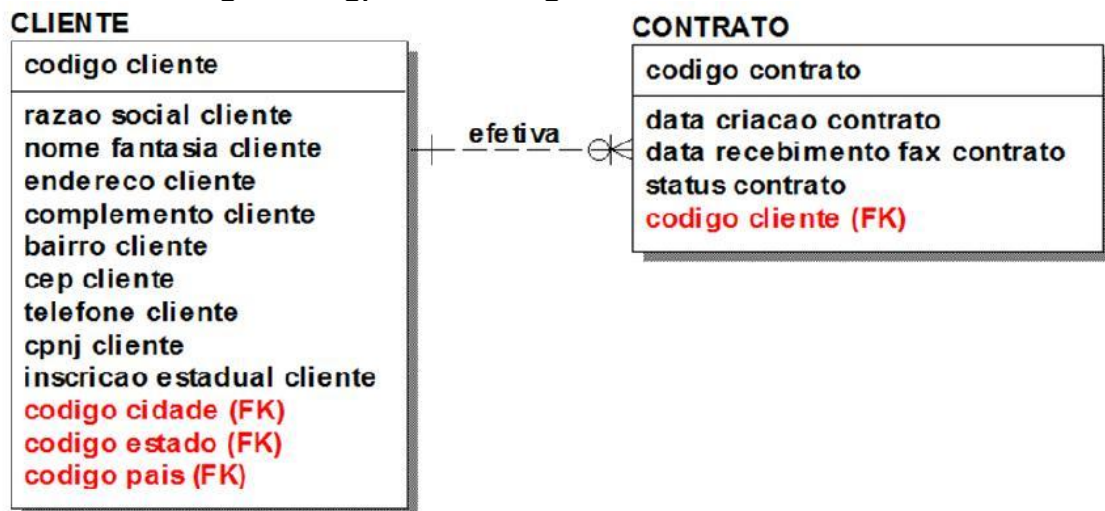
(Integrated DEFINition for information Modeling – Method 1 eXtended)

Modelo Lógico e Físico:



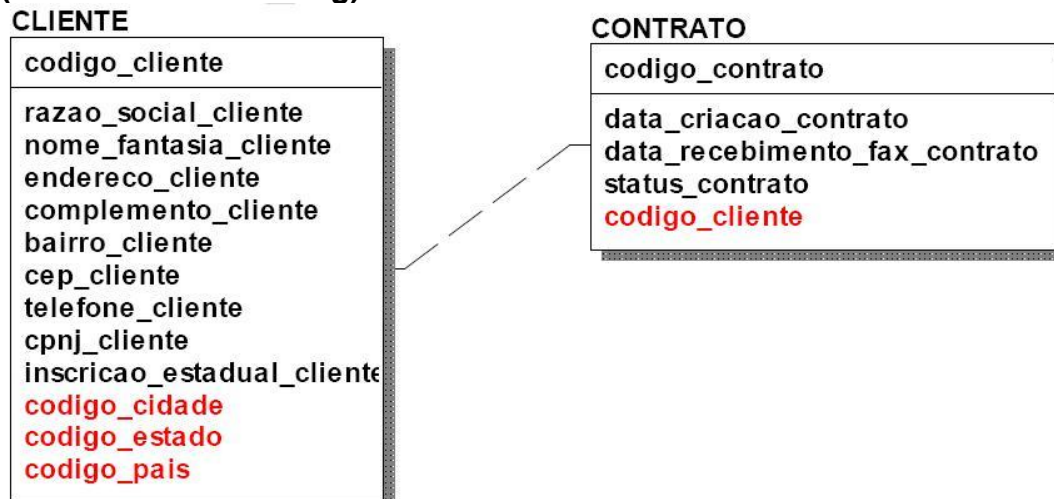
## Engenharia da Informação

(Information Engineering): Modelo lógico e físico



## Modelagem dimensional

(Dimensional Modeling): Modelo físico dimensional



## Tipos de Modelos de Dados

### - Lógico

- Orientado ao Negócio e à corporação
- Definições e regras de negócios
- Representação estruturada
- Independente do SGBD

### - Físico

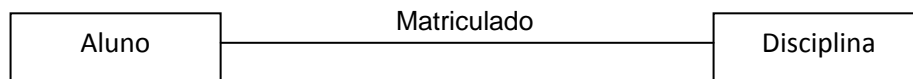
- a) Estrutura de dados no contexto de seu design e de sua implementação
- b) Dependente e limitado pelo SGBD selecionado
- c) Implementado na forma de DDL, um subconjunto de SQL
- d) Projetado para atender os requisitos de performance

### - Dimensional

- Projeto de Data Warehouse de Data Marts

**Relacionamento** - Representa a associação entre os elementos do conjunto de uma entidade com outra entidade. Exemplo: O João está matriculado na disciplina de Banco de Dados, onde:

- João - Elemento do conjunto de valores do atributo Nome do aluno da entidade Aluno;
- Banco de Dados - Elemento do conjunto de valores do atributo Nome da disciplina da entidade Disciplina;
- matriculado - Ligação existente entre um aluno e uma disciplina.

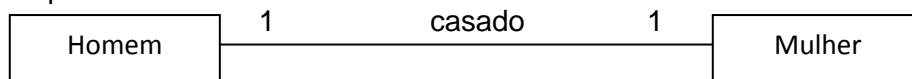


### 3.3 - Cardinalidade de Relacionamentos

Representa a frequência com que existe o relacionamento.

