**2. conceitos e arquitetura do sistema de banco de dados**

O recente crescimento na quantidade de dados exigiu que os sistemas de banco de dados tivessem uma arquitetura distribuida, composta de milhares de computadores que gerenciam os armazenamentos de dados.

Uma SGDB básico é distribuída em dois modulos, o cliente e o servidor. O **módulo cliente** é projetado para executar em uma dispositivo móvel, estação de trabalho ou PC, e geralmente os programas de aplicação e interfaces com o usuário que acessam o banco de dados executam no módulo cliente, assim, se encarrega da interação do usuário e oferece interfaces amigaveis, como GUIs. O **módulo servidor** é responsável pelo armazenamento de dados, acesso, pesquisa e outras funções.

**2.1 Modelos de dados, esquemas e instâncias**

A abstração de dados se refere a supressão de detalhes de organização e armazenamento, destacando os recursos essenciais para melhor conhecimento desse dados, e esta característica é uma das principais da abordagem de BD.

Uma **modelo de dados** é uma coleção de conceitos que podem ser usados para descrever a estrutura de uma banco de dados.Com estrutura de banco de dados, pode se entender “tipos”, os relacionamentos e as restrições que se aplicam aos dados. A maioria desses modelos de dados, incluem uma conjunto de **operações básicas** para especificar recuperações e atualizações no banco de dados.

É comum também incluir conceitos no modelo de dados para especificar o **aspecto dinâmico** ou **comportamento** de uma aplicação de banco de dados.Oque permite ao projetista, especificar uma conjunto de operações válidas, definidas pelo usuário sobre os objetos do banco de dados.

Uma exemplo de operação definida pelo usuário é a CALCULA\_MEDIA que pode ser aplicada sobre uma objeto ALUNO, por sua vez, operações de inserir,excluir,modificar ou recuperar qualquer tipo de objeto, está incluso nas operações básicas.

2.1.1 Categorias de modelos de dados

**Modelos de dados de alto nível ou conceituais** oferecem conceitos próximos ao modo como muitos usuários percebem os dados, enquanto **modelos de dados de baixo nível ou físicos** oferecem conceitos que descrevem os detalhes de como os dados são armazenados no computador, em geral em discos magnéticos. Esses conceitos oferecidos pelos modelos de dados físicos costumam ser voltados para especialistas e não para usuários finais.

A classe intermediária é a de **modelos de dados representativos ou implementação** que oferece conceitos que podem ser facilmente entendidos pelos usuários finais,, e não estão muito longe do modo como os dados são organizados e armazenados no computador, esse tipo de modelo oculta muitos detalhes do armazenamento de dados em disco, mas pode ser implementado diretamente em uma sistema de computador.

Os modelos de dados conceituais ultilizam conceitos como entidades, atributos e relacionamentos. Uma **entidade** representa uma objeto do mundo real, como uma funcionário em uma projeto do minimundo que é descrito no banco de dados. Uma **atributo** representa alguma propriedade de interesse que descreve melhor uma entidade, como nome ou salário do funcionário. Uma **relacionamento** entre duas ou mais entidades representa uma associação entre elas, como por exemplo uma relacionamento “trabalha em” entre uma funcionário e uma projeto.

Os modelos de dados representativos ou de implementação são os usados com maior frequência nos SGBDs comerciais tradicionais. Estes incluem o amplamente utilizado **modelo de dados relacional**, e os **modelos de rede e hierárquicos**.

Os **modelos de dados de objeto** são uma exemplo de uma nova família de modelos de dados de implementação de nível mais alto e que são mais proximos dos modelos de dados conceituais.

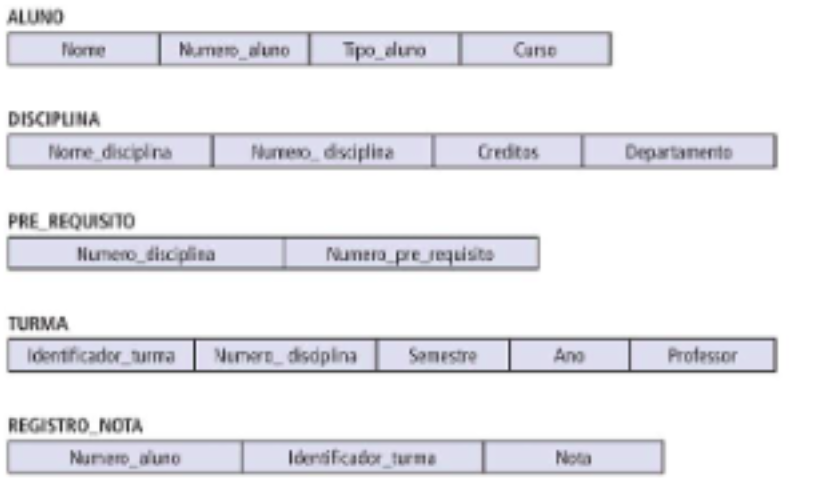
Os modelos de dados físicos descrevem como o dados é armazenado como arquivos no computador, com informações como formatos de registros, ordenações de registro e caminhos de acesso. Uma **caminho de acesso** é uma estrutura que torna eficiente a busca por registros de uma banco de dados, em particular, como indexação ou hashing.Uma **indice** é uma exemplo de uma caminho de acesso que permite o acesso direto aos dados, usando uma termo de índice ou uma palavra-chave.

