

## Stephen Willian Hawking (1942 - 2018)



Hawking morreu no mesmo dia do nascimento de Albert Einstein 14 de março. Nasceu em 8 de janeiro de 1942, 300 anos depois da morte de Galileu Galilei, pai da ciência moderna. Galileo morreu em 8 de janeiro de 1642, Arcetri, na Itália.



Universidade Católica de Brasília

Universidade Católica de Brasília



**Banco de Dados**

Prof. Dr. Milton Pombo da Paz

**TODOS OS DIREITOS RESERVADOS.**

**PROIBIDA A PUBLICAÇÃO E DIVULGAÇÃO DESTE MATERIAL EM QUALQUER TIPO DE MÍDIA.**



**Universidade  
Católica de Brasília**

**Universidade Católica de Brasília**

**Pró-Reitoria de Graduação**


**Escola de Educação, Tecnologia e Comunicação**

**Curso de BSI, BCC, ADS, GTI**

**Banco de Dados**

**Prof. Dr. Milton Pombo da Paz**

 Copyright  
Todos os Direitos Reservados



## Plano de Ensino

Apresentação da Disciplina e Plano de Ensino.

Definição dos temas de pesquisas e projetos.

**Unidade 1 – Introdução:**  
Dado, Informação e Conhecimento. Conceito de banco de dados (BD). Propriedades implícitas de BD. Dados, tipo de dados e descrição de dado (meta-dados).

**Unidade 1 – Introdução:**  
Abordagem tradicional de arquivos e abordagem de banco de dados. Sistema Gerenciador de Banco de Dados (SGBD).

**Unidade 2 - Sistema Gerenciador de Banco de Dados:**  
Modelos, Esquemas e Instâncias, Arquitetura. Independência de Dados. Linguagens, Ambiente de banco de dados.

**Unidade 3 – Modelo Entidade-Relacionamento:**  
Modelo de dados conceitual. Entidades e atributos, Tipos de entidade. Atributos chave, composto, multivalorado, derivado. Relacionamentos, Tipos de relacionamento. Grau de um tipo de relacionamento. Papel (Função) de tipos de entidade em tipos de relacionamento. Restrição de cardinalidade. Restrição de participação. Modelo Entidade-Relacionamento (MER). Generalização e especialização. Utilização de Ferramenta CASE.

**Unidade 3 – Modelo Entidade-Relacionamento.**

**Unidade 3 – Modelo Entidade-Relacionamento.**

**Unidade 3 – Modelo Entidade-Relacionamento.**

**Unidade 4 - Modelo Relacional - Modelos de Banco de Dados:**  
Conceitos básicos. Domínio; tupla; atributo; relação. Restrições de Integridade.

**Unidade 4 - Modelo Relacional - Modelos de Banco de Dados:**  
Transição do Modelo Entidade Relacionamento para o Modelo Relacional.

Recesso.

Reunião geral dos colaboradores Docentes e Administrativos.

**Unidade 4 - Modelo Relacional - Modelos de Banco de Dados:**  
Definição de restrições de integridade (constraints) no MR.

**Unidade 5 - Dependência Funcional e Normalização:**  
Redundância e anomalias de atualização. Dependência funcional, 1FN, 2FN e 3FN.

**Unidade 5 - Dependência Funcional e Normalização:**  
BCNF, 4FN e 5FN.

Prova 1.

**Unidade 6 - Álgebra Relacional:**  
Operadores básicos. Seleção. Projeção. Produto Cartesiano. Operadores complementares da Álgebra relacional.

Seminário:


Entrega dos projetos.

Prova 2.

Prova Substitutiva.

Aula Síntese e divulgação de resultados.

Revisão da matéria e análise da disciplina no contexto do curso e da sociedade.

 Copyright  
Todos os Direitos Reservados

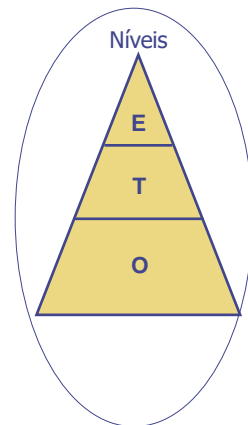
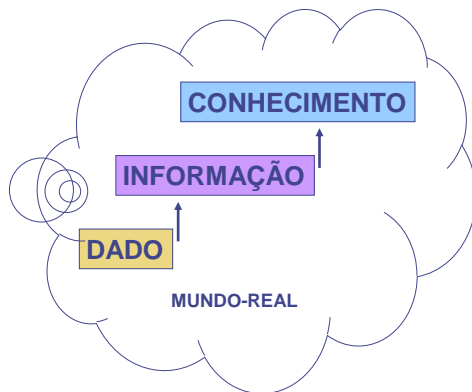
## Início

Apresentação da Disciplina e Plano de Ensino.

Definição dos temas de pesquisas e projetos.

## Fenomenologia - [Johannes Hessen](#)

Dado ➡ Informação ➡ Conhecimento ↻



## Necessidades

- Necessidade humana de armazenar as experiências

- Pedra
- Madeira
- Papel
- Máquinas

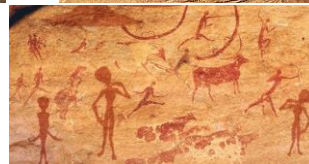
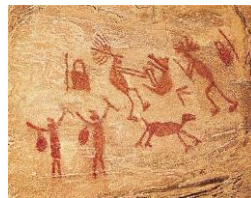


© Copyright  
Todos os Direitos Reservados

UCB – Banco de Dados - Prof. Milton

## Necessidades

- Necessidade humana de armazenar as experiências



© Copyright  
Todos os Direitos Reservados

UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

### Conceito de banco de dados (BD).

Foi Date (2004) quem realizou a primeira implementação na *International Business Machine (IBM)* do *SystemR*, quando esta resolveu validar o Modelo Relacional (MR) proposto por Codd (1970). Para Date (2004), o Sistema de Banco de Dados é basicamente um sistema de manutenção de registros por computador – ou seja, um sistema cujo objetivo global é manter as informações e torná-las disponíveis, quando solicitadas.

Segundo Elmasri (2005):

Um banco de dados é uma coleção de dados relacionados. Os dados são fatos que podem ser gravados, e tem significados implícitos e possuem as seguintes propriedades:

Um banco de dados representa alguns aspectos do mundo real, sendo chamados às vezes de mini-mundo ou de universo de discurso. As mudanças em um mini-mundo são refletidas em um banco de dados.



## Unidade 1 - Introdução

### Conceito de banco de dados (BD).

Martin (1977) define um Banco de Dados como um modelo de dados que representa a estrutura inerente dos dados e, assim, é independente de aplicações individuais e, também, dos mecanismos de software e hardware utilizados para a representação e uso dos dados. Ainda afirma que é uma coleção de dados relacionados, armazenados com redundância controlada, para servir a múltiplas aplicações. Os dados são armazenados de forma a serem independentes dos programas que o usam.

Para Batini (1992), um banco de dados é qualquer coleção extensa de dados armazenada em um sistema baseado em computador.



## Unidade 1 - Introdução

### Conceito de banco de dados (BD).

Segundo Korth e Silberschatz (1999), o gerenciamento de bancos de dados tem evoluído de um aplicativo específico de computador para um componente central de um moderno ambiente computacional.

É uma coleção de dados inter-relacionados e um conjunto de programas para acessá-los. O principal objetivo do SGBD é prover um ambiente conveniente para recuperar e armazenar informações. (KORTH, 1999, p. 1)

Segundo Korth e Silberschatz (1999), um banco de dados deve prover as seguintes funcionalidades:

- ✓ **Consistência dos dados** – Os bancos de dados modelam aspectos do mundo real. Uma vez acessados ou atualizados por uma ou mais aplicações, os bancos de dados devem manter a consistência dos dados, mesmo em caso de falhas.
- ✓ **Facilidade de acesso aos dados** – Os bancos de dados devem permitir consultas simples e sofisticadas de uma maneira simples, conveniente e eficiente.
- ✓ **Isolamento dos dados** – Mesmo que os dados estejam espalhados por diversas tabelas ou arquivos, em ambientes diferentes e com formatos diferentes, uma transação não pode interferir no processo de outra transação e o isolamento entre os dados deve ser mantido.
- ✓ **Suporte a transações distribuídas e concorrentes** – Os sistemas de bancos de dados devem suportar múltiplos acessos concorrentes e distribuídos, além de prover suporte às aplicações distribuídas.
- ✓ **Segurança** – Outra característica desejável em qualquer banco de dados é o suporte à segurança, tanto para o acesso às informações, quanto para as rotinas de cópias (*backups*) e restaurações.



## Unidade 1 - Introdução

### Base de Dados

É o conjunto de todos os dados da organização, estejam eles armazenados em computador ou não e o banco de dados é um subconjunto da base de dados.



## Unidade 1 - Introdução

### Propriedades implícitas de BD.

#### Objetivos da Utilização de Banco de Dados

São objetivos claros da utilização de bancos de dados:

- ✓ produzir informações com segurança sobre uma estrutura de dados estável;
- ✓ representar a estrutura intrínseca das entidades da corporação;
- ✓ desenvolver novas aplicações sobre dados já existentes, através do compartilhamento de dados;
- ✓ facilitar o acesso direto aos dados (consultas) por pessoas não técnicas em Informática;
- ✓ proteger os sistemas já implantados contra obsolescência;
- ✓ possibilitar o compartilhamento, a integração e a distribuição de dados;
- ✓ desenvolver aplicações de modo mais rápido, simples e flexível;
- ✓ criar mecanismos de proteção aos dados.



## Unidade 1 - Introdução

Desse modo, é preciso considerar que um banco de dados:

- ✓ não é um mecanismo de acesso direto;
- ✓ representa a empresa e o seu ambiente (o negócio);
- ✓ é apenas uma parte da solução do problema da automatização de um “negócio”;
- ✓ exige que um projeto seja desenvolvido e que uma metodologia seja adotada com vistas à sua obtenção. Tal projeto requer a adoção de uma modelagem conceitual, lógica e física dos dados e de processos (ou funções);
- ✓ requer o uso de uma ferramenta CASE para apoio, principalmente, na fase de Projeto;
- ✓ requer, após a disponibilização para os usuários (fim da fase de implementação), uma constante vigilância, pelo DBA, para observação de seu desempenho (estatísticas, entre outros);
- ✓ ser revisto pelo DBA que deve, periodicamente, realizar a descarga e carga dos dados do Banco, para melhorar seu desempenho (*back-up* e *restore* do Banco – pelo *software* específico do SGBD\_R e não pelo *back-up* do Sistema Operacional).



## Unidade 1 - Introdução

Dados, tipo de dados e descrição de dado (meta-dados).



## Unidade 1 - Introdução

- Abordagem tradicional de arquivos e abordagem de banco de dados.
- Time-line:
  - Pedra.
  - Papel.
  - SBA.
  - Hierárquico.
  - Rede.
  - Relacional.
  - Relacional Estendido.
  - OO.





## Unidade 1 - Introdução

Algumas características de um banco de dados:

- ✓ Centralização de informações – O canal de informações é o único dentro da organização.
- ✓ Integração das necessidades da empresa – A necessidade é estabelecida uma única vez, embora possa estar presente em mais de um ponto na estrutura da empresa.
- ✓ Independência de dados – Qualquer alteração na estrutura das informações, por características de qualquer ordem, não implica alteração de aplicações que não estejam diretamente afetadas

Podemos apresentar ainda alguns dos objetivos para a existência de um banco de dados:

- ✓ controle centralizado;
- ✓ controle da redundância e proliferação de dados;
- ✓ integridade de dados;
- ✓ controle sobre as inconsistências;
- ✓ compartilhamento das informações;
- ✓ estabelecimento de padronização;
- ✓ flexibilidade de estrutura proporcionada pela independência de dados física e lógica;
- ✓ facilidade no desenvolvimento de novas aplicações;
- ✓ facilidade no acesso às informações;
- ✓ privacidade e segurança;
- ✓ otimização do espaço de armazenamento;
- ✓ facilidade na avaliação de desempenho;
- ✓ flexibilidade de reorganização da estrutura.



## Unidade 1 - Introdução

### Modelos de Dados

Um modelo de dados é uma definição abstrata, autônoma e lógica dos objetos, operadores e outros elementos que, juntos, constituem a máquina abstrata com a qual os usuários interagem. Os objetos nos permitem modelar a estrutura dos dados. Os operadores nos permitem modelar seu comportamento. Podemos, então, distinguir de modo útil o modelo de sua implementação: uma implementação de um determinado modelo de dados é uma representação física sobre uma máquina real dos componentes da máquina abstrata que juntos constituem esse modelo (DATE, 2000, p.13).



## Unidade 1 - Introdução

### Modelos de Dados - Categorias

A identificação de diversas inconsistências nos sistemas existentes, tais como a necessidade de um modo melhor para descrever relações de um-para-muitos, entre diferentes registros, e a maneira mais fácil e rápida de conduzir as pesquisas aos dados, induziram à necessidade de percepção de que se deveria subir mais ainda o nível de abstração na representação da realidade a ser computável, dando origem aos Modelos de Dados e estes aos SBD (Sistema de Banco de Dados).

Esses modelos podem ser categorizados, cronologicamente, de acordo com suas estruturas de dados e de acordo com os operadores apresentados aos usuários. Assim, temos: Modelo Hierárquico (MH); Modelo de Rede (MRede), também chamados de Sistemas CODASYL (*Conference on Data System Languages*) ou Sistemas DBTG (*Database Task Group*) da CODASYL, que escreveu a primeira especificação-padrão de banco de dados, chamada de Relatório CODASYL DBTG 1971; Modelo Relacional (MR); Modelo Relacional Estendido (MRE ou Relacional-Objeto ou Pós-Relacional) e, finalmente, o Modelo Orientado a Objetos (MOO).



## Unidade 1 - Introdução

### Modelos de Dados - Categorias

#### SBA – Sistema Baseado em Arquivos

Os SBA foram os primeiros modos de armazenamentos de dados em computadores. Eram construídos de maneira intuitiva e procuravam respeitar como as organizações tratavam seus dados. Assim, se fosse realizada a automatização das vendas de uma empresa, eram criados arquivos para armazenar todos os seus dados, como: data, valores, vendedores e produtos, entre outros. Pode-se concluir que, por falta de um modelo cientificamente construído, provocava diversos problemas, como: redundância de dados, falta de controle adequado de concorrência e falta de integridade.



## Unidade 1 - Introdução

### Modelos de Dados - Modelo hierárquico

Foi o primeiro modelo de dados formal e surgiu na década de 60. Considera que as estruturas de dados eram o reflexo de sua utilização em uma organização e sendo esta hierárquica, os modelos de solução eram compostos por estruturas hierarquizadas.

Os diagramas da solução consideram as estruturas de dados hierárquicas e seus dados possuíam subordinação *top-down*, tais como: vendedor – clientes – vendas – produtos.

Essas estruturas somente permitem a navegação *top-down*, o que limita sobremaneira as buscas de dados. Os dados nesses SGBD são gravados em uma coleção de estruturas em árvores e são classificados em “pai” e “filho”.

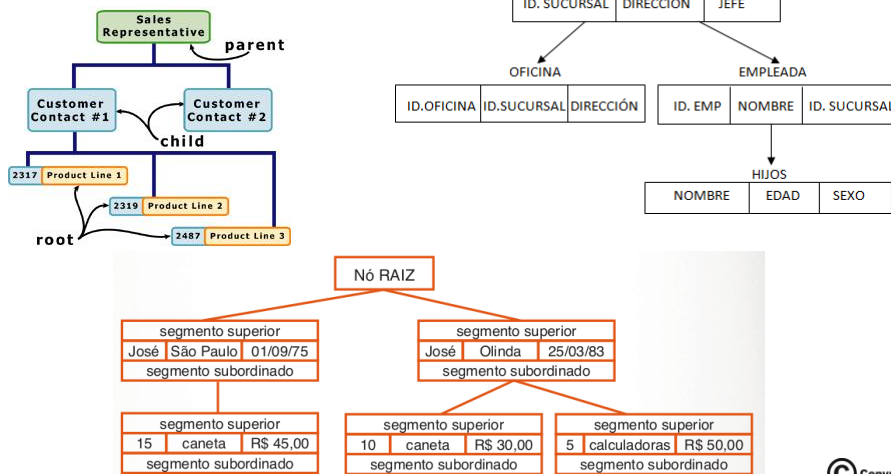
Somente os pais possuem os endereços dos dados dos filhos. Assim, para se localizar os dados de vendas, precisa-se saber o endereço do cliente. A programação utiliza os ponteiros de maneira explícita para poder associar os dados gravados. A mesma utilização de ponteiros é feita para as consultas.



UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

### Modelo hierárquico



UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

### Modelo de rede – sistemas CODASYL (Conference on Data Systems Languages)

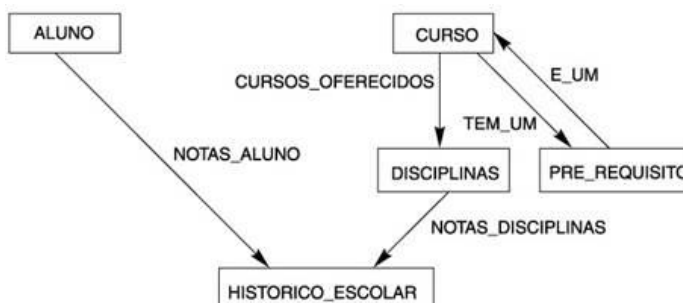
Inicialmente, foi apresentado no relatório do grupo de trabalho CODASYL, em 1971, sendo chamado modelo DBTG. Foi revisado em 1978 e 1981 e apresenta uma evolução do Modelo Hierárquico para resolver suas limitações. Permite a criação de estruturas mais flexíveis, em que os dados podem ser associados entre si, por exemplo: cliente/vendas; e vendas/ cliente. Assim, pode-se navegar entre essas estruturas considerando qualquer uma delas. Os dados são organizados em vários tipos de registros e o relacionamento dos registros pai e filho são explícitos. Em seu diagrama de estrutura de dados, os tipos de registros são organizados na forma de um grafo arbitrário, enquanto no diagrama de estrutura de árvore os tipos de registros são organizados na forma de árvore com raiz. A navegação ficou mais flexível, porém, ainda se utilizam os ponteiros explicitamente.



UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

### Modelo de rede – sistemas CODASYL



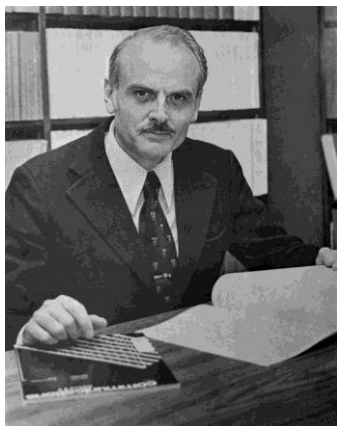
Fonte: Elmasri (2005).



UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

### Modelo Relacional



Edgar Frank "Ted" Codd (19 de agosto de 1923 - 18 de abril de 2003) foi um cientista da computação inglês que, ao trabalhar para a IBM, inventou o modelo relacional para gerenciamento de banco de dados, a base teórica para bancos de dados relacionais e sistemas de gerenciamento de banco de dados relacionais.

Ele fez outras contribuições valiosas para a ciência da computação, mas o modelo relacional, uma teoria geral muito influente de gerenciamento de dados, continua a ser a conquista mais mencionada, analisada e celebrada.

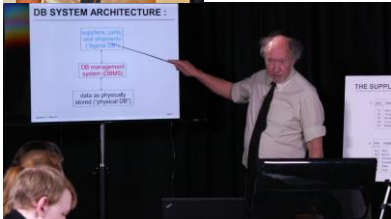
**Dr. Edgar Frank Codd, o pai do Modelo Relacional**



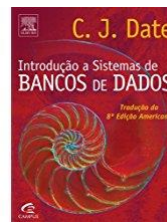
UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

### Modelo Relacional



Chris Date Em 1966, obteve o mestrado em Cambridge e, em 1967, ingressou na IBM Hursley (Reino Unido) como instrutor de programação de computadores. Entre 1969 e 1974, foi instrutor principal no programa de educação europeia da IBM.



**Dr. Christopher J. Date (C. J. Date), o implementador do Modelo Relacional**



UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

### Modelo Relacional

Foi introduzido por Codd (1970); sua utilização comercial, no entanto, se deu por volta do ano de 1978. Comparando-o com o modelo de dados hierárquico e o modelo de redes, é o mais simples, uma vez que apresenta uma estrutura de dados mais uniforme e menos física; é também o mais formalizado, pois foi inteiramente baseado na teoria dos conjuntos.

Possui como características:

- ✓ Consiste em uma coleção de tabelas, cada uma das quais associadas a um único nome. A tabela consiste em vários atributos. Para cada atributo há um conjunto de valores permitidos, chamados de domínios desse atributo.
- ✓ Base de dados visualizada como um conjunto de tabelas, cada uma representando uma relação.
- ✓ Relacionamentos representados por valores de dados; simetria nas consultas.
- ✓ Várias linguagens definidas (álgebra relacional, DDL e DML em SQL).
- ✓ É uma desvantagem do modelo relacional a representação não natural de objetos complexos.
- ✓ É um conjunto de dados armazenados em tabelas bidimensionais que possuem linhas e colunas. Tem como características permitir a criação de estruturas de dados organizadas em tabelas que possuem relacionamento entre si. Também possuem uma linguagem nativa para utilização do banco de dados SQL.



## Unidade 1 - Introdução

### Modelo Relacional

A fundamentação conceitual do Modelo Relacional (MR) deu origem a uma reformulação na mentalidade computadorizada comercial, impulsionando o ambiente de pesquisas em sistemas de armazenamento e tratamento de dados, capacidades de processamento de hardware e proposição de modelos abstratos para representação do conhecimento, como os modelos semânticos de dados.

Entretanto, novas aplicações, limitadas por restrições impostas pelo MR, fizeram surgir necessidades de bancos de dados que suportassem novos tipos de dados. “Essas novas aplicações de bancos de dados não foram consideradas nos anos 70, quando muitos dos sistemas de bancos de dados comerciais foram projetados” (SILBERSCHATZ, 1999, p.250). Exemplos: Sistemas de Suporte à Decisão; CAD; CAM; CASE – Engenharia de Software Apoiada por Computador; Bancos de Dados Espaciais; Banco de Dados Multimídia; Bancos de Dados Móveis (*Mobile Database*); Resgate (*Retrieval*) de Informações; *Office Information System* (OIS) (Resgates de Informações Distribuídas); Banco de Dados Hipertexto.



