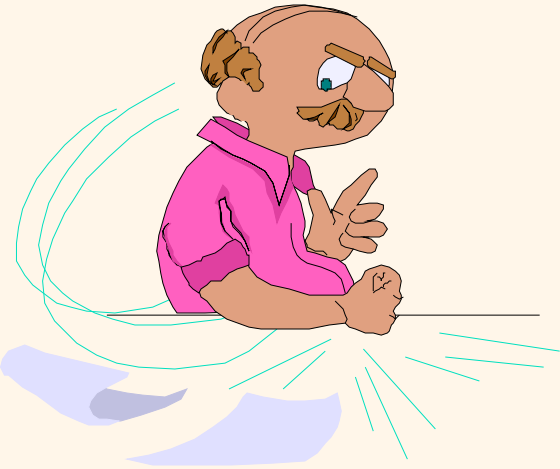
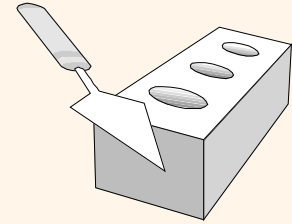


# *Sistemas de Gerência de BD*



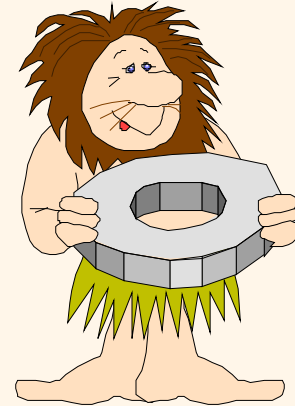
## *Capítulo 1*



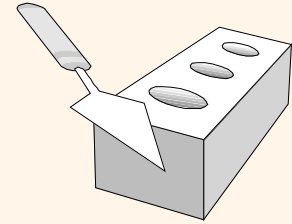
# *Créditos*

- ❖ Este conjunto de slides foram resultado de traduções/adaptações realizadas pelo Prof. Geovane Cayres Magalhães e seus alunos da disciplina MO410 – Banco de Dados oferecida pelo Departamento de Sistemas de Informações do Instituto de Computação da UNICAMP nas turmas de 2003.1, 2004.1 e 2005.1.

# O que é um SGBD?



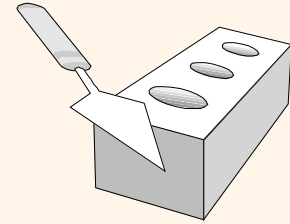
- ❖ Uma coleção (muita grande) integrada de dados.
- ❖ Modela empreendimentos do mundo real
  - Entidades (e.g., alunos, turmas)
  - Relacionamentos (e.g., Madonna está cursando MO410)
- ❖ Um SGBD é um pacote de software projetado para armazenar e gerenciar bancos de dados.



# Arquivos vs. SGBD

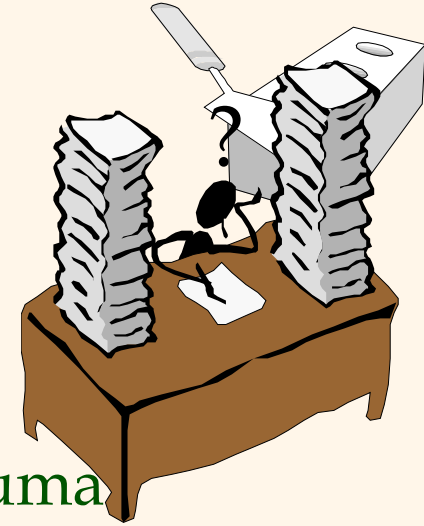
- ❖ Uma aplicação deve carregar grandes conjuntos de dados entre a memória principal e a memória secundária (e.g., *buffer*-ização, acesso orientado à páginas, endereçamento de 32-bit, etc.)
- ❖ Código especial para consultas diversas
- ❖ Deve proteger os dados de inconsistências devido a múltiplos usuários concorrentes
- ❖ Recuperação de *crash*-es
- ❖ Segurança e controle de acesso