2.1.2 Esquemas, instâncias e estado do banco de dados

A **descrição de uma banco de dados** é cahmada de **esquema do banco de dados**, que é especificado durante o projeto, e não se espera que mude com frequência, geralmente esses esquemas são representados como diagramas, essa representação é chamada de **diagrama de esquema**.

A figura abaixo exemplifica esse diagrama de esquema, apresentando a estrutura de cada tipo de registro, mas não as instancias reais dos registros, cada objeto(ALUNO/DISCIPLINA/…) é chamado de **construtor do esquema**.

Nomes de tipos de registro, itens de dados, restrições, são representados no esquema.



Os dados no BD em determinado momento no tempo são chamados de **estado** ou **instante do banco de dados**. Também podem ser chamados de conjunto atual de  **ocorrências** ou **instancias**. Em determinado estado do banco de dados, cada construtor de esquema tem o próprio conjunto de instâncias atuais, por exemplo, o construtor ALUNO terá o conjunto e entidades de cada aluno como suas instâncias. Ao adicionar, excluir ou atualizar o valor de uma item de dados em uma registro, o estado do banco de dados é alterado para outro.

Ao **definir** uma novo banco de dados, seu esquema é especificado para o SGBD, e nesse momento o estado correspondente é o estado vazio. O estado inicial do banco de dados é obtid quando ele é **populado** ou **carregado** com os dados iniciais. O SGBD é responsável por garantir que todos estado do banco de dados seja uma **estado válido**, ou seja, que satisfaça a estrutura e as restrições especificadas no esquema.

**Evolução do esquema** é quando uma esquema precisa ser alterado.

**2.2 Arquitetua de três esquemas e independência de dados**

1)Uso de uma catálogo para armazenar a descrição do banco de dados para torná-lo autodescritivo

2)Isolamento de programa e dados

3)Suporte para múltiplas visões

Características importantes na abordagem de banco de dados

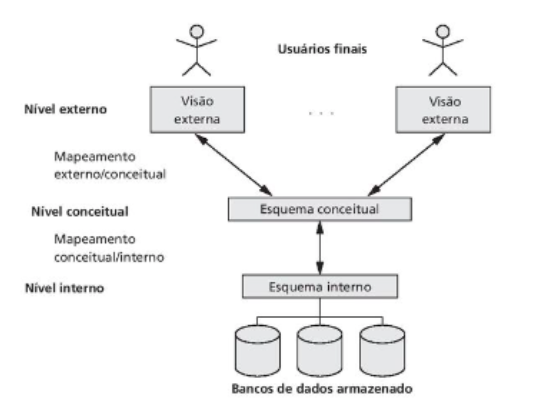
2.2.1 A arquitetura de três esquemas

O objetivo desse tipo de arquitetura é separar as aplicações do usuário do banco de dados físico.Os esquemas podem ser definidor nos três níveis a seguir:

1)Nível interno: Descreve a estrutura do armazenamento físico do banco de dados, utilizando uma modelo de dados físico e descreve os detalhes completos de armazenamento de dados e os caminhos de acesso para o BD

2)Nível conceitual: Descreve a estrutura do banco de dados inteiro para uma comunidade de usuários. Esse esquema oculta detalhes as estruturas de armazenamento físico e se concentra na descrição de entidades, tipos de dados, relacionamentos, operações do usuário e restrições.

3)Nível externo: Cada esquema descreve a parte do banco de dados em que uma grupo de usuários em particular está interessado e oculta o restante do banco de dados do grupo de usuários. Cada esquema externo é implementado com o uso de uma modelo de dados representativo.



Esse tipo de arquitetura é uma ferramente conveniente, com a qual o usuário pode visualizar os níveis de esquema em uma sistema de banco de dados.

