Banco de Dados I

Prof. Allan

Aula Inaugural

Bibliografia Básica

- CHEN, P. Gerenciamento de Banco de Dados. São Paulo: McGraw-Hill, 1990.
- DATE, C. J. Introdução a Sistemas de Banco de Dados. 7ª. Edição. São Paulo: Campus, 2000.
- ELMASRI, R.. NAVATHE, S. B., Sistemas de Banco de Dados – Fundamentos e Aplicações. 3ª. Edição. Rio de Janeiro: LTC, 2000.
- HEUSER, C. A. Projeto de Banco de Dados, 2001.
- SILBERSCHATZ, A: KORTH, H. F.; SUDARSHAN, S. Sistema de Banco de Dados, 2005.

Bibliografia Complementar

- Batini, C., Ceri. S., Navathe S.B. Conceptual Database Design: An Entity-relationship Approach. 1992
- Ramakrishnan, R e Gehrke, J. Sistemas de Gerenciamento de Banco de Dados. 3ª Ed., 2008

Conceitos Gerais

- Dado x Informação x Conhecimento

Introdução - BDs

persistência

Evolução: memória → arquivos → banco de dados

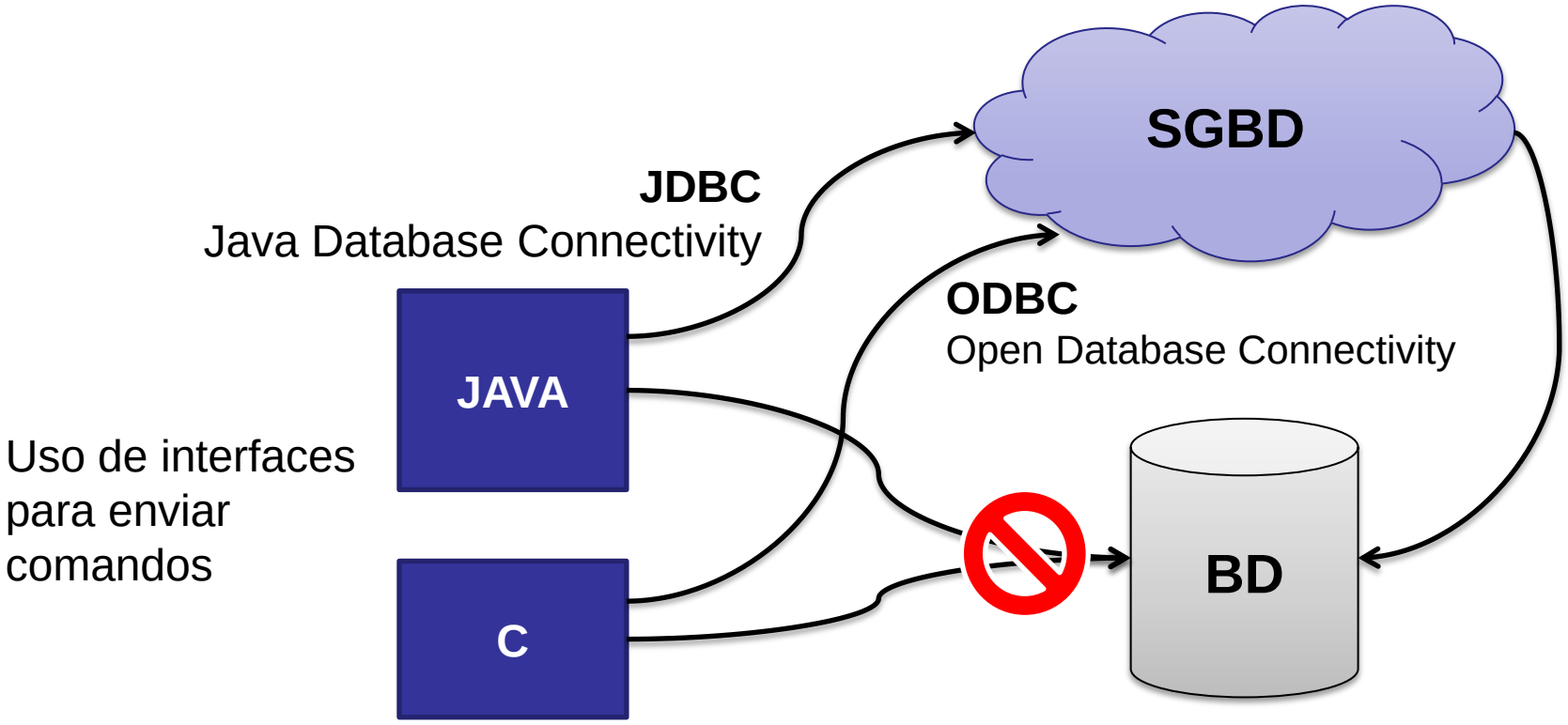
- um Banco de Dados (BD) é um conjunto de dados que representa aspectos do mundo real (universo de discurso) de forma organizada e que permite extrair informações [dado vs. informação]
  - composto por dados e metadados!
  - **Arquivo de dados**: conjunto de registros relacionados [entidade, tabela, relação]
  - **Registro**: conjunto de dados relacionados [tupla, linha]
  - **Campo / dado**: valor armazenado [atributo, coluna]

Introdução - SGBDs

Sistema Gerenciador de Banco de Dados (SGBD)

- É um software que gerencia o banco de dados:
  - criar e manter Banco de Dados
  - realizar consultas eficientes no Banco de Dados
  - fazer interface com aplicação
  - garantir desempenho, segurança e confiabilidade
- Aplicações:
  - Informação empresarial
  - Bancos e finanças
  - Universidades, companhias aéreas, telecomunicações
  - (...)