**Relacionamento 1:1** - O João é casado com a Maria, onde:

- João - Elemento do conjunto de valores do atributo Nome da entidade Homem.
- Maria - Elemento do conjunto de valores do atributo Nome da entidade Mulher.
- casado - Ligação entre um homem e uma mulher, sendo que um homem pode ser casado com uma e apenas uma mulher, assim como uma mulher pode ser casada com um e apenas um homem.

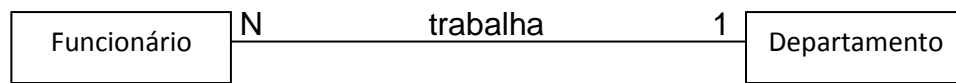


**Relacionamento 1:N ou N:1** - O Pedro trabalha no Departamento Pessoal, onde:

- Pedro - Elemento do conjunto de valores do atributo Nome da entidade Funcionário.
- Departamento Pessoal - Elemento do conjunto de valores do atributo Nome do departamento da entidade Departamento.

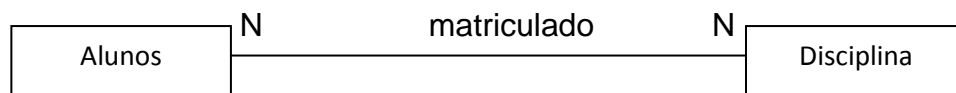


- trabalha - Ligação entre um Funcionário e um Departamento, onde um funcionário pode trabalhar em um e somente um departamento e um departamento pode ter vários funcionários.



**Relacionamento N : N (ou N : M)** - O Antônio está matriculado na disciplina Banco de Dados, onde:

- Antônio - Elemento do conjunto de valores do atributo Nome da entidade Aluno.
- Banco de Dados - Elemento do conjunto de valores do atributo Nome da Disciplina da entidade Disciplina.
- matriculado - Ligação existente entre um aluno e uma disciplina, onde um aluno pode estar matriculado em várias disciplinas e cada disciplina pode ter vários alunos matriculados.

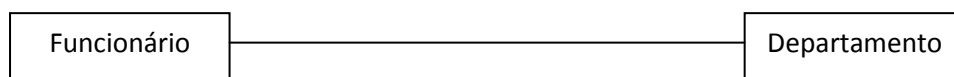


### 3.4 - Atributos do Relacionamento

Quando um determinado relacionamento possui atributos, também conhecido como relacionamento valorado. Esta situação ocorre apenas em relacionamento N : M.

Ex. Pedro trabalha no projeto Alfa 30 horas.

- Pedro - Elemento do conjunto de valores do atributo Nome da entidade Funcionário.
- Alfa - Elemento do conjunto de valores do atributo Nome do Projeto da entidade Projeto.
- trabalha - Ligação existente entre um funcionário e um projeto. Neste caso, este funcionário trabalha 30 horas neste projeto, porém este mesmo funcionário poderá trabalhar outro número de horas em outro projeto, assim como outro funcionário trabalha outro número de horas no mesmo projeto Alfa. Podemos concluir que 30 horas é o atributo que pertence ao Pedro no projeto Alfa.



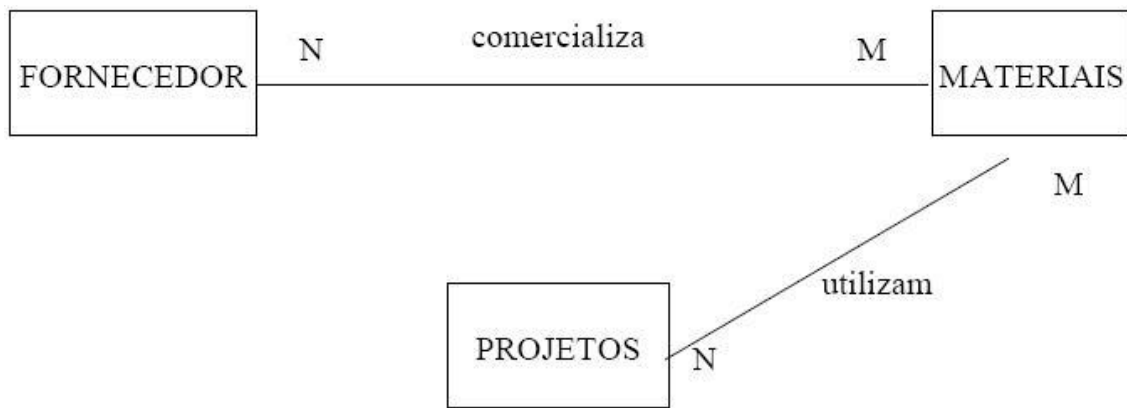
### 3.5 - Grau do Relacionamento

Indica o número de entidade que se relacionam.

#### Relacionamento Binário

Quando existe o relacionamento entre apenas duas entidades.