## Unidade 1 - Introdução

### Modelo Relacional

CORRENTISTA

Cod_corr	Nome	Rua	Cidade
----------	------	-----	--------

CORRENTISTA-CONTA

Cod_corr	Nro_conta
----------	-----------

CONTA

Nro_conta	Saldo
-----------	-------

CORRENTISTA

Cod_corr	Nome	Rua	Cidade
15	José	Figueiras	Campinas
21	João	Laranjeiras	Campinas
37	Antônio	Ipê	São Paulo

CONTA

Nro_conta	Saldo
900	55
556	1000
647	5366
801	10533

CORRENTISTA-CONTA

Cod_corr	Nro_conta
15	900
21	556
21	647
37	647
37	801



## Unidade 1 - Introdução

### Modelo objeto-relacional

Também designado como Modelo de Dados Relacional Estendido (MRE) e, no original, é RM/T, tem como finalidade ser mais representativo em semântica e construções de modelagens do que os tradicionais. Isso porque, além de fixar as características da modelagem tradicional, os modelos relacionais estendidos incorporaram as características do modelo orientado a objetos também.

A seguir, apresentamos algumas diferenças imediatas identificadas, conforme compara Date (2004), entre os modelos relacionais e os estendidos:

O MRE não diferencia entidades e relacionamentos como o MR tradicional.

O MRE define mais precisamente os aspectos estruturais e de integridade a serem utilizados do que o MR tradicional.

O RM/T possui as suas próprias definições e operações a serem utilizadas, mas utiliza também as operações definidas e utilizadas pelo MR.



## Unidade 1 - Introdução

### Modelo orientado a objeto

Segundo Silberschatz (1999, p.7), esses modelos:

[...] são usados na descrição de dados no nível lógico e de visões. São caracterizados por dispor de recursos de estruturação bem mais flexíveis e por viabilizar a especificação explícita das restrições dos dados. Existem vários modelos nessa categoria, e outros ainda estão por surgir. Alguns são amplamente conhecidos, como: Modelo Entidade-Relacionamento; Modelo Orientado a Objetos; Modelo Semântico de Dados; e Modelo Funcional de Dados.

Desses modelos, somente interessa ao escopo deste item conceitual o Modelo Orientado a Objetos.

### Modelo de dados orientado a objetos

Segundo Silberschatz (1999, p.8), esses modelos têm a seguinte caracterização:

- ✓ tem por base um conjunto de objetos;
- ✓ um objeto contém valores armazenados em variáveis de instâncias dentro do objeto;
- ✓ um objeto contém um conjunto de códigos que o operam, chamados métodos (permite que clientes solicitem serviços a um objeto por meio de uma mensagem).



## Unidade 1 - Introdução

### Banco de Dados Distribuídos

Um sistema de banco de dados distribuídos consiste num conjunto de nós, cada qual mantendo um banco de dados local. Cada nó é capaz de realizar transações locais, além da possibilidade de executar transações globais. Podemos citar inúmeras vantagens da utilização de bancos de dados distribuídos, entre elas: o particionamento dos dados, o aumento da confiabilidade e disponibilidade, a tolerância a falhas, a redundância controlada e a aceleração significativa no processamento de consultas. Entretanto, também podemos observar algumas desvantagens: custo do desenvolvimento, gerenciamento das tecnologias envolvidas e, principalmente, aumento da complexidade requerida para assegurar a coordenação entre os nós envolvidos (KORTH & SILBERSCHATZ, 1999).





## Unidade 1 - Introdução

### Introdução à Banco de Dados:

- É necessário organizar o negócio antes de se realizar qualquer automatização.
- Isso se dá por meio de um Planejamento Estratégico.



## Unidade 1 - Introdução

### Planejamento Estratégico Organizacional.

- Objetivos Estratégicos.
  - Macro-metas
    - Metas.
  - Macro-ações
    - Ações.



## Unidade 1 - Introdução

Planejamento Estratégico Organizacional evita que:

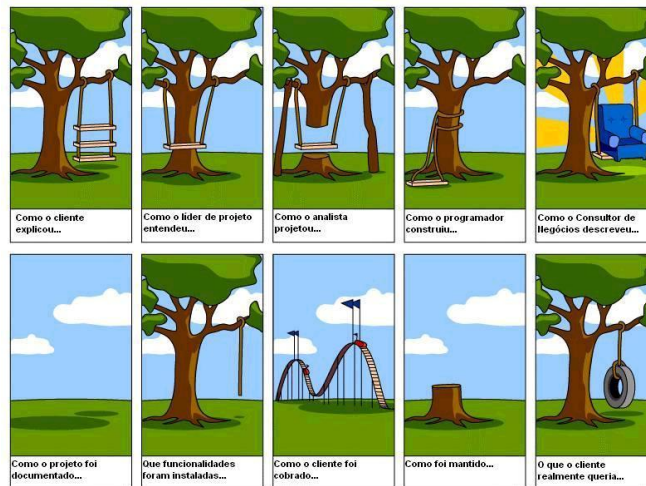
- Haja diversas linguagens na organização.
- Os interesses sejam conflitantes.
- Haja desperdício de recursos.
- Gere muito calor para realizar um trabalho.
- Haja confusão nas ações, metas, objetivos, táticas e estratégias.
- Vazio e interferência de poder.
- Haja ausência de definição de objetivos.
- Haja diversos protocolos na organização.
- Os setores tenham indefinição de suas obrigações.
- O cliente fique sem atenção devida.
- Haja ausência de método, processo, ciclo de vida, técnicas e ferramentas.



UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

PEO evita que:



<http://blog.mundopm.com.br/wp-content/uploads/2012/05/Escopo-do-Projeto.jpg>



UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

Planejamento Estratégico Organizacional:

- Razão de tudo – Ciclo PDCA.

Criado por Walter A. Shewhart - 1940.

Usado por Willian E. Deming no Japão pós 2ª guerra mundial.

Revolucionou a administração.

Revolucionou a gestão.

Revolucionou a engenharia.



<http://logisticaatual.files.wordpress.com/2010/06/00361.jpg>

UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 1 - Introdução

Planejamento Estratégico Organizacional:

- Estratégia: escolhas.
- Tática: como desenvolver e implantar as escolhas estratégicas.
- Operação: realização das táticas.

[http://tangrammarketing.com.br/wp-content/uploads/2013/09/Estrategia\\_capa.jpeg](http://tangrammarketing.com.br/wp-content/uploads/2013/09/Estrategia_capa.jpeg)

UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 1 - Introdução

Planejamento Estratégico Organizacional:

- Requisitos de alto nível.

Ética

Agilidade

Integração

Interação

Parceria

GI



[http://h4.ggpht.com/-NqYh0AfcDg/TD\\_ralOH11/AAAAAAAAAE/ovXtH8sn1Bk/image%25255B4%25255D.png?imgmax=800](http://h4.ggpht.com/-NqYh0AfcDg/TD_ralOH11/AAAAAAAAAE/ovXtH8sn1Bk/image%25255B4%25255D.png?imgmax=800)

© Copyright  
Todos os Direitos Reservados

UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

Planejamento Estratégico Organizacional:

- Gestão da Informação: integração de todas as informações da organização para fins de otimizar as escolhas estratégicas, táticas e operações.

© Copyright  
Todos os Direitos Reservados

UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

### Planejamento Estratégico Organizacional:

- AGREGAR VALOR AO NEGÓCIO.
- OFERECER VANTAGEM COMPETITIVA.

## Unidade 1 - Introdução

### Planejamento Estratégico Organizacional:

- Missão.
- Visão de Futuro.
- Valores.
- Código de ética.
- Desenvolvimento:
  - Dimensões de análise.
  - Objetivos Estratégicos.
  - Macro-metas
  - Metas.
  - Macro-ações.
  - Ações.
  - Projetos
- Mapa de Rotas Estratégicas.
- Mapa e Rotas Tecnológicas.



<http://www.sbcoaching.com.br/blog/wp-content/uploads/2013/11/mulher-binoculos-visao.jpg>

## Unidade 1 - Introdução

Planejamento Estratégico da TI.

O alinhamento estratégico do projeto com as diretrizes organizacionais.

-Alinhado às estratégias de negócio.

Alinhado ao PE.



UCB – Banco de Dados - Prof. Milton

## Unidade 1 - Introdução

Tipos de Sistemas: operacionais, táticos e estratégicos.

Sistemas de Informação Executiva.

Sistemas de Informação Gerencial.

Sistemas de Informação Gerencial.



UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Sistemas de Bancos de Dados

Um sistema de banco de dados é basicamente um sistema computadorizado de armazenamento de registros; isto é, um sistema computadorizado cujo propósito geral é armazenar informações e permitir ao usuário buscar e atualizar essas informações quando solicitado. As informações em questão podem ser qualquer coisa que tenha significado para o indivíduo ou para a organização a que o sistema deve servir – em outras palavras, tudo o que seja necessário para auxiliar no processo geral de tomada de decisões de negócios desse indivíduo ou dessa organização (DATE, 2004, p.4).



UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Ambiente com Banco de Dados

Date (2004) apresenta a visão de um ambiente com banco de dados.



Fonte: Date (2004).



UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

O SGBD ou DBMS (*Data Base Management System*) é o *software* que gerencia os dados do Banco de Dados. Ele oferece recursos para gravar, alterar e retirar dados, segurança e integridade, facilita a recuperação, entre outros.

Pode-se definir um SGBD como:

[...] o software que trata de todo acesso ao banco de dados e possui funções que incluem o suporte a pelo menos os itens: definição de dados; manipulação de dados (*Data Manipulation Language* – DML); otimização e execução; segurança e integridade dos dados; recuperação e concorrência de dados; dicionário de dados (metadados, descritores, repositórios de dados ou enciclopédias de dados); e desempenho (DATE, 2004, p. 37).



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Tipos SGBD

Existem vários tipos de SGBD no mercado:

- Sistema de Gerenciador de Banco de Dados Relacional (SGBDR)

Temos como exemplo de SGBDR o DB2 da IBM, Microsoft SQL Server, Informix, PostgreSQL e MySQL, entre outros.

- Sistema de Gerenciador de Banco de Dados Orientado a Objetos (SGBDOO)

Um SGBDOO é um sistema de banco de dados que segue a Modelagem Orientada a Objetos.

- Sistema de Gerenciador de Banco de Dados Objeto Relacional (SGBDOR)

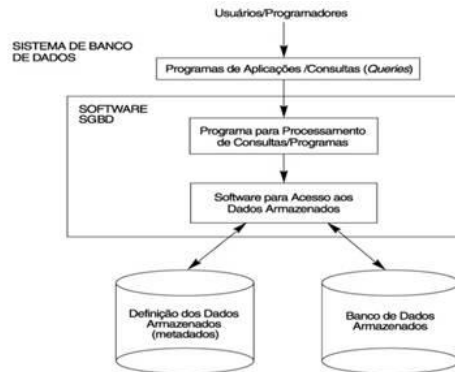
Também conhecido como Modelo de Dados Relacional Estendido (MRE) não diferencia entidades e relacionamentos como o Modelo Relacional tradicional. Ele define mais precisamente aspectos estruturais e de integridade. Ele possui suas próprias definições e operações; entretanto, utiliza também operações da Modelagem Relacional.





## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Sistema Gerenciador de Banco de Dados (SGBD):



Fonte: Elmasri (2005).

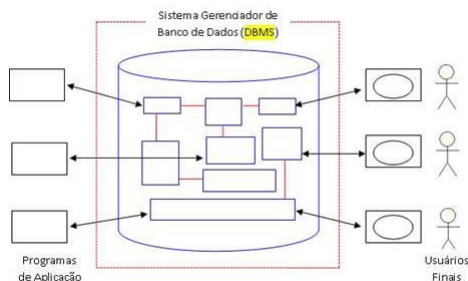
Copyright

UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Componentes Principais de um Banco de Dados

“Um SBD é composto de dados, *hardware*, *software* e usuários” (DATE, 2000, p.5). Veja a Figura 2.2.



Fonte: Date (2004).

Copyright

UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

Pode-se entender que um SGBD é um produto de *software* complexo e deve ser tratado como uma poderosa tecnologia de armazenamento e gerenciamento de dados.

Outros *softwares* dos SBD são: ferramentas para desenvolvimento de aplicações, auxílio em projetos, geradores de relatórios, gerenciador de transações ou monitor de TP (*Transaction Processing* – Processamento de Transações) e gerenciador de comunicações.



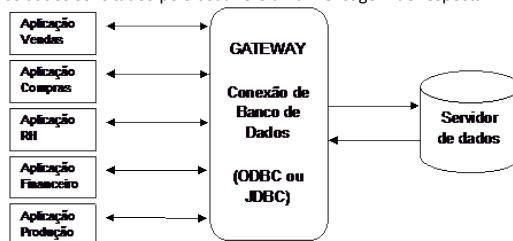
UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Conexão de Banco de Dados

Os passos genéricos de acesso ao banco de dados são identificados como:

- ✓ O usuário clica em uma funcionalidade de sua aplicação.
- ✓ A aplicação de usuário emite uma solicitação ao banco de dados.
- ✓ O *Gateway* intercepta e traduz a solicitação e endereça ao SGBD e ao banco de dados correspondente.
- ✓ O SGBD recebe a solicitação, identifica o banco de dados, identifica o usuário solicitante e valida a autorização de acesso segundo os critérios de acesso.
- ✓ O SGBD inspeciona os esquemas externos (ou sub-esquemas) relacionados para aquele usuário, os mapeamentos entre os três níveis e a definição da estrutura de armazenamento.
- ✓ O SGBD realiza as operações solicitadas no BD.
- ✓ O SGBD retorna os dados solicitados pelo usuário e uma mensagem de resposta.



UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Tarefas e Classificação de um SGBD

São tarefas de um SGBD:

- ✓ Interação com o sistema de arquivos do sistema operacional.
- ✓ Cumprimento da integridade.
- ✓ Cumprimento da segurança.
- ✓ Cópias de segurança (*backup*) e recuperação.
- ✓ Controle da concorrência.

Os SGBDs se classificam em:

- ✓ **Modelos de dados** – rede, hierárquico, relacional ou orientado a objeto.
- ✓ **Número de usuários** – monousuário ou multiusuário.
- ✓ **Localização** – centralizado ou distribuído.
- ✓ **Homogeneidade** – homogêneo ou heterogêneos.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

São exemplos de SGBD\_R (SGBD Relacional):

MSSQL Server.

SQLAnyWhere.

Oracle, DB2.

PostgreSQL.

MS Access.

MySQL.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### **Dados**

São os valores fisicamente armazenados no banco de dados e informação é o significado desse valor para um determinado usuário.

Os dados tanto podem estar armazenados em um único banco de dados como podem estar divididos em diversos bancos de dados distintos. Em um banco de dados, os dados deverão ser integrados e compartilhados. Isso representa a maior vantagem dos sistemas de banco de dados de ambientes grandes e pelo menos a integração também pode ser significativa em ambientes pequenos. Dessa forma, os dados aparecem em um mesmo banco de dados:

- ✓ **Integrados** – Um banco de dados pode ser imaginado como a unificação de diversos arquivos de dados que, de outra forma, seriam distintos. Tal integração elimina, total ou parcialmente, qualquer redundância entre os mesmos.
- ✓ **Compartilhados** – Parcelas isoladas de dados podem ser compartilhadas por diversos usuários em um banco de dados, no sentido de que todos os usuários possam ter acesso à mesma parcela de dados e usá-los com finalidades diferentes no mesmo momento ("acesso concorrente").



## Unidade 2 - Sistema Gerenciador de Banco de Dados

Modelos, Esquemas e Instâncias, Arquitetura.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Compartilhamento e Integração



UCB – Banco de Dados - Prof. Milton

© Copyright  
Todos os Direitos Reservados

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Hardware

Compõe-se dos volumes de memória secundária (principalmente discos rígidos), nos quais reside o banco de dados, juntamente com os processadores, memória principal, dispositivos associados de entrada/saída, dispositivos de controle, canais de entrada/saída, e assim por diante.

### Software

Já o componente de *software* diz respeito à camada entre o banco de dados físico e os usuários do sistema, conhecida como Gerenciador ou Servidor do Banco de Dados ou Sistema de Gerenciamento de Banco de Dados ou, ainda, Sistema Gerenciador de Banco de Dados (SGBD).

### Usuários

Existem cinco grandes classes: Analista/Projetista, Programador de Aplicações, Usuário Final, Administrador de Dados (DA – *Data Administrator*), Administrador do Banco de Dados (DBA – *Database Administrator*).

UCB – Banco de Dados - Prof. Milton

© Copyright  
Todos os Direitos Reservados

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Por que Banco de Dados?

São inúmeras as vantagens do sistema de banco de dados em relação aos métodos tradicionais, baseados em papéis e arquivos. Entre elas, destacamos: é compacto, é rápido, implica menos trabalho braçal e tem fluxo corrente. Essas vantagens são mais significativas em ambientes de usuários múltiplos, em que o banco de dados é maior e mais complexo do que o de usuário único, pois o sistema de banco de dados proporciona à empresa o controle centralizado de seus dados operacionais, o qual pode:

- ✓ reduzir as redundâncias;
- ✓ evitar (até certo ponto) a inconsistência;
- ✓ compartilhar os dados;
- ✓ reforçar os padrões;
- ✓ aplicar restrições de segurança;
- ✓ manter a integridade; e
- ✓ equilibrar as necessidades conflitantes.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Gerenciador de Comunicações de Dados

Possibilita a comunicação entre usuários e o banco de dados persistente. As requisições a banco de dados de usuário são transmitidas da sua estação de trabalho para alguma aplicação *on-line* e daí até o SGBD sob a forma de *mensagens de comunicação*. De modo semelhante, as respostas do SGBD e da aplicação *on-line* para esta estação de trabalho também são transmitidas sob a forma de mensagens. Essas transmissões são controladas por outro componente de *software* denominado gerenciador de comunicações de dados (gerenciador DC).

### Finalidade do SGBD

O SGBD tem como propósito armazenar dados de diversos tipos, de maneira que possam ser acessados segundo normas de segurança e compartilhamento previamente acordadas entre os utilizadores e os operadores. Assim, mesmo tendo grandes volumes de dados, esse acesso é facilitado aos usuários.

O SGBD deve possibilitar que outros sistemas possam acessar seus dados, como as aplicações de usuário escritas em linguagens de programação (Cobol, Java, Delphi, Visual Basic C++, entre outras). Isso é possível por meio de outros *softwares*, denominados *Gateway* (protocolos de comunicação como ODBC ou JDBC), que permitem a conexão da aplicação com o banco de dados.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Arquiteturas

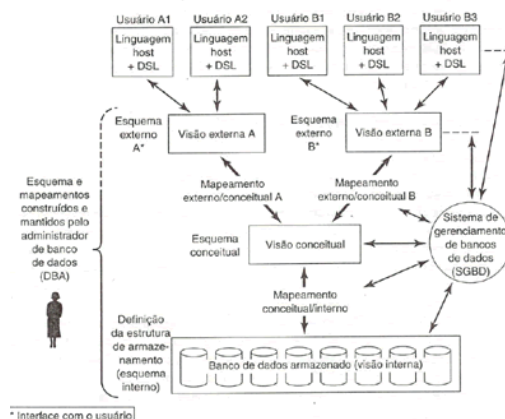
A arquitetura a ser esclarecida aqui é basicamente idêntica à apresentada pelo ANSI/SPARC (*Study Group on Data Base Management System*), mas sem seguir todo o seu rigor.

### Níveis de Arquitetura

A arquitetura ANSI/SPARC divide-se em três níveis: interno, conceitual e externo (Figura 2.5).

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Arquitetura Detalhada do Sistema



Fonte: Date (2004).

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### O SGBD

Segundo Date (2004), o SGBD é o *software* que trata de todo acesso ao banco de dados, o que ocorre da seguinte forma:

- ✓ Um usuário faz um pedido de acesso usando uma determinada sublinguagem de dados (geralmente a SQL).
- ✓ O SGBD intercepta o pedido e o analisa.
- ✓ O SGBD, por sua vez, inspeciona o esquema externo (ou as versões objeto desse esquema) para esse usuário, o mapeamento externo/conceitual correspondente, o esquema conceitual, o mapeamento conceitual/interno e a definição do banco de dados armazenado.
- ✓ O SGBD executa as operações necessárias sobre o banco de dados armazenado.

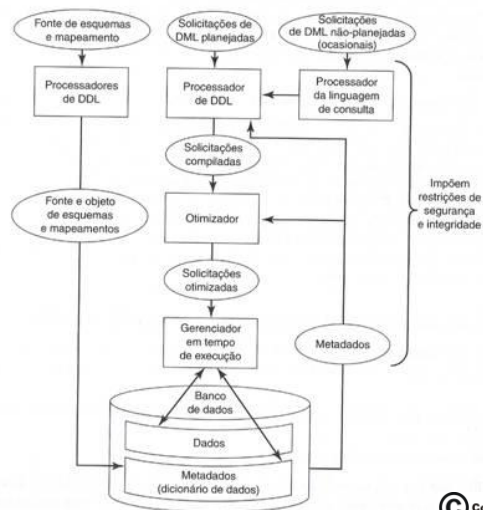


UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### O SGBD

Detalhamento dessas funções.



Fonte: Date (2004).

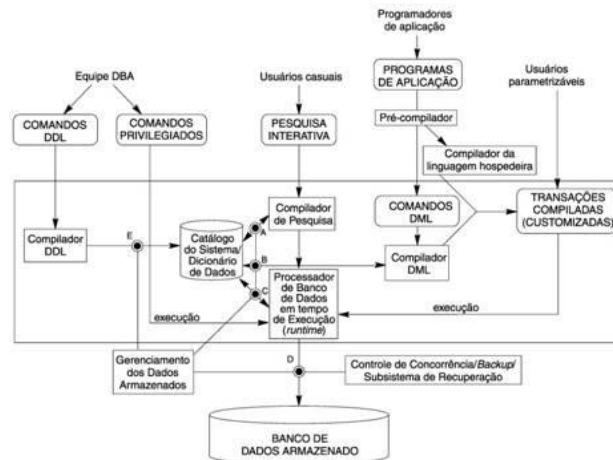


UCB – Banco de Dados - Prof. Milton



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Módulos Componentes de um SGBD e suas Interações



Fonte: Elmasri (2005).

UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 2 - Sistema Gerenciador de Banco de Dados

Para que o SGBD trate de todo acesso ao banco de dados, ele provê as seguintes funções:

**Definição de dados** – Deve ser capaz de aceitar definições de dados em formato fonte e convertê-lo para o formato de objeto apropriado. Para isso, um SGBD deve incluir um processador ou compilador de DDL. Além disso, deve ser capaz de entender os objetos.

**Manipulação de dados** – Deve ser capaz de lidar com requisições do tipo busca, atualização ou exclusão. Inclui um processamento ou compilador DML e, em geral, essas requisições podem ser:

**Planejadas** – Aquelas previstas pelo DBA, onde geralmente são feitos ajustes para atender com um desempenho melhor. Exemplo: requisições operacionais.

**Não-planejadas** – Também conhecidas como consulta *ad-hoc*, são aquelas não planejadas e, por isso, possuem um desempenho geralmente um pouco mais baixo.

**Otimização e execução** – As requisições DML são processadas por um otimizador que determina a maneira mais eficiente de implementar a requisição. Em seguida, são executadas sob controle do gerenciador em tempo de execução.

**Segurança e integridade dos dados** – Deve monitorar as requisições de usuário e rejeitar toda tentativa de violar as restrições de segurança ou integridade.

