Bases Dados vs Sistema de Ficheiros

Ficheiros

Por causa de
- Dados embebidos na aplicação e não separados;
- Ausência de controlo sobre o acesso e manipulação dos dados, para além da restrições da aplicação em si;

Existem estes problemas
1. Poucas preocupações com integridade, segurança e recuperação de dados;
2. Separação e Isolamento dos dados: acesso + difícil, para filtrar alguns dados seria preciso sincronizar o processamento de 2 ou mais ficheiros;
3. Redundância de dados: sistema de dados descentralizado, encoraja/necessita de redundância de informação, difícil de controlar, pode levar a perda de integridade;
4. Mudar a estrutura ou formato dos ficheiros de dados implica alterar a aplicação;
5. Incompatibilidade de Ficheiros: estrutura de um ficheiro depende da linguagem que o gera, se se usarem linguagens diferentes....
6. Acesso restrito a 1 utilizador de cada vez;

Base Dados - Coleção partilhada de dados (e sua descrição) relacionados lógicamente, para responder às necessidades de informação de uma organização
- Resolve os problemas dos sistemas de ficheiros;
- Dicionário de dados permite a independência da base de dados relativamente à aplicação, permite ABSTRAÇÃO DE DADOS.

Gestor de Base de Dados - Software que permite aos utilizadores criarem e interagirem com a base de dados. Permite:
1. Que utilizadores definam a base de dados, os tipos e estruturas de dados, assim como as constraints dos dados a serem guardados;
2. Que os utilizadores adicionem, atualizem, removam e consultem dados, através de uma query language como a SQL;
3. Acesso controlado à base de dados:
    a. sistema de segurança, impede utilizadores não autorizados de acederem;
    b. sistema integrado, que mantém a consistência dos dados;
    c. sistema de controlo de concorrência;
    d. sistema de recuperação de dados;

Vantagens de um Gestor de Base de Dados

- Controlo da redundância de dados;
- Redução do risco de inconsistência de dados, maior integridade de dados;
- Maior facilidade no cruzamento de dados (devido à sua centralização);
- Partilha de dados e concorrência;
- Maior Segurança (permissões de utilizadores);
- Backup e recuperação de dados;
- Maior facilidade na manutenção (data independence);

Desvantagens

- Maior Complexidade;
- Espaço ocupado;
- Custos, do DBMS e hardware adicional;
- Possível menor performance do que os ficheiros;
- Falhas são mais custosas (um erro pode afetar toda a gente);

Views - Subconjunto de dados, que mostram ao utilizador apenas a informação pertinente, dando às aplicações independência de dados. Pode ser usada também por motivos de segurança na consulta de dados;

Nível Externo - Utilizadores da base de dados;
Nível Conceptual - estrutura lógica da base de dados, a view com todos os dados;
Nível Interno - Representação física dos dados num computador (como são guardados);

Responsabilidades do DBMS:

-Permitir que os utilizadores guardem, consultem e atualizem informação da BD;

-Possuir um dicionário de dados atualizado com as descrições do itens de dados guardados e sobre quais estão acessíveis aos utilizadores;

-Suporte a transações (todas as ações são completas ou nenhuma é);

-Serviços de controlo de concorrência;

-Serviços de Recuperação;

-Serviços de Autenticação;

-Serviços de Comunicação de dados;

-Serviços de Integridade (regras de integridade);

-Serviços que promovem Independência de dados (views e subshcema no Físico);

-Serviços auxiliares (reorganização de índices, garbage collection, estatísticas….);

Perguntas

Indique as responsabilidades de administraçao do gestor de bases de dados.

R: O gestor é responsável por gerir a realização física da base de dados, o que incui o design e implementação do esquema físico, tendo em conta a segurança e controlo de integridade. Um gestor de base de dados deve também monitorizar a performance do sistema, reorganizando e optimizando a BD se necessário.

Explique de forma sucinta o que entende por um cursor e que tipo de operações podemos realizar com eles.

R: Um cursor é uma estrutura de controlo que permite percorrer registos numa base de dados e efetuar operações sobre estes registos. Os cursores facilitam as operações como adição, remoçao e pesquisa de registos na base de dados, e podem ser pensados como se fosses iterators. Os cursores são usados para processar registos individuais sobre resultados de queries sobre a base de dados. Um cursor permite então efetuar operações logicas em termos de linha a linha de uma tabela. Um cursor pode ser visto como uma apontador para uma linha de uma tabela, sendo que apenas pode referenciar uma linha de cada vez, mas pode ser movido para outras linhas da tabela se necessário.