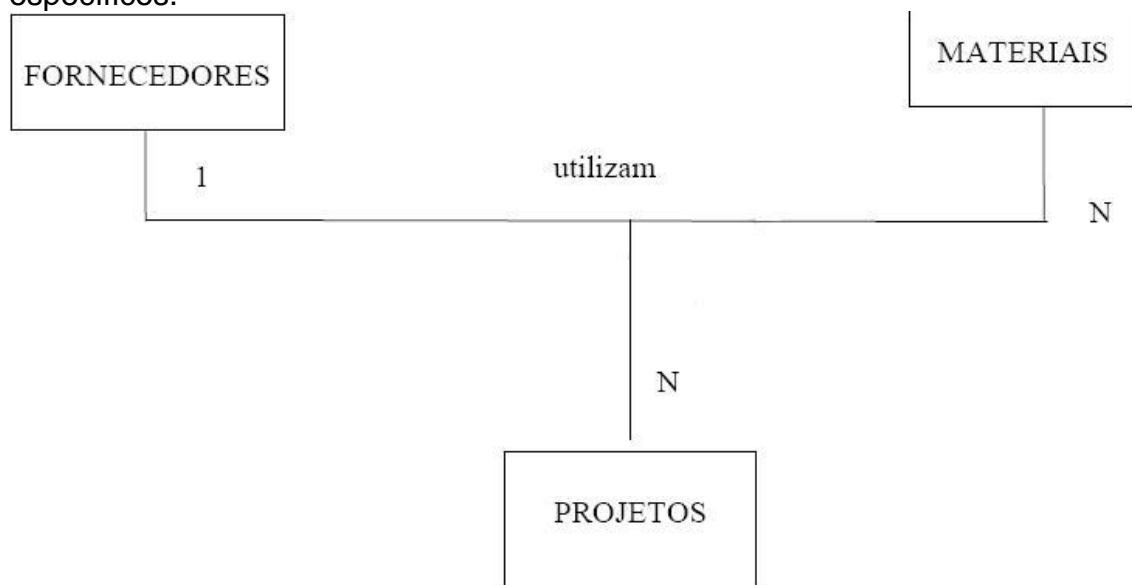
Ex. Um fornecedor comercializa materiais que são utilizados em diversos projetos.



### Relacionamento Ternário

Quando existe o relacionamento entre três entidades.

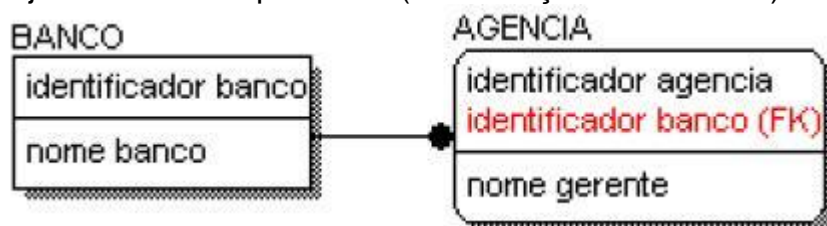
Ex. Um fornecedor comercializa materiais que são utilizados em projetos específicos.



### **3.6 Tipos de Relacionamento**

Os relacionamentos podem ser de dois tipos: Identificados e não Identificados, conforme segue abaixo:

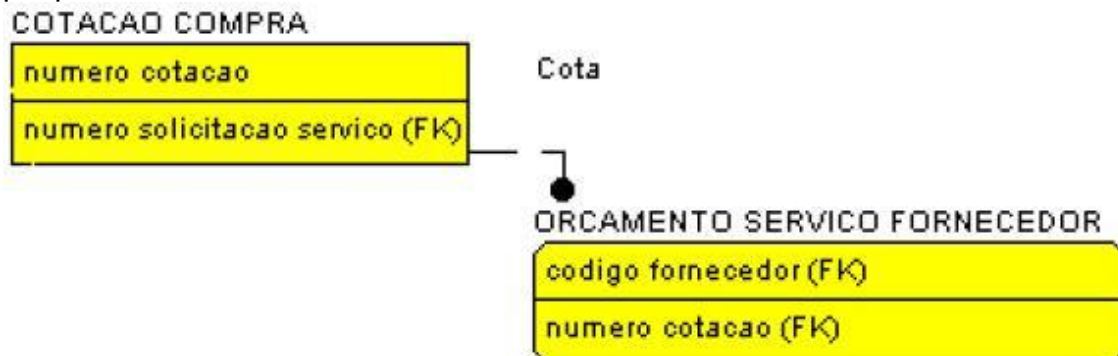
Relacionamento Identificado: No relacionamento identificado, a chave primária da entidade pai migra para a entidade filha como uma chave estrangeira, compondo a chave primária da entidade filha. Isto faz com que a entidade filha seja totalmente dependente (Identificação e Existência).



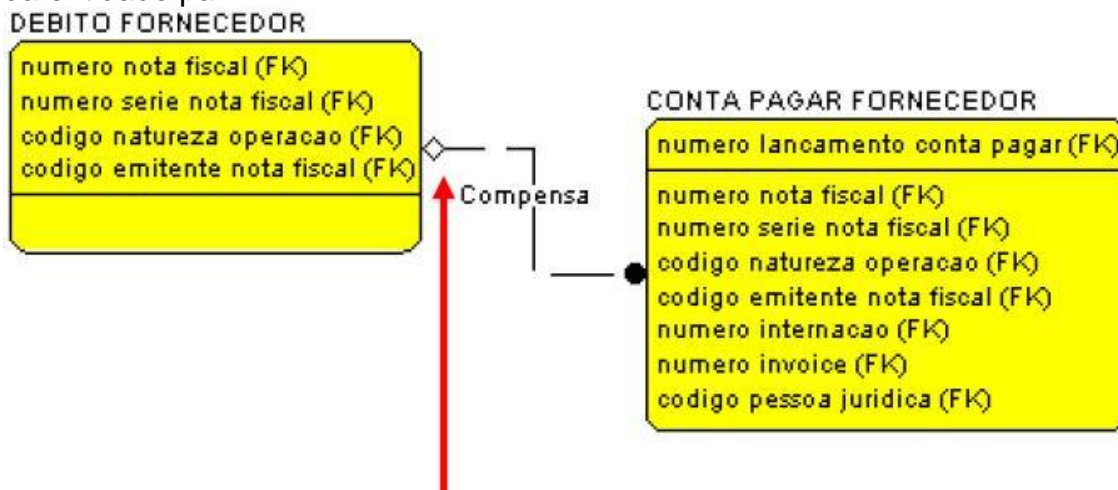
Relacionamento Não Identificado: No relacionamento Não Identificado, a chave primária da entidade pai migra para a entidade filha como atributo

simples e não identifica a entidade filha, portanto a entidade filha é independente da entidade pai quanto à sua identificação. Este tipo de relacionamento pode se apresentar em duas formas: Mandatário e Não Mandatário.

a. Não Identificado Mandatário: a entidade filha necessita de informações do pai para existir.



b. Não Identificado Não Mandatário: a entidade filha independe da existência da entidade pai.