**Recuperação de dado e concorrência** – Deve-se manter um gerenciador de transação para impor certos controles de recuperação e concorrência.

**Dicionário de dados** – O SGBD deve fornecer uma função de dicionário de dados que pode ser considerado como um banco de dados para o sistema ou um banco isolado. Nele, estão contidos os metadados, que são “dados sobre dados”.

**Desempenho** – O SGBD deve realizar todas as funções acima descritas de forma mais eficiente possível.

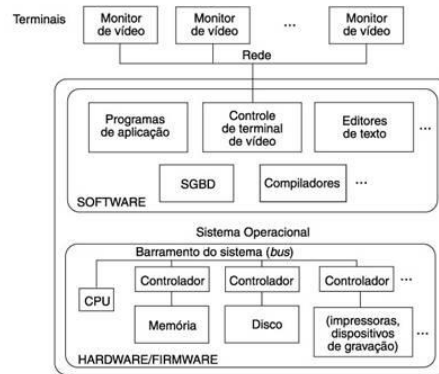
UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Arquitetura Física Centralizada

Observe o esquema abaixo, que sintetiza a ideia de uma arquitetura centralizada:



Fonte: Elmasri (2005).

UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Arquitetura Cliente/Servidor

Segundo Date (2004), o principal objetivo dessa arquitetura é o de fornecer suporte ao desenvolvimento e à execução de aplicações de banco de dados e, sob um ponto de vista de alto nível. Essa arquitetura está dividida em duas partes, sendo uma delas o servidor (*back end*) e a outra o cliente (*front end*):

- ✓ O servidor é o próprio SGBD. Ele admite todas as funções básicas de um SGBD. Em outras palavras, o termo servidor é apenas um outro termo para SGBD.
- ✓ Os clientes são as diversas aplicações executadas com base no SGBD, que podem ser escritas por usuários ou aplicações internas (*built-in*, ou seja, aplicações fornecidas pelo fabricante do SGBD).

UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Arquitetura Lógica Cliente/Servidor de Duas Camadas



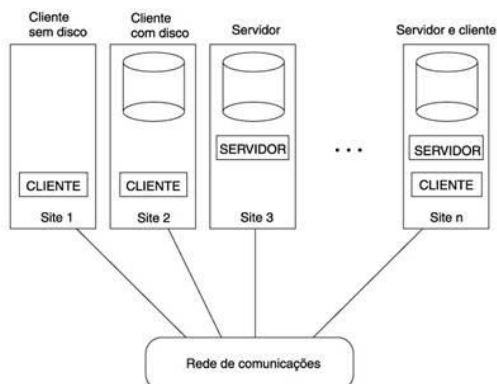
Fonte: Elmasri (2005).



UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Arquitetura Física cliente/servidor de duas Camadas



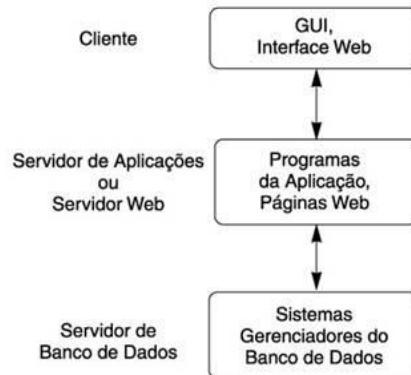
Fonte: Elmasri (2005).



UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Arquitetura Lógica Cliente/Servidor de três Camadas



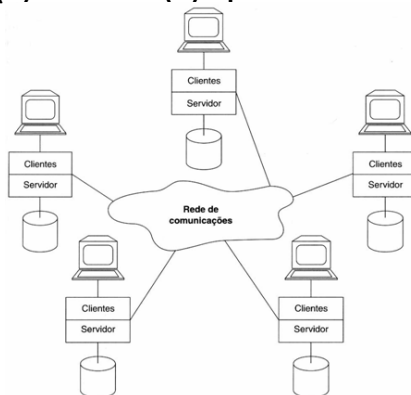
Fonte: Elmasri (2005).



UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Arquitetura Lógica Cliente/Servidor de três Camadas - Cada máquina pode executar tanto o(s) cliente(s) quanto o servidor.



Fonte: Elmasri (2005).



UCB – Banco de Dados - Prof. Milton

## Unidade 2 - Sistema Gerenciador de Banco de Dados

Outro aspecto importante, quando tratamos de SGBD, é pensarmos nos **utilitários**, programas projetados para auxiliar o DBA com diversas tarefas administrativas. Segundo Date (2004) esses são alguns exemplos de utilitários:

- ✓ **Rotinas de carga** – Cria a versão inicial do banco de dados a partir de um ou mais arquivos do sistema operacional.
- ✓ **Rotinas de descarga/recarga (ou *dump/restore*)** – Descarrega o banco de dados ou parte dele para o meio de armazenamento de *back-up* e recarrega dados dessas cópias de *back-up*.
- ✓ **Rotinas de reorganização** – Arruma novamente os dados no banco de dados armazenados por vários motivos, entre eles desempenho.
- ✓ **Rotinas de estatísticas** – Calcula estatísticas de desempenho (tamanhos de arquivos e distribuição de valores, contagens de E/S).
- ✓ **Rotinas de análise** – Analisa as estatísticas acima.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Profissionais de Banco de Dados

#### Usuários

Segundo Date (2004), existem cinco grandes classes de usuários de bancos de dados: analista/ projetista; programador de aplicações; usuário final; administrador de dados (DA – *Data Administrator*); administrador do banco de dados (DBA – *Database Administrator*).

- ✓ **Analista/Projetista**
- ✓ **Programador de Aplicações**
- ✓ **Usuário Final**
- ✓ ***Data Administrator* (DA)**



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Database Administrator (DBA)

Segundo Date (2004), o DBA é a pessoa – ou grupo de pessoas, gerentes e técnicos – responsável pelo controle geral do sistema em um nível técnico; também fornece o suporte técnico necessário para implementar essas decisões. Sua posição deve ser de alto nível gerencial, requerendo: um alto grau de capacitação técnica; capacidade de entender e interpretar as necessidades da empresa em termos de gerência executiva. Date (2004) afirma que o DBA desempenha o papel de fornecer os meios para que as decisões tomadas pelo DA sejam efetuadas, tendo as seguintes responsabilidades:

- ✓ **Definição do esquema conceitual** – Cabe ao DBA identificar quais entidades são de interesse da empresa e quais informações a respeito da mesma devem ser armazenadas. Essa atribuição normalmente é desempenhada durante o processo de desenvolvimento do projeto lógico de dados e, após a definição do mesmo, o DBA cria o esquema conceitual correspondente, usando a DDL conceitual. **Definição do esquema interno** – O DBA deve decidir como serão representados os dados no banco de dados armazenado. Esta atribuição normalmente é desempenhada durante a elaboração do projeto físico do banco de dados.
- ✓ **Contato com os usuários.**
- ✓ **Definição das restrições de segurança e integridade.**
- ✓ **Definição das normas de descarga e recarga.**
- ✓ **Monitoramento do desempenho e resposta a requisitos de mudança.**



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Structured Query Language (SQL)

SQL (Linguagem de Consulta Estruturada) é a linguagem suportada pelos Sistemas Gerenciadores de Bancos de Dados. Ela é composta por uma parte DDL e outra DML:

- ✓ DDL (Linguagem de Definição de Dados) é a parte da linguagem SQL que permite a criação da estrutura do BD, através de instruções como: *Create Database, Create Table, Create Index, Create Constraint*, entre outras.
- ✓ DML (Linguagem de Manipulação de Dados) é uma parte da linguagem SQL que trabalha com os dados no BD, através de instruções como *Select, Insert, Update, Delete*.

A linguagem SQL pode ser definida como sendo uma “linguagem de consulta”, sendo utilizada em Sistemas Gerenciadores de Bancos de Dados Relacionais. Contudo, ela avança em outras direções além de mera consulta ao banco de dados, servindo também como recurso para a definição da estrutura de dados, para a modificação de dados e para a especificação de restrições de segurança.

A SQL é conhecida hoje, além da linguagem padrão de banco de dados, como sendo a linguagem mais utilizada no mercado. Ela usa uma combinação de construtores em álgebra e cálculo relacional.

O modelo relacional é o modelo mais usado pelos estudiosos da área de informática e tem por base uma coleção de tabelas, onde cada linha desta tabela é denominada tupla. O usuário de um sistema de banco de dados pode desejar consultar essas tabelas, inserindo novas tuplas, eliminando, atualizando ou modificando-as.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Structured Query Language (SQL)

O cálculo relacional é a base para uma linguagem relacional de consulta. A álgebra relacional define as operações básicas usadas dentro das linguagens relacionais de consulta.

Segundo Silberschatz (1999, p.109), a SQL usa uma combinação de construtores em álgebra e cálculo relacional. A SQL tem representado o padrão para linguagens de banco de dados relacionais, sendo a linguagem comercial mais utilizada no mercado. Outras linguagens utilizadas são: QBE e QUEL.

A versão original foi desenvolvida pela IBM no Laboratório de Pesquisa de San José. Foi originalmente chamada de Sequel e implementada como parte do projeto do System\_R no início dos anos de 1970. A linguagem foi evoluindo e seu nome foi alterado para SQL.

Em 1986, o American National Standards Institute (ANSI) publicou o padrão SQL e, em 1987, foi a vez da International Standards Organization (ISO), a Systems Application Architecture Database Interface (SAA-SQL). Uma extensão para o padrão SQL, a SQL-89 foi publicada em 1989. Revisões foram realizadas em 1992 (ISO/IEC 9075:1992 ou SQL-92), em 1999 (ISO/IEC 9075:1999 ou SQL-1999 ou SQL3), e em 2003 (ISO/IEC 9075:2003 ou SQL:2003), incluindo extensões para a XML (Extensible Markup Language). -SQL:2006: Lançada em 2006, não inclui mudanças significativas para as funções e comandos SQL. Contempla basicamente a interação entre SQL e XML. O nome formal do padrão SQL é ISO/IEC 9075 Database Language SQL.

O padrão SQL: 2003 está dividido nas seguintes partes:

- ✓ ISO/IEC 9075-1 Framework (SQL/Framework)
- ✓ ISO/IEC 9075-2 Foundation (SQL/Foundation)
- ✓ ISO/IEC 9075-3 Call Level Interface (SQL/CLI)
- ✓ ISO/IEC 9075-4 Persistent Stored Modules (SQL/PSM)
- ✓ ISO/IEC 9075-9 Management of External Data (SQL/MED)
- ✓ ISO/IEC 9075-10 Object Language Bindings (SQL/OLB)
- ✓ ISO/IEC 9075-11 Information and Definition Schemas (SQL/Schemata)
- ✓ ISO/IEC 9075-13 Routines and Types using the Java Language (SQL/JRT)
- ✓ ISO/IEC 9075-14 XML-related specifications (SQL/XML)

A maioria dos SGBD Relacionais do mercado utiliza a linguagem SQL, como o SQL/DS e DB2 da IBM, Oracle e Oracle/RDB da Oracle Corporation, SQLAnywhere, PostgreSQL, MySQL, Microsoft SQLServer, entre outros.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### Data Definition Language (DDL)

Corresponde à LDD (Linguagem de Definição de Dados), tem comandos para criação, alteração e exclusão de objetos de banco de dados, por meio de instruções como Create, Alter e Drop, como: banco de dados, tabelas, índices, restrições de integridades, visões, entre outros. Cada SGBD possui sua própria biblioteca de comandos, baseadas no padrão SQL. A seguir, são apresentados alguns comandos baseados no SGBD MySQL.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### **Data Definition Language (DDL)**

A DDL (Linguagem de Definição de Dados) possui diversos comandos:

- ✓ *CREATE DATABASE* - Para criar bancos de dados.
- ✓ *CREATE E ALTER TABLE* - Para definição de tabelas.
- ✓ *CREATE VIEW* - Para definição de visões do banco de dados.
- ✓ *GRANT e REVOKE* - Para para definição de permissões.

Tais comando podem definir e estruturar os objetos do banco de dados e suas relações e regras de integridade. A DDL inclui, ainda, comandos para a especificação de transações (*begin and transaction*). Algumas implementações também permitem explicitar bloqueios de dados para controle de concorrência (*locks*), sendo capaz de coordenar o compartilhamento de dados entre usuários concorrentes, sem interferir na ação de cada um deles.

Por meio de comandos SQL, os usuários podem montar consultas poderosas sem a necessidade da criação de um programa.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### **Criar bancos de dados**

```
CREATE DATABASE [IF NOT EXISTS] Nome_do_banco;
```

### **Criar Tabela**

```
CREATE TABLE PRODUTO ( ID_PRODUTO INT NOT NULL AUTO_INCREMENT, NOME VARCHAR (100), DATA_FAB DATE, DATA_VENC DATE, QTDE INT, PRIMARY KEY (ID_PRODUTO));
```

### **Criar View**

A visão pode ser vertical, quando o comando SELECT apresenta algumas colunas da tabela e todos os dados.

```
CREATE VIEW funcionario_v AS SELECT matricula, nome FROM funcionario;
```

Pode ser horizontal quando o SELECT apresenta alguns dados de todas as colunas – cláusula WHERE.

```
CREATE VIEW funcionario_v AS SELECT * FROM funcionario WHERE matricula > 100;
```

Pode ser mista quando o SELECT apresenta algumas colunas e algumas linhas da tabela – usando a cláusula WHERE.

```
CREATE VIEW funcionario_v AS SELECT matricula, nome FROM funcionario WHERE matricula > 100;
```

Já nas *materialized view* a coisa é diferente. Serve para guardar dados e efetuar cálculos.

### **Criar Índice**

```
CREATE UNIQUE INDEX I_NOME ON FUNCIONARIO (NOME);
```

### **Alterar tabela**

```
ALTER TABLE CLIENTE ADD TELEFONE number(9) not null AFTER ENDereco;
```

### **Excluir objetos**

```
DROP DATABASE NOME_BANCO; DROP TABLE CLIENTE; DROP INDEX I_CPF; DROP CONSTRAINT fk_Cliente_ID_Pedido; ALTER TABLE CLIENTE DROP COLUMN TELEFONE
```





## Unidade 2 - Sistema Gerenciador de Banco de Dados

### **Data Manipulation Language (DML)**

Corresponde à LMD (Linguagem de Manipulação de Dados), parte da linguagem SQL que manipula os dados no BD, por meio de instruções como Select, Insert, Update, Delete.

Segundo Korth & Silberschatz (1999), a estrutura básica de uma consulta em SQL consiste em três cláusulas: *select*, *from* e *where*.

A cláusula *select* corresponde à operação de projeção da álgebra relacional. Ela é usada para relacionar os atributos desejados no resultado de uma consulta.

A cláusula *from* corresponde à operação de produto cartesiano da álgebra relacional. O produto cartesiano permite combinar informações de duas relações quaisquer. A cláusula *from* associa as relações que serão pesquisadas durante a evolução de uma expressão.

A cláusula *where* corresponde à seleção do predicado (qualificação ou restrição em uma relação qualquer) da álgebra relacional. Ela consiste em um predicado envolvendo atributos da relação que aparece na cláusula *from*.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### **Data Manipulation Language (DML)**

O formato de uma consulta típica em SQL é:

```
select A1, A2, ..., An
from r1, r2,..., rm
where <condição>
```

Em síntese, pode-se dizer que a SQL forma um produto cartesiano das relações indicadas na cláusula *from*; executa uma seleção em álgebra relacional, usando o predicado da cláusula *where*; e projeta o resultado sobre os atributos da cláusula *select*.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### **Data Manipulation Language (DML)**

#### **Selecionar dados**

```
SELECT * FROM CLIENTE;
SELECT NOME, MATRICULA FROM FUNCIONARIO;
SELECT t1.NOME, t2.NOME_DEP FROM FUNCIONARIO t1, DEPENDENTE t2 WHERE t1.ID_FUNCIONARIO =
t2.ID_FUNCIONARIO;
SELECT FUNCIONARIO.NOME, DEPENDENTE.NOME_DEP FROM FUNCIONARIO, DEPENDENTE WHERE FUNCIONARIO.
ID_FUNCIONARIO = DEPENDENTE.ID_FUNCIONARIO and FUNCIONARIO.CIDADE = 'Rio de Janeiro';
SELECT NOME, SALARIO FROM FUNCIONARIO ORDER BY NOME DESC;
SELECT NOME, MAX(SALARIO) FROM USUARIO GROUP BY NOME HAVING MAX(SALARIO)>10;
SELECT NOME, SALARIO INTO OUTFILE "c:/diretorio/consulta.txt" FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY
"" LINES TERMINATED BY "\n" FROM FUNCIONARIO;
SELECT DISTINCT CIDADE FROM CANDIDATO; SELECT NOME AS APROVADOS FROM CANDIDATO;
```

#### **Cláusulas**

As cláusulas são condições de modificação utilizadas para definir os dados que deseja selecionar ou modificar em uma consulta.

FROM – Utilizada para especificar a tabela que se vai selecionar os registros.

WHERE – Utilizada para especificar as condições que devem reunir os registros que serão selecionados.

GROUP BY – Utilizada para separar os registros selecionados em grupos específicos.

HAVING – Utilizada para expressar a condição que deve satisfazer cada grupo.

ORDER BY – Utilizada para ordenar os registros selecionados com uma ordem específica.

DISTINCT – Utilizada para selecionar dados sem repetição.

UNION – combina os resultados de duas consultas SQL em uma única tabela para todas as linhas correspondentes.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### **Data Manipulation Language (DML)**

#### **Inserir dados**

```
INSERT INTO FUNCIONARIO (ID, NOME) VALUES (1,
'Maria');
```

#### **Atualizar dados**

```
UPDATE FUNCIONARIO SET NOME = 'Maria' WHERE
ID = 10;
```

#### **Excluir dados**

```
DELETE FROM FUNCIONARIO WHERE ID = 30;
```



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### *Data Manipulation Language (DML)*

#### **Operadores de comparação**

#### **Operadores de lógicos**

AND - e  
OR - ou  
NOT - não

SELECT nome FROM aluno WHERE cidade='Rio de Janeiro' AND matricula='444';

#### **Operadores Lógicos**

BETWEEN – Utilizado para especificar valores dentro de um intervalo fechado.

LIKE – Utilizado na comparação de um modelo e para especificar registros de um banco de dados. "Like" + extensão % significa buscar todos resultados com o mesmo início da extensão.

IN - Utilizado para verificar se o valor procurado está dentro de um « » a lista. Ex.: valor IN (1,2,3,4).

Operador	Descrição
<	Menor
>	Maior
<=	Menor ou igual
>=	Maior ou igual
=	Igual
<>	Diferente

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### *Data Manipulation Language (DML)*

#### **Funções de Agregação**

Usadas em cláusula SELECT que retorna um valor que se aplica a um grupo de registros.

AVG – Utilizada para calcular a média dos valores de um campo determinado.

COUNT – Utilizada para devolver o número de registros da seleção.

SUM – Utilizada para devolver a soma de todos os valores de um campo determinado.

MAX – Utilizada para devolver o valor mais alto de um campo especificado.

MIN – Utilizada para devolver o valor mais baixo de um campo especificado.

## Unidade 2 - Sistema Gerenciador de Banco de Dados

### ***SGBD que usam SQL***

Sybase Advantage Database Server	Microsoft Access
Apache Derby	Microsoft SQL Server
Caché	MySQL
DB2	Oracle
Firebird	PointBase Micro (banco de dados relacional implementado em Java)
HSQldb (BD implementado em Java)	PostgreSQL
IDMS (banco de dados hierárquico)	SQLite
IMS (banco de dados hierárquico)	LiteBase Mobile (dedicado à plataformas móveis como: Palm OS, Pocket PC, WinCE, Symbian)
Informix	Sybase Adaptive Server Enterprise
Ingres	Teradata (primeiro RDBMS com arquitetura paralela do mercado)
InterBase	



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### ***Data Control Language (DCL)***

Corresponde à DCL (Linguagem de Controle de Dados), parte da linguagem SQL que trabalha com as autorizações e revogações de acesso dos usuários aos dados no banco de dados, por meio de instruções como Grant, Revoke, Alter Password e Create Synonym.

O comando GRANT dá permissão a determinados usuários poderem operar sobre determinados recursos do banco de dados como views e tabelas e pelo comando REVOKE que remove essas permissões. Também possui os comandos Alter Password para alteração da senha de acesso ao banco e Create Synonym para criar apelidos para objetos, tais como: tabelas, *views*, storage procedure, entre outros.

Por exemplo, para dar acesso para Maria na *view*:

GRANT SELECT ON ESTOQUE\_V TO Maria; Para Maria visualizar os dados:

SELECT \* FROM ESTOQUE\_V; Para remover a permissão de Maria na *view*:

REVOKE SELECT ON ESTOQUE\_V TO Maria;



## Unidade 2 - Sistema Gerenciador de Banco de Dados

### **Data Transaction Language (DTL)**

Corresponde à LTD (Linguagem de Transação de Dados), parte da SQL que trata do início e do fim das transações e suas finalizações, por meio dos comandos Begin Work (ou Begin Transaction ou Start Transaction), Commit e Rollback.

### **Data Query Language (DQL)**

Corresponde à LCD (Linguagem de Consulta de Dados), parte da SQL que trata das consultas aos bancos de dados por meio do comando Select.

```
SELECT * FROM ALUNO;
```

### **Embedded DML (Incorporação DML)**

Comandos SQL incorporados são projetados para aplicação em linguagens de programação de uso geral como PL/I, Cobol, Pascal, Fortran e C.



## Unidade 2 - Sistema Gerenciador de Banco de Dados

Ambiente de banco de dados.

- Modelagem e projeto de BD.
- Uso de SGBD.
- Administração de BD.
- Administração de Dados.
- Segurança da Informação



## Características de Banco de Dados

### Segurança

Segundo Date (2004), significa proteger os dados contra usuários não autorizados e garantir que os usuários tenham permissão para fazer aquilo que estão tentando fazer. Os seguintes aspectos devem ser considerados quanto à segurança:

- ✓ **Aspectos legais, sociais e éticos** – Por exemplo, a pessoa que faz a requisição, digamos, quanto ao crédito de um cliente, tem direito legal à informação solicitada?
- ✓ **Controles físicos** – Por exemplo, a sala do computador do terminal é trancada ou protegida de algum modo?
- ✓ **Questões de política** – Por exemplo, como a empresa proprietária do sistema decide quem deve ter acesso a quê?
- ✓ **Problemas operacionais** – Por exemplo, se é usado um esquema de senha, como é conservado o segredo das próprias senhas e com que frequência elas são trocadas?
- ✓ **Controles de hardware** – Por exemplo, o servidor fornece recursos de segurança, tais como chaves de proteção de armazenamento ou um modo de operação protegido?
- ✓ **Suporte do sistema operacional** – Por exemplo, o sistema operacional sendo utilizado apaga o conteúdo da memória principal e dos arquivos de disco quando termina de trabalhar com eles – e o que dizer do *log* de recuperação?
- ✓ **Questões de interesse específico do próprio sistema de banco de dados** – Há um conceito de propriedade de dados?