Introdução - SGBDs



Introdução - SGBDs

Vantagens

- Fornece processamento eficiente de consulta (índices, *caching*, otimização de consulta)
- Oferece mecanismos de backup e recuperação
- Pode assegurar restrições de integridade (valor único, integridade referencial, verificação de restrições)
- Criação de gatilhos e procedimentos

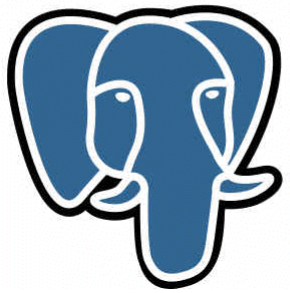
Desvantagens

- Requer algum investimento inicial ($, tempo, hardware)
- Pode não ser necessário caso aplicações sejam simples e bem definidas, sem mudanças

Principais SGBDs disponíveis no mercado

INGRES

PostgreSQL



ORACLE

MySQL

Histórico e Modelo de Dados

Modelos de Dados descrevem a semântica, os relacionamentos, as restrições e as operações com dados

- PRIMEIRA GERAÇÃO
  - Época: década de 1960 e 1970
  - Modelo: sistema de arquivos
  - Características:
    - ❖ gerenciamento de registros sem relacionamentos
    - ❖ utilizado principalmente em sistemas de mainframes da IBM
  - Exemplos: MVS / VSAM (sist. de arq. orientados a registro)

Histórico e Modelo de Dados

- SEGUNDA GERAÇÃO
  - Época: final da década de 1960 e década de 1970
  - Modelo: hierárquico e em rede (grafo)
  - Características:
    - ❖ primeiros sistemas de bancos de dados
    - ❖ acesso navegacional
    - ❖ modelo hierárquico organizado como estrutura de árvore (registro pode ter múltiplos filhos vinculados por *links*, mas um filho só pode ter um registro pai)
    - ❖ deficiência: falta de uma linguagem de consulta de alto nível
  - Exemplos: IMS, ADABAS, IDS-II

Histórico e Modelo de Dados

- TERCEIRA GERAÇÃO
  - Época: Meados da década de 1970 até hoje
  - Modelo: relacional
  - Características:
    - ❖ baseado em: entidades, atributos e relacionamentos / tabelas
    - ❖ simplicidade conceitual e de modelagem
    - ❖ teve grande aceitação e se transformou em um padrão
  - Exemplos: DB2, Oracle, MS SQL Server, MySQL

Histórico e Modelo de Dados

- QUARTA GERAÇÃO
  - Época: Meados da década de 1980 até hoje
  - Modelo: orientado a objetos e relacional estendido
  - Características:
    - ❖ suporte a conceitos e visão de orientação a objetos
    - ❖ suporte a dados complexos (dados multimídia, geográficos)
    - ❖ ajudam no problema da divergência de impedância (diferença entre a representação na aplicação e no BD)
    - ❖ deficiência: estrutura continua rígida
  - Exemplos: Versant, Objectivity/DB, DB/2 UDB, Oracle 10g

Histórico e Modelo de Dados

- QUINTA GERAÇÃO
  - Época: Meados da década de 1980 até hoje
  - Modelo: não estruturado ou semiestruturado (NoSQL)
  - Características:
    - ❖ representação dos dados (estrutura) flexível [ex XML]
    - ❖ não garante algumas propriedades (ex atomicidade)
    - ❖ desempenho melhor
    - ❖ voltado para aplicações de BigData
  - Exemplos: Google BigTable, CouchDB, MongoDB

Por que usar um SGBD?

Problemas da primeira geração de BDs (sem SGBDs):

- Redundância e Inconsistência: informação replicada com possíveis dados divergentes [ex: aluno de dois cursos]
- Dificuldade de consulta: dificuldade em filtrar dados de maneiras diferentes [ex: filtrar alunos que moram em uma determinada área e filtrar alunos em exame neste semestre]
- Isolamento dos dados: dificuldade em escrever programas para filtrar dados que estão em arquivos e possivelmente formatos/estruturas diferentes [ex: arquivos XML e CSV]
- Problemas de integridade: dificuldade em alterar todos os programas existentes a fim de garantir certas restrições de consistência [ex: saldo ≥ 0].

Por que usar um SGBD?

Problemas da primeira geração de BDs (sem SGBDs):

- Problema de atomicidade: dificuldade em garantir que toda uma transação é feita por completo ou nada é feito [ex: apagão]
- Problemas com acesso concorrente: o uso inadequado de múltiplos processos ou *threads* pode gerar dados inconsistentes ao trabalhar simultaneamente com um mesmo dado
  [ex: transferências bancárias]
- Problemas de segurança: garantir que apenas os usuários corretos possam ver determinados dados [ex: secretária não deveria poder ver os dados das folhas de pagamento; dados nos arquivos devem estar criptografados]

Por que usar um SGBD?

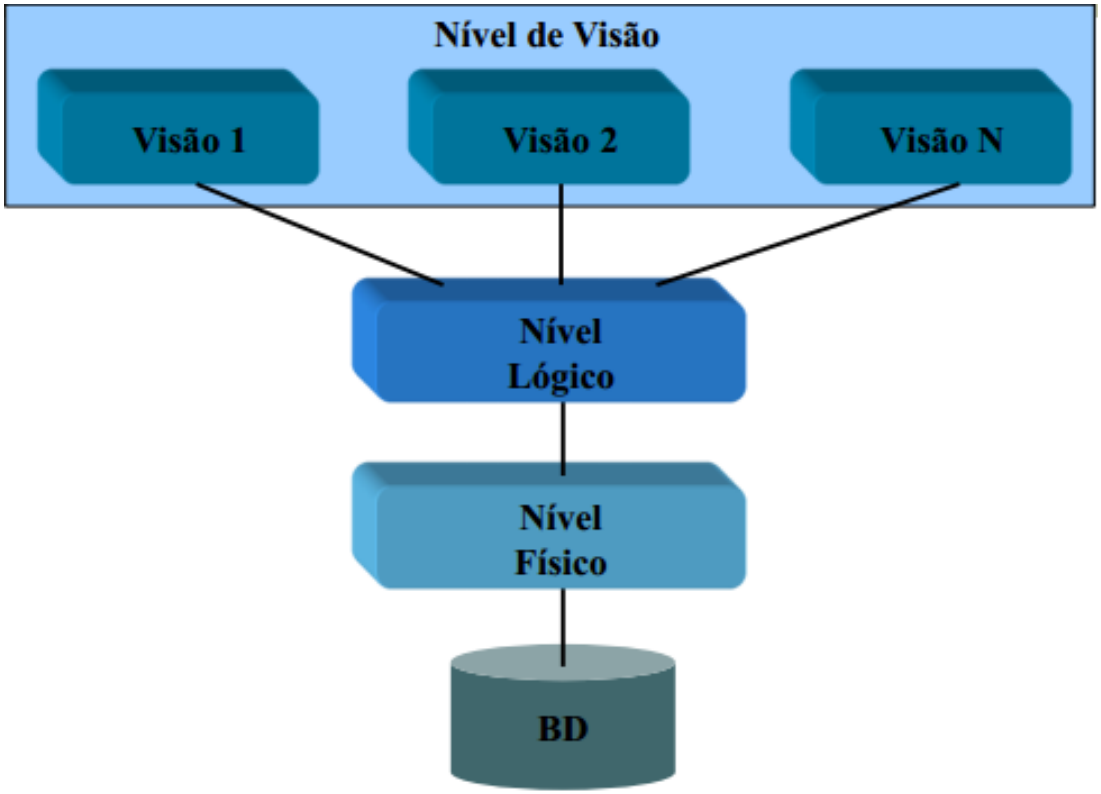
Contrapartida com o uso de um SGBD:

