

Programação II

Professores:

Claudio Esperança

Inês Dutra

Aula 1

Professor:

Claudio Esperança

Aplicações Web

Conteúdo:

- 1.1 Aplicações Web
- 1.2 Html e Http
- 1.3 Arquiteturas em 3 Camadas
- 1.4 Apache e PHP
- 1.5 Sistemas de Gerenciamento de Banco de Dados

1.1 Aplicações Web

- Vários tipos de aplicações acessadas via web:
 - ▶ Desenho
 - ▶ Processamento de texto
 - ▶ Engenharia
- Estamos interessados em aplicações de bancos de dados, como por exemplo:
 - ▶ Fóruns de discussão
 - ▶ Compras via Internet
 - ▶ Controle de estoque
 - ▶ Movimentação de contas bancárias via Internet

A Web

- Vamos estudar aplicações em rede montadas sobre a estrutura da *www*
 - ▶ *www* = *World Wide Web* (rede mundial)
 - ▶ Mais conhecida simplesmente como *web*
- Quando falamos da "web", queremos dizer uma infraestrutura de rede calcada em três padrões:
 - ▶ *html* = *HyperText Markup Language* (Linguagem de Marcação Hipertexto)
 - ▶ *http* = *HyperText Transfer Protocol* (Protocolo de transferência de Hipertexto)
 - ▶ *TCP/IP* = *Transmission Control Protocol / Internet Protocol* (Protocolo de Controle de Transmissão / Protocolo Internet)

1.2 Html e Http

- A linguagem de hipertexto *html* constitui uma forma bastante comum de estruturar informações que são comunicadas através de uma rede
- A apresentação de um documento *html* é feita através de um navegador (*ou browser*)
- O navegador constitui a parte da arquitetura mais próxima do usuário
 - ▶ Se comunica com o provedor de documentos (servidor) através do protocolo *http*
- O protocolo *http* normalmente utiliza a estrutura da Internet para o transporte de dados através da suite de protocolos *TCP-IP*

Exemplo Http

- Protocolo simples no modelo cliente / servidor

- Cliente envia ao servidor um *pedido*

GET / HTTP/1.1

Host: www.lcg.ufrj.br

User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US;
rv:1.8.0.1) Gecko/20060124 Firefox/1.5.0.1

Connection: close

- Servidor retorna a informação solicitada

HTTP/1.1 200 OK

Date: Mon, 01 May 2006 09:04:44 GMT

Server: Apache

...

Content-Type: text/html

<HTML>

...

