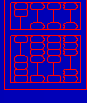




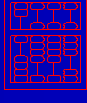
Conceitos básicos



banco de dados ou **base de dados** =
coleção de dados ou de informações
relacionadas entre si
representa aspectos do mundo real com
significado próprio que desejamos armazenar.
Usaremos os dois termos de forma
intercambiável.



Conceitos básicos



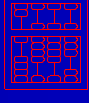
Bases de dados podem ser mantidas manualmente:

- relação dos nomes e telefones de pessoas conhecidas
- relação de bens e valores de uma pessoa física

Nos interessam as mantidas num computador, sejam elas pequenas ou grandes.



Exemplos

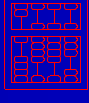


- estoque de uma farmácia ou de uma padaria (megabytes)
- catálogo de todos os livros publicados nos Estados Unidos (dezenas de megabytes)
- base de dados de todas as fotos recolhidas ao longo dos anos pelo programa espacial americano (centenas de terabytes?)

Projeto deve levar em conta o uso principal que se quer fazer dos dados



Exemplos

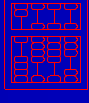


Bases de dados complexas:

- muitos tipos diferentes de dados,
- interdependentes e inter-relacionados
- podem chegar a centenas ou milhares de tipos distintos de dados:
dados sobre funcionários, sobre fábricas, sobre produtos, sobre matérias-primas, sobre componentes, sobre finanças etc.



Exemplos

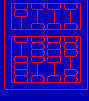


Exemplo: dados corporativos da Boeing ou General Motors:
cada tipo de dados pode conter milhares ou até milhões de itens:

- número de funcionários,
- número de componentes de um avião



Exemplos

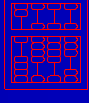


O nome da base de dados nos dá uma idéia do seu uso principal:

- Sistema de reserva de passagens aéreas
- Base de dados de biblioteca
- Base de dados dos contribuintes da Receita
- Base de dados acadêmicos da UNICAMP
- BD de cadeia de distribuição de bens de consumo
- Sistema bancário do banco BXY
- Base de dados dos cabos da Telefônica



Exemplos



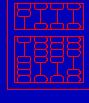
Bases de dados convencionais: armazenam dados alfanuméricos (cadeias de caracteres e valores numéricos).

Atualmente BD armazenam também imagens, gráficos, e até objetos multimídia (som e vídeo) gerando necessidades maiores de armazenamento e de complexidade de recuperação e processamento dos dados.

Exemplo: *bancos de dados geográficos.*



BDs x coleção de arquivos



BD = *coleção de dados relacionados entre si.*

Problemas com coleção de arquivos :

- inter-relacionamento que entre dados,
- regras de consistência que garantem esses inter-relacionamentos

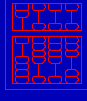
Exemplo: funcionários de um departamento devem estar na lista de funcionários do setor de RH

Regras de consistência complexas: entre produtos, componentes, matérias-primas e fornecedores:

são as chamadas *regras de negócio.*



Requisitos de Bancos de Dados



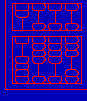
- *transações atômicas*: operações sobre os dados de forma conjunta e indivisível a fim de preservar a sua consistência
 - na ocorrência de falta de energia,
 - de falhas comunicação com a base de dados,
 - de falha nos discos de armazenamento

Exemplo: correntista de um banco transfere um valor da sua poupança para sua conta-corrente;

após a dedução na poupança, uma falha na comunicação ou no sistema impede o crédito na conta-corrente!



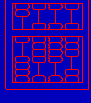
Requisitos de Bancos de Dados



- *Controle de concorrência*: o acesso simultâneo por vários usuários pode gerar inconsistências. Exemplo: dois agentes de viagem podem reservar a última vaga num vôo para passageiros distintos!
- *Persistência ou durabilidade* dos dados em meio de armazenamento confiável (discos magnéticos/fitas magnéticas)

Em inglês: **ACID** – *atomicidade, consistência, isolamento e durabilidade*

Sistema gerenciador de bases de dados (SGBD)

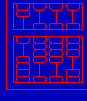


Coleção de programas:

- suficientemente genérica para permitir a criação e manutenção de qualquer base de dados independentemente da aplicação.
- descreve a estrutura dos dados de cada BD específica: *catálogo* contém *metadados* que “descrevem os dados” da BD
- permite apenas acesso autorizado aos dados por usuários registrados com senha.



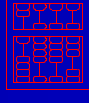
Requisitos de um SGBD



- prover recursos para armazenamento (*backup*) e recuperação da base de dados em meio magnético *off-line*:
backup, criação e autorização de usuários são executadas por um usuário especial: o *administrador da base de dados* (em inglês *DBA*).
- *independência de dados*:
forma de acesso a uma BD por uma aplicação deve ser independente da organização interna dos dados : se esta organização mudar, a aplicação não precisa mudar.