- Redundância controlada (requer bom uso do administrador)
- Eficiência na manipulação e consulta de dados
- Independência de dados (permite modificações estruturais)
- Pode garantir integridade dos dados (basta especificar)
- Pode garantir atomicidade nas operações
- Pode garantir segurança no acesso concorrente
- Pode garantir controle de acesso aos dados
- Proporciona diferentes níveis de abstração

Níveis de Abstração

Os SGBDs fornecem aos usuários uma visão abstrata dos dados e não exatamente como estão armazenados.

É comum o uso de uma abstração de 3 níveis:



Níveis de abstração

- Nível externo: possui as diversas descrições do BD de acordo com os grupos de usuários.

- Nível conceitual: descreve a estrutura de todo o BD para uma determinada comunidade de usuários, ocultando detalhes sobre a organização física dos dados e apresentando a descrição lógica dos dados e das ligações existentes entre eles.

Níveis de abstração

- Nível interno: descreve a estrutura de armazenamento físico dos dados do BD, descreve o modelo físico dos dados que inclui detalhes sobre os caminhos de acesso aos dados internamente.

Níveis de abstração

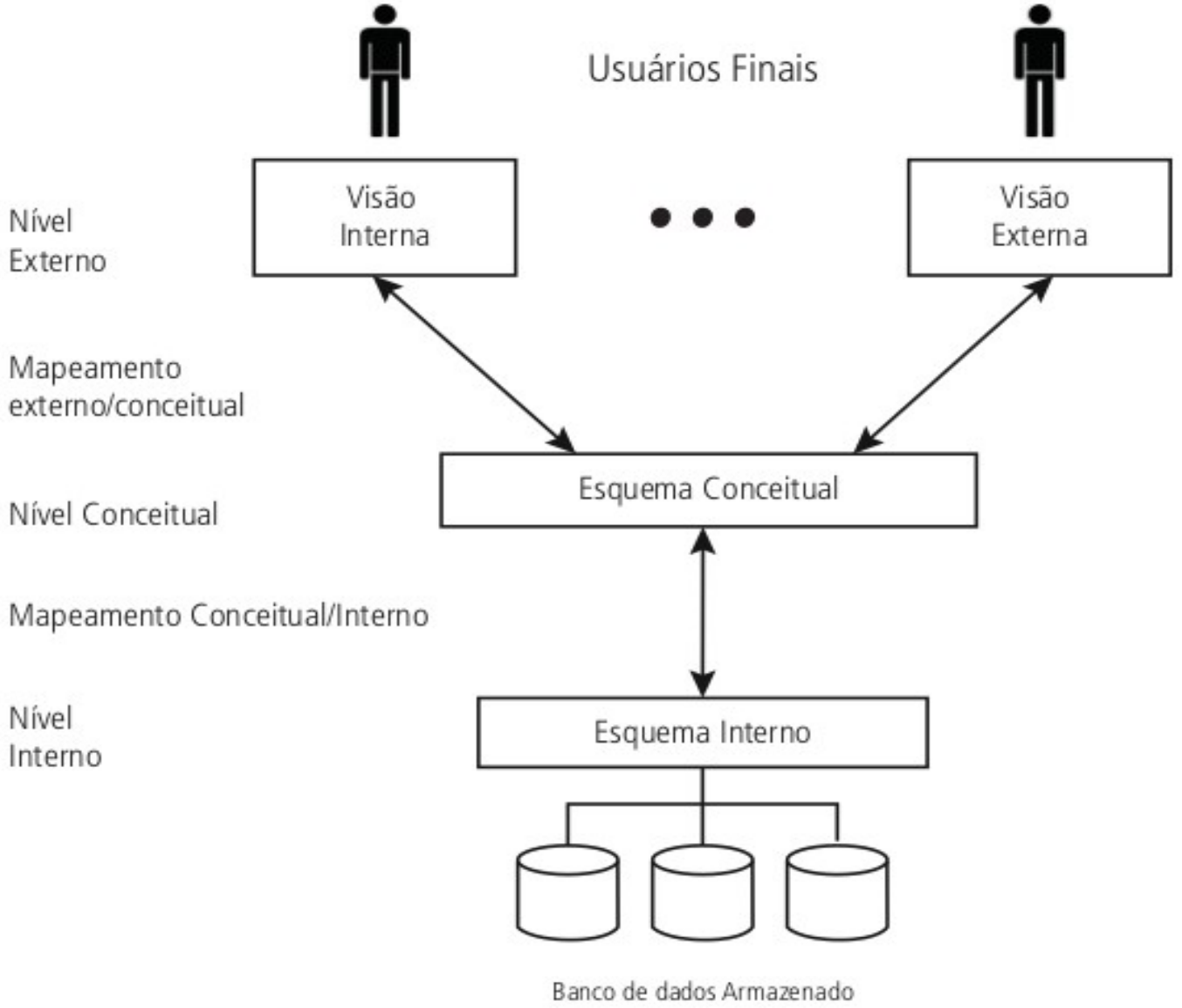


Figura 1.4: Visões de um banco de dados
Fonte: ELMASRI & NAVATHE, 2005

Níveis de Abstração

Independência de Dados: os níveis de abstração devem manter um grau de independência uns dos outros:

- Independência Lógica: capacidade de modificar o esquema lógico sem necessidade de reescrever as aplicações/visões
- Independência Física: capacidade de modificar o esquema físico sem necessidade de reescrever o modelo lógico.