# *Porque usar um SGBD?*

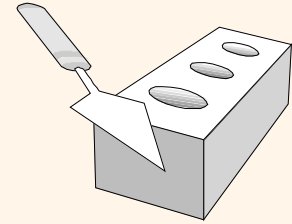


- ❖ Independência de dados e acesso eficiente.
- ❖ Desenvolvimento de aplicações em tempo reduzido.
- ❖ Integridade de dados e segurança.
- ❖ Administração de dados uniforme.
- ❖ Acesso concorrente, recuperação após falhas.

# Porque estudar banco de dados??

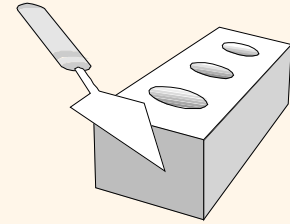


- ❖ Evolução de computação para informação
  - de um lado: embaralhamento na internet (uma bagunça!)
  - do outro lado: aplicações científicas
- ❖ Conjunto de dados aumentando em diversidade e volume.
  - Biblioteca digitais, vídeos interativos, Projeto Genoma Humano, Projeto EOS
  - ... explosão da necessidade de SGBD
- ❖ SGBD engloba muito da CC
  - SO, linguagens, teoria, IA, multimídia, lógica



# Modelos de Dados

- ❖ Um modelo de dados é um conjunto de conceitos que descrevem dados.
- ❖ Um esquema é a descrição de uma conjunto particular de dados, utilizando um específico modelo de dados.
- ❖ O modelo de dados relacional é o modelo mais utilizado atualmente no mundo.
  - Conceito principal: relação, basicamente uma tabela com linhas e colunas.
  - Cada relação tem um esquema, o qual descreve as colunas, ou campos.

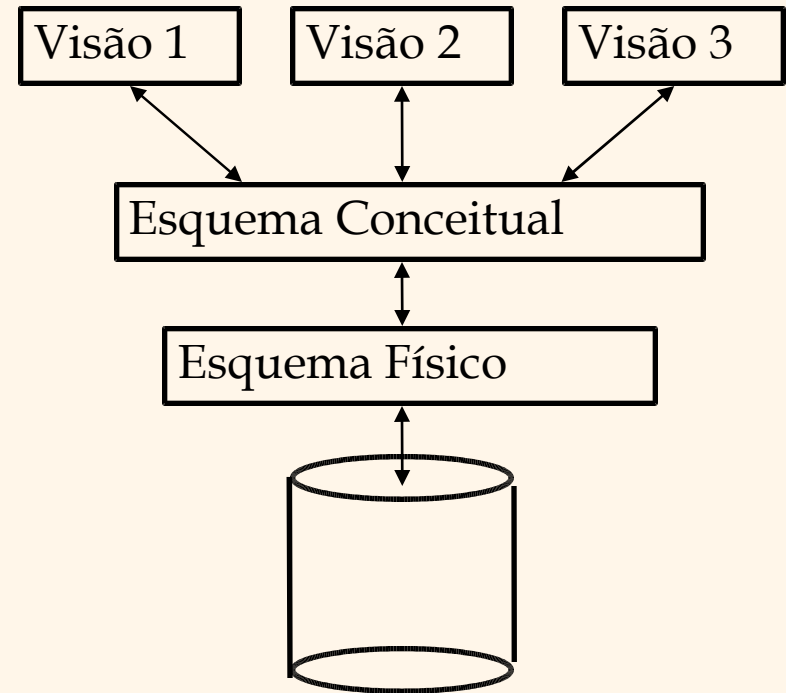


# Níveis de Abstração

❖ Várias visões, um único esquema conceitual (lógico) and esquema físico.