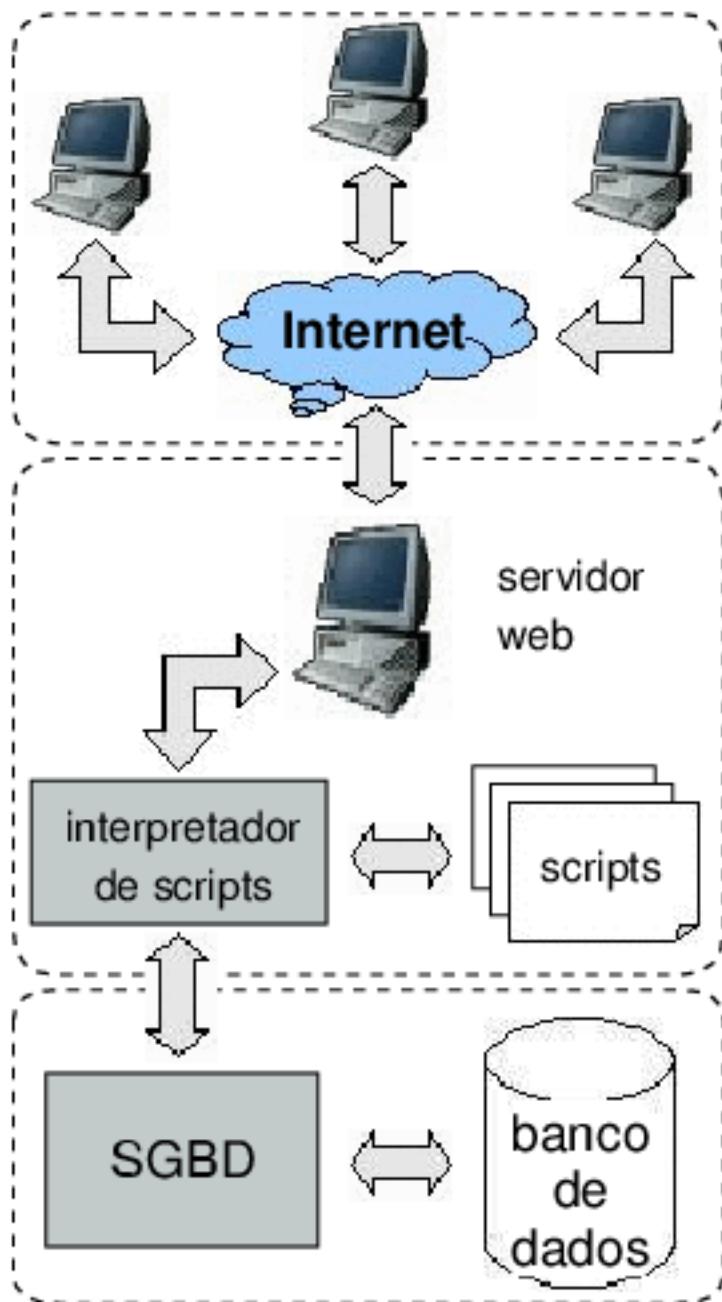
</html>

Cabeçalho

Conteúdo

- Ou uma mensagem de erro (Ex.: Erro 404)

1.3 Arquiteturas em 3 Camadas



- Modelo conceitual para aplicações web
- 1a camada: Cliente
 - ▶ Navegador
- 2a camada: Camada de Aplicação
 - ▶ Servidor Web
 - ▶ Interpretador
 - ▶ Scripts (PHP)
- 3a camada: Sistema de Gerenciamento de Banco de Dados
 - ▶ SGBD (MySQL)
 - ▶ Banco de Dados

1a Camada: Cliente

- Normalmente consiste de um navegador acessando documentos via web
 - ▶ Cliente "leve": não requer que software especializado seja instalado na máquina do usuário
 - Em contraste com clientes "pesados", que requerem a instalação de software especializado
- Navegadores enviam pedidos e exibem as respostas de servidores Web
 - ▶ Exibem documentos *html* com texto, imagens, hiper-links
 - ▶ Alguns podem exibir sons e vídeos
 - ▶ Suporte para documentos dinâmicos usando javascript
 - ▶ Conteúdo especializado pode ser suportado com "applets" em Java, por exemplo

2a Camada: Camada Intermediária

- Também é chamada de camada de aplicação
- Onde a maior parte da lógica da aplicação é processada
- Transforma os pedidos lógicos do usuário em consultas ao Banco de Dados
- Integra os dados obtidos do BD em documentos enviados ao Cliente para ser exibido
- Executada em um servidor web (ou servidor http)
 - ▶ Responsável por receber pedidos *http* de clientes, formular respostas e enviá-las de volta aos clientes
 - Pedidos de conteúdo estático (normalmente html)
 - Pedidos de conteúdo dinâmico criado por programas
 - ▶ Conteúdo dinâmico requer o uso de uma linguagem apropriada (PHP no nosso caso)

Apache e PHP

- O servidor web mais comumente usado é conhecido como servidor Apache ou simplesmente Apache
 - ▶ Software livre produzido pela *Apache Foundation* (www.apache.org)
- Conteúdo dinâmico processado por aplicações externas ou módulos
 - ▶ O mais popular é o módulo PHP
- Pedidos de conteúdo dinâmico
 - ▶ Entregues pelo Apache a um engenho PHP
 - ▶ Engenho PHP
 - Recupera *script* do disco
 - Compila o *script* (se necessário)
 - Executa o *script*
 - ▶ Resultado do processamento é entregue ao Apache que o envia ao cliente

1.4 Apache e PHP

PHP e aplicações web

- PHP é rápido, escalável e muito popular
- Fonte livre e gratuito
 - ▶ Mantido pela comunidade de usuários
 - ▶ Não é influenciado por interesses do fabricante
- Multiplataforma
 - ▶ Diversos Sistemas Operacionais
 - ▶ Pode ser usado em outros servidores além do Apache

PHP e aplicações web (continuação)