Esquema: é a descrição dos dados do BD; seu projeto geral; costuma ser pouco alterado.

Instância ou Estado: coleção de dados armazenados no BD em um determinado momento (fotografia dos dados)

Propriedade ACID

BDs relacionais foram introduzidos com a proposta de garantir certas características. Estas se tornaram uma referência para os SGBDs, que podem implementá-las ou não. São elas:

- **Atomicidade:** realizar operações não divisíveis; transação é executada por completa ou não é executada

Transação é um conjunto de operações que realiza uma única função lógica

Propriedade ACID

- **Consistência:** o BD deve respeitar certas regras impostas (restrições de consistência) de tal forma que o BD esteja consistente antes e após qualquer transação

  - **Restrição de chave**: restringe duplicidade de chaves
  - **Restrição de domínio**: restrições de tipos
  - **Integridade de vazio**: restrição de obrigatoriedade de valor
  - **Integridade referencial**: garante o vínculo entre dois ou mais registros (chave estrangeira) em alterações ou exclusões
  - **Assertivas**: outras condições referentes aos dados que precisam ser satisfeitas

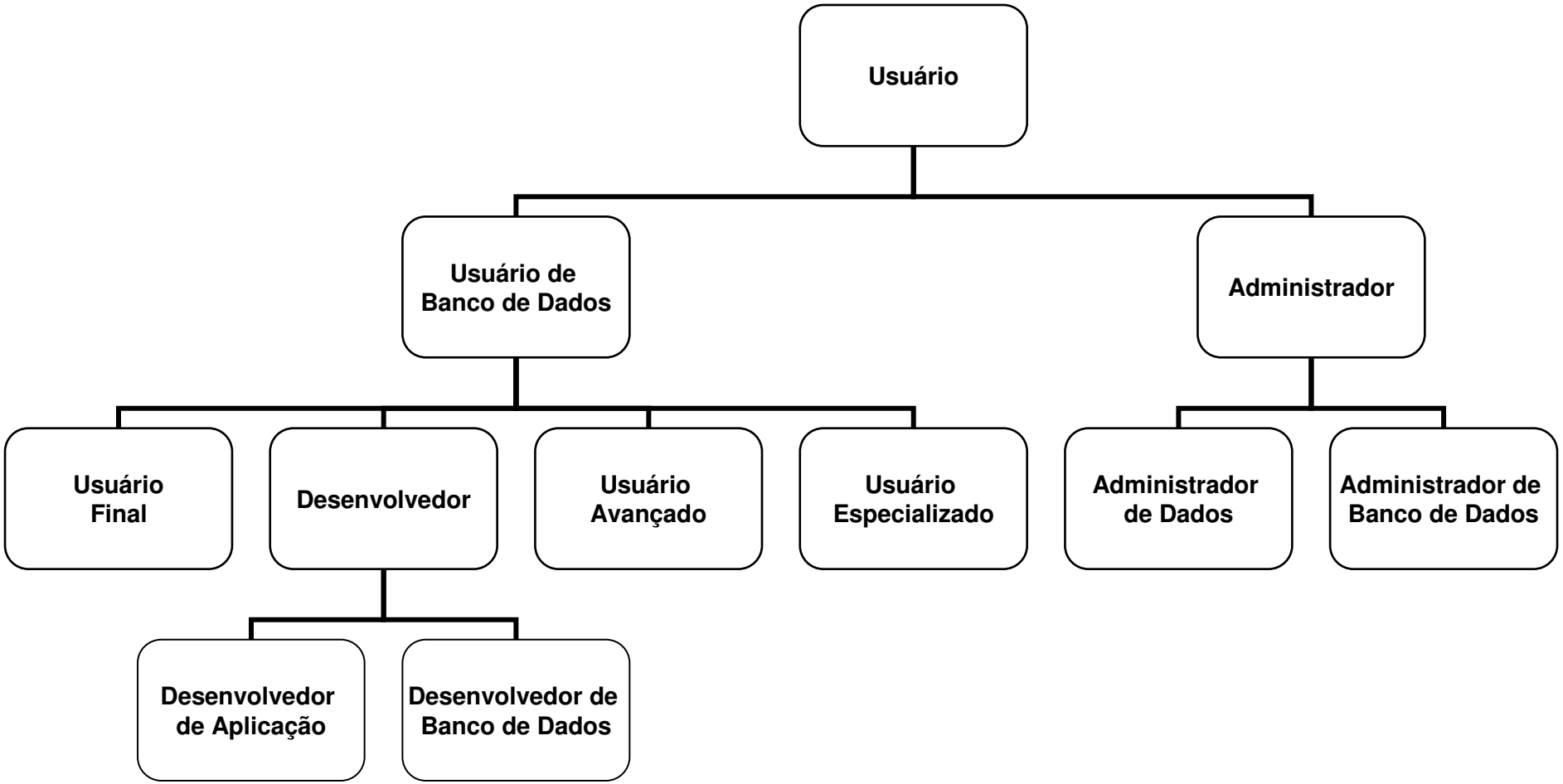
Propriedade ACID

- **Isolamento:** duas transações não podem se interferir gerando um resultado inconsistente (ex: transferência bancária). O SGBD pode, contudo, paralelizar transações que não atuam sobre um mesmo item

- **Durabilidade:** valores criados ou alterados em uma transação devem persistir, mesmo havendo falha no sistema