Obs.: O losango ao lado da entidade pai indica que o relacionamento é Não Mandatário

### 3.7 Redundância

Certas redundâncias parecem inevitáveis. Entretanto, redundâncias podem gerar inconsistência de dados. Para evitar inconsistências e, como consequência, informações de baixa qualidade, efetuar formalmente, para todo caso de redundância, os seguintes procedimentos:

- Analisar a necessidade da redundância, explicitando a causa da mesma.
- Analisar os benefícios trazidos pela redundância, a fim de se ter total ciência dos verdadeiros ganhos conquistados, e desta forma, poder avaliar o seu custo/ benefício real .
- Documentar e implementar uma forma de controle que garanta a consistência entre as informações redundadas.

## 4 - LINGUAGEM SQL

A linguagem **SQL** (Structured Query Language) representa um conjunto de comandos responsáveis pela definição das tabelas, comandos e atualização dos dados em um S.G.B.D.

Os comandos existentes nesta linguagem são subdivididos em dois grupos:

- Comandos **DDL** (Data Definition Language) - Conjunto de comandos responsáveis pela criação, alteração e deleção da estrutura das tabelas e índices de um sistema.
- Comandos **DML** (Data Manipulation Language) - Conjunto de comandos responsáveis pela consulta e atualização dos dados armazenados em um banco de dados.

### 4.1 - Comandos DDL

#### 4.1.1 - Create Table

Objetivo: Criar a estrutura de uma tabela (arquivo) definido as colunas (campos) e as chaves primárias e estrangeiras existentes.

Sintaxe:

```
CREATE TABLE <nome-tabela>
(<nome-coluna> , <tipo-do-dado>          [NOT NULL]
                                     [NOT NULL WITH DEFAULT] )

PRIMARY KEY      (nome-coluna-chave)
FOREIGN KEY      (nome-coluna-chave-estrangeira) REFERENCES
                                     (nome-tabela-pai) ON DELETE [RESTRICT]
                                                         [CASCADE]
                                                         [SET NULL]
```

onde:

- a) nome-tabela - Representa o nome da tabela que será criada.
- b) nome-coluna - Representa o nome da coluna que será criada. A definição das colunas de uma tabela é feita relacionando-as uma após a outra.
- c) tipo-do-dado - Cláusula que define o tipo e tamanho dos campos definidos para a tabela. Os tipos de dados mais comuns serão definidos mais à frente.
- d) NOT NULL - Exige o preenchimento do campo, ou seja, no momento da inclusão é obrigatório que possua um conteúdo.
- e) NOT NULL WITH DEFAULT - Preenche o campo com valores pré-definidos, de acordo com o tipo do campo, caso não seja especificado o seu conteúdo no momento da inclusão do registro. Os valores pré-definidos são:  
Campos numéricos - Valor zero.  
Campos alfanuméricos - Caracter branco.  
Campo formato Date - Data corrente.  
Campo formato Time - Horário no momento da operação.

- f) PRIMARY KEY (nome-coluna-chave) - Definir para o banco de dados a coluna que será a chave primária da tabela. Caso ela tenha mais de um coluna como chave, elas deverão ser relacionadas entre os parênteses.
- g) FOREIGN KEY (nome-coluna-chave-estrangeira) REFERENCES (nome-tabela-pai) - Definir para o banco de dados as colunas que são chaves estrangeiras, ou seja, os campos que são chaves primárias de outras tabelas. Na opção REFERENCES deve ser especificado a tabela na qual a coluna é a chave primária.
- h) ON DELETE - Esta opção especifica os procedimentos que devem ser feitos pelo SGBD quando houver uma exclusão de um registro na tabela pai quando existe um registro correspondente nas tabelas filhas. As opções disponíveis são:
  - RESTRICT - Opção default. Esta opção não permite a exclusão na tabela pai de um registro cuja chave primária exista em alguma tabela filha.
  - CASCADE - Esta opção realiza a exclusão em todas as tabelas filhas que possuam o valor da chave que será excluída na tabela pai.
  - SET NULL - Esta opção atribui o valor NULO nas colunas das tabelas filhas que contenha o valor da chave que será excluída na tabela pai.

### **Tipos de dados mais comuns:**

#### **1) Numéricos:**

- Smallint - Armazena valores numéricos, em dois bytes binários, compreendidos entre o intervalo -32768 a +32767.
- Integer - Armazena valores numéricos, em quatro bytes binários, compreendidos entre o intervalo -2147483648 a +2147483647
- Decimal (n,m) - Armazena valores numéricos com no máximo 15 dígitos. Nesta opção deve ser definida a quantidade de dígitos inteiros (n) e casas decimais (m) existentes no campo.