A segurança de bancos de dados se relaciona ainda, diretamente, com aspectos como: controle de acesso; trilhas de auditoria; bancos de dados estatísticos; criptografia de dados; recursos de SQL; identificação e autorização de usuário e perdas. Vejamos agora um pouco sobre cada um deles.



## Características de Banco de Dados

### Controles de acesso

Os SGBD modernos admitem uma ou as duas abordagens gerais para a segurança de dados, conhecidas como controle **discriminatório** e **mandatário**.

No caso do controle **discriminatório**, determinado usuário terá, em geral, direitos diferentes de acesso sobre objetos diferentes (também conhecidos como **privilégios**). Além disso, há bem poucas limitações inerentes a respeito de quais usuários podem ter mais direitos sobre quais objetos. Por exemplo: o usuário U1 pode ser capaz de ver A, mas não B, enquanto o usuário U2 pode ser capaz de ver B, mas não A. Assim, os esquemas discriminatórios são muito flexíveis.

No caso de controle **mandatário**, ao contrário, cada objeto de dados é assinalado como certo nível de **classificação**, e cada usuário recebe certo nível de **liberação**. O acesso a determinado objeto de dados só pode ser feito por usuários com a liberação apropriada. Os esquemas mandatários tendem assim a ser hierárquicos por natureza e, desse modo, comparativamente rígidos. Por exemplo: se o usuário U1 pode ver A, mas não B, então a classificação de B deve ser maior que a de A, então nenhum usuário U2 poderá vê B sem poder ver A.



## Características de Banco de Dados

Date (2004) diz que é mais fácil declarar o que é permitido do que o que não é permitido. Considerando uma linguagem hipotética para atribuir uma autoridade, o comando cria uma autoridade (AF3) que permite a consulta de código, nome e cidade e excluir dados da tabela T para os usuários Maria, José e João:

AUTHORITY AF3

GRANT RETRIEVE (TCODIGO, TNAME, TCIDADE), DELETE

ON T

TO Maria, José, João;

Date (2004) afirma que os controles de acesso mandatários são aplicáveis a bancos de dados e que tais dados possuem uma estrutura de classificação bastante estática e rígida, como em ambientes militares ou governamentais. Nesse caso, cada objeto de dados tem um nível de classificação (ultra-secreto, secreto, confidencial, reservado e ostensivo) e cada usuário tem um nível de liberação. Suponha que os níveis formem uma ordenação estrita (ultra-secreto > secreto > confidencial > reservado > ostensivo).



## Características de Banco de Dados

### *Trilha de auditoria*

Segundo Date (2004) é, em essência, um arquivo ou banco de dados especial, em que o sistema automaticamente acompanha todas as operações realizadas por usuários sobre os dados normais. Isso é normalmente relevante em sistemas confidenciais ou quando se tornar necessário. Essa trilha pode estar fisicamente integrada ao *log* de recuperação.

Uma entrada de trilha de auditoria pode conter: requisição (texto de origem); terminal de onde a operação foi chamada; usuário que chamou a operação; data e hora da operação; variáveis de relação, tuplas, atributos afetados; imagens antes (valores antigos); imagens depois (valores novos).



## Características de Banco de Dados

### ***Bancos de dados estatísticos***

Segundo Date (2004), é um banco de dados que permite consultas que derivam informações de agregação (totais, médias), mas não consultas que derivam informações individuais. Por exemplo: a consulta “qual é a média de salário dos programadores?” poderia ser permitida, enquanto a consulta “qual é o salário da programadora Maria?” não seria. Os totais contêm vestígios das informações originais; um usuário curioso poderia ser capaz de (re)construir essas informações processando totais em número suficiente. Isso se chama “obtenção de informações confidenciais por dedução”.

### ***Criptografia de dados***

Para Date (2004), considerando um usuário que tenta contornar ilegalmente o sistema (retirando o disco rígido), a contramedida mais eficaz seria a criptografia de dados, isto é, o armazenamento e a transmissão de dados confidenciais em forma criptografada, quando um texto comum é submetido a um algoritmo de criptografia cujas entradas são o texto comum e uma chave de criptografia; a saída desse algoritmo é chamada de texto cifrado (ou criptografado). Os detalhes do algoritmo são mantidos públicos, porém, a chave é secreta.



## Características de Banco de Dados

### ***Recursos de SQL (Structured Query Language)***

São recursos de SQL as visões e as autorizações:

- ✓ **Visões (views)** – Segundo Silberschatz (1999), é qualquer relação que não faça parte do modelo lógico, mas seja visível para o usuário com uma relação virtual. As visões são recursos utilizados em segurança de dados para permitir ao usuário ver apenas os dados de uma determinada tabela a que tem direito, ou seja, a fim de proteger os dados contra acessos indevidos de usuários não autorizados. Uma view é um recurso de banco de dados para se criarem visões parciais de uma ou mais tabelas e definir os dados sobre os quais a autorização deve ser concedida; funciona como uma tabela virtual, pois não possui linhas próprias na medida em que as obtém em tempo de execução, ficando na memória para pronto uso. Pode ser utilizada em comandos da SQL, como: SELECT, INSERT, UPDATE e DELETE, para recuperação e manipulação de dados (com restrições), porém, não armazena esses dados.
- ✓ **Autorizações/privilegios** – São concessões dadas pelo DBA aos usuários de maneira a permitir que os mesmos operem no banco de dados com autorizações de: leitura; inserção; atualização; exclusão. São compostas pelos comandos GRANT que dão permissão a determinados usuários para poderem operar sobre determinado recurso do banco de dados como views e tabelas e pelo comando REVOKE que remove essas permissões.





## Características de Banco de Dados

Para dar acesso para Maria na *view*:

```
GRANT SELECT ON ESTOQUE_V TO Maria;
```

Para Maria visualizar os dados:

```
SELECT * FROM ESTOQUE_V;
```

Para remover a permissão de Maria na *view*:

```
REVOKE SELECT ON ESTOQUE_V TO Maria;
```



## Características de Banco de Dados

### **Identificação e autenticação do usuário**

É obtida por meio de identificação do usuário, utilizando um nome e uma senha autenticados quando este realiza o *logon* no sistema. Outra forma é a utilização de identificação biométrica (digitais, retina, sangue).

### **Perdas**

Podem ser do tipo:

- ✓ **Acidentais** – Podem ocorrer devido a avarias durante o processamento de transações, anomalias causadas por acesso concorrente aos dados e anomalias causadas por distribuição do banco de dados.
- ✓ **Intencionais** – Podem ocorrer devido à leitura não autorizada de dados, modificação não autorizada de dados, destruição não autorizada de dados e inserção não autorizada de dados.



## Características de Banco de Dados

A contramedida a ser adotada para se evitarem as perdas acidentais e intencionais considera os aspectos:

- ✓ **Físico** - O local do computador deve ter controle contra entradas de pessoas não autorizadas.
- ✓ **Controles de acesso ao dado** - Mapear o direito de acesso ao dado por meio de controles que fazem a validação sempre que um usuário tente acessar os dados (administrador, operador, suporte, vendas, RH, contas a pagar e receber entre outros).
- ✓ **Sistema operacional** - O DBA deve considerar os mecanismos de segurança fornecidos pelos sistemas operacionais para dobrar a segurança no acesso ao banco de dados.
- ✓ **Humano** - Usuários devidamente autorizados a acessarem os dados.



## Características de Banco de Dados

### Integridade

Segundo Date (2004), manter a integridade significa proteger os dados contra usuários autorizados. Assim, significa precisão, correção e validade, com o objetivo de preservar dados contra atualizações inválidas, normalmente, causadas por erros na entrada de dados do usuário, do programador, da aplicação, de falha do sistema ou de falsificação deliberada.

### Tipos de Integridade

A integridade em banco de dados deve ser implementada no SGBD e, posteriormente, pode ser replicada na aplicação. Primeiramente, vamos apresentar os conceitos de:

- ✓ **Chave primária (primary key)** - São colunas nas tabelas que aceitam somente um valor único e não podem ter valores nulos. Por exemplo, CPF do funcionário na tabela Funcionário.



## Características de Banco de Dados

- ✓ **Chave estrangeira ou externa (*foreign key*)** - São colunas nas tabelas que são cópias de outras colunas de tabelas relacionadas e que podem ter valores nulos. Por exemplo, CPF do funcionário na tabela Dependente que é relacionada à tabela Funcionário.
- ✓ **Integridade de entidade** - Nenhum componente da chave primária de qualquer tabela deve ter permissão para aceitar valores nulos.
- ✓ **Integridade referencial** - O banco de dados não pode conter quaisquer valores de chaves estrangeiras sem correspondência de chaves primárias nas tabelas relacionadas. A integridade referencial é implementada por meio de chaves estrangeiras.
- ✓ **Gatilhos (*triggers*)** - É um procedimento de ativação automática, dependente de alguma transação particular no banco de dados. No projeto de um *trigger*, devem-se considerar as condições em que serão executados e especificar as ações que deverão ser tomadas.



## Características de Banco de Dados

Para criar um *trigger* TRG\_LOGON, após o usuário se logar no banco de dados, uma linha é inserida em uma tabela TAB\_LOGON, contendo os dados USUARIO e SYSDATE (Data do sistema).

```
CREATE OR REPLACE TRIGGER TRG_LOGON
  AFTER LOGON ON DATABASE
BEGIN
  INSERT INTO TAB_LOGON (QUEM, QUANDO) VALUES (USUARIO,
SYSDATE);
END;
```



## Características de Banco de Dados

Eis outros conceitos fundamentais (asserções e restrições):

- ✓ **Asserções** - Para Silberschatz (1999), uma asserção é um predicado que expressa uma condição que deseja ser sempre válida no banco de dados. Durante qualquer modificação na base de dados, as asserções são verificadas e a modificação será permitida quando não houver violação. O uso exagerado de asserções pode prejudicar muito o desempenho do banco de dados. Uma asserção é uma condição para que o banco de dados insira determinado valor em uma tabela. Podemos enxergar as restrições de domínio como uma forma especial de asserção. Quando uma asserção é criada, o sistema verifica sua validade, se for válida qualquer modificação posterior, no banco, deverá satisfazer as asserções confirmadas; a atualização só será efetuada caso a asserção não seja violada, fato pelo qual se devem utilizar as asserções com muito cuidado.
- ✓ **Restrição de tipo de dado** - Esse tipo de restrição existe na medida em que, ao se definir um tipo de dados para uma coluna da tabela, por exemplo, inteiro (*integer*), essa coluna não aceita a digitação de outros tipos de dados.
- ✓ **Restrição NOT-NULL** - Define que uma coluna de uma tabela não pode aceitar valores nulos. Por exemplo, a coluna Nome do funcionário não pode receber valor nulo.
- ✓ **Restrição de domínio de atributo** - Define um conjunto de valores para determinada coluna de uma tabela. Por exemplo, a coluna Sexo na tabela de funcionário pode conter apenas os dados: masculino e feminino. Ao se inserir o dado sobre o sexo, o SGBD verifica se esse dado pertence ao conjunto de valores predeterminados por meio de uma restrição do tipo *check*.



## Características de Banco de Dados

- ✓ **Restrição de variáveis de relação** - Para Date (2004), uma restrição de variável de relação é uma restrição sobre uma variação de relação individual. Por exemplo, o comando a seguir cria uma restrição em que os fornecedores de Londres devem ter o status 20.  
`CONSTRAINT Fornecedor_Cidade  
IS_EMPTY ( FORNECEDOR WHERE CIDADE = "Londres" AND STATUS != 20 );`
- ✓ **Restrição de banco de dados** - Para Date (2004), uma restrição de banco de dados é uma restrição que relaciona entre si duas ou mais variáveis de relações distintas. A restrição a seguir afirma que nenhum fornecedor com status menor que 20 pode oferecer uma quantidade de peças maior que 500.  
`CONSTRAINT Fornecedor_C  
IS_EMPTY ( ( FORNECEDOR JOIN FORNECEDOR_PECA )  
WHERE STATUS < 20 AND QDE > 500 );`
- ✓ **Restrição de estado e restrição de transição** - A restrição de transição de dados define os estados possíveis que um determinado dado pode assumir após uma transição, de modo que não danifique a exatidão da informação.  
 Transições Corretas: Solteiro -> Casado; Casado -> Viúvo; Casado -> Divorciado.  
 Transições incorretas: Solteiro -> Viúvo; Solteiro -> Divorciado.



## Características de Banco de Dados

### Regra Áurea

Segundo Date (2004), não deve ser permitida nenhuma operação de atualização que deixe qualquer variável de relação em um estado que viole seu próprio predicado. Da mesma forma, não deve ser permitida nenhuma transação de atualização que deixe o banco de dados em um estado que viole seu próprio predicado.



## Características de Banco de Dados

### Compartilhamento

Segundo Date (2004), o banco de dados pode ser compartilhado entre diferentes usuários, no sentido de que podem ter acesso aos mesmos dados, possivelmente ao mesmo tempo (acesso concorrente). Tal compartilhamento, concorrente ou não, em parte, é uma consequência do fato de que o banco de dados é integrado.

### Integração

Date (2004) afirma que por integrado se quer dizer que o banco de dados pode ser considerado como uma unificação de vários arquivos que, de outro modo, seriam distintos, com eliminação de qualquer redundância parcial ou total entre esses arquivos.

### Recuperação

A recuperação em um sistema de banco de dados para Date (2004) significa recuperar o próprio banco de dados, restaurando-o a um estado correto, utilizando-se aplicativos do SGBD que realizam *back-up* e recuperação em banco de dados.

### Confiabilidade

Segundo Date (2004), é a probabilidade de o sistema funcionar sem queda, em qualquer momento, e a disponibilidade é a probabilidade de o sistema estar pronto e funcionando continuamente, sem queda, durante um período especificado.

### Desempenho

É a qualidade de um sistema que permite que as operações sobre os bancos de dados ocorram conforme negociado com o cliente em termos de tempo de realização. Por exemplo: pode-se definir que as operações de inserção devem ser de no máximo dois segundos, e que as consultas devem ser realizadas em no máximo cinco segundos.



## Características de Banco de Dados

### Controle de Concorrência

Para compreender como se dá o controle de concorrência, precisamos antes compreender o que é uma transação.

Date (2004) diz que uma transação é uma unidade lógica de trabalho que não é necessariamente uma única operação sobre o banco de dados. Para Silberschatz (1999), uma transação é uma unidade de execução de programa que acessa vários itens de dados. A transação consiste em todas as operações executadas entre o começo e o fim da transação. Para garantir a integridade dos dados, uma transação deve conter as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade).

Segundo Date (2004), concorrência é a capacidade de um SGBD permitir que diversas transações acessem o banco de dados ao mesmo tempo; porém, é necessário que haja algum tipo de mecanismo de controle da concorrência para assegurar que transações concorrentes não interfiram umas nas outras. As transações que concorrem ao mesmo dado no mesmo instante podem provocar inconsistências no banco e um gerenciador de controle de transações atua sobre esses processos para evitar as inconsistências. Esse controle pode ser implementado por meio de bloqueios que definem níveis de isolamento rígidos que garantam a consistência dos dados do banco.

Os SGBD devem garantir que todas as operações sejam executadas de forma segura, mantendo uma seqüência e deixando sempre o banco em um estado consistente. Os SGBD podem ter um monitor de processamento de transações (Monitor TP) que aguarda por um *commit* (término de uma operação bem sucedida) ou *rollback* (término de uma operação mal sucedida).



UCB – Banco de Dados – Prof. Milton

## Características de Banco de Dados

### Controle de Concorrência

Para Silberschatz (1999), há duas razões para permitir a concorrência e garantir a consistência dos dados em operações seqüenciais sobre os bancos de dados:

- ✓ Uma transação consiste em diversos passos, os quais envolvem atividades de entrada e saída e atividades na CPU. Uma vez que esses recursos podem ser explorados em paralelo, podemos explorá-los para que haja um aumento do *throughput* do sistema, ou seja, o número de transações que podem ser executadas em um determinado tempo.
- ✓ Algumas transações são longas, outras curtas. Se a execução das transações for seqüencial, uma transação curta, concorrendo com outra longa, seria obrigada a esperar um tempo desnecessário, caso estivessem operando em diferentes partes do banco de dados. Assim, a execução concorrente pode reduzir os atrasos imprevisíveis na execução das transações e reduzir o tempo médio de resposta.

Os sistemas de bancos de dados permitem que muitas transações tenham acesso ao mesmo banco de dados ao mesmo tempo e, portanto, devem dispor de algum tipo de mecanismo de controle de concorrência para assegurar que essas transações não interfiram entre si. Segundo Date (2004), existem três modos pelos quais uma transação, mesmo correta, pode produzir uma resposta errada se sofrer alguma forma de interferência de outra transação: o problema da atualização perdida (*lost update*); o problema da dependência sem COMMIT (*incommited dependency*) ou leitura suja (*dirty read*); o problema da análise inconsistente.



UCB – Banco de Dados – Prof. Milton

## Características de Banco de Dados

### Atualização Perdida

Ocorre quando duas transações A e B têm que atualizar a mesma tupla. A transação A lê a informação contida na tupla no instante t1. A transação B lê a informação contida na mesma tupla no instante t2. A transação A atualiza a tupla no instante t3 com as informações obtidas em t1. A transação B atualiza a tupla no instante t4 com os dados obtidos no instante t2, ignorando a transação A e sobrescrevendo a atualização feita por ela.

### Dependência sem COMMIT

Esse problema ocorre quando uma transação tem permissão de leitura e/ou gravação sobre uma tupla que foi atualizada por outra transação, mas que ainda não foi validada, ou seja, se o COMMIT ainda não ocorreu.

### Análise Inconsistente

Ocorre quando uma transação A faz a leitura de uma tupla e obtém dados para acumulação de valores, enquanto uma transação B realiza atualizações sobre as mesmas tuplas lidas por A. A transação A somente finaliza a análise de seus dados após a validação da atualização feita por B sobre a tupla (COMMIT), porém com dados inconsistentes lidos antes do UPDATE.



## Características de Banco de Dados

### Formas de solução de problemas de concorrência

- ✓ **Serializabilidade** - Os problemas de atualização perdida podem ser resolvidos utilizando a ordenação de processamento, que deve seguir uma seqüência. Para Date (2004), a serializabilidade é o critério de correção geralmente aceito para a execução de um conjunto de transações. Uma transação individual é considerada correta quando transforma um estado correto do banco de dados em outro estado correto. Se uma execução intercalada entre transações é equivalente a alguma execução serial, diz-se que ela é considerada correta e, portanto, serializável. O método mais usado para implementar a serialização é permitir o acesso a um item de dados somente se ele estiver bloqueado.
- ✓ **Níveis de isolamento** - O nível de isolamento deve ser o máximo possível para garantir a serializabilidade e que nenhuma interferência seja aceita. A SQL define quatro níveis de isolamento: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ e SERIALIZABLE, em que o padrão é SERIALIZABLE que permite a execução intercalada de qualquer conjunto de transações concorrentes. O padrão define três formas específicas pelas quais a serializabilidade poderia ser violada: leitura suja; leitura não repetível e fantasmas. A Tabela apresenta os diversos níveis de isolamento, indicando quais violações de serializabilidade são permitidas:

Nível de Isolamento	Leitura suja	Leitura não-repetível	Fantasma
READ UNCOMMITTED	S	S	S
READ COMMITTED	N	S	S
REPEATABLE	N	N	S
SERIALIZABLE	N	N	N

Onde:

**S** - Violação pode ocorrer

**N** - Violação não pode ocorrer



## Características de Banco de Dados

**Bloqueio** - Os problemas de serialização descritos anteriormente, ocasionados por concorrência entre transações, podem ser resolvidos por meio da técnica de bloqueio. Há dois tipos de bloqueio, compartilhados (C) e exclusivos (X), os quais são mostrados numa matriz de compatibilidade para obtenção de bloqueio.

Quando uma transação necessita de uma garantia de que o item de dados no qual ela irá operar não mudará enquanto ela estiver ativa, a transação adquire um bloqueio sobre esse item. Assim, ela é capaz de executar seu processamento tendo certeza de que este item sobre o qual ela opera permanecerá intacto durante o tempo que ela desejar. Dependendo do tipo de operação que será realizada, a transação pode solicitar ao gerenciador de controle de concorrência os tipos de bloqueio: Compartilhado (C - permissão de leitura, mas não de escrita) e Exclusivo (X - permissão de leitura e escrita).

**Matriz de compatibilidade de bloqueio** - Uma transação A possui no momento algum bloqueio sobre t representado nas colunas e uma transação distinta B requer uma solicitação de bloqueio sobre t, conforme representado na parte esquerda:

Onde:

**X** - Bloqueio exclusivo

**C** - Bloqueio compartilhado

**N** - Conflito (estado de espera)

**S** - Compatibilidade (solicitação satisfeita)

	X	C	-
X	N	N	S
C	N	S	S
-	S	S	S



## Características de Banco de Dados

Se uma transação A mantiver um bloqueio exclusivo (X) sobre a tupla t, então uma solicitação feita por uma transação distinta B de um bloqueio de qualquer tipo sobre t será negada.

Se uma transação A mantiver um bloqueio compartilhado (C) sobre a tupla t, então: uma solicitação de uma transação distinta B de um bloqueio X sobre t será negada; e uma solicitação de alguma transação distinta sobre B de um bloqueio C sobre t será concedida (ou seja, ambos mantêm um bloqueio C sobre t).

Quando nenhuma outra transação mantém algum bloqueio conflitante com o que está sendo solicitado sobre um determinado item de dado, o bloqueio pode ser concedido.

### Consequência da concorrência

**Deadlock** - Quando duas transações estão em estado de espera permanente pelo mesmo recurso esperando que a outra libere um bloqueio antes de prosseguir suas operações. Em caso de *deadlock*, é necessário que o sistema o detecte e o interrompa, o que implica em escolher uma das transações e fazer o ROLLBACK, liberando assim os seus bloqueios e permitindo o prosseguimento do processamento da outra transação.





## Características de Banco de Dados

### Distribuição

É a capacidade de um sistema de banco de dados permitir a criação de bancos de dados distribuídos para atender às demandas das organizações que têm seus negócios distribuídos. Reveja o conceito apresentado na Aula 2, item 2.4.5 Arquitetura Cliente Servidor (processamento distribuído).

### Portabilidade

É a capacidade de o sistema poder ser transferido para uma plataforma de SGBD de outro fabricante sem sofrer danos em sua estrutura e massa de dados. Por exemplo, migrar do ambiente Microsoft para o ambiente Oracle.

### Escalabilidade

É a capacidade de o sistema evoluir para uma plataforma de SGBD mais moderno do mesmo fabricante, sem sofrer danos em sua estrutura e massa de dados. Por exemplo: evoluir da versão 8.3 para a versão 10g do ambiente Oracle.

Como se pode ver, é preciso conhecer profundamente tais características para manter a eficácia do banco de dados.



