

Abordagem de SGBDs versus Abordagem Tradicional

>Porque utilizar SGBD

- Facilidade na manipulação dos dados
- Evitar redundância de informações e esforço repetitivo (ter que voltar na aplicação sempre pra manipular o banco de dados)
- Evitar inconsistência nos dados

Natureza Auto-descritiva da Abordagem de banco de dados

>Natureza Auto-descritiva

- Possui descrição própria
- DB schema: Estrutura bem definida dos dados
- Natureza auto-descritiva definida por Metadados e o Schema
- Dificilmente acharemos empresas que não usam SGBD

Isolamento Program/data e Abstração

>Isolamento e abstração

- Isolamento é ótimo para manutenção dos dados e do sistema
- Facilidade na alteração dos dados
- Abstração é um pilar na independência do programa e dados
- Transparência em como executa as tarefas, no SGBD você sabe claramente como as tarefas estão sendo executadas

Suporte a Múltiplas Visões(views) dos dados

>Table Views

- Raramente você vai ser o único a estar usando o SGBD
- Várias pessoas de vários setores de uma empresa podem precisar utilizar o SGBD
- Esses dados podem ser relevantes para alguns setores e restrito a outros
- Isso gera perspectivas distintas de um mesmo contexto
- A view para os setores são read-only o que não permite a manipulação dos dados

Compartilhamento de dados e Processamento de Transações Multiusuários

>Design - Múltiplos acessos

- Teremos no nosso SGBD múltiplos acessos de vários setores de uma empresa
- Integração é um fator importante já que serão vários acessos de pessoas diferentes
- A manutenção do SGBD é outro fator importante
- Concurrency Control é o nome do atributo que permite que várias pessoas utilizem o SGBD de forma concorrente

- Para essa concorrência acontecer de forma coerente é necessário que os dados só se alterem caso seja feita uma atualização desses dados e não por qualquer pessoa

>OLTP

- Acesso simultâneo com a finalidade de completar uma ação
- Exemplo de uma reserva de viagem
- Insert, Delete, Update
- Online Transaction Processing
- App multiuser
- Gerenciador: transações concorrentes
- Atomicidade: 8 ou 80 (executa tudo ou volta e não executa nada)
- Sistema operacional que tomará decisões mediante transações

Abordagem de banco de dados - Quais são os Atores em Banco de Dados

>Atores

- SGBDs mais simples: BDs simples, poucas pessoas acessam
- Big Organizations: Muitas pessoas, um setor específico para a manipulação dos dados
- Isso acarreta em outras preocupações como: Manutenção, Design e Usabilidade
- Atores do dia-a-dia: Administrador, Designers, Usuários finais
- Designer (modelagem): identificar dados e requisitos; representação e estrutura; fase preliminar
- Administrador (DBA - Database Administrator): Gerencia recursos; Orquestração; Autorização de acesso; geralmente tem uma DBA Staff o acompanhando e o designer faz parte dessa Staff
- Usuários Finais: Interessados em consumir as informações; são categorizados
- Classificação dos Usuários Finais: Casuais (acessos ocasionais e faz uso de APIs); Ingênuos (maior parte dos usuários finais, uso de Canned Transactions - transações encapsuladas por uma linguagem de programação, raramente ocorrem erros); Sofisticados (tem conhecimento prévio do sistema); Standalone (pessoa que tem um BD pessoal mas somente ele acessa)
- Quem programa as Canned Transaction (o ato de curtir uma publicação em uma rede social é um exemplo de Canned Transaction)? Os Engenheiros de Software

Workers em background - Banco de Dados

>Background

- Fora do contexto de BD: Designer do sistema de SGBD (não do SGBD em si, mas do sistema que dá suporte); Implementação (instalação) do sistema de SGBD; Pessoal de operação e manutenção; Desenvolvedores de ferramentas

Vantagens de Utilizar a Abordagem de SGBDs

>Vantagens do SGBD:

- Controle de Redundância (também de forma concorrente)
- Restrição de acesso (read-only)
- Storage - prove persistência
- Storage - prove estrutura
- Backup e Recovery

Ganhos em utilizar a Abordagem de SGBDs

>Ganhos com SGBD

- Padronização (tipos de dados por exemplo)
- Redução de tempo no desenvolvimento da aplicação
- Flexibilidade
- Disponibilidade de info atualizadas (update imediato)
- Economia com escalabilidade

Quando não usar SGBDs

- Custo-benefício
- Custo de overhead
- Verificar a sua necessidade e se essa necessidade é sanada com algo mais simples que um SGBD

> Custo

- Investimento inicial
- Generalidade na definição e processamento
- Segurança, controle de concorrência, recovery, funções de integridade