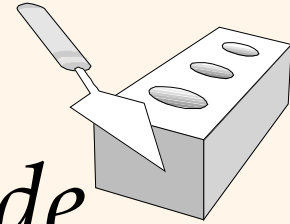
- Visões descrevem como usuários vêem os dados.
- O esquema conceitual define a estrutura lógica.
- O esquema físico descreve os arquivos e índices usados.

☞ *Esquemas são definidos através de uma LDD; dados são modificados/consultados através de uma LMD.*





# Exemplo: BD de uma Universidade



## ❖ Esquema conceitual:

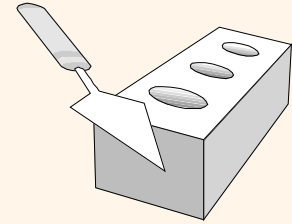
- *Students*(*sid*: string, *name*: string, *login*: string, *age*: integer, *gpa*:real)
- *Courses*(*cid*: string, *cname*:string, *credits*:integer)
- *Enrolled*(*sid*:string, *cid*:string, *grade*:string)

## ❖ Esquema físico:

- Relações armazenadas como arquivos sem ordenação.
- Índice na primeira coluna de *Students*.

## ❖ Esquema externo (Visão):

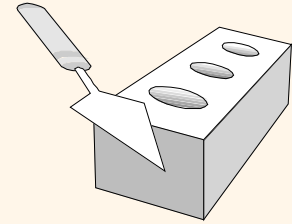
- *Course\_info*(*cid*:string,*enrollment*:integer)



# *Independência de Dados \**

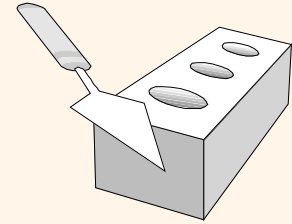
- ❖ Aplicações isoladas de como os dados são estruturados e armazenados.
- ❖ *Independência Lógica de Dados*: Proteção contra mudanças na estrutura *lógica* dos dados.
- ❖ *Independência Física de Dados*: Proteção contra mudanças na estrutura *física* dos dados.

☞ *Um dos mais importantes benefícios dos SGBD!*



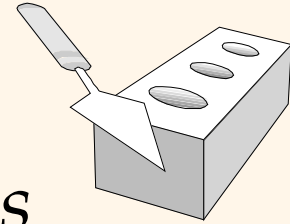
# *Controle de Concorrência*

- ❖ A execução concorrente de programas de usuários é essencial para um bom desempenho do SGBD.
  - Porque acesso a discos são freqüentes, e relativamente devagar, é importante manter a UCP ocupada trabalhando com vários programas concorrentemente.
- ❖ Ao intercalar ações de diferentes usuários os programas podem causar inconsistências: e.g., cheque descontado enquanto o saldo da conta é calculado.
- ❖ SGBD garante que tais problemas não aconteçam: usuários podem pensar que els estão utilizando um sistema mono-usuário.



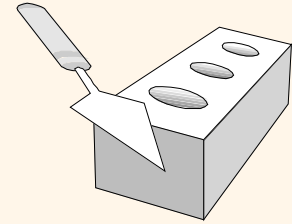
# Transação: Uma Execução de um Programa de DB

- ❖ O conceito chave é transação, a qual é uma seqüência *atômica* de ações do banco de dados (leituras/escritas).
- ❖ Cada transação, executada completamente, deve deixar o BD num estado consistente se BD estiver consistente quando a transação começou.
  - Usuários podem especificar algumas restrições de integridade simples, e o SGBD deve garantir estas restrições.
  - Além disto, o SGBD não entende a semântica do dado. (e.g., ele não entende como os juros numa conta bancária é calculada).
  - Então, a garantia de que uma transação (processando sozinha) preserva consistência é de responsabilidade do **usuário!**