#### **2) Alfanuméricos:**

- a) Varchar (n) - Definir um campo alfanumérico de até n caracteres, onde n deve ser menor ou igual a 254 caracteres.
- b) Char (n) - Definir um campo alfanumérico de n caracteres, onde n deve ser menor ou igual a 254 caracteres.
- c) Long Varchar - Definir um campo alfanuméricos de comprimento maior que 254 caracteres.

#### **3) Campo Date - Definir um campo que irá armazenar datas.**

#### **4) Campo Time - Definir um campo que irá armazenamento de horário.**

#### 4.1.2 - Alter Table

Objetivo: Alterar a estrutura de uma tabela(arquivo) acrescentando, alterando, retirando e alterando nomes, formatos das colunas e a integridade referencial definidas em uma determinada tabela.

Sintaxe:

```
ALTER TABLE <nome-tabela>
    DROP <nome-coluna>
    ADD <nome-coluna> <tipo-do-dado> [NOT NULL]
                                     [NOT NULL WITH DEFAULT]
    RENAME <nome-coluna> <novo-nome-coluna>
    RENAME TABLE <novo-nome-tabela>
    MODIFY <nome-coluna> <tipo-do-dado> [NULL]
                                     [NOT NULL]
                                     [NOT NULL WITH DEFAULT]

    ADD PRIMARY KEY <nome-coluna>
    DROP PRIMARY KEY <nome-coluna>
    ADD FOREIGN KEY (nome-coluna-chave-estrangeira)
    REFERENCES (nome-tabela-pai) ON DELETE [RESTRICT]
                                     [CASCADE]
                                     [SET NULL]

    DROP FOREIGN KEY (nome-coluna-chave-estrangeira)
    REFERENCES (nome-tabela-pai)
```

onde:

- a) nome-tabela - Representa o nome da tabela que será atualizada.
- b) nome-coluna - Representa o nome da coluna que será criada.
- c) tipo-do-dado - Cláusula que define o tipo e tamanho dos campos definidos para a tabela.
- d) **DROP** <nome-coluna> - Realiza a retirada da coluna especificada na estrutura da tabela.
- e) **ADD** <nome-coluna> <tipo-do-dado> - Realiza a inclusão da coluna especificada na estrutura da tabela. Na coluna correspondente a este campo nos registros já existentes será preenchido o valor NULL (Nulo). As definições NOT NULL e NOT NULL WITH DEFAULT são semelhantes à do comando CREATE TABLE.
- f) **RENAME** <nome-coluna> <novo-nome-coluna> - Realiza a troca do nome da coluna especificada.
- g) **RENAME TABLE** <novo-nome-tabela> - Realiza a troca do nome da tabela especificada.
- h) **MODIFY** <nome-coluna> <tipo-do-dado> - Permite a alteração na característica da coluna especificada.

Opções:

Além das existentes na opção ADD (NOT NULL e NOT NULL WITH DEFAULT), temos a opção NULL que altera a característica do campo passando a permitir o preenchimento com o valor Nulo.

- i) **ADD PRIMARY KEY** <nome-coluna> - Esta opção é utilizada quando é acrescido um novo campo como chave primária da tabela.
- j) **DROP PRIMARY KEY** <nome-coluna> - Esta opção é utilizada quando é retirado um campo como chave primária da tabela.

- k) **ADD FOREIGN KEY** <nome-coluna> - Esta opção é utilizada quando é acrescentado um novo campo sendo ele uma chave estrangeira.
- l) **DROP FOREIGN KEY** <nome-coluna> - Esta opção é utilizada quando é retirado uma chave estrangeira da estrutura da tabela.

#### 4.1.3 - Drop Table

Objetivo: Deletar a estrutura e os dados existentes em uma tabela. Após a execução deste comando estarão deletados todos dados, estrutura e índices de acessos que estejam a ela associados.

Sintaxe:

**DROP TABLE** <nome-tabela>

onde:

- a) nome-tabela - Representa o nome da tabela que será deletada.

#### 4.1.4 - Create Index

Objetivo: Criar uma estrutura de índice de acesso para uma determinada coluna em uma tabela. Um índice de acesso permite um acesso mais rápido aos dados em uma operação de seleção. Os índices podem ser criados a partir de um ou mais campos de uma tabela.

Sintaxe:

**CREATE [UNIQUE] INDEX** <nome-índice>  
**ON** <nome-tabela> (<nome-coluna> [ASC ], [<nome-coluna> [ASC ] ])  
[DESC] [DESC]

onde:

- a) nome-índice - Representa o nome da estrutura de índice que será criada.
- b) nome-tabela - Representa o nome da tabela que contém a coluna na qual será criada o índice de acesso.
- c) nome-coluna - Representa o nome da coluna que será criada.
- d) Opção ASC/DESC - Representa a criação do índice ordenada crescentemente (ASC) ou decrescentemente (DESC).

#### 4.1.5 - Drop Index

Objetivo: Deletar uma estrutura de índice de acesso para uma determinada coluna em uma tabela.

Sintaxe:

**DROP INDEX** <nome-índice>

onde:

- a) nome-índice - Representa o nome da estrutura de índice que será deletada.

### 4.2 - Comandos DML

#### 4.2.1 – Insert

Objetivo: Incluir um novo registro em uma tabela do Banco de Dados.

Sintaxe:

```
INSERT INTO <nome-tabela> [(<nome-coluna>, [<nome-coluna>])]  
VALUES (<relação dos valores a serem incluídos>)
```

onde:

- a) nome-tabela - Representa o nome da tabela onde será incluída o registro.
- b) nome-coluna - Representa o nome da(s) coluna(s) terão conteúdo no momento da operação de inclusão.