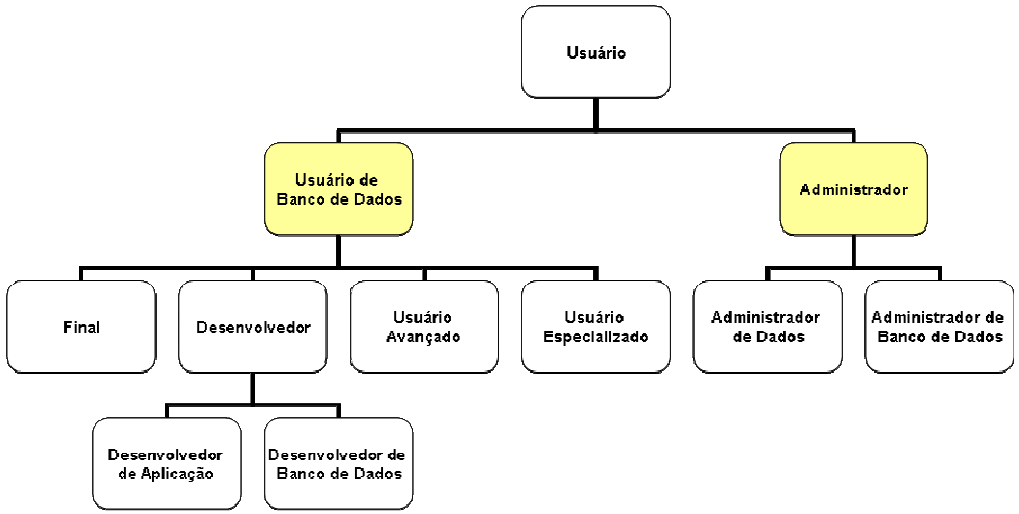
[ex: transação terminou, alteração no buffer, mas não alterou arquivo de dados e ... caiu a energia / ocorreu um erro no sistema]

Tipos de Usuário



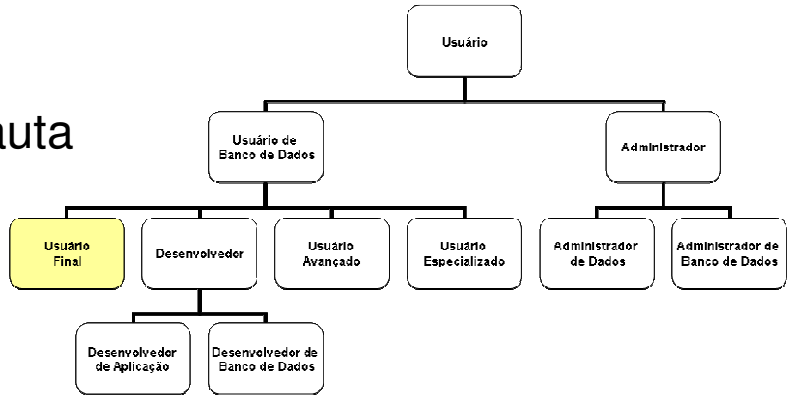
Tipos de Usuário

- Usuário de Banco de Dados
  - Interage direta ou indiretamente com o SGBD
- Administrador
  - Interage diretamente com o SGBD
  - Atende as necessidades dos usuários de banco de dados



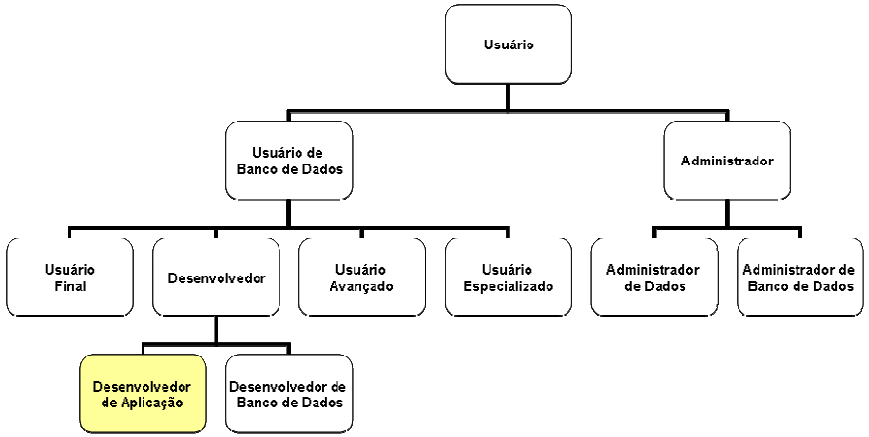
Tipos de Usuário

- Usuário de Banco de Dados
  - Usuário Final
    - Interage com o SGBD utilizando diferentes aplicativos
    - Desconhece completamente a existência do SGBD, portanto só vê telinhas!
    - É para atender suas necessidades que um sistema de banco de dados é desenvolvido
    - A utilidade de um sistema de banco de dados é medida através dele
    - Exemplos
      - Executivo, secretária, internauta



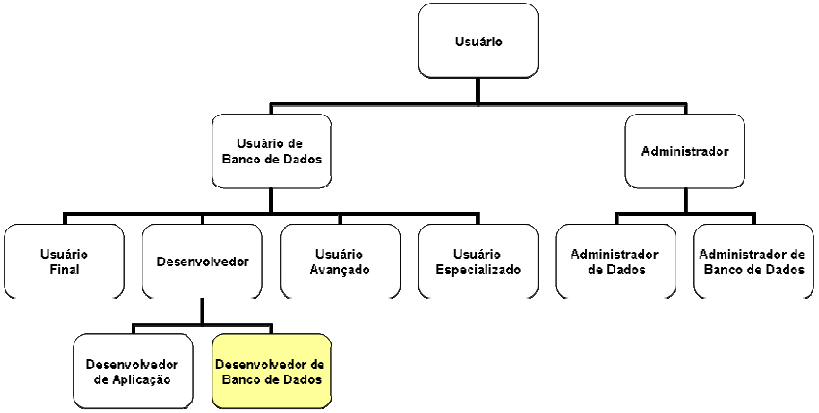
Tipos de Usuário

- Usuário de Banco de Dados
  - Desenvolvedor de Aplicação
    - Interage indiretamente com o SGBD escrevendo aplicações que submetem comandos de manipulação de dados
    - Boa capacidade de programação
    - Conhecer várias linguagens de programação (Java, C++, C#)



Tipos de Usuário

- Usuário de Banco de Dados
  - Desenvolvedor de Banco de Dados
    - Interage diretamente com o SGBD
    - Parte de uma aplicação pode ser desenvolvida utilizando a linguagem de programação do SGBD
      - Motivos: desempenho, gerenciamento do código, etc
    - Essa parte do código fica armazenada no banco de dados e é executada no servidor