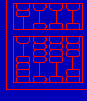
Requisitos de um SGBD



- prover meios para a *descrição* ou *representação conceitual* dos dados independente da sua representação interna.
- prover ferramentas para:
 - definição dos conjuntos de dados que constituirão a BD,
 - para construir e preencher a BD,
 - para manipular o seu conteúdo através de consultas e atualização
- para gerar relatórios.



Requisitos de um SGBD



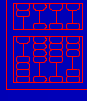
A linguagem SQL (*Structured Query Language* provê uma *interface de aplicação (API)* para criar e manipular os dados da base de dados.

SQL contém uma linguagem para definição e especificação dos dados (*Data Definition Language*) e uma linguagem para manipulação de dados, isto é, atualizações e consultas (*Data Manipulation Language*).

SQL tornou-se o padrão dos SGBDs relacionais e provê as funções citadas (exceto relatórios).



Requisitos de um SGBD



A interface de uma aplicação com o SGBD deveria ser independente do SGBD específico. Existe uma padrão ISO-ANSI para SQL com esta finalidade.

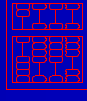
Vantagens:

- transportar uma aplicação de um SGBD para outro;
- aplicações poderiam utilizar de forma transparente dados armazenados em diferentes SGBDs.

A realidade: há diferenças incompatíveis nas APIs dos fornecedores de SGBDs e diferentes disponibilidades de recursos.



BD Torneios de Tênis da ATP



Coleta informações sobre: torneios patrocinados pela ATP, jogadores de tênis associados à ATP

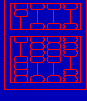
Jogadores: nome, sobrenome, país, ano e cidade de nascimento, ano de profissionalização, cidade de residência, número de títulos ganhos em jogos “simples”, em “duplas”, número total de jogos ganhos e de jogos perdidos.

Jogadores *ativos*: participam dos principais torneios da ATP, possuem número de registro unívoco;

Jogadores *aposentados*: não participam mais de torneios e têm o registro cancelado.



BD Torneios de Tênis da ATP



Torneios: nome do torneio, país, categoria, tipo da quadra, número inicial de participantes.

Três grandes categorias de torneios:

Grand Slam, Master Series e International Series.

Tipo da quadra: saibro, grama, material sintético poroso e tapete.

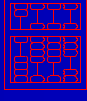
Essas informações não possuem dados históricos: refletem a situação atual de cada torneio.

Registros históricos dos principais torneios ao longo dos últimos anos:

vencedor do torneio e prêmio recebido.



BD Torneios de Tênis da ATP

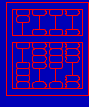


Esta descrição informal da base de dados é ambígua e incompleta; não esclarece:

- quais informações queremos realmente registrar sobre o histórico de um torneio?
- quais são os “torneios principais” sobre os quais queremos registrar seus dados?
- os torneios são anuais ou podem se realizar mais de uma vez por ano?
- um dado torneio, num dado ano é sempre realizado no mesmo lugar/país?



BD Torneios de Tênis da ATP

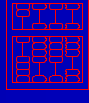


- qual a janela de tempo sobre a qual queremos registrar os dados?
- quais dados queremos armazenar sobre o histórico de um jogador? Por exemplo, se o país que ele representa muda, isso deve ser registrado?
- se um jogador se *aposenta*, os dados dos torneios que venceu devem permanecer na base de dados?
- que informações (relatórios) desejamos extrair da base de dados?

Ela corresponde ao que chamaremos de *visão externa dos dados*.



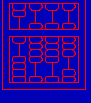
Conexões lógicas a SGBDs



- cinco formas principais para uma aplicação se conectar a um SGBD,
- são evoluções de como múltiplos usuários se conectam a um sistema de grande porte,
- refletem evoluções dos sistemas operacionais respectivos.
- Sistemas Unix suportam simultaneamente esses tipos de conexão.



Conexão tipo *time sharing*

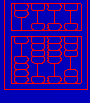


Conexão estrela: vários terminais acessam uma única máquina *Servidora*:

- estações: tipicamente terminais *burros*;
- toda a aplicação executa no servidor, inclusive o SGBD;
- existe um alto *overhead* de comunicação;
- acesso remoto via linhas dedicadas;
- não escalável com número de terminais;
- atualmente: rede local interligando microcomputadores a servidor SGBD.



Conexão tipo servidor de arquivos



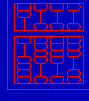
Surgiu com as redes locais de microcomputadores;

redes Novell com aplicações desenvolvidas em *clipper* são típicas dessa arquitetura.

- estações de usuários são inteligentes
- aplicação e SGBD executam na estação
- arquivos residem no servidor e transitam pela rede quando acessados pela aplicação
- tráfego na rede proporcional ao número de estações e ao tamanho da BD



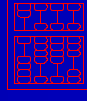
Conexão tipo servidor de arquivos



- escalável com número de terminais
- não é escalável com o tamanho da BD, devido ao tráfego de dados pela rede.



Conexão tipo *cliente-servidor*



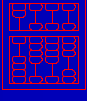
Surgiu com estações gráficas na década de 80 e redes locais de alto desempenho (Ethernet, Fast-Ethernet) conectando as estações aos servidores.