- Integração extremamente simples com Html
 - ▶ Apenas a parte dinâmica do documento precisa ser escrita em PHP, com o restante usando html
- Excelente biblioteca
 - ▶ Fácil acesso a diversos SGBDs
 - Integração com BDs MySQL muito simples e elegante
 - ▶ Funções de acesso a diversas facilidades
 - Expressões regulares, correio eletrônico, compressão de dados, etc

3a Camada: Banco de Dados

- É a camada base sobre a qual repousa a estrutura de uma aplicação web de banco de dados
- A camada de banco de dados é responsável pela gerência das informações
 - ▶ Armazenamento e acesso aos dados
 - ▶ Permite atualização dos dados
 - ▶ Assegura integridade dos dados
 - ▶ Controla o acesso *concorrente* aos dados
 - ▶ Provê serviços como estatísticas e cópias de segurança (*backup*)

3a Camada: Banco de Dados

- Normalmente implementada por um SGBD (Sistema de Gerenciamento de Banco de Dados) relacional
 - ▶ Dados são armazenados sob a forma de relações (tabelas)
 - ▶ Detalhes de como o SGBD são escondidos do usuário
 - ▶ Maior parte da funcionalidade acessada por uma *linguagem de consulta*
 - SQL (*Structured Query Language*) é a linguagem de consulta mais usada

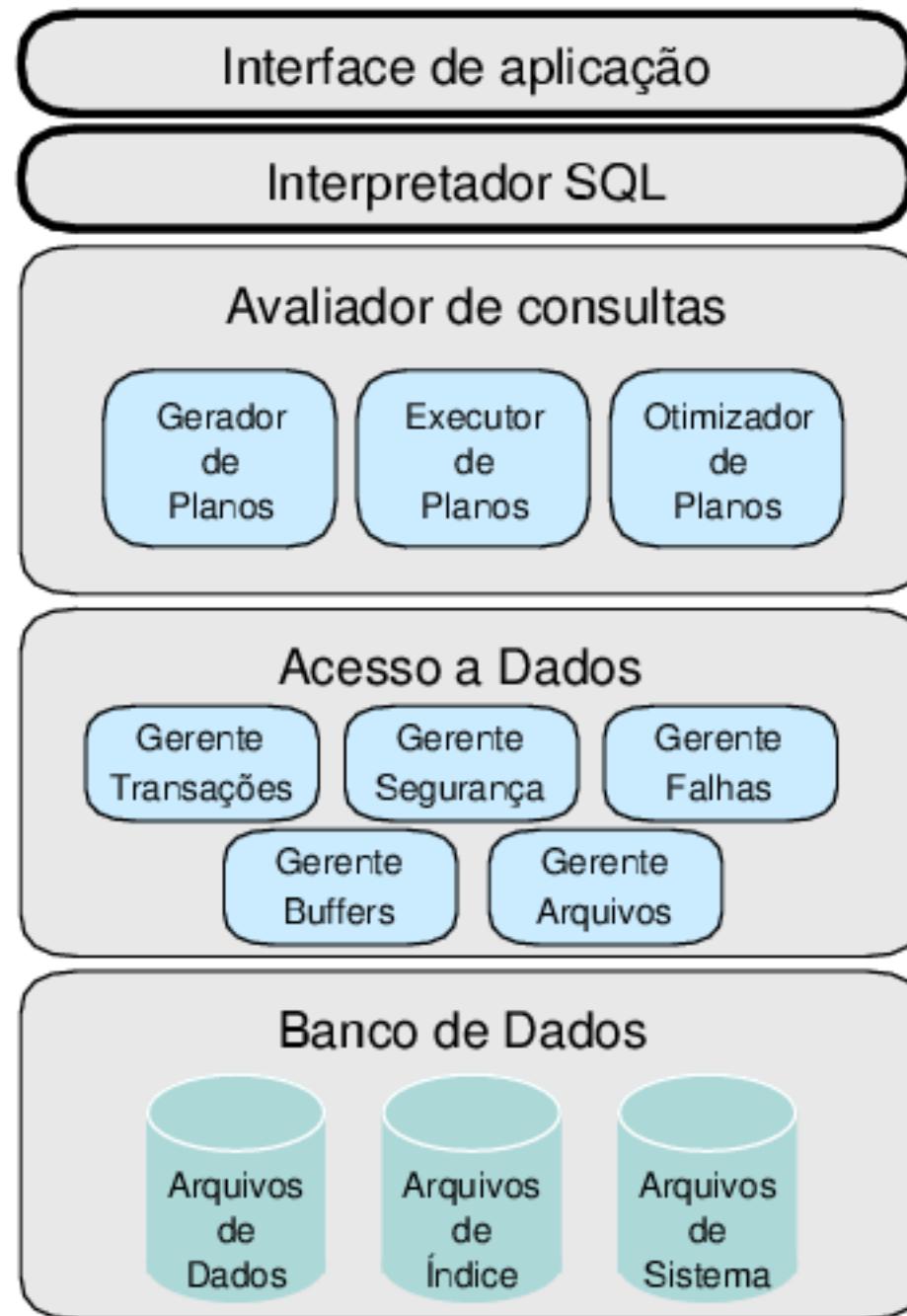
1.5 Sistemas de Gerenciamento de Banco de Dados