## Projeto de Banco de Dados

O Projeto de Banco de Dados é um processo de determinação da organização, incluindo sua estrutura, conteúdo e as aplicações que o acessarão. Entretanto, muitos progressos foram feitos e, hoje, a área de Projetos é considerada uma disciplina estável, com seus métodos e técnicas que apoiam as necessidades das corporações em definir e atingir seus requisitos de informações, incluindo: definir, criar, armazenar, distribuir e recuperar dados; produzir e compartilhar informações; planejar e implementar um ambiente integrado, seguro e eficiente para criar e utilizar em aplicações distribuídas.

Um banco de dados deve ser modelado e projetado de acordo com os requisitos desejados. Depois, deve-se projetar sua arquitetura de funcionamento, de preferência, utilizando uma ferramenta de apoio ao desenvolvimento denominada CASE (*Computer-Aided Software Engineering*).



## Projeto de Banco de Dados

Requer alguns cuidados, como:

- ✓ adotar de uma metodologia;
- ✓ entender que o processo mental para construção do Modelo Conceitual de Dados, deve ser levado a sério, pois envolve técnicas de abstração necessárias para representar o que é essencial no negócio que se está modelando;
- ✓ entender que o processo mental para a obtenção do Modelo de Dados é diferente daquele para obtenção do Modelo de Processos;
- ✓ entender que as fases do ciclo de vida de um Sistema devem ser respeitadas;
- ✓ entender que uma fase só deve começar quando a outra terminar e for homologada pela alta cúpula da corporação interessada;
- ✓ haver disciplina no cumprimento das fases;
- ✓ todos os envolvidos (técnicos, usuários, direção etc.) devem ter o mesmo objetivo – uma empresa com modernos recursos de hardware e software;
- ✓ todos devem ter em mente que, hoje, no mundo moderno, “o pequeno veloz engole o grande lerdo” (Joelmir Betting, em palestra da Comdex Brasil – 1989);
- ✓ nunca os interesses pessoais (orgulho e vaidade) devem prevalecer sobre os interesses da organização.



UCB – Banco de Dados - Prof. Milton

## Projeto de Banco de Dados

### Características do Projeto de Banco de Dados

#### ***Independência de dados***

É a imunidade das aplicações à estrutura de armazenamento e à estratégia de acesso; isto é, as aplicações não dependem de qualquer estrutura de armazenamento ou de estratégia de acesso.

A provisão da independência de dados é o maior objetivo dos sistemas de banco de dados. É mais um objetivo dos sistemas de banco de dados do que, necessariamente, uma vantagem.

#### ***Dados não redundantes***

Dados que não se repetem desnecessariamente ou que deem uma mesma informação são chamados dados não redundantes. É preciso observá-los para:

- ✓ garantir que um item de informação apareça apenas uma vez no banco de dados;
- ✓ simplificar o processo de atualização de dados;
- ✓ permitir o acesso aos dados por meio de visões (*views*) da informação;
- ✓ garantir a consistência dos dados durante processos concorrentes ou atualizações;
- ✓ promover uma validação de dados definida apenas uma vez e executada pelo gerenciador do banco de dados, em vez de ser executada várias vezes dentro do código dos diversos programas.



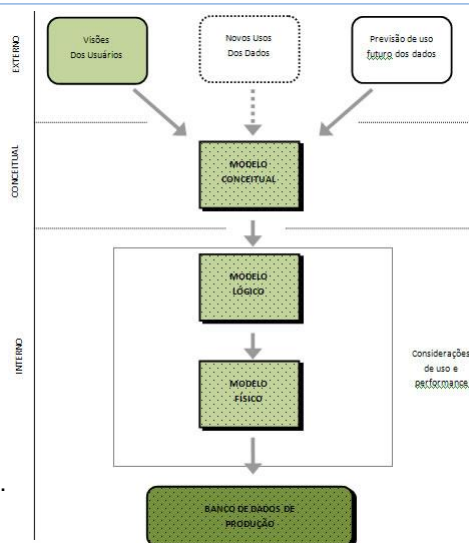
UCB – Banco de Dados - Prof. Milton

## Projeto de Banco de Dados

### Arquitetura de Projeto de Banco de Dados

Arquitetura com base em uma realizada pela CODASYL e ANSI/ SPARC – 1978, que diz que um projeto de banco de dados deve ter três níveis:

## Projeto de Banco de Dados



Fonte: DATE, 2004.

## Projeto de Banco de Dados

### Ciclo de Vida de um Projeto

O processo de criação de um banco de dados pode ser dividido em atividades distintas, desempenhadas por pessoas diferentes; porém, algumas pessoas podem assumir mais de uma atividade, como: planejador, analista (sistema, negócio, requisitos); projetista, implementador; testador, implantador; DBA e DA.

O ciclo de vida de um projeto é dividido em fases bem definidas, como:

- ✓ **Planejamento** – São adquiridos os conhecimentos necessários sobre o que se pretende desenvolver.
- ✓ **Análise** – São adquiridos os conhecimentos de negócio necessários sobre o que se pretende desenvolver.
- ✓ **Projeto** – São definidas as estruturas de dados e os processos que irão compor o sistema.
- ✓ **Implementação** – O sistema é implementado, instalado no ambiente final de produção, e é dado início a sua operação.
- ✓ **Teste** – São feitos os testes de segurança, integridade e usabilidade.
- ✓ **Implantação** – A aplicação é instalada para uso.
- ✓ **Manutenção** – São feitos ajustes e gerência do sistema.
- ✓ **Documentação**.

Esse processo é cíclico e novas necessidades darão início a uma nova fase de planejamento, iniciando assim um novo ciclo. Esse ciclo de vida pode ser o espiral por ser iterativo e incremental, ou seja, divide o problema em partes menores e, ao se encerrar cada uma, o projeto vai ganhando forma. Todas as fases devem ser encadeadas de forma que as saídas de uma fase sejam as entradas da fase subsequente.



## Modelagem de Dados

### Modelos de Dados

Um modelo de dados é uma ferramenta de modelagem que tem como propósito permitir a especificação das estruturas de dados e das operações permitidas sobre as mesmas, ou seja, deve permitir a captura do aspecto estrutural e do aspecto comportamental de um mini-mundo.

Pelo uso de modelo de dados, os analistas/projetistas constroem esquemas, os quais são representações de um mini-mundo. A qualidade do esquema resultante depende não univocamente da habilidade do projetista de banco de dados, mas das qualidades do modelo de dados selecionado. A seguir, descrevemos dois modelos de dados possíveis:

- ✓ **Modelo de dados semânticos** – Tiveram como origem comum o Modelo Entidade-Relacionamento, proposto por Chen (1976). Esse modelo incorporou alguns dos mais importantes conceitos semânticos sobre o mundo-real, adotando uma visão natural de que este consiste em entidades, relacionamentos e atributos.
- ✓ **Modelo conceitual de dados** – São classificados como um tipo particular de Modelos de Dados Semânticos para a representação do conhecimento, voltados para a Modelagem de Dados em nível conceitual. Entre esses modelos destaca-se, em ambiente computacional, o Modelo Entidade-Relacionamento (MER).



## Modelagem de Dados

### Mundo-Real

O mundo-real, do ponto de vista formal, é ainda muito nebuloso. Vários cientistas e leigos têm a fé de que, um dia, o mundo-real será todo formalizável, pois a sua visão do Homem e do Universo é mecanicista. Mas mesmo os que têm essa moderna religiosidade concordarão que o conhecimento científico atual do mundo é ínfimo e, portanto, permanece nebuloso em termos científicos clássicos, daí ser representado por uma nuvem. **Os objetos do mundo-real são os seres, os fatos, as “coisas” e os organismos sociais.** Não se deseja aqui dar precisão ao termo “mundo-real”; estamos cientes de que existem várias questões filosóficas e de percepção ligadas a esse conceito como, por exemplo, se os pensamentos, sentimentos e a vontade são ou não reais ou, em outras palavras, se há distinção entre o concreto e o abstrato. **Com toda essa imprecisão, será deixado para os analistas e projetistas delimitar o que lhes interessa como mundo-real, para fins de tratamento da informação** (SETZER, 1989).

Setzer (1989) diz que o **mini-mundo** é uma parcela do mundo-real, ou uma “porção” do mesmo sobre a qual temos algum tipo de interesse, seja ele o de observar, manipular ou “informatizar”, entre outros.



Copyright  
Todos os Direitos Reservados

UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

### Modelo de dados conceitual.

- Modelo semântico de representação do conhecimento.
- Considera objetos conceituais no ambiente de modelagem.
  - Cliente, Fornecedor, Veículo, Produto, Aluno.

Esse modelo usa uma representação simples, e tem como componentes os símbolos das Entidades (retângulo) que representam os objetos de interesse, os Relacionamentos (linhas) entre as entidades e as Cardinalidades (números) dos relacionamentos.

Copyright  
Todos os Direitos Reservados

UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

- Modelo Entidade-Relacionamento (MER).
  - Representa todas as entidades com seus atributos e relacionamentos.



Um cientista da computação americano e professor de Ciência da Computação da Louisiana State University, conhecido pelo desenvolvimento da Modelagem de Entidade-Relacionamento em 1976. Nascido em Taichung, Taiwan, ele recebeu um B.S. em engenharia elétrica em 1968 na Universidade Nacional de Taiwan, e um Ph.D. em informática / matemática aplicada na Universidade de Harvard em 1973.

Dr. Peter Pin-Shan Chen, criador do Modelo ER



UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

### **Entidades**

Segundo Bourbaki (1970), um conjunto é formado por elementos suscetíveis de possuírem certas propriedades e de terem, entre si, ou com elementos de outros conjuntos, certas relações.

Entidade é "qualquer coisa" do mundo real que possua uma existência independente, individual, separada, ou que possua características específicas relevantes para o sistema que está sendo modelado. Uma Entidade pode ser: um objeto com existência física – uma pessoa em particular, um carro, uma casa, um empregado, lugares; um objeto com uma existência conceitual – uma companhia, um curso numa universidade, eventos.




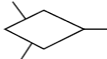


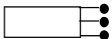
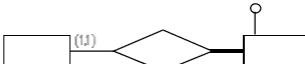

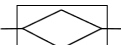
UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

- Entidades.
  - Entidade é a representação de um objeto do mundo-real.
  - Cliente, Fornecedor, Veículo, Produto, Aluno.

## Unidade 3 – Modelo Entidade-Relacionamento

### Notações

Conceito	Símbolo
Entidade	
Relacionamento	
Atributo	
Atributo identificador	 
Relacionamento identificador	
Generalização/especialização	
Entidade associativa	

## Unidade 3 – Modelo Entidade-Relacionamento

### Notações

#### ENTIDADES

As entidades são objetos ou conceitos que representam dados importantes. Geralmente elas são substantivos, ex.: *cliente*, *supervisor*, *local* ou *promoção*.

- ✓ **As entidades fortes** existem independentemente de outros tipos de entidades. Elas sempre possuem um ou mais atributos que distinguem de modo único cada ocorrência da entidade.
- ✓ **Entidades fracas** dependem de algum outro tipo de entidade. Elas não possuem atributos únicos (também conhecidos como chave primária) e não têm significado no diagrama se não dependerem de uma outra entidade. Esta outra entidade é conhecida como proprietária.
- ✓ **Entidades associativas** são entidades que associam as instâncias de um ou mais tipos de entidades. Elas também contêm atributos que são únicos no relacionamento entre tais instâncias de entidades.

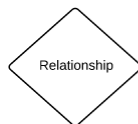


## Unidade 3 – Modelo Entidade-Relacionamento

### Notações

#### RELACIONAMENTOS

- ✓ **Os relacionamentos** são associações significativas entre uma ou mais entidades. Geralmente eles são verbos, ex.: *atribuir*, *associar* ou *monitorar*. Um relacionamento fornece informações úteis que não poderiam ser diferenciadas apenas com os tipos de entidade.
- ✓ **Relacionamentos fracos**, ou relacionamentos de identificação, são conexões que existem entre um tipo de entidade fraca e seu proprietário.





## Unidade 3 – Modelo Entidade-Relacionamento

### Notações

#### ATRIBUTOS

- ✓ **Os atributos** ou são características de uma entidade, ou de um relacionamento de muitos para muitos ou de um relacionamento de um para um.
- ✓ **Atributos multivalorados ou multivalores** são aqueles capazes de assumir mais do que um valor.
- ✓ **Os atributos derivados** são atributos cujo valor pode ser calculado a partir de valores de atributos relacionados.



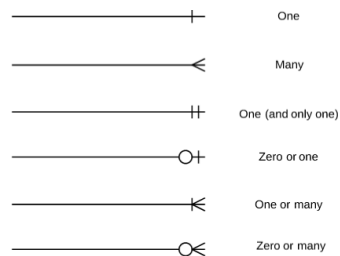
## Unidade 3 – Modelo Entidade-Relacionamento

### Notações

**Os relacionamentos** ilustram uma associação entre duas tabelas. No modelo de dados físico, os relacionamentos são representados por linhas estilizadas.

**Cardinalidade e ordinalidade** referem-se, respectivamente, ao número máximo de vezes que uma instância em uma entidade pode ser associada às instâncias em uma entidade relacionada e ao número mínimo de vezes que uma instância em uma entidade pode ser associada à uma instância na entidade relacionada. A cardinalidade e a ordinalidade são representadas pelo estilo de uma linha e seu final, como denotado pelo estilo da notação.

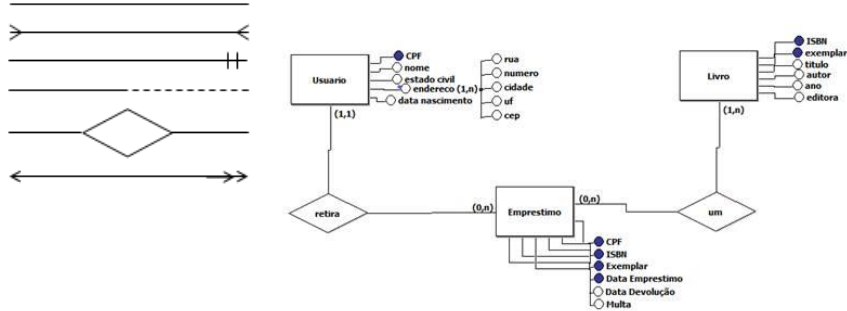
Quando se trata de notação, os modeladores de dados possuem muitas opções de escolha. Enquanto a notação pé de galinha é amplamente aceita como o estilo mais intuitivo, alguns desenvolvedores usam a notação OMT, IDEF, Bachman ou UML para indicar cardinalidade. Uma vez que a notação pé de galinha mostra tanto a cardinalidade mínima como a máxima em um formato gráfico de fácil leitura, o Lucidchart oferece a notação pé de galinha como o estilo preferido.



## Unidade 3 – Modelo Entidade-Relacionamento

### Notações

Os **relacionamentos** São as associações lógicas entre as entidades. Pode adotar as seguintes notações.



UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 3 – Modelo Entidade-Relacionamento

### Entidades-tipo

É uma classe ou conjunto de entidades que possuem as mesmas características. É representada graficamente por meio de um retângulo e identificada pelo seu nome no singular.



UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 3 – Modelo Entidade-Relacionamento

**Identificação de entidades** – É a tradução dos itens relevantes ao negócio e que possuem informações que serão utilizadas pelo sistema. Para conhecê-las, pergunta-se: Qual é o negócio da empresa? Que fatos ou itens são importantes? Há informações relacionadas com o item? O item possui significado em si? É entrada para o sistema?

### Categorias especiais de entidade-tipo

**Entidade-tipo fraca** – Representa um conjunto de entidades que não possuem existência independente, ou seja, dependem da existência de alguma outra entidade-tipo "forte" relacionada com a mesma. Exemplo: um funcionário possui dependentes. Um funcionário (entidade-tipo forte) pode "existir" independentemente da "existência" de dependentes (entidade-tipo fraca).

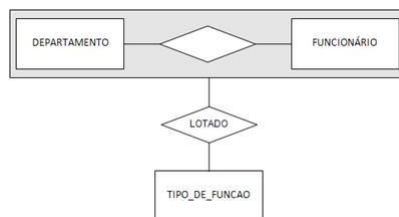


© Copyright  
Todos os Direitos Reservados

UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

**Entidade-tipo agregada** – Representa um relacionamento-tipo com o propósito de permitir que o mesmo se relacione com outra entidade-tipo ou com outro relacionamento-tipo. Exemplo: um funcionário lotado em determinado departamento pode assumir mais de uma função.

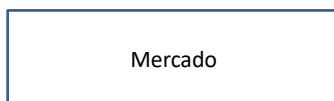


© Copyright  
Todos os Direitos Reservados

UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

**Entidade-tipo meramente conceitual** – Representa uma Entidade-tipo sobre a qual não existe interesse sobre os seus atributos, mas que deve ser representada para melhorar a expressividade de um Modelo Entidade-Relacionamento.



## Unidade 3 – Modelo Entidade-Relacionamento

### Atributos.

É uma propriedade elementar (característica) de uma Entidade ou de um Relacionamento que, de alguma forma, os descreve ou identifica. Por exemplo: funcionário tem nome, endereço etc.; livro tem ISBN, autor, título etc..

**Identificação de atributos** – Deve-se considerar: a análise de formulários já existentes; se há informações relacionadas com o item; se tem significado em si.

### Categorias especiais de atributos

**Atributo simples ou atômico** – Não pode ser dividido em subpartes, uma vez que não haveria um significado associado a essas subpartes.

**Atributo composto ou molecular (agregado)** – Pode ser decomposto em subpartes (outros atributos), com algum significado associado a essas subpartes. O valor de um atributo composto é a concatenação ou agregação de valores dos seus atributos simples.



## Unidade 3 – Modelo Entidade-Relacionamento

**Atributos monovalorado** – Quando determina no máximo um valor de um determinado domínio (conjunto de valores). Exemplo: um funcionário tem, no máximo, um nome, uma identidade, um endereço etc.

**Atributo multivalorado** – Quando determina diversos valores (N) de um domínio. Exemplo: Um Funcionário pode ter diversos telefones.

**Atributo opcional** – Quando uma dada entidade ou relacionamento não tiver valores de um domínio a eles associados. Atributos opcionais são os que aceitam o valor nulo que representa "valor desconhecido" ou "valor não aplicável".

**Atributo mandatório (obrigatório)** – Quando uma dada entidade ou relacionamento tiver pelo menos um valor de um domínio a eles associado.



## Unidade 3 – Modelo Entidade-Relacionamento

- Atributos chave.
  - CPF do Cliente.
  - RG do Cliente.
- Atributos simples: Nome do Cliente.
- Atributos compostos.
  - Endereço do Cliente: Rua, Logradouro, Cidade, Estado.
- Atributos derivados.
  - Idade é derivado de data de nascimento.
  - Quantidade de empregados é derivada da contagem de empregados em um setor.



## Unidade 3 – Modelo Entidade-Relacionamento

### Relacionamento

São as associações lógicas entre as entidades.

**Relacionamento-tipo** – Um relacionamento-tipo é uma classe ou conjunto de relacionamentos entre entidades pertencentes a Entidades-tipo não necessariamente distintas.

### **Tipos de relacionamentos**

**Relacionamento simples** – Exemplo: Alguns médicos são lotados em hospitais.

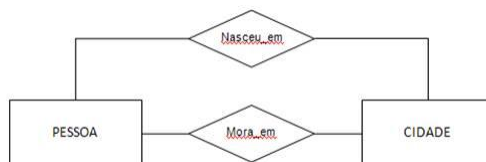


UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 3 – Modelo Entidade-Relacionamento

**Relacionamento binário** – Exemplo: Nasceu\_em, que relaciona pessoa e cidade de nascimento. Outro exemplo é Mora\_em, que indica a cidade que a pessoa mora atualmente.

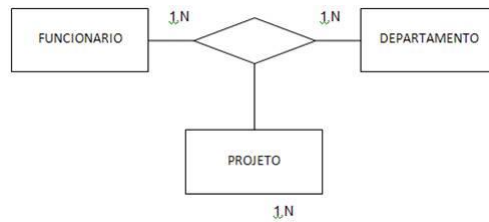


UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

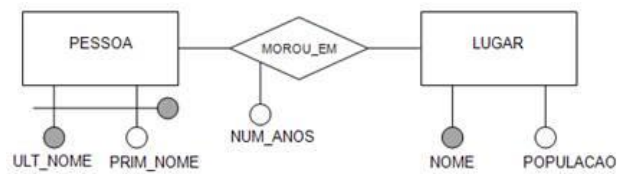
## Unidade 3 – Modelo Entidade-Relacionamento

### Relacionamento ternário (n-ário)



## Unidade 3 – Modelo Entidade-Relacionamento

### Relacionamento com atributo



Obs.: Relacionamento MOROU\_EM tem atributo porque é N,N.



## Unidade 3 – Modelo Entidade-Relacionamento

### Cardinalidade

É a quantificação dos relacionamentos entre as entidades.

### **Restrição de integridade de cardinalidade de atributo**

As cardinalidades devem ser representadas pela notação de um par de valores (mínimo, máximo), onde o mínimo impõe a “restrição de integridade de cardinalidade de atributo”.

#### **1º Valor (Cardinalidade Mínima):**

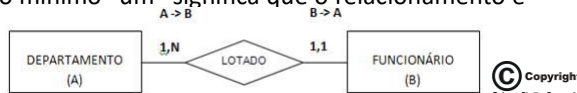
- 0 - Participação de Entidade em Relacionamento Opcional
- 1- Participação de Entidade em Relacionamento Obrigatório
- n - Participação mínima indefinida

#### **2º Valor (Cardinalidade Máxima):**

- 1- Participação em no máximo um Relacionamento
- n - Participação máxima em Relacionamentos indefinida

Exemplo: (0,1), (0,N), (1,1), (1,N), (M,N).

Quando se adota essa notação, o mínimo “zero” significa que o relacionamento é opcional e o mínimo “um” significa que o relacionamento é obrigatório (mandatório).



UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

As cardinalidades também podem usar outros limites numéricos mínimos e/ou máximos, de maneira que imponham a “Restrição de Integridade de Cardinalidade de Atributos”. Exemplos: um funcionário que pode ter, no máximo, três dependentes (0,3); o caso de um aluno que é obrigado a se matricular, em cada semestre letivo, no mínimo, em três disciplinas e, no máximo, em seis (3,6).

### **Tipos de cardinalidades**

**Um para um (1,1)** – Implica que uma ocorrência particular de uma entidade só pode estar relacionada a uma e somente uma ocorrência da outra entidade.

**Um para vários (1,N)** – Implica que uma ocorrência particular de uma entidade pode estar relacionada a uma ou mais ocorrências da entidade relacionada.

**Vários para vários (N,N)** – Implica que qualquer ocorrência de uma entidade pode estar relacionada a várias ocorrências da outra.

**Múltiplos** – Quando há mais de duas entidades participando do relacionamento. Pode ser ternário, quaternário, n-ário.

**Exclusivo** – Quando, em uma associação, a primeira entidade deve ser ligada a uma, e somente uma, de várias possíveis.