# Agendamento de Transações Concorrentes

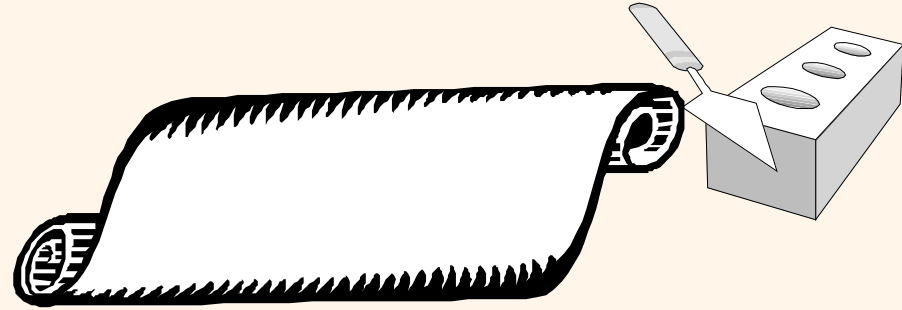
- ❖ O SGBD garante que a execução de  $\{T_1, \dots, T_n\}$  é equivalente a alguma execução serial  $T_1' \dots T_n'$ .
  - Antes de ler/escrever um objeto, uma transação requisita um lock sobre o objeto, e espera até o SGBD dar o lock. Todos os locks são liberados no final da transação. (protocolo de lock Strict 2PL.)
  - **Idéia:** Se uma ação de  $T_i$  (digamos, escrevendo  $X$ ) afeta  $T_j$  (que talvez lê  $X$ ), uma delas, digamos  $T_i$ , deve obter o lock sobre  $X$  primeiro e  $T_j$  é forçada a esperar até que  $T_i$  complete; isto efetivamente ordena as transações.
  - E se  $T_j$  já tem um lock sobre  $Y$  e  $T_i$  mais tarde requisita um lock sobre  $Y$ ? (Deadlock!)  $T_i$  ou  $T_j$  é abortado e reiniciado!



# Garantindo Atomicidade

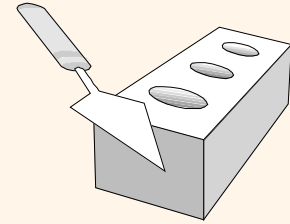
- ❖ O SGBD garante *atomicidade* (propriedade tudo-ou-nada) mesmo que o sistema falhe no meio de uma Xact.
- ❖ **Idéia:** Manter um log (história) de todas as ações executadas pelo SGBD enquanto executando um conjunto de Xacts:
  - **Antes** que uma mudança seja feita no BD, o registro do *log* correspondente é forçado para uma localização segura. (protocolo WAL; o suporte do SO para isto é normalmente inadequado.)
  - Depois de uma falha, os efeitos de um transação parcialmente executada são desfeitas usando o log. (Graças ao WAL, se o registro do log não tiver sido salvo antes da mudança, a correspondente mudança não terá sido aplicada ao BD!)

# O Log (Diário)



- ❖ As seguintes ações são registradas no *log*:
  - *Ti escreve um objeto*: O valor antigo e o valor novo.
    - O registro do *Log* deve ir para o disco *antes* da mudança na página!
  - *Ti commit(a)/abort(a)*: Um registro no *log* indica esta ação.
- ❖ Os registros do *Log* são encadeadas pela Xact id, de forma que fica fácil desfazer uma Xact específica (e.g., para resolver um *deadlock*).
- ❖ O *Log* é normalmente *duplicado* e *arquivado* num meio “estável”.
- ❖ Todas as atividades relativas ao *Log* (e de fato, todas as atividade relativas ao Controle de Concorrência tais como *lock/unlock*, tratamento de *deadlocks* etc.) são manipuladas de forma transparente pelo SGBD.