- Um banco de dados é uma coleção de dados que têm relação entre si
 - ▶ Um caderno de endereços
 - ▶ Dados relativos a clientes de um banco
 - ▶ Estoque de uma loja
- Um BD pode ser pequeno ou grande, mas é manipulado de forma unificada por um SGBD

Sistemas de Gerenciamento de Banco de Dados

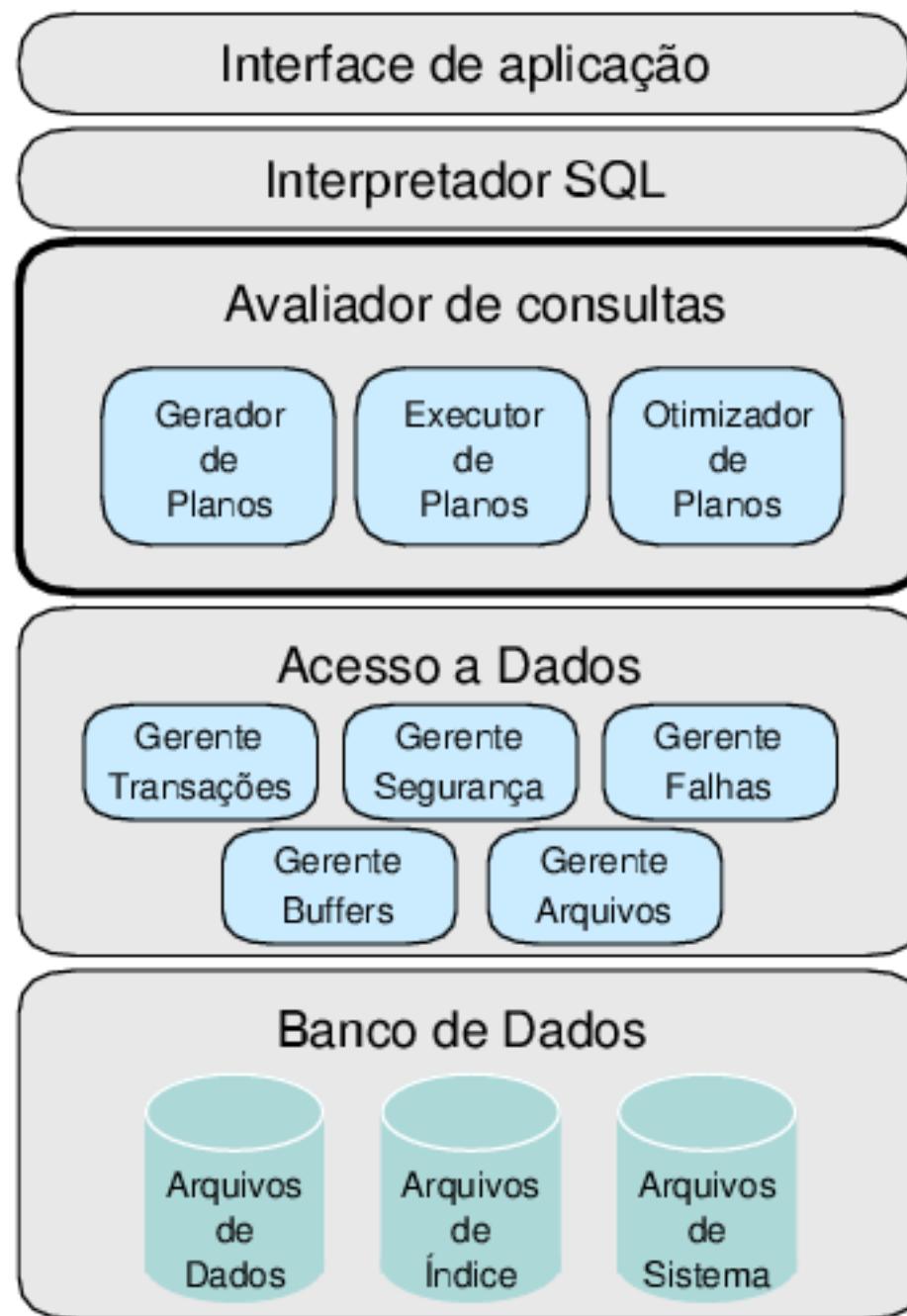
- Um SGBD armazena, recupera e gerencia dados em um BD
- Vamos nos concentrar apenas em SGBDs *relacionais*
 - ▶ Dados são organizados em tabelas ou *relações*
 - ▶ Um SGBD relacional é capaz de gerenciar relacionamentos entre as diversas tabelas de um BD
- O SGBD que iremos empregar é o MySQL