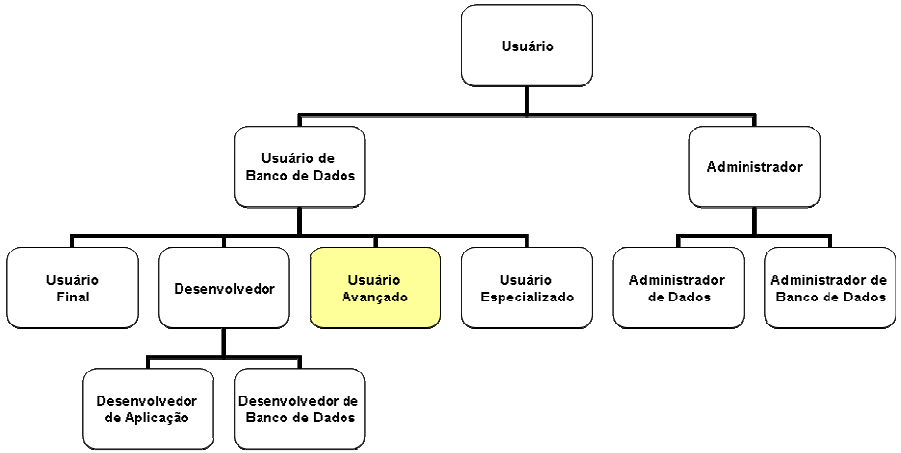


Tipos de Usuário

- Usuário de Banco de Dados
  - Desenvolvedor de Banco de Dados (cont.)
    - Boa capacidade de programação
    - Conhecer várias linguagens de programação de banco de dados
      - PL/SQL (Oracle), Transact SQL (SQL Server), PL/pgSQL (PostgreSQL)
    - Muitas vezes, nas empresas, o desenvolvedor de aplicação e de banco de dados são a mesma pessoa

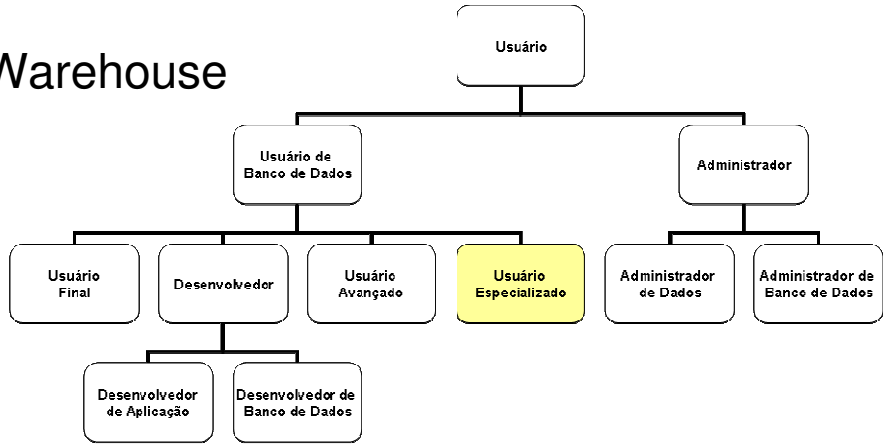
Tipos de Usuário

- Usuário de Banco de Dados
  - Usuário Avançado
    - Interage diretamente com o SGBD sem escrever aplicações
    - Conhece a sintaxe da linguagem de acesso e manipulação de dados
    - Conhece o esquema do banco de dados
    - Exemplos
      - Gerente de informática, Analista de Sistemas, Consultor



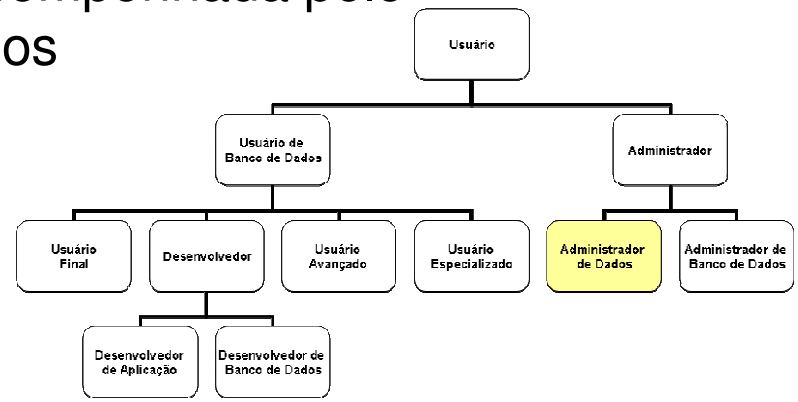
Tipos de Usuário

- Usuário de Banco de Dados
  - Usuário Especializado
    - Interage diretamente com o SGBD escrevendo aplicações de banco de dados especializadas
    - Aplicação Especializada: Data Warehouse (DW)
      - Um DW é um banco de dados “especial” que armazena dados integrados oriundos de vários outros bancos de dados de uma empresa
      - É preciso extrair os dados, transformá-los e carregá-los no DW
    - Exemplo
      - Desenvolvedor de Data Warehouse



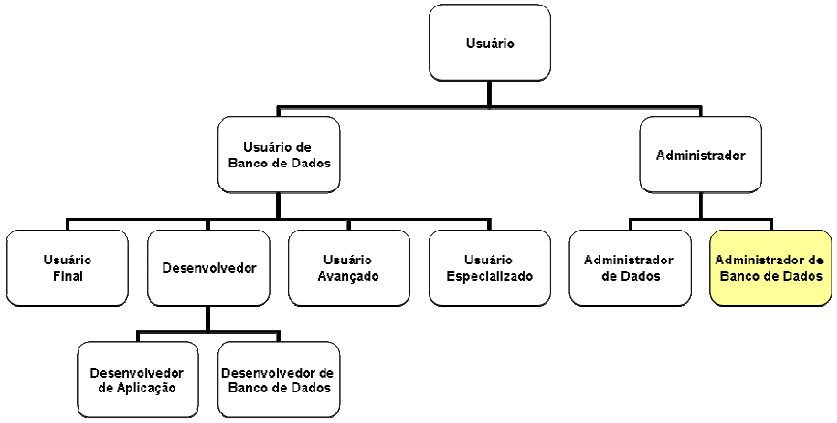
Tipos de Usuário

- Administrador
  - Administrador de Dados
    - Cuida da “saúde mental dos dados”
    - Conhece a semântica dos dados e como eles estão relacionados
    - Mantém a consistência das informações
    - Determina o modo com que as aplicações compartilham suas informações
    - Sua função é muitas vezes desempenhada pelo administrador de banco de dados