Copyright

UCB – Banco de Dados - Prof. Milton



## Unidade 3 – Modelo Entidade-Relacionamento

### Mecanismos de Abstração – Mental

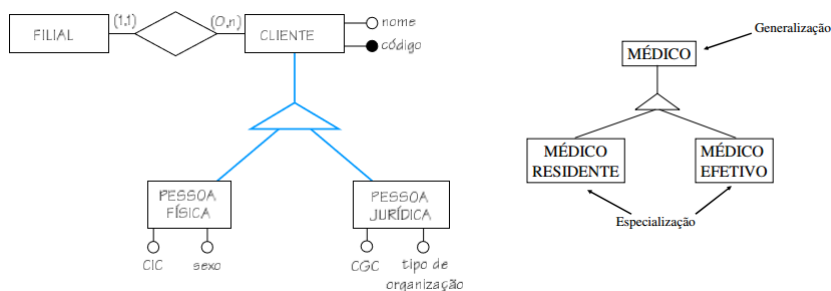
- ✓ Abstração.
- ✓ Associação.
- ✓ Agregação – composição.
- ✓ Classificação - Generalização.
- ✓ Classificação – Especialização.
- ✓ Classificação – Instanciação.



UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

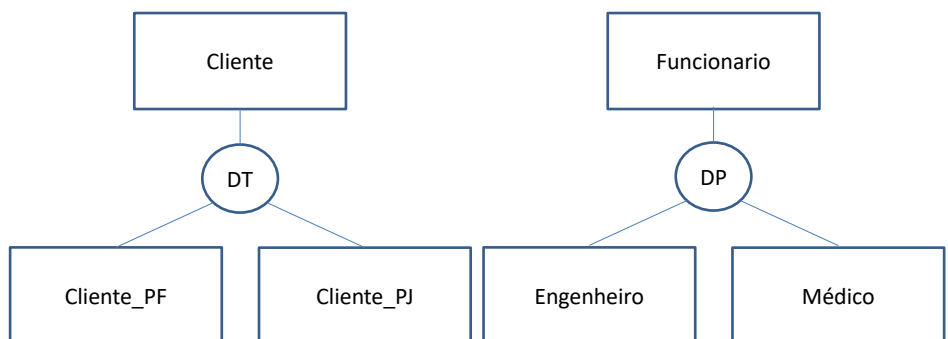
### Generalização x Especialização: Total ou Parcial Herança



UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

- Generalização e especialização.
  - Representação hierárquica.
  - Disjunção total e parcial.

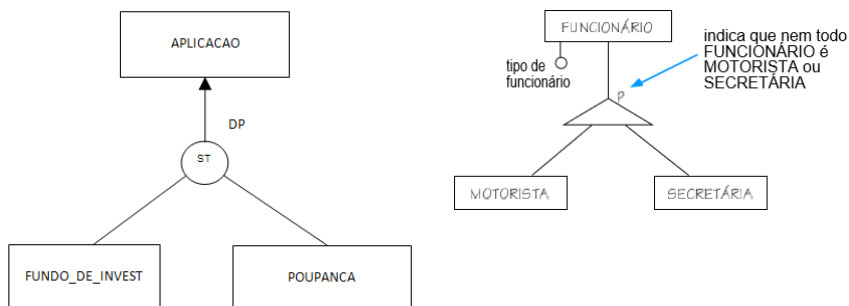


UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 3 – Modelo Entidade-Relacionamento

**DP (Disjunção Parcial)** – Uma ocorrência da entidade pai pode ser definida como uma das ocorrências da entidade filho ou nenhuma delas. Impõe a hierarquia/ herança entre as entidades. A entidade super-tipo pode ser uma das entidades subtipo ou outra que venha a existir no futuro. As entidades subtipo herdam os atributos da entidade supertipo. Exemplo: investimento bancário.



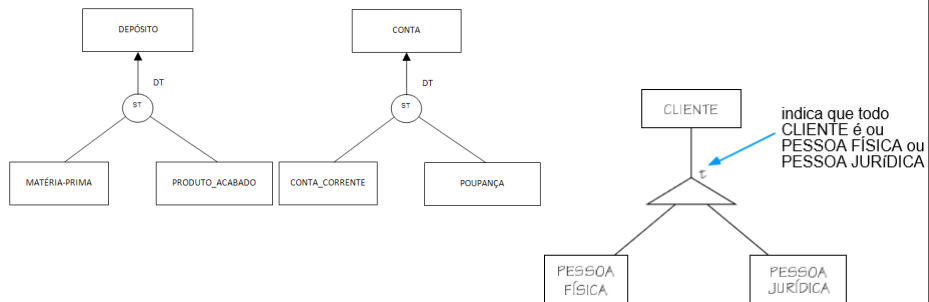
UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 3 – Modelo Entidade-Relacionamento

**DT (Disjunção Total):** Uma ocorrência da Entidade Pai deve ser definida como uma das ocorrências da Entidade filho, somente. Impõe a hierarquia / herança entre as entidades.

A entidade super-tipo pode ocorrer somente como uma das entidades sub-tipo. As entidades sub-tipo herdam os atributos da entidade super-tipo. Exemplo: Depósito de material ou conta em um banco.



UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### Restrições de Integridade.

- Definição de restrições de integridade (constraints) no MR.
  - Limitações no Modelo Relacional:
    - Cardinalidade do relacionamento: quantificação de um relacionamento (0, 1).
    - Chave: primária – unívoca.
    - Atributo: sexo M ou F.

UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

### Unidade 3 – Modelo Entidade-Relacionamento

- Grau de um tipo de relacionamento (Cardinalidade).
  - Par ordenado: m, M (mínimo, máximo).
  - 1,1: mínimo de 1 e máximo de 1.
  - 1,N: mínimo de 1 e máximo de N.



### Unidade 3 – Modelo Entidade-Relacionamento

- Restrição de cardinalidade.
  - Cardinalidade mínima:
    - 0: opcional.
    - 1: obrigatório.

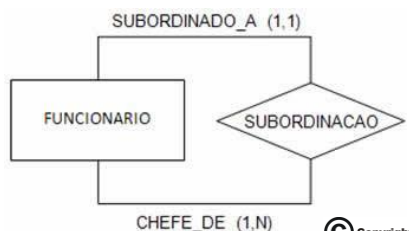


## Unidade 3 – Modelo Entidade-Relacionamento

### Categorias especiais de relacionamento-tipo

**Auto-relacionamento ou relacionamento-tipo recursivo** – É um relacionamento-tipo que envolve a mesma entidade-tipo.

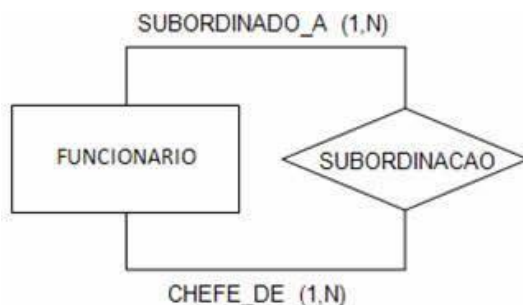
**Representa a chefia imediata** – Um funcionário é subordinado a outro funcionário.



UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

**Representa a cadeia hierárquica** – Um funcionário é subordinado a outro funcionário.



UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

### Observações

- ✓ Alguns atributos podem tornar-se relacionamento entre Entidades;
- ✓ Alguns atributos podem tornar-se novas Entidades. Isso ocorre às vezes por razões de projeto físico, quando então os atributos menos acessados são separados, constituindo-se em uma nova Entidade (Dados históricos, entre outros);
- ✓ Muitos relacionamentos podem ser reduzidos a atributos, possivelmente na segunda Entidade do relacionamento;
- ✓ Relacionamentos que se tornem desnecessários ou redundantes devem ser retirados do Modelo.



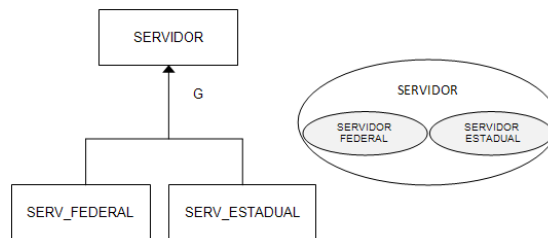
UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

Devem ser observadas as seguintes restrições com relação às entidades-tipo:

**Restrição de integridade de identificação:** Toda e qualquer entidade-tipo tem um conjunto de atributos cujos valores são capazes de identificar suas entidades de forma unívoca. São as chaves candidata.

**Restrição de integridade de tipo de generalização:** O Mecanismo de generalização é o originador do conceito matemático subconjunto. Uma Hierarquia de Generalização pode ser representada, segundo a teoria dos conjuntos, da seguinte forma:



UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

Em função dessa característica, as seguintes situações podem ocorrer:

Interseção entre os Subconjuntos (Especializações). Nesse caso diz-se que existe uma **SOBREPOSIÇÃO (S)** na Hierarquia de Generalização;

Caso não exista a **SOBREPOSIÇÃO** diz-se que a Hierarquia de Generalização é **EXCLUSIVA (E)**;

Se a união dos Subconjuntos formar o “Conjunto principal” diz-se que a Hierarquia de Generalização é **TOTAL (T)**; e

Caso contrário, diz-se que é **PARCIAL (P)**.

Tais símbolos são representados por meio de um par de valores da seguinte maneira :

1º Valor:

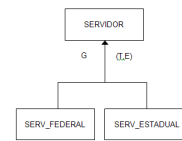
P - parcial

T - total

2º Valor:

E - exclusiva

S - sobreposição



Copyright  
Todos os Direitos Reservados

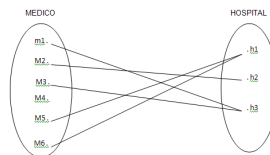
UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

**Restrição de integridade de cardinalidade de relacionamento-tipo** – Permite o estabelecimento da quantidade máxima e mínima de participações de entidades em relacionamentos. Uma restrição de integridade de cardinalidade é uma categoria de um relacionamento, baseada em regras de negócio. Por exemplo:

Um médico pode não estar lotado em nenhum hospital, caso do médico, ou estar lotado em no máximo um hospital.

Um hospital deve ter pelo menos um médico lotado, podendo ter um número máximo indefinido de médicos lotado.



Copyright  
Todos os Direitos Reservados

UCB – Banco de Dados - Prof. Milton

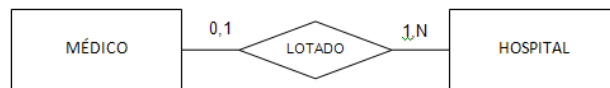
### Unidade 3 – Modelo Entidade-Relacionamento

A situação acima representa pode ser expressa da seguinte forma:

Um médico pode não estar lotado em nenhum hospital, caso do médico “m4”, ou estar lotado em no máximo um hospital; e

Um hospital deve ter pelo menos um médico lotado, podendo ter um número máximo indefinido de médicos lotado.

A mesma situação pode ser representada segundo o MER da seguinte forma:

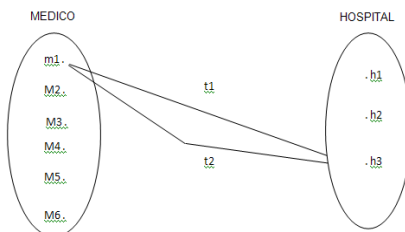


UCB – Banco de Dados - Prof. Milton

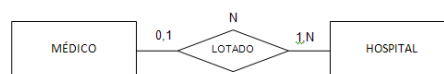
### Unidade 3 – Modelo Entidade-Relacionamento

#### Restrição de integridade de repetição em relacionamento-tipo

– Permite representar situações em que uma entidade volta a se relacionar com uma mesma entidade em outro momento no tempo. Exemplo: um médico pode voltar a trabalhar em um mesmo hospital ao longo de sua carreira profissional.



Tal situação pode ser representada pelo MER onde o N significa que um médico pode voltar a trabalhar em um mesmo hospital um número indefinido de vezes.



UCB – Banco de Dados - Prof. Milton



### Unidade 3 – Modelo Entidade-Relacionamento

**Restrição de integridade de função** – Essa restrição é utilizada nos relacionamentos-tipo recursivos, servindo para melhorar o seu entendimento.



No exemplo acima pode-se observar que a Entidade-Tipo Funcionário assume dois papéis ou funções no auto-Relacionamento SUBORDINAÇÃO, os papéis de CHEFE e SUBORDINADO.



UCB – Banco de Dados - Prof. Milton

### Unidade 3 – Modelo Entidade-Relacionamento

**Restrição de integridade de domínio de atributo** – Permite que se defina para cada atributo um conjunto de valores válidos (domínio) em cima do qual o atributo irá atuar.

Exemplo: 1- O atributo estado civil só pode assumir os seguintes valores: C – Casado; S - Solteiro; V - Viúvo; D – Divorciado. 2 - O atributo idade de funcionário só pode variar de: idade mínima - X anos a idade máxima - Y anos.

**Restrição de integridade de cardinalidade de atributo** – O conceito de cardinalidade de atributo é similar ao de cardinalidade de relacionamento-tipo, chegando-se então à seguinte classificação para os Atributos: OPCIONAL - (Cardinalidade Mínima = 0); OBRIGATÓRIO - (Cardinalidade Mínima = 1); MONO-VALORADO - (Cardinalidade Máxima = 1); MULTI-VALORADO - (Cardinalidade Máxima = N).



UCB – Banco de Dados - Prof. Milton

### Unidade 3 – Modelo Entidade-Relacionamento

Utilização de Ferramenta CASE.

- Computer-Aided Software Engineering

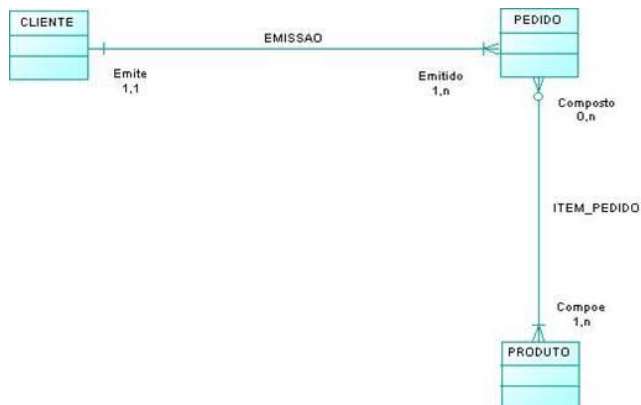
WebModeler, Rational Rose, DBMain, ErWin, PowerDesigner, Oracle Designer, Together J 5.5, ER/Studio, Dr. Case, DBDesigner, ERWIN/ERX, S-DESIGNOR, SYSTEM ARCHITECT, JM Designer.



UCB – Banco de Dados - Prof. Milton

### Unidade 3 – Modelo Entidade-Relacionamento

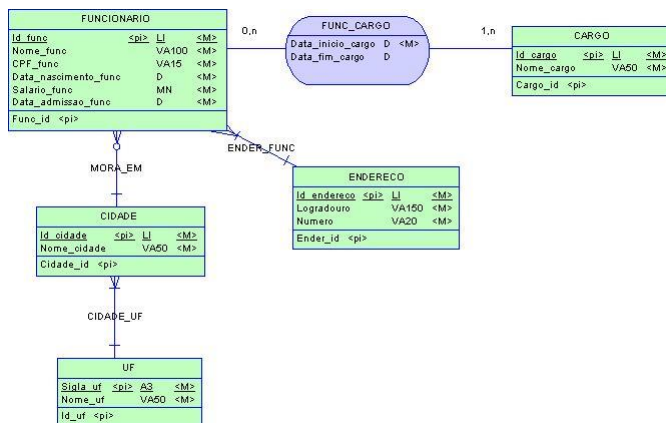
Este modelo foi elaborado utilizando a ferramenta CASE PowerDesigner 12.5 da Sybase.



UCB – Banco de Dados - Prof. Milton

### Unidade 3 – Modelo Entidade-Relacionamento

Este modelo foi elaborado utilizando a ferramenta CASE PowerDesigner 12.5 da Sybase.



UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

### Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

Transição do Modelo Entidade Relacionamento para o Modelo Relacional.

- Entidades viram Tabelas.
- Atributos viram Colunas.
- Chaves candidatas viram Chaves primárias.
- Chaves primárias podem virar chaves estrangeiras.

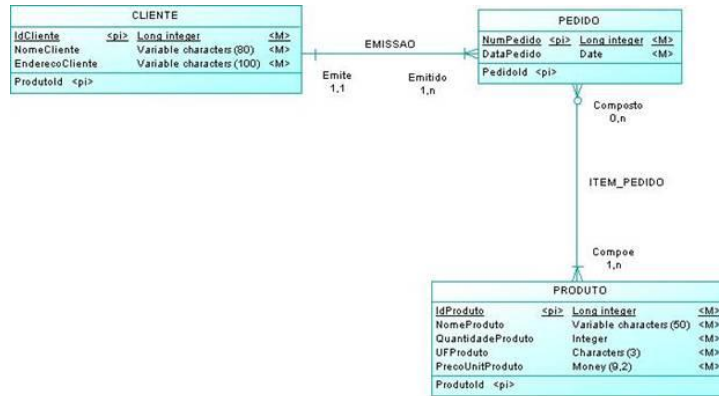
UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 3 – Modelo Entidade-Relacionamento

### Modelo Lógico de Dados

Este modelo foi elaborado utilizando a ferramenta CASE PowerDesigner 12.5 da Sybase.



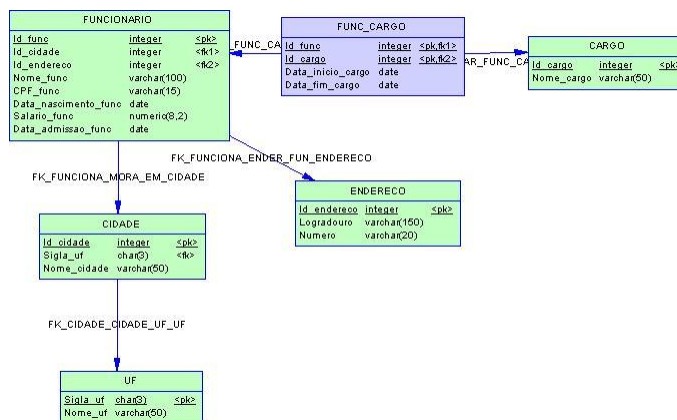
UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 3 – Modelo Entidade-Relacionamento

### Modelo Lógico de Dados

Este modelo foi elaborado utilizando a ferramenta CASE PowerDesigner 12.5 da Sybase.



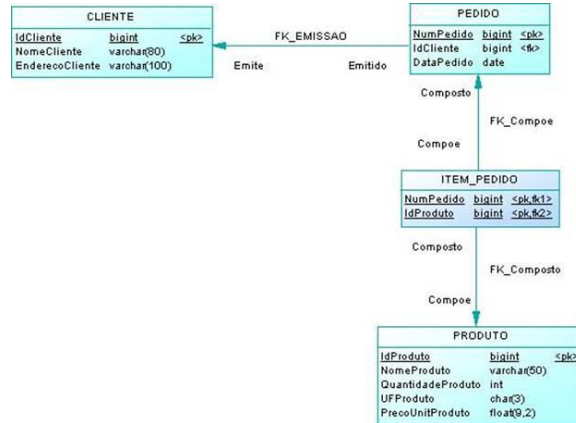
UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 3 – Modelo Entidade-Relacionamento

### Modelo Físico de Dados

Este modelo foi elaborado utilizando a ferramenta CASE PowerDesigner 12.5 da Sybase, sendo gerado a partir do MER para o SGBD MySQL 5.0



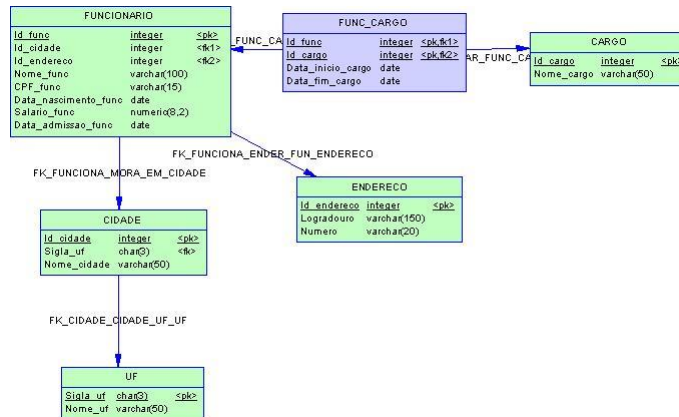
UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 3 – Modelo Entidade-Relacionamento

### Modelo Físico de Dados

Este modelo foi elaborado utilizando a ferramenta CASE PowerDesigner 12.5 da Sybase, sendo gerado a partir do MER para o SGBD MySQL 5.0



UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 3 – Modelo Entidade-Relacionamento

### Schema do BD – Script

Este script foi elaborado utilizando a ferramenta CASE PowerDesigner 12.5 da Sybase, para o SGBD MySQL 5.0.



UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

```

/*=====*/
/* DBMS name: MySQL 5.0 */
/* Created on: 15/6/2008 20:15:50 */
/*=====*/
drop table if exists CLIENTE;
drop table if exists ITEM_PEDIDO;
drop table if exists PEDIDO;
drop table if exists PRODUTO;
/*=====*/
/* Table: CLIENTE */
/*=====*/
create table CLIENTE ( IDCLIENTE bigint not null, NOMECLIENTE varchar(80) not null, ENDerecoCLIENTE varchar(100) not null, primary key (IDCLIENTE));
/*=====*/
/* Table: ITEM_PEDIDO */
/*=====*/
create table ITEM_PEDIDO ( NUMPEDIDO bigint not null, IDPRODUTO bigint not null, primary key (NUMPEDIDO, IDPRODUTO) );
/*=====*/
/* Table: PEDIDO */
/*=====*/
create table PEDIDO ( NUMPEDIDO bigint not null, IDCLIENTE bigint not null, DATA PEDIDO date not null, primary key (NUMPEDIDO) );
/*=====*/
/* Table: PRODUTO */
/*=====*/
create table PRODUTO ( IDPRODUTO bigint not null, NOMEPRODUTO varchar(50) not null, QUANTIDADEPRODUTO int not null, UFPRODUTO char(3) not null, PRECOUNITPRODUTO float(9,2) not null, primary key (IDPRODUTO) );
alter table ITEM_PEDIDO add constraint FK_COMPOE foreign key (NUMPEDIDO) references PEDIDO (NUMPEDIDO) on delete restrict on update restrict;
alter table ITEM_PEDIDO add constraint FK_COMPOSTO foreign key (IDPRODUTO) references PRODUTO (IDPRODUTO) on delete restrict on update restrict;
alter table PEDIDO add constraint FK_EMITAO foreign key (IDCLIENTE) references CLIENTE (IDCLIENTE) on delete restrict on update restrict;

```



UCB – Banco de Dados - Prof. Milton

## Unidade 3 – Modelo Entidade-Relacionamento

### Índices

São recursos utilizados para agilizar as pesquisas em bancos de dados, porém dificultam as atualizações, pois são armazenados em estruturas diferentes das estruturas dos dados. Os campos inicialmente candidatos são os da cláusula Where em um comando SQL. Podem utilizar uma ou mais colunas. Observe os conceitos relacionados a índice:

- ✓ **Primary** – É o índice principal criado a partir da chave primária. Por exemplo, matrícula do aluno na tabela aluno. Esse índice não pode repetir valor.
- ✓ **Unique** – Criado a partir de um atributo único da tabela. Por exemplo, CPF do aluno na tabela aluno. Esse índice não pode repetir valor.
- ✓ **Index** – É um índice normal que pode repetir valor. Por exemplo, nome do aluno na tabela aluno.
- ✓ **Word index** – Criado a partir de um atributo do tipo *string*.
- ✓ **Clustered** – Determina a ordem física dos dados na tabela.
- ✓ **NonClustered** – Os dados são armazenados em locais diferentes dos dados.
- ✓ **Btree** – Árvore de pesquisa balanceada para trabalhar em qualquer dispositivo de armazenamento de acesso direto em memória secundária.
- ✓ **Rtree** – Utilizado para indexar estrutura de dados multidimensionais, limitado a dados com até 8Kbytes. Utilizado normalmente com dados do tipo box, circle, point e outros.
- ✓ **Hash** – Implementa as dispersões lineares de Litwin. Possui fraco desempenho em alta concorrência.
- ✓ **Fulltext Search** – Permite obter registros que contenham certa palavra em uma coluna ou conjunto de colunas.

UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### Conceitos básicos.

- Edgar Frank Codd – IBM – 1969 - A Relational Model for Large Shared Data Banks.
- Teoria dos Conjuntos.
- Relações.
- Regras do Modelo Relacional.
- C. J. Date: validou o SGBD Relacional – IBM.

UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

Conceitos básicos.

O Modelo de Dados Relacional foi introduzido por Codd (1970). Sua utilização comercial, no entanto, se deu por volta do ano de 1978, quando a IBM resolveu validar o Modelo Relacional, realizando a primeira implementação do *SystemR*.

Comparando-o com os Modelos de Dados Hierárquico e de Redes, ele é o mais simples. Uma vez que apresenta uma estrutura de dados mais uniforme e menos física, é também o mais formalizado, pois foi inteiramente baseado na teoria dos conjuntos.



## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### Características dos Sistemas Relacionais

A maioria dos Sistemas de Banco de Dados desenvolvidos nos últimos anos é relacional. Um Sistema Relacional é aquele em que:

Os arquivos convencionais são tratados como Tabelas Relacionais ou simplesmente Tabelas. Os dados são percebidos pelos usuários como Tabelas.

Os operadores à disposição do usuário são operadores que geram novas tabelas, a partir das antigas.

As linhas (tuplas) das Tabelas são comparadas aos registros dos arquivos (registros lógicos).

As colunas das tabelas são comparadas aos campos desses registros lógicos.

As operações SELECT, INSERT, UPDATE, DELETE são exemplos de instruções da linguagem de Banco de Dados chamada SQL (*Structured Query Language*).

Segue um exemplo de banco de dados simplificado (DATE, 2004).





## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

Um banco de dados de adega de vinhos, que contém somente uma Tabela – ADEGA:

Codigo	Tipo	Produtor	Safra	Estoque	Comentario
2	Chardonnay	Buena Vista	83	1	
3	Chardonnay	Louis Martini	81	5	
6	Chardonnay	Chappellet	82	4	Thanks giving
11	Jo. Riesling	Jeken	84	10	
12	Jo. Riesling	Buena Vista	82	1	Late Harvest
16	Jo. Riesling	Sattui	82	1	Very Dry
21	Fume Blanc	Ch. St. Jean	79	4	Napa Valley
22	Fume Blanc	Robt. Mondavi	78	2	
25	Wh. Burgundy	Mirassou	80	6	
30	Gewurztraminer	Buena Vista	80	3	
43	Cab. Sauvignon	Robt. Mondavi	77	12	
50	Pinot Noir	Mirassou	77	3	Harvest
51	Pinot Noir	Ch. St. Jean	78	2	
64	Zinfandel	Mirassou	77	9	Anniversary
72	Gamay	Robt. Mondavi	78	2	



UCB – Banco de Dados - Prof. Milton

## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

Exemplo de comando SQL para recuperação dos dados nesta Tabela:

```
SELECT Codigo, Tipo, Produtor
FROM ADEGA
WHERE Safra = 82;
```

Resultado na tela:

Codigo	Tipo	Produtor
6	Chardonnay	Chappellet
12	Jo. Riesling	Buena Vista
16	Jo. Riesling	Sattui



UCB – Banco de Dados - Prof. Milton

## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

Distinção de um sistema relacional do não-relacional:

- ✓ Em um sistema relacional, o usuário vê os dados como Tabelas.
- ✓ Em um sistema não-relacional, o usuário vê outras estruturas de dados. Essas estruturas necessitam de outros operadores para manipulá-las. Por exemplo: um sistema hierárquico apresenta os dados aos usuários sob a forma de um conjunto de estrutura de árvores e os operadores, para manipulá-las, incluem operadores para percorrerem caminhos hierárquicos para cima e para baixo das árvores.

Como dissemos antes, o Modelo Relacional se baseia na Teoria dos Conjuntos. Vamos relembrar algumas informações importantes para nossa compreensão sobre esse modelo.



## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### Teoria dos Conjuntos

Dado os conjuntos  $A = \{a,b\}$  e  $B = \{c,d\}$ , temos :

O Produto Cartesiano de A com B, simbolizado por  $A \times B$ , é um conjunto formado pelos elementos dos pares ordenados  $(x,y)$ , onde  $x \in A$  e  $y \in B$ .

$$A \times B = \{ (x,y) / x \in A \text{ e } y \in B \}$$

$$\text{De outra forma, } A \times B = \{ (a,c), (a,d), (b,c), (b,d) \}$$

Suponha os conjuntos:

- ✓  $A = \text{nomes (domínio de nomes)} = \{\text{Leonardo, Felipe, Bianca}\}$
- ✓  $B = \text{estados (domínio de estados)} = \{\text{RJ, BSB}\}$
- ✓  $C = \text{idades (domínio de idades)} = \{20, 25\}$

Temos o Produto Cartesiano  $A \times B \times C = \{(\text{Leonardo, RJ, 20}), (\text{Leonardo RJ, 25}), (\text{Leonardo, BSB, 20}), (\text{Leonardo, BSB, 25}), (\text{Felipe, RJ, 20}), (\text{Felipe, RJ, 25}), (\text{Felipe, BSB, 20}), (\text{Felipe, BSB, 25}), (\text{Bianca, RJ, 20}), (\text{Bianca, RJ, 25}), (\text{Bianca, BSB, 20}), (\text{Bianca, BSB, 25})\}$ .



## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### Teoria dos Conjuntos

Essa teoria apresenta as possibilidades (domínio) de combinações (ocorrências) entre os elementos dos três conjuntos.

Observando o Produto Cartesiano como Objetos de Dados, temos:

Leonardo	RJ	20
Leonardo	RJ	25
Leonardo	BSB	20
Leonardo	BSB	25
Felipe	RJ	20
Leonardo	RJ	20
Leonardo	RJ	25
Leonardo	BSB	20
Leonardo	BSB	25
Felipe	RJ	20

UCB – Banco de Dados - Prof. Milton



## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

Dessa forma, Codd (1970) sugeriu que os Objetos de Dados do Modelo de Dados Relacional possam ser definidos, informalmente, como contendo:

- ✓ Relação corresponde a uma Tabela.
- ✓ Tupla corresponde a uma linha da Tabela.
- ✓ Atributo corresponde a uma coluna da Tabela.

UCB – Banco de Dados - Prof. Milton



## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

Exemplo ilustrativo:

TABELA A

↓

RELÇÃO (AxBxC)

ATRIBUTO,  
COLUNA OU  
CAMPO

↓

Leonardo	RJ	20
Leonardo	RJ	25
Leonardo	BSB	20
...	...	...
Nome N	Cidade N	Idade N

← TUPLA, LINHA  
OU REGISTRO

UCB – Banco de Dados - Prof. Milton

Copyright

## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### Banco de Dados com Informações de Aluno e Curso

A Figura apresenta um banco de dados com informações sobre aluno e curso.

Fonte: Elmasri (2005).

ALUNO	Nome	Numero	Turma	Curso	Hab
	Smith	17	1	CC	
	Brown	8	2	CC	

CURSO	Nome do Curso	Numero do Curso	Creditos	Departamento
	Introdução à Ciência da Computação	CC1310	4	CC
	Estruturas de dados	CC3320	4	CC
	Matemática Discreta	MAT2410	3	MATH
	Banco de dados	CC3380	3	CC

DISCIPLINA	Identificador de Disciplina	Numero do Curso	Semestre	Ano	Instrutor
	85	MAT2410	Segundo Semestre	96	King
	92	CC1310	Segundo Semestre	96	Anderson
	102	CC3320	Primeiro Semestre	99	Knuth
	112	MAT2410	Segundo Semestre	99	Chang
	119	CC1310	Segundo Semestre	99	Anderson
	135	CC3380	Segundo Semestre	99	Stone

HISTORICO_ESCOLAR	Numero do Aluno	Identificador de Disciplinas	Nota
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PRE_REQUISITO	Numero do Curso	Numero do Pre_requisito
	CC3380	CC3320
	CC3380	MAT2410
	CC3320	CC1310

UCB – Banco de Dados - Prof. Milton

Copyright

## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### Domínio.

- Valores assumidos na relação.
  - Ex: Idade do Funcionário: 18 a 80 anos.

Um domínio D é um conjunto de valores atômicos do mesmo tipo, a partir do qual um ou mais atributos retiram os seus valores.

Um método comum para a especificação de domínio é a especificação do tipo do dado para o qual os valores do domínio são válidos. É também usual a especificação de um nome de domínio para auxiliar a interpretação de seus valores. Um domínio pode ser mais bem definido ainda por meio da especificação adicional do formato dos elementos do domínio, limites inferiores e superiores ou valores discretos assumidos.

### Exemplo:

Domínio Número-de-peça: INT  
 Domínio Nome-de-peça: CHAR(20)  
 Domínio Cor: CHAR(6)  
 Domínio peso: SMALLINT  
 Relação PEÇAS  
 (#P : domínio Número-de-peça,  
 PNAME : domínio Nome-de-peça,  
 COR : domínio cor,  
 PESO : domínio Peso).



## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### Atributo.

- São as características de uma relação:
  - Nome do Cliente, Telefone do Cliente.

Um atributo significa o uso de um domínio em uma relação, estabelecendo sobre esse domínio os valores que estão sendo “utilizados” na relação. Os atributos dentro de uma relação podem representar atributos de entidades-tipo ou de relacionamentos-tipo.

Os atributos no Modelo de Dados Relacional são atômicos; isto é, são indivisíveis, no sentido de não poderem ser decompostos sem perda de significado, sendo mono valorados.

Dessa forma, podemos afirmar que o domínio é o conjunto de valores de uma relação (Leonardo, RJ, 20) e atributo é o uso de domínio dentro de uma relação (Leonardo). Em uma relação, os atributos podem ter o mesmo domínio.



## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### Relação.

- É um objeto no modelo relacional de dados:
  - Cliente, Funcionário, Aluno, Veículo.

Podemos definir relação de maneira informal ou formal:

**Definição informal** – Uma relação é um conjunto de tuplas, ou seja, é uma tabela de valores em que as linhas representam um agregado de valores de dados relacionados. Esses agregados podem ser interpretados como um fato que descreve uma entidade ou um relacionamento. O nome da tabela e da coluna é usado para ajudar a interpretação do significado dos valores de cada linha da tabela.

Uma relação R de A para B é um subconjunto de  $A \times B$ .

PESSOA = {(Leonardo, RJ, 20), (Leonardo, BSB, 20), (Felipe, BSB, 25), (Bianca, RJ, 25)}

**Tupla:** Ocorrência conceitual em uma relação:

- Nome do Cliente, Telefone do Cliente.
- 'Maria José', '61-998989898'.



## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

**Definição formal** – É o subconjunto formado por elementos (domínio) pertencentes ao conjunto decorrente do Produto Cartesiano ( $A \times B$ ) entre dois ou mais conjuntos.

Toda relação apresenta determinadas propriedades, as quais veremos a seguir:

**Cardinalidade** – É o conjunto de valores de um Domínio em uma associação entre relações.

**Grau** – É o número de atributos de uma relação:

- Unárias** – grau um.
- Binárias** – grau dois.
- Ternárias** – grau três.
- N-árias** – grau N.

### Ordenação

**Tuplas** – Não são ordenadas, pois uma relação é um conjunto e estes, matematicamente, não são ordenados.

**Atributos** – Matematicamente, podemos considerar que os domínios de uma relação têm uma ordem definida.

**Repetição de tuplas** – Como, matematicamente, os elementos de um conjunto não são repetidos e as tuplas representam esses elementos, elas não se repetem.



## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### As 12 Regras para um SGBD\_R

Para encerrar esta aula, veremos que, segundo Codd (1985), há 12 regras para que um Sistema Gerenciador de Banco de Dados (SGBD) seja considerado relacional. São elas:

**Regra fundamental** – Um SGBD relacional deve gerenciar seus dados usando apenas suas capacidades relacionais.

**Regra da informação** – Toda informação deve ser representada de uma única forma, como dados em uma Tabela.

**Regra da garantia de acesso** – Todo dado (valor atômico) pode ser acessado logicamente (e unicamente) usando o nome da Tabela, o valor da chave primária da linha e o nome da coluna.

**Tratamento sistemático de valores nulos** – Os valores nulos (diferente do zero, da string vazia, da string de caracteres em brancos e outros valores não nulos) existem para representar dados não existentes de forma sistemática e independente do tipo de dado.

**Catálogo dinâmico on-line baseado no modelo relacional** – A descrição do banco de dados é representada no nível lógico como dados ordinários (isto é, em Tabelas), permitindo que usuários autorizados apliquem as mesmas formas de manipular dados aplicadas aos dados comuns a os consultá-las.

**Regra da sublinguagem compreensiva** – Um sistema relacional pode suportar várias linguagens e formas de uso; porém, deve possuir ao menos uma linguagem com sintaxe bem definida e expressa por cadeia de caracteres e com habilidade de apoiar a definição de dados, a definição de visões, a manipulação de dados, as restrições de integridade, a autorização e a fronteira

transações.

UCB – Banco de Dados – Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### As 12 Regras para um SGBD\_R

**Regra da atualização de visões** – Toda visão que for teoricamente atualizável será também atualizável pelo sistema.

**Inserção, atualização e eliminação de alto nível** – A capacidade de manipular a relação base ou relações derivadas como um operador único não se aplica apenas à recuperação de dados, mas também à inserção, alteração e eliminação de dados.

#### Independência dos dados físicos:

Programas de aplicação ou atividades de terminal permanecem logicamente inalterados, quaisquer que sejam as modificações na representação de armazenagem ou métodos de acesso internos.

Independência lógica de dados.

Programas de aplicação ou atividades de terminal permanecem logicamente inalterados quaisquer que sejam as mudanças de informação que permitam teoricamente a não alteração das tabelas base.

**Independência de integridade** – As relações de integridade específicas de um banco de dados relacional específico devem ser definidas em uma sublinguagem de dados e armazenadas no catálogo (e não em programas).

**Independência de distribuição** – A linguagem de manipulação de dados deve possibilitar que as aplicações permaneçam inalteradas, estejam os dados centralizados ou distribuídos fisicamente.

**Regra da não-subversão** – Se o sistema relacional possui uma linguagem de baixo nível (um registro por vez), não deve ser possível subverter ou ignorar as regras de integridade e restrições definidas no alto nível (muitos registros por vez).

UCB – Banco de Dados – Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

### Características dos Sistemas Relacionais

A maioria dos Sistemas de Banco de Dados desenvolvidos nos últimos anos é relacional. Um Sistema Relacional é aquele em que:

- ✓ Os arquivos convencionais são tratados como Tabelas Relacionais ou simplesmente Tabelas. Os dados são percebidos pelos usuários como Tabelas.
- ✓ Os operadores à disposição do usuário são operadores que geram novas tabelas, a partir das antigas.
- ✓ As linhas (tuplas) das Tabelas são comparadas aos registros dos arquivos (registros lógicos).
- ✓ As colunas das tabelas são comparadas aos campos desses registros lógicos.
- ✓ As operações SELECT, INSERT, UPDATE, DELETE são exemplos de instruções da linguagem de Banco de Dados chamada SQL (*Structured Query Language*).
- ✓ Segue um exemplo de banco de dados simplificado (DATE, 2004).
- ✓ Um banco de dados de adega de vinhos, que contém somente uma Tabela – ADEGA:



UCB – Banco de Dados - Prof. Milton

## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

Segue um exemplo de banco de dados simplificado (DATE, 2004).

Um banco de dados de adega de vinhos, que contém somente uma Tabela – ADEGA:

Código	Tipo	Produtor	Safrá	Estoque	Comentário
2	Chardonnay	Buena Vista	83	1	
3	Chardonnay	Louis Martini	81	5	
6	Chardonnay	Chappellet	82	4	Thanks giving
11	Jo. Riesling	Jeken	84	10	
12	Jo. Riesling	Buena Vista	82	1	Late Harvest
16	Jo. Riesling	Sattui	82	1	Very Dry
21	Fume Blanc	Ch. St. Jean	79	4	Napa Valley
22	Fume Blanc	Robt. Mondavi	78	2	
25	Wh. Burgundy	Mirassou	80	6	
30	Gewurztraminer	Buena Vista	80	3	
43	Cab. Sauvignon	Robt. Mondavi	77	12	
50	Pinot Noir	Mirassou	77	3	Harbest
51	Pinot Noir	Ch. St. Jean	78	2	
64	Zinfandel	Mirassou	77	9	Anniversary
72	Gamay	Robt. Mondavi	78	2	



UCB – Banco de Dados - Prof. Milton



## Unidade 4 - Modelo Relacional - Modelos de Banco de Dados

Distinção de um sistema relacional do não-relacional:

- ✓ Em um sistema relacional, o usuário vê os dados como Tabelas.
- ✓ Em um sistema não-relacional, o usuário vê outras estruturas de dados. Essas estruturas necessitam de outros operadores para manipulá-las. Por exemplo: um sistema hierárquico apresenta os dados aos usuários sob a forma de um conjunto de estrutura de árvores e os operadores, para manipulá-las, incluem operadores para percorrerem caminhos hierárquicos para cima e para baixo das árvores.

Como dissemos antes, o Modelo Relacional se baseia na Teoria dos Conjuntos. Vamos relembrar algumas informações importantes para nossa compreensão sobre esse modelo.



## Unidade 5 - Dependência Funcional e Normalização

### Teoria da Normalização

Proposta por Codd (1970), a normalização é baseada na matemática relacional, sendo o processo de decomposição de um esquema de relação em outros esquemas de relação (executado sobre as definições de dados ou atributos, para evitar inconsistências nos dados) que: não tem informação redundante; poderia gerar problemas de atualização; mantém a semântica original dos dados: mesmo conjunto de restrições de integridade sem perda de dados e seus relacionamentos. Principais conceitos envolvidos:

- ✓ **Dependências funcionais (FDs)** – Capturam a semântica dos dados.
- ✓ **Junção natural** – Ajuda a capturar o conceito de perda de dados.
- ✓ **Formas Normais (NFs)** – Definem quando um esquema de relação está livre de (certas) redundâncias.



## Unidade 5 - Dependência Funcional e Normalização

### Dependência funcional

Dada uma relação R, o atributo Y de R é funcionalmente dependente do atributo X de R ( $R, X \rightarrow R.Y$ ) (X determina funcionalmente Y) se, e apenas se, cada valor de X em R for associado ao menos a um valor Y em R. (Obs. X e Y podem ser um grupo de atributos de R).



## Unidade 5 - Dependência Funcional e Normalização

Normalização - Redundância e anomalias de atualização.

### Forma Normal

Uma das maneiras de se classificar as relações que compõem um banco de dados relacional é quanto à forma normal na qual a mesma se encontra.

Codd (1970), quando propôs o Modelo de Dados Relacional, apresentou em artigo correlato as seguintes formas normais: 1FN, 2FN, e 3FN. Ao longo do tempo, novas formas normais foram acrescentadas: BCNF, 4FN, 5FN,..., com o propósito de se dar mais precisão à atividade de projeto lógico de dados.

O processo de normalização é feito por camadas, isto é, só se pode passar para uma forma normal x se tiver passado pela normalização da camada x-1.

Para normalizar atributos, na Análise de Dados, devemos derivar o MER, buscando uma forma mais detalhada e eliminando anomalias comuns no Mundo-Real. Na prática, o grau de normalização necessário depende da aplicação. Estruturas de dados normalizadas até o terceiro grau, são consideradas normalizadas.

São benefícios da normalização: eliminar redundâncias; eliminar anormalidades nos dados; evitar potenciais inconsistências; tornar futuras alterações no modelo mais simples; tornar o modelo mais inteligível para o pessoal técnico encarregado da revisão do modelo; montar um modelo para a fase de projeto físico que seja independente da implementação do banco de dados escolhido.



## Unidade 5 - Dependência Funcional e Normalização

### Primeira Forma Normal (1FN)

- ✓ A entidade não contém grupos de dados repetitivos (um funcionário tem várias informações de função e salário).
- ✓ Existe uma chave candidata que identifica univocamente cada ocorrência da entidade.
- ✓ Todas as ocorrências da entidade têm exatamente o mesmo número de atributos.

#### Os passos para se chegar a 1FN são:

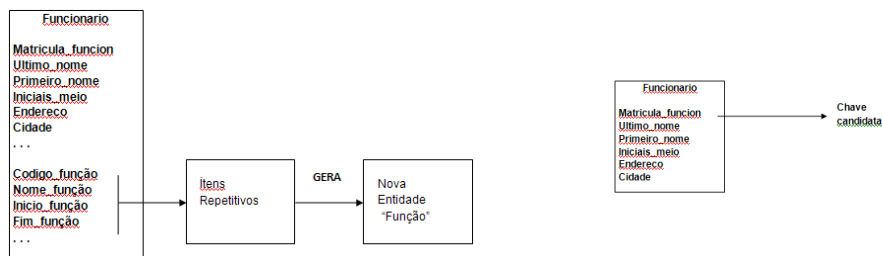
- ✓ Identificar grupos repetitivos.
- ✓ Separar os grupos repetitivos da entidade principal.
- ✓ Criar novas entidades para cada um dos grupos repetitivos.
- ✓ Identificar uma chave candidata que descreva a entidade.



UCB – Banco de Dados - Prof. Milton

## Unidade 5 - Dependência Funcional e Normalização

### Primeira forma normal (1FN)



Neste passo da normalização sobraram alguns atributos (**Nome\_funcao**, **Código\_funcao**, **Início\_funcao**, **Fim\_funcao**), que futuramente serão transformadas em novas Entidades (**Função** e **Evolucao\_funcional**).



UCB – Banco de Dados - Prof. Milton

## Unidade 5 - Dependência Funcional e Normalização

### Segunda Forma Normal (2FN)

Os dois próximos graus de forma normal são relativos aos relacionamentos entre atributos que não sejam Chaves Candidatas. Na 2FN, um atributo que não seja chave não pode fornecer informações relativas a outros atributos que não sejam chave, nem a atributos que sejam parte da chave.