Estrutura de um SGBDR



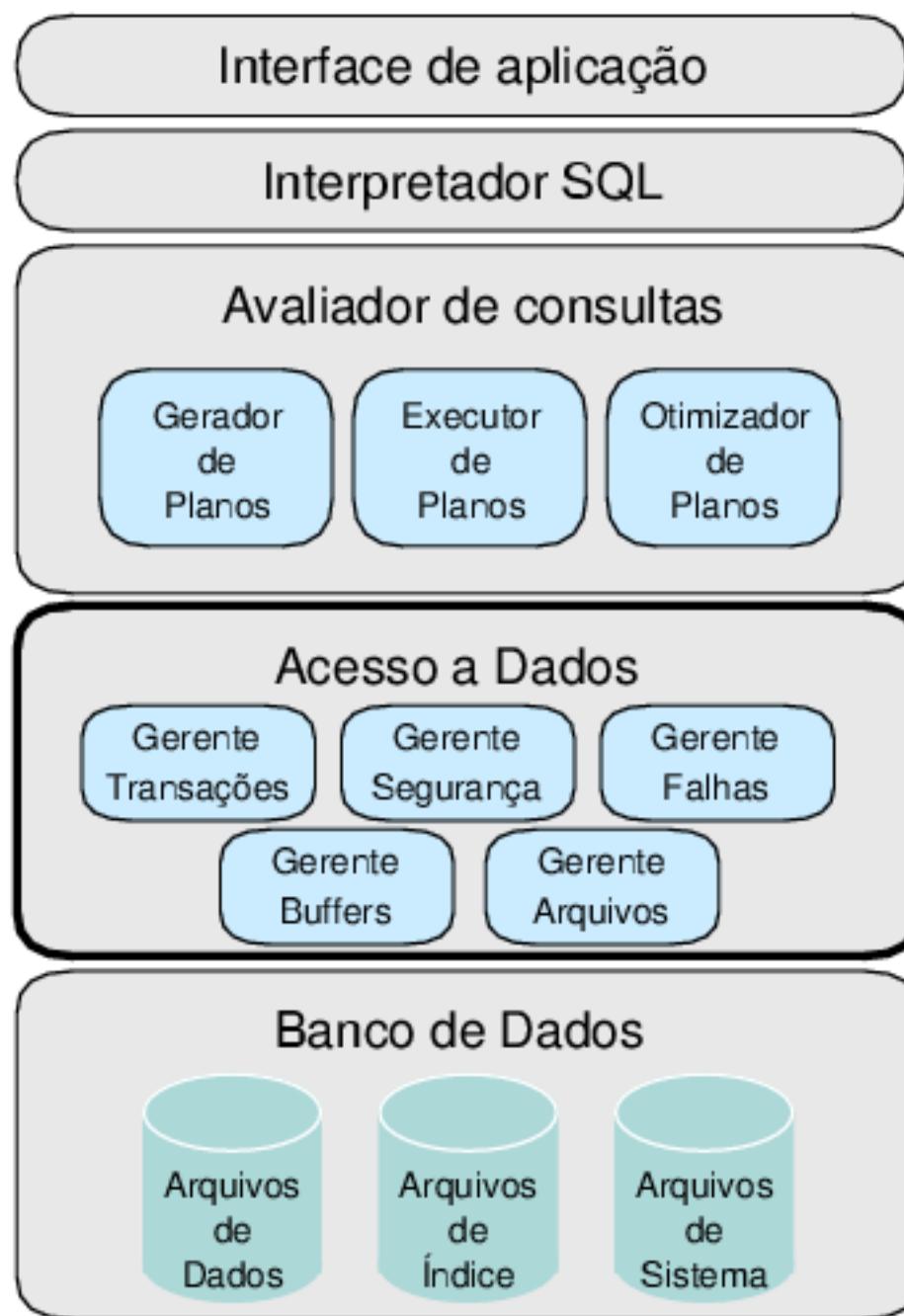
- Interface de Aplicação
 - ▶ Bibliotecas para acesso ao SGBD
 - ▶ Interface de linha de comando
 - ▶ Biblioteca de funções dentro da linguagem script
- Interpretador SQL
 - ▶ Verifica a sintaxe de comandos SQL
 - ▶ Transforma os comandos em uma representação interna