Obs.: Este comando pode ser executado de duas maneiras:

- 1) Quando todos os campos da tabela terão conteúdo - Neste caso não é necessário especificar as colunas, entretanto a relação dos valores a serem incluídos deverão obedecer a mesma seqüência da definição da tabela.
- 2) Quando apenas parte dos campos da tabela terão conteúdo - Neste caso devem ser especificadas todas as colunas que terão conteúdo e os valores relacionados deverão obedecer esta seqüência. Para os campos que não tem conteúdo especificado será preenchido o valor NULL.

#### 4.2.2 - Update

Objetivo: Atualizar os dados de um ou um grupo de registros em uma tabela do Banco de Dados.

Sintaxe:

```
UPDATE <nome-tabela>  
SET <nome-coluna> = <novo conteúdo para o campo>  
    [<nome-coluna> = <novo conteúdo para o campo>]  
WHERE <condição>
```

onde:

- a) nome-tabela - Representa o nome da tabela cujo conteúdo será alterado.
- b) nome-coluna - Representa o nome da(s) coluna(s) terão seus conteúdos alterados com o novo valor especificado.
- c) condição - Representa a condição para a seleção dos registros que serão atualizados.

Esta seleção poderá resultar em um ou vários registros. Neste caso a alteração irá ocorrer em todos os registros selecionados.

#### 4.2.3 - Delete

Objetivo: Deletar um ou um grupo de registros em uma tabela do Banco de Dados.



Sintaxe:

```
DELETE FROM <nome-tabela>  
WHERE <condição>
```

onde:

- a) nome-tabela - Representa o nome da tabela cujos registros serão deletados.
- b) condição - Representa a condição para a deleção dos registros. Esta seleção poderá resultar em um ou vários registros. Neste caso a operação irá ocorrer em todos os registros selecionados.

#### 4.2.4 - Select

Objetivo: Selecionar um conjunto de registros em uma ou mais tabelas que atenda a uma determinada condição definida pelo comando.

Sintaxe:

```
SELECT ALL FROM <nome-tabela> [, <nome-tabela>]  
DISTINCT  
WHERE <condição>  
GROUP BY <nome-coluna>  
HAVING <condição>  
ORDER BY <nome-campo> ASC  
DESC
```

onde:

- a) nome-tabela - Representa o nome da(s) tabela(s) que contem as colunas que serão selecionadas ou que serão utilizadas para a execução da consulta.
- b) condição - Representa a condição para a seleção dos registros. Esta seleção poderá resultar em um ou vários registros.
- c) nome-coluna - Representa a(s) coluna(s) cujos resultados são agrupados para atender à consulta.
- d) ALL - Opção default. Mostra todos os valores obtidos na seleção.
- e) DISTINCT - Opção que mostra os valores obtidos na seleção eliminando as duplicidades.
- f) WHERE - Especifica o critério de seleção dos registros nas tabelas especificadas.
- g) GROUP BY - Especifica o(s) campo(s) que serão agrupados para atender a consulta.
- h) HAVING - Especifica uma condição para seleção de um grupo de dados. Esta opção só é utilizada combinada com a opção GROUP BY.
- i) ORDER BY - Esta opção quando utilizada apresenta o resultado da consulta ordenado de forma crescente ou decrescente pelos campos definidos.

## Algumas funções utilizadas no comando Select.

a) **COUNT**(\*)  
(DISTINCT <nome-campo>)

Objetivo: Retorna a quantidade de registros existentes no campo especificado. Quando a opção \* é utilizada o resultado é a quantidade de registros existentes. Quando é referenciado o nome de um campo retorna a quantidade de valores existentes na coluna.

b) **SUM** (ALL <nome-campo>)  
DISTINCT

Objetivo: Retorna a soma dos valores existentes no campo especificado. Quando a opção DISTINCT é utilizada são consideradas apenas os diferentes valores existentes no campo.

c) **AVG** (ALL <nome-campo>)  
DISTINCT

Objetivo: Retorna a média dos valores existentes no campo especificado. Quando a opção DISTINCT é utilizada são consideradas apenas os diferentes valores existentes no campo.

d) **MAX** (ALL <nome-campo>)  
DISTINCT

Objetivo: Retorna o maior valor existente no campo especificado. Quando a opção DISTINCT é utilizada são consideradas apenas os diferentes valores existentes no campo.

e) **MIN** (ALL <nome-campo>)  
DISTINCT

Objetivo: Retorna o menor valor existente no campo especificado. Quando a opção DISTINCT é utilizada são considerados apenas os diferentes valores existentes no campo.