# *Bancos de Dados fazem felizes ...*



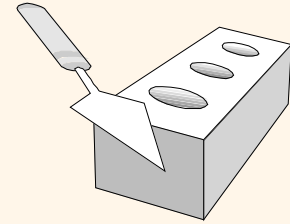
- ❖ Usuários Finais e Vendedores de SGBD
- ❖ Programadores de Aplicações
  - E.g., webmasters espertos
- ❖ Administradores de Banco de Dados (ABD)
  - Projetam esquemas lógicos / físicos
  - Manipulam segurança e autorizações
  - Disponibilidade de Dados, Recuperação de Falhas
  - Tuning de SGBD quando as aplicações evoluem



*Devem entender como um SGBD funciona!*

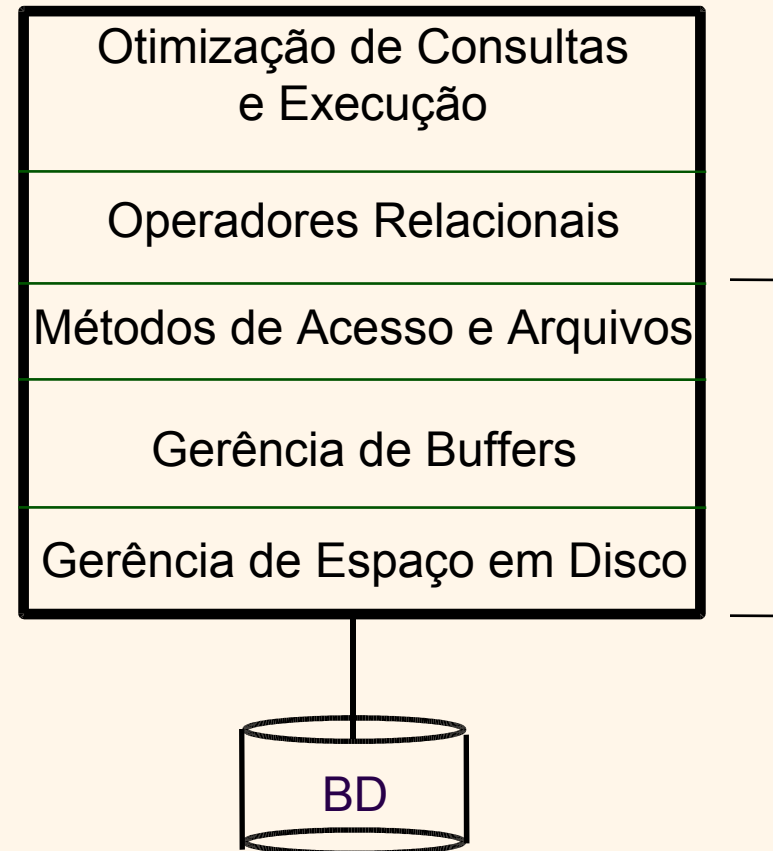


# Estrutura de um SGBD

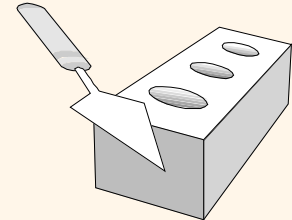


**Estas camadas devem considerar o controle de conc. e recuperação.**

- ❖ Um típico SGBD tem uma arquitetura em camadas.
- ❖ A fig. ao lado não mostra os componentes de recuperação e controle de concorrência.
- ❖ Esta é uma de várias possíveis arquiteturas; cada sistema tem sua própria variação.



# Sumário



- ❖ SGBD são utilizados para manter e consultar grandes BD.
- ❖ Benefícios incluem recuperação de falhas do sistema, acesso concorrente, desenvolvimento rápido de aplicações, integridade de dados and segurança.
- ❖ Níveis de abstração fornece independência de dados.
- ❖ Um SGBD normalmente tem uma arquitetura em camadas.
- ❖ O ABD têm um emprego de responsabilidade e são **bem pagos!** 😊
- ❖ DBMS P&D em SGBD é uma das mais abrangentes e interessantes áreas em CC.