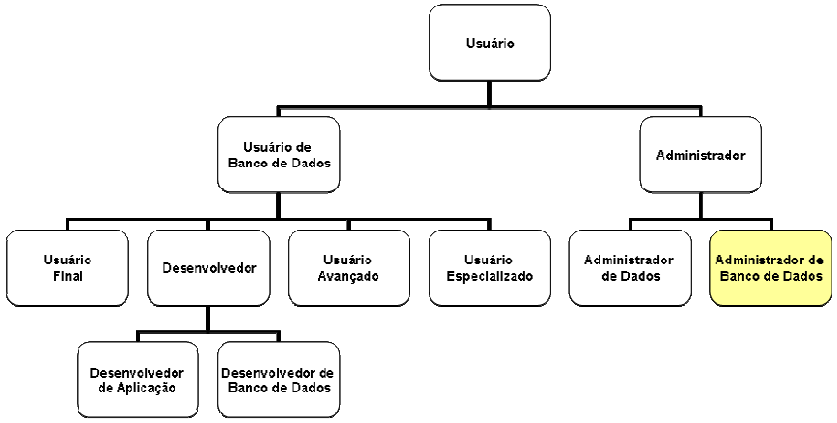
Tipos de Usuário

- Administrador
  - Administrador de Banco de Dados (DBA)
    - Cuida da “saúde física dos dados”
    - Participa da elaboração do projeto lógico juntamente com os analistas de projetos
    - Executa o projeto físico dos bancos de dados
    - Coordena atividades de manutenção dos bancos de dados



Tipos de Usuário

- Administrador
  - Administrador de Banco de Dados (DBA) (cont.)
    - Define as políticas de segurança e planos de contingências para os bancos de dados
    - Importante possuir bons conhecimentos em sistemas operacionais e redes



O que é SQL?

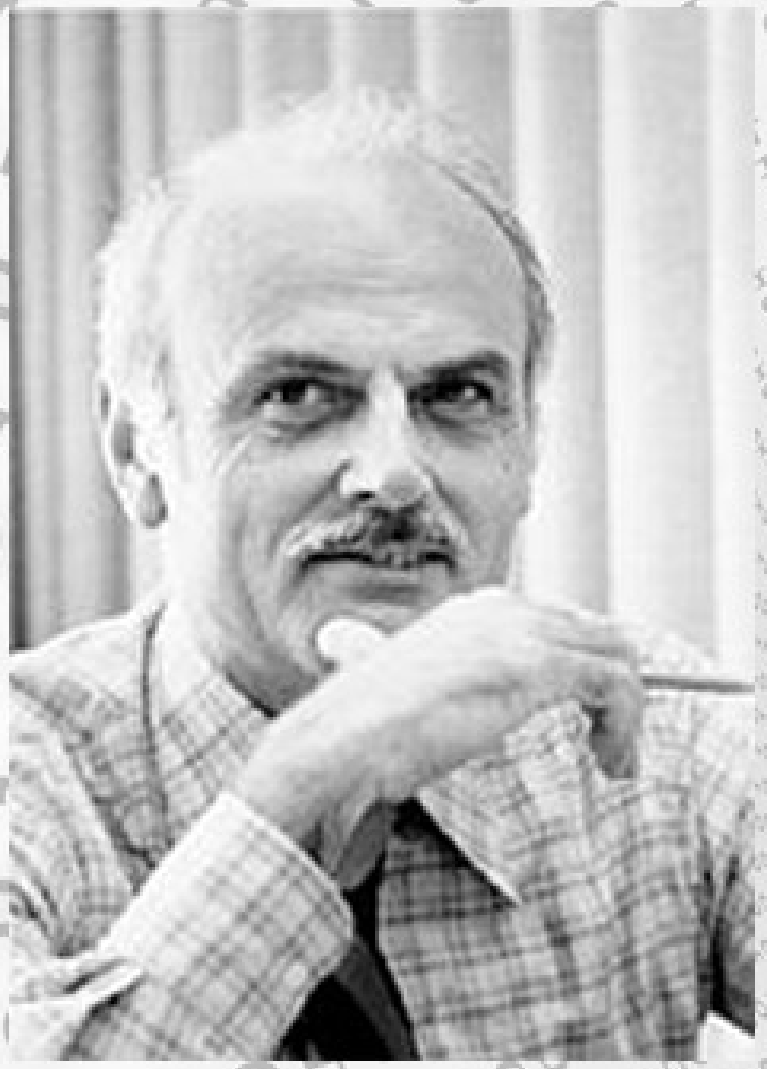
Structured Query Language, ou **Linguagem de Consulta Estruturada** ou **SQL**, é uma linguagem de pesquisa declarativa para banco de dados relacional.

Muitas das características originais do SQL foram inspiradas na álgebra relacional

Foi desenvolvido originalmente no início dos anos 70 nos laboratórios da IBM.

Tinha por objetivo demonstrar a viabilidade da implementação do modelo relacional proposto por Edgar Codd.

Seu nome original era **SEQUEL**, acrônimo para **"Structured English Query Language"**



Edgar F. Codd

1923-2003

Codd invented the relational database while working for IBM.

He revolutionised the way in which data was stored and retrieved.

SQL (principais características)

A linguagem SQL é um grande padrão de banco de dados.

Por ser uma linguagem declarativa (não procedural), uma consulta SQL especifica a **forma** do resultado e não o **caminho** para chegar a ele.

Apesar de ser originalmente criada pela IBM, muitos desenvolvedores foram criando "dialetos" para ela. Essa expansão levou à necessidade de ser criado e adaptado um padrão para a linguagem.

Em 1986/87 a linguagem SQL foi padronizada pela ANSI e ISO sendo revisada nos anos de 1992, 1999 e 2003.

Normalmente a linguagem pode ser aportada de plataforma para plataforma sem mudanças significativas em sua estrutura.

Estrutura da linguagem SQL

A linguagem SQL é dividida em subconjuntos de acordo com as operações que se deseja efetuar sobre um banco de dados. Os principais subconjuntos são:

DDL - Data Definition Language *(Linguagem de Definição de Dados)*

Principais comandos: CREATE, ALTER e DROP

DML - Data Manipulation Language *(Linguagem de Manipulação de Dados)*

Principais comandos: SELECT, INSERT, UPDATE, DELETE, TRUNCATE e outros.

