



Modelagem e Projeto de Banco de Dados

ARQUITETURA DE UM SISTEMA GERENCIADOR DE BANCO DE DADOS (SGBD) — COM EXEMPLOS PRÁTICOS

Prof. Gustavo Dias

Segundo Heuser (2009), um Sistema Gerenciador de Banco de Dados (SGBD) é um conjunto de programas responsáveis por definir, armazenar, manipular, proteger e recuperar dados, atuando como intermediário entre as aplicações e os dados armazenados fisicamente.

Para cumprir essas funções, o SGBD é estruturado em módulos especializados, cada um com responsabilidades bem definidas.

1. Pré-compilador e Compilador DML

O pré-compilador DML identifica comandos DML inseridos em programas de aplicação escritos em linguagens de programação tradicionais e os separa do código-fonte da aplicação.

Exemplo:

Um sistema acadêmico escrito em Java possui um trecho que executa uma instrução SELECT para consultar todos os registros de uma determinada tabela.

O pré-compilador reconhece esse comando SQL dentro do código Java e o envia ao SGBD para tratamento.

O compilador DML então analisa esse comando, verifica sua sintaxe e semântica e gera um plano de execução, definindo como os dados serão acessados.

Caso de uso:

Decidir se a busca pelo aluno será feita por varredura sequencial da tabela ou utilizando um índice sobre o campo id.

2. Interpretador DDL

O interpretador DDL processa comandos responsáveis pela definição e modificação da estrutura do banco de dados.

Exemplo: instrução CREATE para criar uma determinada tabela.

Nesse caso, o interpretador DDL cria a estrutura da tabela e registra suas características no dicionário de dados.

Caso de uso:

Quando um novo módulo do sistema exige o cadastro de disciplinas, o interpretador DDL cria a tabela disciplina e suas restrições antes que qualquer dado seja inserido.

3. Controle de Execução

O controle de execução coordena a execução das operações solicitadas, acionando os módulos necessários e garantindo que cada etapa seja realizada corretamente.

Exemplo:

Ao executar um UPDATE em uma tabela, o controle de execução:

- verifica permissões,
- aciona o gerenciador de transações,
- solicita páginas ao gerenciador de memória,
- coordena a escrita dos dados.

Caso de uso:

Em um sistema hospitalar, uma atualização de prontuário dispara múltiplos acessos controlados para evitar inconsistências.

4. Gerenciador de Transações

O gerenciador de transações assegura as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), fundamentais para a confiabilidade do banco de dados.

Atomicidade (Atomicity): Garante que a transação seja tratada como uma unidade única, onde todas as operações são concluídas com sucesso ("commit") ou nenhuma delas é aplicada ("rollback"), prevenindo falhas no meio do processo.

Consistência (Consistency): Assegura que o banco de dados evolua de um estado válido para outro, seguindo todas as regras, restrições e regras de integridade predefinidas.

Isolamento (Isolation): Permite que múltiplas transações ocorram simultaneamente sem que uma interfira na outra, garantindo que o resultado seja o mesmo que se as transações fossem executadas sequencialmente.

Durabilidade (Durability): Garante que, uma vez que a transação foi confirmada (commit), seus dados serão armazenados permanentemente, mesmo em caso de falha de energia ou sistema.

4.1 Subsistema de Recuperação

Exemplo:

Durante a transferência de um aluno de curso, o sistema atualiza várias tabelas. Se ocorrer uma queda de energia no meio do processo, o subsistema de recuperação desfaz todas as alterações parciais.

Caso de uso:

Garantir que uma matrícula seja registrada por completo ou não seja registrada de forma alguma.

4.2 Controle de Concorrência

Exemplo:

Dois usuários tentam alterar o mesmo registro de aluno simultaneamente. O controle de concorrência impede que um sobrescreva o trabalho do outro.

Caso de uso:

Evitar que duas secretarias acadêmicas alterem a situação de matrícula do mesmo aluno ao mesmo tempo.

5. Gerenciador de Memória

O gerenciador de memória reduz o custo de acesso ao disco, melhorando o desempenho do sistema.

5.1 Gerenciador de Buffer Pool

Exemplo:

Registros frequentemente acessados (como alunos ativos) permanecem em memória, evitando leituras repetidas em disco.

Caso de uso:

Consultas frequentes à tabela aluno são respondidas mais rapidamente durante o horário de pico.

5.2 Gerenciador de Arquivos

Exemplo:

Define onde e como os registros da tabela aluno são armazenados fisicamente no disco.

Caso de uso:

Gerenciar o crescimento da base de dados ao longo dos semestres sem perda de desempenho.

6. Índices

Os índices são estruturas auxiliares que permitem acesso rápido aos dados.

Exemplo:

Um índice sobre cpf permite localizar um aluno específico sem percorrer toda a tabela.

Caso de uso:

Busca rápida por pacientes em um sistema de saúde ou por clientes em um sistema bancário.

7. Arquivo de Dados

O arquivo de dados armazena os registros das tabelas de forma permanente.

Exemplo:

A tabela aluno com milhares de registros está fisicamente armazenada em arquivos gerenciados pelo SGBD.

Caso de uso:

Armazenamento histórico de dados acadêmicos mesmo após a formatura do aluno.

8. Dicionário de Dados

O dicionário de dados mantém informações sobre a estrutura e as regras do banco.

Exemplo:

Registrar que o campo id da tabela aluno é um identificador e não aceita valores nulos.

Caso de uso:

Permitir que o SGBD valide automaticamente consultas e restrições sem depender das aplicações.

Referência

HEUSER, Carlos Alberto. Projeto de banco de dados. 6. ed. Porto Alegre: Bookman, 2009.