Em que tipo de situações poderemos ver como viável uma possível desnormalização de um modelo de dados. Apresente um exemplo duma dessas situações.

R:Desnormalização é o processo de tentar otimizar a performance de leitura numa base de dados adicionando dados redundantes ou agrupando dados. Um modelo normalizado irá normalmente guardar pedaços de informação relacionados em diversas relações, se estas relações estão guardadas em diferentes discos, completar uma query à base de dados pode ser lento e custos.
Existem duas abordagens para melhorar isto:
- Manter a base de dados normalizada, mas permitir que o SGBD adicione informação redundante no disco para otimizar a resposta das querys. Neste caso é responsabilidade do software manter os dados consistentes. (acho que isto é o mesmo que criar views... não é bem criar views, mas sim manter mesmo uma tabela com a informação mais acedida)
- Desnormalizar a base de dados. Agora é da responsabilidade do designer manter a base de dados consistente. Ao desnomalizar estamos a aumentar a performance dos reads e a diminuir a performace dos writes.
A introdução de redundância controlada apenas deve ser considerada quando se prevê que a Base de Dados criada não cumprirá os requisitos de performance pretendidos, dado que esta contribui para o surgimento de dados inconsistentes. Quando se pretende evitar o processamento simultâneo de várias relações, manter um qualquer sistema de arquivo ou manter as relações a um nível (simples) que permita satisfazer as “queries” dos seus

utilizadores, desnormalizar o sistema em causa pode ser uma opção viável. Apesar de a desnormalização contribuir positivamente para o desempenho das operações de pesquisa, acontece o contrário relativamente às operações de atualização. Deste modo, poderá fazer sentido considerar a realização deste processo para relações cuja frequência de atualizações seja pequena e bastante menor do que a frequência de pesquisas efetuadas. Uma possível medida a tomar com vista à desnormalização de um sistema, é a combinação de relacionamentos unários. Tomando como exemplo o relacionamento Projeto-Candidatura, poderíamos combinar estas duas relações numa só. No entanto, visto que se prevê que a relação Candidatura possua bastante mais tuplos que a relação Projeto e o relacionamento mantido pelas duas entidades ser opcional, conclui-se que esta combinação não seria favorável, já que se desperdiçaria bastante espaço com valores NULL.

Objetivos de uma arquitetura em 3 camadas. Nome disto?
Uma camada: O SGBD é a unica entidade onde o utilizador interage. Qualquer mudanças feitas aqui irão afetar diretamente a base de dados. Não é grande coisa para utilizadores e preferencialmente apenas programadores e designers utilizam.

Tres camadas: Arquitetura mais usada.
Camada de dados: Nesta camada apenas se encontra a base de dados, juntamente com as suas querys. Contêm todas as relações e restrições.

Camada aplicacional: Nesta camada encontra-se a funcionalidade da aplicação que acede à base de dados. Esta camada funciona como uma abstração da base de dados. Os utilizadores não sabem da existência da base de dados para lá da aplicação. A base de dados também não sabe de nada para além da camada aplicacional. Esta camada funciona como um mediador entre a camada de apresentação e de dados.

Camada de dados: Camada que interage com o utilizador. Do ponto de vista do utilizador, esta camada constitui toda a aplicação. Toda a informaçao apresentada nesta camada foi buscada à base de dados pela camada aplicacional.

Este tipo de arquitetura é altamente modificavel visto que todos os componentes são independentes uns dos outros, e abstrai a funcionalidade da aplicação do utilizador. Melhora a escalabilidade e é menos afetada por erros, visto que se uma das camadas falhar, as outras não falham obrigatoriamente.

Quais as funções de um processador de queries de um SGBD?

Um processador de queries de um SGDB recebe como input um pedido de query em forma de texto SQL, efetua o parsing deste texto, gera um plano de execução e completa o processamento executando o plano de execução e returnando os resultados ao cliente.

Detalhes: Num sistema de base de dados relacional, o processador de queries é o modulo responsável por executar queries de bases de dados. O processador de queires recebe como input queries na forma de texto SQL, efetua o parsing e otimiza-as, e completa a sua execução aplicado metodos de acesso a dados especificos e implementaçoes de operadores de bases de dados. O processador de queries comunica com o motor de armazenamento, o qual lê e escreve dados do disco, gere registos, contra concorrência e mantém ficheiros de log.