Também denominada de “arquitetura em duas camadas” (*two tiered architecture*),

- SGBD executa num servidor;
- estações inteligentes possuem alta capacidade de processamento;
- aplicação dividida entre a estação (*cliente*) e o SGBD;



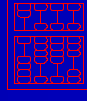
Conexão tipo *cliente-servidor*



- o SGBD executa comandos SQL vindos de clientes, e apenas resultados trafegam pela rede;
- carrega a rede menos que a solução servidor de arquivos para bases de dados grandes;
- carrega o servidor menos que a solução *time sharing*, devido ao menor processamento na máquina servidora.



Conexão tipo *cliente-servidor*

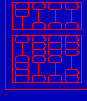


Também tem seus problemas:

- disseminação e atualização de aplicações entre muitas estações,
- estações não devem guardar dados corporativos, por razões de segurança e consistência dos dados,
- acesso remoto complexo a telas gráficas e tráfego moderado de dados (minorado com o uso da Internet para acesso a BD corporativas - Intranets),
- depuração do software complexa pela divisão do processamento entre a máquina cliente e o SGBD.



Conexão via servidor de aplicações

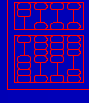


Uma única máquina, o servidor de aplicações, concentra a aplicação:

- resolve o problema da disseminação e atualização de aplicações entre estações,
- recebe pedidos dos clientes e envia ao SGBD,
- cliente apenas formata telas gráficas,
- denominada de “arquitetura em três camadas” (*three tiered architecture*),
- agrega as vantagens da arquitetura *cliente-servidor*.



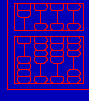
Conexão via servidor de aplicações



- garante o sigilo da aplicação, agora residente em máquina fisicamente controlada.
- em sistemas grandes, o processamento da aplicação pode ser distribuído replicando o servidor de aplicações e implementando um mecanismo dinâmico de distribuição de carga entre eles.



Conexão via servidor Web (Internet)



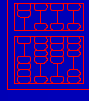
Desvantagem do servidor de aplicações convencional: acesso gráfico ao SGBD de forma transparente tanto local como remotamente.

A Internet para empresas viabilizou este tipo de acesso (embora nem sempre com tempos de resposta razoáveis):

- a aplicação reside num servidor WWW que também faz o papel de servidor de aplicações,
- único software requerido no cliente é um navegador (*browser*) padrão,



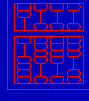
Conexão via servidor Web (Internet)



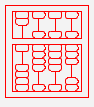
- o problema de atualização de software nos clientes praticamente desaparece (exceto por incompatibilidades entre diferentes navegadores, especialmente a execução de código Javascript),
- problema remanescente: diferentes graus de resolução gráfica dos clientes, em especial clientes móveis: *notebooks, palm-tops etc.* e celulares,
- uma nova versão fica imediatamente acessível quando um navegador abre a página de entrada da aplicação.



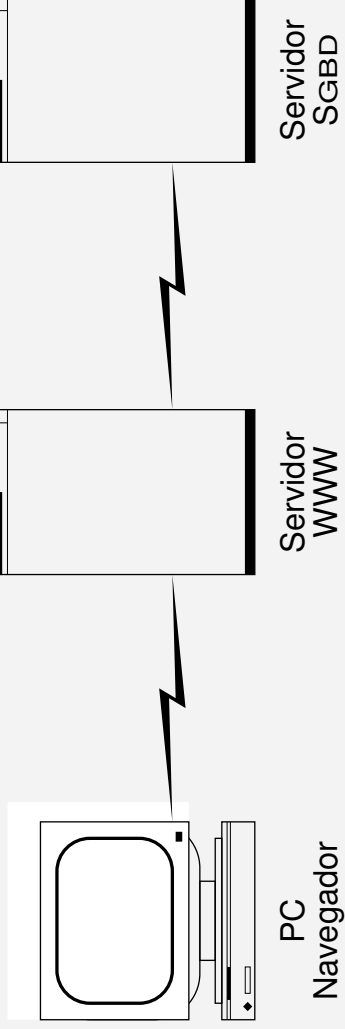
Conexão via *servidor Web* (Internet)



- mudanças nas linguagens para aplicações: agora são linguagens interpretadas como Java, PHP, Perl, Python, C# e ASP.
- aplicação gera formulários HTML para a estação cliente, acessa o SGBD usando os dados do formulário e devolve o resultado sob a forma de páginas HTML dinamicamente geradas,
- o SGBD normalmente reside numa outra servidora. Em aplicações menores pode residir na própria máquina do servidor WWW.

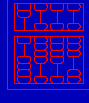


Conexão via servidor Web (Internet)





Conexão via servidor *Web* (Internet)



Pela sua generalidade e baixo custo, esta forma de conexão a SGBDs relacionais tem se tornado predominante.

Os problemas de segurança, no entanto, são mais críticos devido ao acesso remoto,