Os três esquemas são descrições de dados, os dados que realmente estão armazenados estão no nível físico.Como na arquitetura de três esquemas cada grupo de usuários recorre ao seu próprio esquema externo, o SGBD precisa transformar uma solicitação especificada em uma esquema externo em uma solicitação no esquema conceitual, e depois em uma solicitação no esquema interno para o processamento no banco de dados.

Os processos de transformação de requisições e os resultados entre os níveis são chamados de **mapeamentos**, e podem ser demorados.

2.2.2 Independência de dados

A capacidade de alterar o esquema em uma nível do sistema sem ter de alterar o esquema no próximo nível mais alto.

1)**Independência lógica de dados**: A capacidade de alterar o esquema conceitual sem ter de alterar os esquemas externos ou os programas de aplicação.

2)**Independência física de dados**: A capacidade de alterar o esquema interno sem ter de alterar o esquema conceitual, logo os esquema externos também não precisam ser alterados, isso pode acontecer pois caso alguns arquivos físicos forem reorganizados, por exemplo, ao criar estruturas de acesso adicionais.

A independencia física de dados existe na maioria dos banco de dados e ambientes de arquivo, nos quais os detalhes físicos, como a localização exata dos dados no disco, e detalhes de hardware sobre codificação do armazenamento são ocultados do usuário. Mas a independencia lógica de dados é mais dificil de ser alcançada pois permite alterações estruturais e de restrição sem afetar os programas de aplicação.

**2.3 Linguagens e interfaces do banco de dados**

2.3.1 Linguagens do SGBD

É muito importante especificar esquemas conceituais e internos para o banco de dados e quaisquer mapeamentos entre os dois, e em muitos SGBDs em que não há uma separação estrita de níveis, uma linguagem chamada **linguagem de definição de dados(DDL)** é usada pelo DBA e pelos projetista de BD para definir os dois esquemas. O SGBD terá uma compilador da DDL cuja a função é processar instruções da DDL a fim de identificar as descrições dos construtores de esquema, e armazenar a descrição de esqeuma no catálogo do SGBD.

Nos SGBDs que há uma separação clara entre os níveis conceituais e interno a DDL é usada apenas para especificar o esquema conceitual, a **linguagem de definição de armazenamento(SDL)** é utilizada para especificar o esquema interno.(O mapeamento entre os dois esquemas podem ser especificados em qualquer uma dessas linguagens). Nos SGBDs relacionais não existe uma linguagens especifica que realize o papel de SDL, então o esquema interno é especificado por uma combinação de funções,parametros e especificações relacionadas ao armazenamento de arquivos. Isto permite aos DBAs controlar opções de indexação e mapeamento de dados.

Para uma arquitetura de três esquemas é importante que haja uma **linguagem de definição de visão(VDL)**, para especificar visões do usuário e seus mapeamentos ao esquema conceitual, entretanto na maioria dos SGBDs a DDL é usada para definir o esquema conceitual e o externo.

As manipulações de banco de dados podem ser realizas pela **linguagem de manipulação de dados(DML)**, que como o nome sugere, possibilita a inserção, remoção, atualização e recuperação de dados.

A SQL é uma combinação de **DDL, VDL** e **DML.**

Existem dois tipos de DML:

1)**DML de alto nível/não procedural:** Especifica operações de banco de dados complexas de forma muito concisa.

2)**DML de baixo nível/procedural:** Embutida em uma linguagem de programação de uso geral, e consegue recuperar registros individuais ou objetos de banco de dados e processa cada uma deles separadamente, portanto precisa de construções de linguagens de programação como laços.

A DML de alto nível é capaz de recuperar diverso registros com apenas uma instrução enquanto a de baixo nível não.

Quando comandos da DML são incorporados em uma linguagem de programação, ela é chamada de **linguagem hospedeira** e a DML de **sublinguagem de dados**.Por sua vez, uma DML de alto nível usada em uma maneira iterativa é chamada de **linguagem de consulta**.

Usuários finais casuais, costumam usar uma linguagem de consulta de alto nível para especificar suas solicitações, enquanto os programadores usam a DML em sua forma embutida. Para usuários comuns e paramétricos, normalmente existem interfaces amigáveis para interagir com o banco de dados.

2.3.2 Interfaces de SGBD

1)**Interfaces baseadas em menu para clientes web ou de navegação:** Apresentam ao usuário uma lista de opções que acompanham o usuário na formulação de uma solicitação. Acabam com a necessidade de memorizar os comandos especificos e a sintaxe de uma linguagem de consulta.

2)**Aplicativos para dispositivos móveis:** Dão acesso aos dados para os usuários móveis, por exemplo, uma banco oferece aplicativos para permitir que os usuários acessem seus dados por meio de uma telefone.