## GLOSSÁRIO

AK	Ver Alternate Key
Alternate Key	A <i>Alternate Key</i> , é uma chave usada em uma ou mais Colunas da Tabela, onde tem por definição, não permitir que seu grupo de Colunas sofra repetições, garantindo assim a unicidade da informação.
Atributo	É um objeto lógico, que faz parte do Modelo de Dados. Tem a função de representar algo necessário para compor uma Tabela
Auto-Relacionamento	Quando uma Tabela se relaciona com si mesma, chamamos isso de Auto-Relacionamento ou Relacionamento Recursivo.
Biblioteca do ModelMart	Para melhor organizar os Modelos de Dados no ModelMart, foram criadas uma série de Bibliotecas, que são divididas pelo tipo do Banco de Dados e o Ambiente que este Modelo foi gerado.
Bpwin	O BPwin é uma eficiente ferramenta de modelagem utilizada para analisar, documentar e melhorar complexos processos de negócios. Um modelo do BPwin permite documentar claramente fatores importantes como as atividades que são requeridas, como são desempenhadas e que recursos são necessários.
Cardinalidade	Define o número de vezes, que um relacionamento pode ser usado entre as Tabelas relacionadas. Todo o relacionamento, obrigatoriamente tem sua cardinalidade. Exemplos: <b>1 para N (Um Funcionário tem N Dependentes)</b> <b>N para N (N livros podem ser emprestado para N Pessoas)</b>
Chave Composta	A Chave Composta, é um grupo de Chaves, que juntas tem um significado único.
Chave Estrangeira	Quando uma Tabela, recebe um ou mais campos, de outra, denominamos este campo como Chave Estrangeira. Ela tem a função de manter uma integridade entre as Tabelas que se relacionam por esta Chave. O nome da Chave Primária em ferramentas e livros do assunto <b>Foreign Key</b> ou <b>FK</b> . <i>Obs.: Uma Tabela pode receber um relacionamento dela mesma, isso é chamado de Auto-Relacionamento.</i>
Chave Primária	Um ou mais Campos da Tabela, que devem ser únicos, não nulos e tem a função de identificar um registro. O nome da Chave Primária em ferramentas e livros do assunto é <b>Primary Key</b> ou <b>PK</b> .
Coluna	É um objeto Físico, que compõem uma Tabela, e tem a função de receber um tipo de dado, que agrupado com as outras Colunas da Tabela, vão formar uma informação.

Domínio	<p>O Domínio é um conjunto de informações, agrupadas por assuntos e com uma possibilidade mínima de sofrer alterações. São usados, geralmente, em campos que de TIPOS. Por exemplo:</p> <p><b>Domínio Sexo: (Campo = TIPO_SEXO)</b>  M – Masculino  F – Feminino</p> <p><b>Domínio Estado Civil: (Campo = TIPO_ESTADO_CIVIL)</b>  S – Solteiro  C – Casado  D – Divorciado  V – Viúvo  O – Outros</p>
Entidade	A Entidade é um Objeto lógico que integra o Modelo de Dados, representando uma parte do Negócio. Ela é composta por um ou mais atributos.
Erwin	Ferramenta de Modelagem de Dados, adotada pela Bovespa. Essa ferramenta apóia todos os Analistas de Sistema, na criação e manutenção de Modelos de Dados.
FK	Ver Chave Estrangeira.
Foreign Key	Ver Chave Estrangeira.
Identificante	Ver Tipo de Relacionamento.
Identifying	Ver Tipo de Relacionamento.
Integridade referencial	Regras que se podem impor aos relacionamentos, impedindo que haja referências a coisas que não existem
Mapa de Acesso Lógico	Através do uso de regras específicas que propiciam a visualização de formas de armazenamento simples e que permitiam a evolução das aplicações, com uma maior independência dos dados, sem forte preocupação, neste momento, com aspectos de performance.
Mnemônico	<p>Na língua Portuguesa, Mnemônico significa: “<b>Fácil de reter na memória.</b>” Esse termo é adaptado para a informática, como uma sigla que irá representar um determinado termo, podendo variar entre 2(duas) à 4(quatro) caracteres. Exemplo:</p> <p>TAXA = TAXA  VALOR-FINANCEIRO = VLFN  CEP = CEP</p>
ModelMart	O ModelMart é um repositório centralizado dos modelos do ERwin e do BPwin e, tem como função garantir um ambiente de modelagem coerente, gerenciando o acesso e controle de todos os modelos armazenados.
Modelo Físico	É um conjunto de objetos, que tem por obrigação, representar graficamente um determinado negócio. Sua representação gráfica é transportada para um Script, e será implantado no Banco de Dados para o qual foi projetado. Por se tratar de um produto corporativo, todos os objetos devem seguir por completo, as regras e normas

	definidas no Manual da Administração de Dados, pois só assim teremos Modelos homogêneos, e de fácil compreensão.
Modelo Lógico	É um conjunto de objetos, que tem por obrigação, representar graficamente um determinado negócio. Sua representação gráfica é transportada para o Modelo Físico Por se tratar de um produto corporativo, todos os objetos devem seguir por completo, todas as regras e normas definidas no Manual da Administração de Dados, pois só assim teremos Modelos homogêneos, e de fácil compreensão.
Não Identificante	Ver Tipo de Relacionamento.
Natureza	A Natureza, é obrigatória para a constituição do nome de um campo. Ela define qual será o valor desse Campo. Exemplo: <b>Natureza</b> – <i>Valor</i> <b>Número</b> = <i>Inteiro, Decimal e etc.</i> <b>Código</b> = <i>Inteiro, Char e etc.</i> <b>Descrição</b> = <i>Char, Texto e etc.</i> <b>Data</b> = <i>DateTime, TimeStamp e etc.</i>
Non-Identifying	Ver Tipo de Relacionamento.
Nulidade	Campo que pode estar no estado ou qualidade de nulo.
Padronização	Aplicação das regras e padrões, estabelecidas no Manual de Administração de Dados, que deve ser aplicadas em todos os Modelos de Dados
PK	Ver Chave Primária.
Primary Key	Ver Chave Primária.
Qualificador	Compõe e qualifica o nome do Campo da Tabela, sendo obrigatório no mínimo um para cada Campo. Esse Qualificador, sempre virá após uma Natureza, pois o mesmo sozinho, não dá um sentido para o Campo. Exemplo: USUÁRIO – O Qualificador Usuário não dá uma idéia total do que o Analista de Sistema quer representar em uma determinada coluna, pois existem várias naturezas para este Qualificador, por Exemplo: Nome do Usuário, Código do Usuário e etc.
Redundância	Quando temos dois ou mais objetos no Modelo de Dados, que representam a mesma coisa.
Regras de Validação	São argumentos para equalizar os Modelos de Dados da Bovespa. Com elas, os Modelos de Dados da empresa terão as mesmas características estéticas e funcionais. Essas Regras de Validação podem ser encontradas no Manual da Administração de Dados.
Relacionamento	Ligação entre uma ou mais Tabelas, por intermédio das PKs, as quais garantem a integridade entre as informações das Tabelas envolvida.
Relacionamento Recursivo	Ver Auto-Relacionamento.
RelationShip	Ver Relacionamento
RelationShip Type	Ver Tipo de Relacionamento