Uma entidade está na 2FN se ela está na 1FN e todos os atributos que não sejam chave, estão relacionados à totalidade da chave primária

Os passos para se chegar a 2FN para cada entidade que já esteja na 1FN:

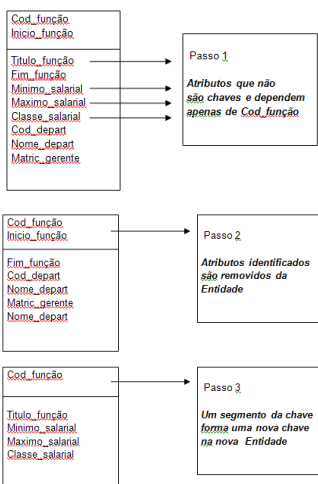
- ✓ Identificar todos os atributos que não sejam dependentes de toda a chave primária.
- ✓ Retirar estes atributos da entidade.
- ✓ Criar uma nova entidade com estes atributos.



UCB – Banco de Dados - Prof. Milton

## Unidade 5 - Dependência Funcional e Normalização

### Segunda Forma Normal (2FN)



Nesta Entidade, que já se encontrava na 1FN, descobrimos que certos atributos que não fazem parte da chave se referem apenas ao código da função (Cod\_funcao).

No exemplo anterior, a nova Entidade inclui o atributo Cod\_funcao. Conceitualmente, esta informação é desnecessária nesta fase do projeto, mas, quando o Modelo Lógico se transformar em um Modelo Físico, ele será necessário para fazer a recuperação da informação. Portanto é comum incluir este atributo neste ponto do Modelo Conceitual



UCB – Banco de Dados - Prof. Milton

## Unidade 5 - Dependência Funcional e Normalização

### Terceira Forma Normal (3FN)

A 3FN continua fornecendo informações sobre o relacionamento entre atributos de cada entidade. Nesse nível, estaremos procurando atributos não chave que descrevem outros atributos não chave.

Uma entidade está na 3FN se:

- ✓ Estiver na 2FN.
- ✓ Possuir atributos não chave que não são dependentes ou não descrevem outros atributos não chave.

Os passos para se chegar a 3FN são:

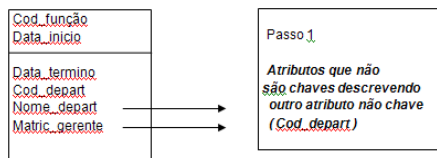
- ✓ Identificar atributos não chave, que descrevem ou dependem de outros atributos que não são chave.
- ✓ Remover estes atributos da entidade.
- ✓ Criar uma nova entidade com os atributos identificados.

Outra recomendação da 3FN é que uma entidade não deve conter atributos derivados de cálculo, por exemplo, valor\_total\_venda.



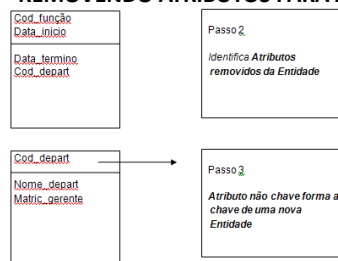
## Unidade 5 - Dependência Funcional e Normalização

### Terceira Forma Normal (3FN)



No exemplo anterior, vemos que os dois atributos descrevem ou associam fatos somente a outro atributo não chave. A informação destes atributos não dependem da Chave Candidata.

#### REMOVENDO ATRIBUTOS PARA A TERCEIRA FORMA NORMAL



Outra recomendação da 3FN é que uma entidade não deve conter atributos derivados de cálculo, por exemplo, valor\_total\_venda.



## Unidade 5 - Dependência Funcional e Normalização

Forma-Normal Boyce-Codd - BCNF.

*uma tabela está na forma normal BCNF quando todo determinante existente na tabela é chave candidata.*

Caso uma relação possua chaves candidatas com atributos sobrepostos, poderemos obter imperfeições

exemplo

abastecimento

nbomb  
capac  
codfun  
nomfun  
data  
hora  
codcar  
placa  
descrVeiculo  
kmOdômetro  
litros

**Forma Normal de Boyce-Codd (BCNF)**

*a tabela teria 3 chaves candidatas*

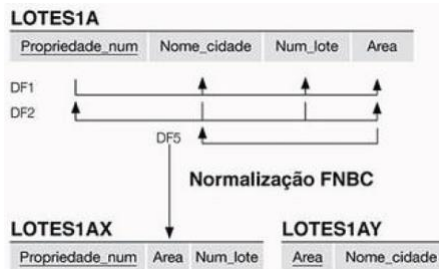
<http://www.dalfovo.com/Bogo/informacoes/programa/bdnormalizacao.htm#bcfn>



## Unidade 5 - Dependência Funcional e Normalização

Forma-Normal Boyce-Codd - BCNF.

Uma coisa interessante é que ela foi proposta como uma forma mais simples que 3FN, porém mais rígida. Se uma relação está na FNBC, também está na 3FN. Sua definição diz o seguinte: uma relação está na FNBC se todo determinante é chave candidata. Abaixo segue uma figura que demonstra uma normalização de uma relação para a FNBC.



<http://www.itnerante.com.br/profiles/blogs/desmistificando-as-formas-normais>



## Unidade 5 - Dependência Funcional e Normalização

Forma-Normal Boyce-Codd - BCNF.

As formas normais vistas até o momento são apoiadas em dependências funcionais (2FN, 3FN e BCNF) e podem ser consideradas para cada relação. Uma base de dados será considerada normalizada para uma dessas formas quando todas as suas relações se apresentarem nessa forma.

Tratamos das formas normais mais “simples” e relacionadas à DF, espero que você tenha entendido. Qualquer dúvida ou comentário pode ser enviado para mim aqui mesmo, nos comentários abaixo, ou no meu perfil do Itnerante. Mas, o objetivo deste artigo é: desmistificar a quarta e a quinta forma normal. Então, prepare sua mente, pois as próximas linhas deixarão todo assunto bem claro. Vem comigo!

Primeira informação que devemos ter que faz parte da base do conhecimento das próximas formas normais é a ideia de **decomposição sem perdas na junção**. A decomposição deve ser feita de maneira que, quando se recompõe a relação original, apenas e exatamente as tuplas existentes na relação original são reobtidas. A decomposição baseada nas dependências funcionais (2 FN, 3FN e FNBC) não causam perdas de junção, portanto a normalização para as formas normais baseadas em dependências funcionais estão livres desse problema.

<http://www.itnerante.com.br/profiles/blogs/desmistificando-as-formas-normais>



UCB – Banco de Dados - Prof. Milton

## Unidade 5 - Dependência Funcional e Normalização

Forma-Normal Boyce-Codd - BCNF.

Contudo, a normalização de relações através de dependências funcionais é apenas uma das maneiras, embora a mais importante, de evitar inconsistências em relações. Outra maneira, advém de uma variação das dependências funcionais, chamadas **dependência multivalorada (DMV)**, **multi-dependência funcional (MDF)**, ou ainda, **dependência multivalor**.

E o que seria essa tal dependência multivalorada? A **dependência multivalorada** caracteriza o fato de que, embora um conjunto de atributos não possa determinar o valor de outro atributo, ainda assim, esse conjunto consegue restringir os valores possíveis para aquele atributo.

Vamos a um exemplo simples. Uma pessoa pode ter vários números de telefones e um número de telefone pode ser compartilhado por várias pessoas. Nesse relacionamento N para M não é possível encontrar uma DF entre esses dois atributos, porém podemos visualizar o que chamamos de dependência multivalorada entre o Id\_pessoa e Num\_telefone que pode ser representada por Id\_pessoa  $\twoheadrightarrow$  Num\_telefone. Diz-se que Id\_pessoa multidetermina Num\_telefone, ou, que Num\_telefone é funcionalmente multidependente de Id\_pessoa.

Aqui cabe uma observação a respeito das DMV. Elas são consequência da 1FN que não permite que um atributo tenha um conjunto de valores. Veja um exemplo entre Funcionário, Projeto e Dependente.

Funcionario	Projeto	Dependente
Thiago	Proj_01	Vinicius
Thiago	Proj_01	Maria Clara
Thiago	Proj_01	Gustavo
Thiago	Proj_02	Vinicius
Thiago	Proj_02	Maria Clara
Thiago	Proj_02	Gustavo

<http://www.itnerante.com.br/profiles/blogs/desmistificando-as-formas-normais>



UCB – Banco de Dados - Prof. Milton

## Unidade 5 - Dependência Funcional e Normalização

Veja que na tabela acima podemos encontrar duas DMV. Funcionário multidetermina Projeto e Funcionário multidetermina Dependente.


Sempre que houver a ocorrência de um atributo multivalorado ocorrerá a multi-dependência funcional. Entretanto, se houver duas ou mais **dependências multivaloradas** independentes entre si na mesma relação então pode ocorrer anomalias de atualização na relação. Tente inserir na tabela a informação que Thiago participará do projeto 03 (proj 03). Para mantermos a DMV, teremos que adicionar 3 linhas, uma para cada dependente de Thiago, caso contrário ao procurarmos os dependentes dos funcionário que participam do projeto 03 retornaremos uma informação incorreta.

Se tivermos um ou mais atributos multivalorados independentes na mesma relação, temos de repetir, gerando todas as combinações entre os atributos para manter a consistência entre as instancias.

Observe a relação (Funcionario, Projeto e Dependente). Veja que, ainda que esteja na FBNC, a mesma ainda apresenta redundância. Observe também que a decomposição não pode se basear em DF, pois não existem DF na relação. É necessário, portanto uma regra para o tratamento dessas situações, que possa ser usada para decompor a relação sem perdas.

Numa relação  $R(A, B, C)$  se existe uma DMV  $A \twoheadrightarrow B$  também existe  $A \twoheadrightarrow C$  (ou seja,  $A \twoheadrightarrow R - AB$ ). Nestes casos, as DMV surgem sempre aos pares e representam-se por:  $A \twoheadrightarrow B|C$ . Apenas reforçando, se  $A \twoheadrightarrow B|C$ , então a relação deve conter todas as combinações possíveis dos conjuntos de valores de B e de C, associados ao mesmo valor de A.

Formalmente podemos pensar da seguinte forma: dada uma relação R com atributos A, B, C, existe uma dependência multivalorada do atributo A no atributo B ( $A \twoheadrightarrow B$ ) se, um valor de A é associado a uma coleção específica de valores de B, independente de quaisquer valores de C.

<http://www.itnerante.com.br/profiles/blogs/desmistificando-as-formas-normais> 

UCB – Banco de Dados – Prof. Milton

## Unidade 5 - Dependência Funcional e Normalização

### Quarta forma normal (4FN)

Começamos com uma definição informal: Uma relação está na 4FN se para qualquer DMV  $X \twoheadrightarrow Y$  a relação não tem outros atributos além dos que fazem parte de X e de Y. Esse tipo de DMV é conhecido como trivial.

Outra definição, desta vez, de acordo com o Navathe. "Um esquema de relação R está na 4FN com relação a um conjunto de dependências funcionais ou multivaloradas F se, para toda **dependência multivalorada não trivial**  $X \twoheadrightarrow Y$  em F+, X for uma superchave de R."

Formalmente, o conjunto de todas as dependências de F, bem como todas as dependências que podem ser inferidas para F, é chamado de clausura de F, que é denotada por F+.

A DMV pode ser classificada como **trivial** ou **não trivial**. Numa relação R, a DMV  $A \twoheadrightarrow B$  é dita trivial se:

- (a) B for subconjunto de A ou
- (b)  $A \cup B = R$ .

Uma DMV que não satisfaz nem a (a) nem a (b) é dita **não trivial**. Em outras palavras, de uma maneira mais intuitiva, pode-se dizer que uma DMV não trivial ocorre sempre que houver mais do que um atributo multivalorado na mesma relação. Assim, o que se procura é a chamada DMV **não trivial**. Observe que as relações que contêm DMVs tendem a ser all-key (tudo é chave) – ou seja, sua chave é formada por todos os seus atributos tomados em conjunto.

O processo de normalização de uma relação envolvendo DMVs não triviais, que não está na 4FN, consiste em decompô-la de modo que cada DMV seja representada por uma relação separada, onde se torna uma DMV trivial. Veja o nosso exemplo para a relação (Funcionario, Projeto, Dependente), que será decomposta em duas relações (Funcionario, Dependente) e (Funcionario, Projeto).

<http://www.itnerante.com.br/profiles/blogs/desmistificando-as-formas-normais> 

UCB – Banco de Dados – Prof. Milton



## Unidade 5 - Dependência Funcional e Normalização

### Quarta forma normal (4FN)

Funcionario	Projeto	Dependente
Thiago	Proj. 01	Vinicius
Thiago	Proj. 01	Maria Clara
Thiago	Proj. 01	Gustavo
Thiago	Proj. 02	Vinicius
Thiago	Proj. 02	Maria Clara
Thiago	Proj. 02	Gustavo

Funcionario	Dependente
Thiago	Vinicius
Thiago	Maria Clara
Thiago	Gustavo

Funcionario	Projeto
Thiago	Proj. 01
Thiago	Proj. 02

Vimos que a propriedade sem perda na junção é uma das diversas propriedades para o projeto de banco de dados. De fato, essa propriedade é essencial: sem ela, há perdas de informação. Quando restringimos o conjunto de relações válidas entre as que satisfazem um conjunto de dependências funcionais e multivaloradas, podemos usar essas dependências para mostrar que certas decomposições são decomposições sem perda na junção.

A próxima e última forma normal tem relação direta com Dependência de Junção (DJ), por isso, ela também é conhecida como Forma Normal Projeção Junção.

Quando é possível restringir um conjunto de relações válidas sobre um esquema R para aquelas relações para as quais uma dada decomposição é uma decomposição sem perdas, podemos definir essa restrição como uma dependência de junção.

Em outras palavras, uma dependência de junção (DJ), denotada por DJ (R1, R2, ..., Rn), em um esquema de relação R, especifica uma restrição nos estados r de R. Essa restrição diz que todo estado legal r de R deveria ter uma decomposição de junção não aditiva para R1, R2, ..., Rn, ou seja, para todo r tenham

$*(\pi R_1(r), \pi R_2(r), \dots, \pi R_n(r)) = r$  ( $\pi$  é uma projeção sobre a relação R)

<http://www.itnerante.com.br/profiles/blogs/desmistificando-as-formas-normais>  Copyright

UCB – Banco de Dados - Prof. Milton

## Unidade 5 - Dependência Funcional e Normalização

### Quinta forma normal (5FN)

A aplicação da 5FN consiste em encontrar a DJ \* [R1, R2, ..., Rp] que permite decompor uma relação sem perdas

Adém das dependências multivaloradas que ocorrem entre os atributos de uma relação. A verificação da 5ª FN somente precisa ser empreendida em relações que tenham três ou mais atributos como parte da chave.

A 5ª FN trata da situação em que a informação permite ser reconstruída a partir de componentes menores que possam ser mantidos com uma redundância menor. Ela generaliza os casos não cobertos pela segunda, terceira e quarta formas normais.

**Definição:** Um esquema de relação R está na quinta forma normal (5ª FN) em relação a um conjunto F de dependências funcionais, multivaloradas e de junção se, para cada dependência de junção não trivial DJ (R1, R2, ..., Rn) de F+ (ou seja implicada por F), todo Ri for uma superchave de R.

Abaixo segue o exemplo da relação Fornece, que não pode ser decomposta em duas relações, pois a junção entre elas geraria tuplas espúrias. Assim, usamos a 5FN para decompor a relação em três (r1, r2, r3), de modo que uma junção feita sobre essa relação mantém a propriedade de junção sem perdas.

<http://www.itnerante.com.br/profiles/blogs/desmistificando-as-formas-normais>  Copyright

UCB – Banco de Dados - Prof. Milton

## Unidade 5 - Dependência Funcional e Normalização

### Quinta forma normal (5FN)

FORNECE

Nome_fornec	Nome_pec	Nome_proj
Silva	Peneira	ProjX
Silva	Porca	ProjY
Adam	Peneira	ProjY
Walter	Porca	ProjZ
Adam	Preço	ProjX
Adam	Peneira	ProjX
Silva	Peneira	ProjY

R1

Nome_fornec	Nome_pec
Silva	Peneira
Silva	Porca
Adam	Peneira
Walter	Porca
Adam	Preço

R2

Nome_fornec	OR_proj
Silva	ProjX
Silva	ProjY
Adam	ProjY
Walter	ProjZ
Adam	ProjX

R3

Nome_pec	OR_proj
Peneira	ProjX
Porca	ProjY
Peneira	ProjY
Porca	ProjZ
Preço	ProjX

<http://www.itnerante.com.br/profiles/blogs/desmistificando-as-formas-normais>



UCB – Banco de Dados - Prof. Milton

## Unidade 5 - Dependência Funcional e Normalização

Ver mais BCNF, 4FN e 5FN.

<http://www.fernandolobo.info/bd1718/slides/bcnf3nf.pdf>

<https://www.ime.usp.br/~jef/bd09.pdf>



UCB – Banco de Dados - Prof. Milton

## Unidade 5 - Dependência Funcional e Normalização

### Tipos de Chave

Chave é um atributo (chave simples) ou conjunto de atributos (chave composta) de uma relação, cujos valores sejam únicos, e que identifiquem univocamente suas tuplas. As chaves em bancos de dados são classificadas em:

- ✓ **Superchave (superkey)** – É uma combinação de atributos capaz de identificar suas tuplas univocamente. Uma relação pode ter diversas superchaves. Uma relação possui pelo menos uma superchave composta por todos os seus atributos.
- ✓ **Candidata (candidate key)** – São os atributos identificadores dos objetos que os distinguem dos demais (identifica univocamente uma Entidade no MER, ou que a distingue dos demais) e que no Modelo Físico de Dados poderão ser usados como Chaves Primárias (*primary key*). As chaves definitivas são obtidas ao final da fase de normalização.
- ✓ **Primária (primary key)** – Atributo(s) que identifica(m) de forma unívoca uma ocorrência na relação. É uma chave candidata escolhida para ser a chave primária. O critério de escolha de uma chave primária se baseia na quantidade de atributos das chaves candidatas, dando-se preferência por aquela que tiver a menor quantidade de atributos. A chave primária é única numa relação. A chave primária é utilizada como base para a restrição de integridade de entidade, como será visto adiante.
- ✓ **Secundária (secondary key)** – Difere de uma chave primária pelo fato de não possuir um valor diferente para cada registro; assim, uma chave secundária identifica um conjunto de registros. É uma combinação de um ou mais atributos de uma relação, com o propósito de se criar uma estrutura de acesso mais otimizada.
- ✓ **Alternadas (alternate key)** – É aquela que não foi aproveitada como chave primária, porém também identifica um registro univocamente. Por exemplo: se a chave primária no arquivo aluno for matrícula, o CPF poderá ser a chave alternativa.
- ✓ **Estrangeira (foreign key)** – É uma combinação de um ou mais atributos de uma relação que funciona como chave primária em outra relação. É por meio da chave estrangeira que se representa logicamente os relacionamentos entre as tabelas. A chave estrangeira é utilizada como base para a restrição de integridade referencial, como será visto adiante.

UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

### Avaliação

#### Prova 1.

UCB – Banco de Dados - Prof. Milton

Copyright  
Todos os Direitos Reservados

## Unidade 6 - Álgebra Relacional

- Operadores básicos.
- Seleção.
- Projeção.
- Produto Cartesiano.
- Operadores complementares da Álgebra relacional.

[http://www.macoratti.net/13/06/sql\\_arcb.htm](http://www.macoratti.net/13/06/sql_arcb.htm)

<https://www.youtube.com/watch?v=a15YNisBKhU>

<http://wiki.icmc.usp.br/images/2/2c/SCC578920131-algebraSQL.pdf>



## Unidade 6 – Cálculo Relacional

- Operadores básicos.
- Seleção.
- Projeção.
- Produto Cartesiano.
- Operadores complementares da Álgebra relacional.

<https://www.ime.usp.br/~jef/calcrelac.pdf>

<https://www.dcc.fc.up.pt/~ricroc/aulas/1011/bd/apontamentos/parteIV.pdf>

<https://www.youtube.com/watch?v=hYogueV5B8Q>



## Unidade 6 - Álgebra Relacional

### Cálculo Relacional

O cálculo relacional de tuplas e o cálculo relacional de domínio são linguagens não-procedurais que representam a base de poder para uma linguagem relacional de consulta. A álgebra relacional é uma linguagem procedural que possui poder equivalente a ambas as formas de cálculo relacional quando é garantida por expressões de segurança. A álgebra define as operações básicas usadas dentro das linguagens relacionais de consulta.

### Álgebra Relacional


A álgebra relacional e o cálculo relacional são linguagens formais e sucintas. Seu uso não é muito apropriado para usuários comuns de sistema de banco de dados. É recomendável que os bancos de dados comerciais usem linguagens mais acessíveis para os usuários, e a linguagem SQL vem garantir essa adequação de maneira fluente no mercado. Como já citamos anteriormente, a SQL usa as estruturas da álgebra e cálculo relacional, mas sua característica é de uma linguagem mais branda, para manusear o sistema de banco de dados; apesar de a mesma ter sido desenvolvida especialmente para o ambiente relacional, pode ser adaptada a qualquer ambiente não relacional.



## Seminário


Seminário: Entrega dos projetos.






**Avaliação**

## Prova 2.


 Copyright  
Todos os Direitos Reservados

UCB – Banco de Dados - Prof. Milton



**Avaliação**

## Prova Substitutiva.

 Copyright  
Todos os Direitos Reservados

UCB – Banco de Dados - Prof. Milton

## Aula Síntese

Aula Síntese e divulgação de resultados.

Revisão da matéria e análise da disciplina no contexto do curso e da sociedade.



## Bibliografia Básica

DATE, C. J. Introdução a sistemas de bancos de dados. Rio de Janeiro, RJ: Elsevier, Campus, c2004. 865 p.

ELMASRI, R., NAVATHE, S. B., Sistemas de Bancos de Dados, 4ª Edição, 2005, Editora Pearson /Addison Wesley.

Korth, H. F. e Silberschatz, A. Sistemas de Banco de Dados, São Paulo, McGraw-Hill.





## Bibliografia Complementar

DATE, C. J. Banco de dados: Tópicos avançados. Rio de Janeiro: Campus, 1988. 361p.

GILLENSON, Mark L. Fundamentos de sistemas de gerência de banco de dados. Rio de Janeiro, RJ: LTC, 2006.

HEUSER, Carlos Alberto. Projeto de banco de dados. 4. ed. Porto Alegre: Editora Sagra Luzzatto, 2001. xvi, 204 p.

BATINI, Carlo; CERI, Stefano; NAVATHE, Sham. Conceptual database design: An entity-relationship approach. Redwood City, Ca.: The Benjamin/Cummings, c1992. 470 p.

SETZER, Valdemar W. Bancos de dados: conceitos, modelos, gerenciadores, projeto lógico, projeto físico. 3. ed São Paulo: Edgard Blücher, 1989. 289 p.