DCL - Data Control Language *(Linguagem de Controle de Dados)*

Principais comandos: GRANT, REVOKE e SET.

DDL Linguagem de Definição de Dados

O conjunto de comandos da linguagem DDL é usado para a definição das estruturas de dados, fornecendo as instruções que permitem a criação, modificação e remoção de objetos de banco de dados (base de dados, esquemas, tabelas, índices etc.).

A maioria dos bancos de dados comerciais tem extensões proprietárias no DDL.

Os comandos básicos da DDL são:

- CREATE: cria um objeto (uma Tabela, por exemplo) dentro da base de dados.
- DROP: apaga um objeto do banco de dados.
- ALTER: permite ao usuário alterar um objeto, por exemplo, adicionando uma coluna a uma tabela existente.

DML Linguagem de Manipulação de Dados

É o grupo de comandos dentro da linguagem SQL utilizado para a recuperação, inclusão, remoção e modificação de informações em bancos de dados.

Os comandos básicos da DML são:

- SELECT: permite ao usuário especificar uma consulta ("query") como uma descrição do resultado desejado.
- INSERT é usada para inserir um registro (formalmente uma tupla) a uma tabela existente.
- UPDATE para mudar os valores de dados em uma ou mais linhas da tabela existente.
- DELETE permite remover linhas existentes de uma tabela.
- TRUNCATE: remove rapidamente todas as linhas da tabela, esvaziando-a.
- COMMIT: efetiva a transação atualmente executada.
- ROLLBACK:desfaz a transação corrente, fazendo com que todas as modificações realizadas pela transação sejam rejeitadas.

DCL Linguagem de Controle de Dados

É o grupo de comandos que permitem ao administrador de banco de dados gerenciar os aspectos de autorização de dados e licenças de usuários para controlar quem tem acesso para ver ou manipular dados dentro do banco de dados.

Alguns exemplos de comandos DCL são:

- GRANT: concede privilégios a um ou mais usuários para acessar ou realizar determinadas operações em um objetos de dados.
- REVOKE: revoga (remove) ou restringe a capacidade de um usuário de executar operações.
- SET: Define parâmetros em tempo de execução, como por exemplo, o tipo de codificação do cliente e o estilo de representação de data e hora.
- LOCK: Bloqueia explicitamente uma tabela fazendo o controle de acessos concorrente.

O que é um Dicionário de Dados ?

Uma documentação do Diagrama Entidade Relacionamento.

Entidade: Cliente				
Atributo	**Classe**	**Domínio**	**Tamanho**	**Descrição**
Codigo_cliente	Determinante	Numérico		
Nome	Simples	Texto	50	
Telefone	Multivalorado	Texto	50	Valores sem as máscaras de entrada
Cidade	Simples	Texto	50	
data_nascimento	Simples	Data		Formato dd/mm/aaaa

Elementos do DD

- **Entidade:** Descreve o nome da entidade que foi definida no MER. A entidade é uma pessoa, objeto ou lugar que será considerada como objeto pelo qual temos interesse em guardar informações a seu respeito.

- **Atributo:** Os atributos são as características da entidade que desejamos guardar.

Elementos do DD

- **Classe:** as classes podem ser: simples, composto, multivalorado e determinante.
  - ***Simples*** indica um atributo normal.
  - ***Composto*** indica que ele poderá ser dividido em outros atributos, como por exemplo, o endereço.
  - ***Multivalorado*** é quando o valor do atributo poderá não ser único
  - ***Determinante*** é um atributo que será usado como chave, como CPF, Código do cliente, etc.

Elementos do DD

- **Domínio:** podem ser numérico, texto, data e boleano. Podemos chamar também de tipo do valor que o atributo irá receber. A definição desses tipos deve seguir um processo lógico.

- Exemplo:
  - **nome** é texto,
  - **salário** é numérico
  - **data de nascimento** é data
  - **aprovado** é boleano, pois só pode ser sim ou não (verdadeiro ou falso).

  ... e assim por diante.

Elementos do DD

- **Tamanho:** define a quantidade de caracteres que serão necessários para armazenar o seu conteúdo. Geralmente o tamanho é definido apenas para atributos de domínio ***texto***.

- **Descrição:** é opcional e pode ser usado para descrever o que é aquele atributo ou dar informações adicionais que possam ser usadas futuramente pelo analista ou programador do sistema.

Exercícios

1. Um conjunto de programas que apoiam a criação, manutenção e operação de um banco de dados são chamadas:_____

2. A linguagem usada para definir tabelas, esquemas e domínios de atributo de dados é chamada:

a) ( ) linguagem de definição de armazenamento.

b) ( ) linguagem de definição de banco de dados.

c) ( ) linguagem de definição de dados.

d) ( ) linguagem de definição de visões.

e) ( ) linguagem de definição de esquema.

3. A linguagem de manipulação do banco de dados permite:

a) ( ) modificar o esquema conceitual.

b) ( ) modificar o esquema interno.

c) ( ) especificar atualizações e recuperações.

d) ( ) modificar o esquema externo.

e) ( ) acessar o conteúdo do banco de dados.

f) ( ) modificar a estrutura do banco de dados.

4. Qual das opções abaixo não consiste em uma vantagem de se usar bancos de dados?

a) ( ) informações atualizadas.

b) ( ) flexibilidade.

c) ( ) redundância controlada.

d) ( ) interfaces múltiplas com o usuário.

e) ( ) Investimentos iniciais baixos em hardware, software e treinamento.

5. Quem ou o que é responsável por manter a consistência do banco de dados?

a) ( ) aplicações de banco de dados.

b) ( ) vendedor do SGBD.

c) ( ) administrador do sistema de banco de dados.

d) ( ) usuários finais.

e) ( ) projetista do banco de dados.

f) ( ) todas as anteriores.

6. Descreva o significado de DBA.

7. Realize uma pesquisa na internet e descreva as funcionalidades de ao menos 3 SGBDs existentes.

8. Na empresa onde trabalha, ou em outro local que possua um sistema de informação que conheça, qual é o SGBD utilizado? Quem interage com o banco de dados? Quais vantagens você considera em sua utilização em contraposição se fosse necessário o armazenamento de forma manual como em um armário de aço.