RoleName	O <i>RoleName</i> , é o nome do Relacionamento entre Tabelas. Geralmente ele tem o mesmo nome da Chave, porém em alguns casos é necessário mudá-lo, pois quando temos um Campo, com o mesmo nome do relacionamento que uma determinada tabela, está recebendo, o nome fica oculto.
Sinônimo	Diz-se de palavra que tem a mesma ou quase a mesma significação que outra, e por este motivo, recebem da Administração de Dados o mesmo Mnemônico.
Stored Procedure	São comandos na linguagem SQL, que tem por objetivo manipular os dados nos Bancos de Dados. <i>SP</i> , se necessária, deve ser implantada no SGDB.
Subject Area	É um recurso da ferramenta ERwin, que permite criar áreas de trabalho dentro do Modelo de Dados, onde é possível selecionar, dentre várias Tabelas, só as que são necessárias para um determinado assunto. Um exemplo disso são as Tabelas Corporativas e Incorporadoras, que podem ser separadas em <i>Subject Areas</i> diferentes em um mesmo Modelo de Dados, não restringindo a existência de uma Tabela em <b>N</b> <i>Subject Areas</i> diferentes.
Tabela	É um objeto físico, pertencente ao Modelo de Dados e tem a função de representar um determinado fato do negócio.
Tabela Auxiliar	Este tipo de tabela, serve apenas para receber dados temporários, cujo os quais só servem para cálculos, carga de dados que devem ser tratados antes de serem colocados em Tabelas, etc.
Tabela Corporativa	Uma Tabela que representa um negócio, o qual será usado por outro Sistema além da sua origem, pode ser considerada uma Tabela Corporativa. Esse tipo de tabela nem sempre nasce com essas características, e pode passar a ser Corporativa, assim que houver a necessidade de usar seu conteúdo em algum Sistema.
Tabela de Log	Para controlar o que está ocorrendo com um grupo de tabelas, é criada uma tabela de <i>Log</i> , que irá informar qual Tabela foi manipulada, por quem e que tipo de manipulação foi aplicado a ela.
Tabela de Negócio	Serve para armazenar todos os dados pertinentes ao negócio.
Tabela Dimensão	Esta tabela representa uma categoria pela qual a tabela fato está sendo analisada.
Tabela Fato	Esta tabela tem a função de representar o assunto da análise.
Tabela Histórica	Esta tabela tem a característica semelhante a tabela de <i>Log</i> , porém, ela só tem a função de controlar a manipulação de uma determinada Tabela.
Tabela Integradora	Quando há necessidade de se incorporar uma Tabela Corporativa, no Modelo de Dados que se está trabalhando, está será uma Tabela Integradora, a que está se incorporando ao grupo de Tabelas locais. Para que esta Tabela não se misture às Tabelas Locais, ela terá o seu fundo com a cor azul

Tipo de relacionamento	<p>Existem três Tipos de Relacionamentos (<i>RelationShip Type</i>)</p> <p><b>Identificante (<i>Identifying</i>):</b> O Campo, que é o vínculo do relacionamento, no caso a FK, obrigatoriamente fará parte do conjunto da PK da Tabela receptora.</p> <p><b>Não-Identificante Não Nulo (<i>Non-Identifying</i>):</b> O Campo, que é o vínculo do relacionamento, no caso a FK, não fará parte do conjunto da PK, porém seu conteúdo é obrigatório.</p> <p><b>Não-identificante Nulo (<i>Non-Identifying</i>):</b> O Campo, que é o vínculo do relacionamento, no caso a FK, não fará parte do conjunto da PK, porém seu conteúdo não é obrigatório.</p> <p>Obs.: Se a FK for composta, ela recebe os mesmos tratamentos acima descritos.</p>
Trigger	<p>É um conjunto de comandos SQL, que é disparado pelo Banco de Dados mediante a algum parâmetro programado no código.</p> <p>A <i>Trigger</i> pode ser implementada no SGDB, junto com o Modelo de Dados, diretamente na Tabela desejada.</p>
Unicidade	Garantia que uma informação não seja replicada.
Valores Default	<p>No Domínio, é necessário deixar uma das opções como Default. Exemplo:</p> <p><b>Domínio Sexo:</b>  M – Masculino  F – Feminino  Valor Default = M – Masculino.</p>
Verb Phrase	É um recurso do Modelo de Dados Lógico, para representar a ação de um relacionamento entre as Tabelas. O Verbo representa isso de uma maneira que, ao ler o mesmo, entre o Relacionamento, o leitor tenha um entendimento da ação que ocorre entre as Tabelas envolvidas.
View	A <b>view</b> comporta-se como uma tabela, porém não ocupa espaço físico para armazenar dados. Uma vez que representa apenas uma visualização de dados, permitindo e facilitando a consulta dos dados nas Tabelas, da melhor maneira para o Sistema.