Estrutura de um SGBDR



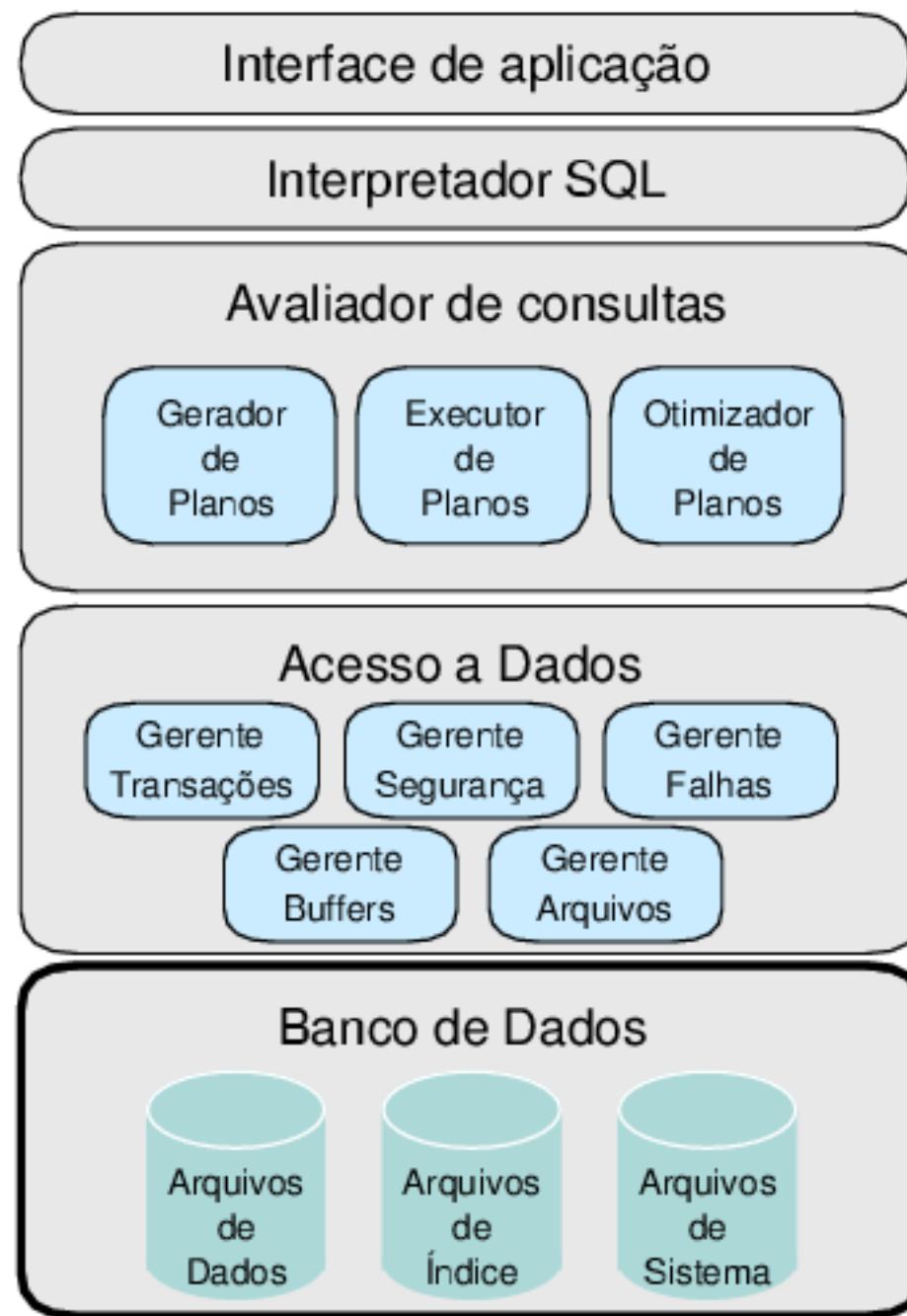
- Avaliador de consultas
 - ▶ Gera planos alternativos para execução da consulta
 - ▶ Escolhe o plano mais apropriado e o otimiza
 - Baseado em características dos algoritmos de acesso e estatísticas sobre o BD
 - ▶ Executa o plano otimizado

