

## EXPLORANDO A ABORDAGEM DE SGBD'S

Por que utilizar um SGBD é mais interessante? (Comparado a uma abordagem tradicional.)

Mais fácil o compartilhamento de informações, menos repetição, menos gastos e consistência de dados.

- Você não precisa ficar modificando a aplicação

SGBD pode ter vários banco de dados, cada um com o seu Schema

Auto-descritivo: SGBD traz consigo a descrição de sua estrutura.

Catálogo: atrelado a essa auto-descrição

Isolamento: é ótimo para a manutenção dos dados. Se for preciso uma modificação ela é feita no SGBD, não mexe/impacta a aplicação.

Na abordagem tradicional uma modificação acarreta reestruturação.

Abstração: generalizar

### Múltiplas Visões - Views

Perspectivas distintas de um mesmo contexto.

Com o SELECT é possível fazer vários tipos de consultas e trazer somente as informações necessárias/solicitadas.

Compartilhamento: consigo compartilhar informações/ dados com diversos grupos distintos.

Controle de concorrência - manter os dados coerentes. O sistema tem que permitir a concorrência no acesso às informações, mas de forma coerente (ex: não deixar que 2 pessoas diferentes reservem o mesmo assento no avião)

OLTP (online transition processing)

Encarregado de registrar todas as operações efetuadas.

A ideia é prover performance para que eu consiga executar uma transação fim a fim sem nenhum problemas - constância na execução

Ou executa tudo ou não executa nada!

## Atores, Workers, Vantagens e Desvantagens

### **Atores**

*Design (modelagem)*: mapeamento da estrutura. Entender o contexto, quais as informações serão retornadas. Como você quer enxergar os dados interfere na modelagem.

*Administrador - DBA*: normalmente tem uma Staff

*Usuários finais*: interessados em consumir as informações. O acesso é Via Query e via API

Casuais

Ingênuos (acesso por uma API gráfica) - ex: sem parar, caixa eletrônico

Sofisticados (noção ou conhecimento prévio do sistema)

Standalone (pessoa que tem um BD pessoal - só ele acessa)

*Engenheiro de Software*

**Workers - relacionados mas não diretamente ligados ao SGBD**

- > *Designer e implementação do SGBD*: São vários requisitos que devem ser instalados na máquina para que o SGBD funcione bem
- > *Pessoal de operação e manutenção*
- > *Desenvolvedores*

Pessoas que dão suporte para que os dados sejam consumidos, possibilitam uma infra estrutura

**Vantagens**

- > Controle de redundância: cada pessoa consulta direto o SGBD
- > Restrição de acesso: o usuário terá login e senha, por exemplo. O acesso aos dados é seguro. DBA faz a restrição de acesso
- > Prover persistência:
- > Quando você realiza a consulta (QUERY) - O SGBD busca os dados, encontra e joga-os na memória e nós conseguimos visualizar
- > Caching, Buffering e Indexação (para tornar a consulta mais rápida)
- > Backup e Recovery
- > Integridade de dados: respeitar sempre as regras de negócio

**Ganhos ao utilizar SGBD**

- Padronizando a utilização do modelo relacional
- Redução de tempo ao desenvolver a aplicação e otimizar recursos
- Flexibilidade em termos de modificação (mas a modificação não é trivial)
- Nova inserção/atualização é refletida quase que automaticamente
- Baixar o custo operacional e de gerenciamento

**Quando não utilizar SGBD**

Se for algo simples, pequeno, precisa ponderar se é necessário mesmo um SGBD.