Estrutura de um SGBDR



- Acesso aos dados: Módulos para gerenciar o acesso aos dados em disco
 - ▶ **Gerente de Transações:** assegura que acessos concorrentes deixem o BD num estado consistente
 - ▶ **Gerente de Segurança:** barra acesso ao BD de pessoas ou programas não autorizados
 - ▶ **Gerente de Buffers:** gerencia a memória usada no acesso ao BD
 - ▶ **Gerente de Falhas:** assegura que falhas não comprometam integridade do BD
 - ▶ **Gerente de Arquivos:** garante acesso eficiente ao BD

Estrutura de um SGBDR



- Banco de Dados
 - ▶ Armazenados em arquivos em disco
 - ▶ Arquivos de Índice permitem acesso direto a informações baseado em conteúdo
 - ▶ Arquivos de sistema contêm dados estatísticos e atuariais sobre os dados armazenados
- Usados pelo otimizador de planos

Por que usar um Banco de Dados?

- Independência do programa que acessa os dados
- Possibilidade de descrever os dados desejados sem especificar os algoritmos para acessá-los
- Gerenciar relacionamentos complexos
- Acesso concorrente a dados compartilhados
 - ▶ Transações
- Escalabilidade
 - ▶ Grandes massas de dados são armazenadas e acessadas de forma eficiente
- Segurança
- Tolerância a falhas
- Interfaces de administração
- Sistema integrado pronto para o uso

O SGBD MySQL

- É um sistema de médio porte mas que suporta bancos de dados de grande porte
 - ▶ Não suporta grandes volumes de acessos concorrentes
 - ▶ Não suporta algumas facilidades para consulta mais sofisticadas
- Mais rápido que muitos sistemas de maior porte
 - ▶ Outro SGBD mais poderoso porém um pouco mais lento é o PostgreSQL
- Pode ser tanto usado sob licença de software livre como sob licença comercial suportado pela MySQL AB
- No momento em que estamos falando, a versão mais recente do software é a versão 5.0

História do SQL

- Linguagem de consulta estruturada inicialmente proposta pela IBM nos anos 70
- Em 1986, os institutos internacionais ISO e ANSI publicaram a primeira especificação padronizada
- A versão do padrão mais comumente implementada e suportada é a conhecida como SQL-89
- Muitos sistemas implementam apenas algumas funcionalidades do padrão seguinte, conhecido como SQL-92 (ou SQL-2)
- O padrão mais recente, SQL-99 (ou SQL-3) ainda é pouco suportado
- O MySQL implementa a versão mais simples do padrão SQL-92

Componentes do SQL

- Linguagem de Definição de Dados
 - ▶ Permite
 - Criar e remover bancos de dados
 - Criar e remover tabelas (alterar o *esquema* de um BD)
 - Criar e remover índices
 - ▶ Usada durante a construção de um banco de dados
- Linguagem de Manipulação de Dados
 - ▶ Permite realizar buscas, consultar e alterar tabelas
 - ▶ É o conjunto de comandos normalmente empregado por uma aplicação no uso normal de um banco de dados

Componentes do SQL

- Controle de transações
 - ▶ Operações para tratar conjuntos de operações como um bloco
 - ▶ Permite, por exemplo, desfazer o efeito de uma transação num BD
- Facilidades avançadas
 - ▶ Definição de *visões*
 - ▶ Controle de direitos de acesso e de integridade