3)**Interfaces baseadas em formulário**: Apresenta uma formulário para cada usuário, os usuários podem preencher todas as entradas do formulário para inserir novos dados ou preencher apenas certas entradas.

4)**Interfaces gráficas com o usuário:** Uma GUI normalmente apresenta uma esquema para o usuário em formato de diagrama, e o usuário pode então especificar uma consulta manipulando o diagrama.

5)**Interfaces de linguagem natural:** Aceitam solicitações escritas em ingles ou em outro idioma, e tentam entende-las, e costuma ter o próprio esquema.

6)**Pesquisa do banco de dados baseada em palavra-chave**: Aceitam sequencias de palavras de linguagem natural e as combinam com documentos em sites específicos

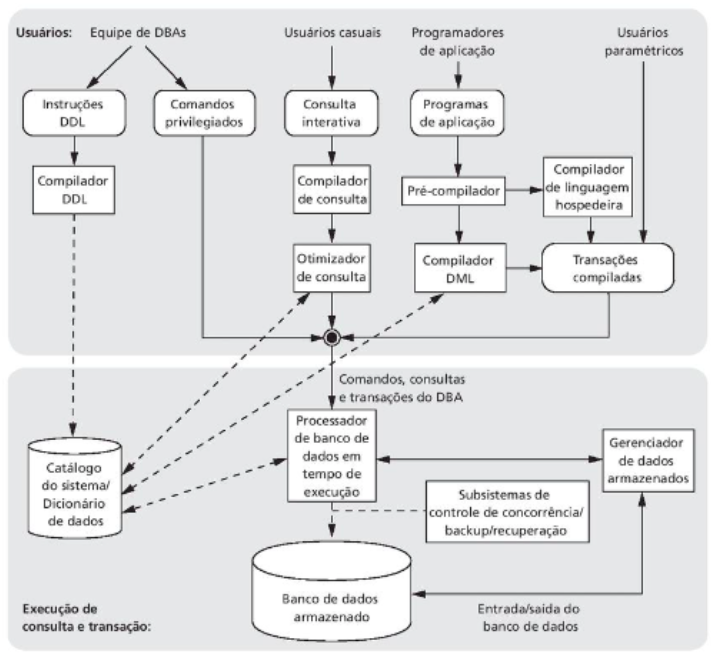
7)**Entrada e saída de voz**: Uso da voz como entrada para uma consulta e como resposta para uma pergunta

8)**Interfaces para usuários paramétricos**: Os usuarios parametricos, como caixas de bancos, possum uma pequeno conjunto de operações que precisam realizar repetidamente, assim, é implementado uma interface especial para cada classe de usuário

9)**Interfaces para o DBA**

**2.4 O ambiente do sistema de banco de dados**

2.4.1 Módulos componentes do SGBD



Essa figura ilusta os componentes típicos do SGBD, e está divida em duas partes, a superior, se refere aos diversos usuários do ambiente do banco de dados e suas interfaces, já a parte inferior mostra os módulos interno do SGBD, responsáveis pelo armazenamento de dados e processamento de transações.

O banco de dados e o catálogo do SGBD geralmente são armazenados em disco, e o acesso ao disco é controlado pelo SO, que escalona a leitura/escrita em disco. Muitos SGBDs possuem o próprio módulo de **gerenciamento de buffer** para planejar a leitura/escrita em disco, visto que isso causa uma efeito considerável no desempenho.

A equipe de DBAs define o banco de dados e faz ajustes, alterando sua definição por meio da DDL e de outros comandos privilegiados, o compilador da DDL então, processa as definições de esquema especificadas e armazena as descrições dos esquemas no catálogo do SGBD.

Os usuarios casuais e as pessoas com necessidade ocasional de informações do banco de dados interagem usando a interface de consulta interativa. Essas constultas são analisadas por uma **compilador de consulta**, que as compila para uma formato interno, o **otimizador de consulta** preocupa-se com o rearranjjo e a possível ordenação de operações, com a eliminação de redundancias.

O pré-compilador extrai comandos DML do programa de aplicação escrito em uma linguagem de programação hospedeira, esses comandos são enviados para serem compilados em código objeto para o acesso ao banco de dados.

O **processador de banco de dados em tempo de execução** exeucta os comandos privilegiados, os planos de consulta executaveis e as transações programadas com parametros em tempo de execução. Trabalha com o **catálogo do sistema** e pode atualiza-lo com estatisticas, também trabalha como o **gerenciador de dados armazenados**.

O programa cliente pode acessar o SGBD executando em uma computador separado daqueles em que o banco de dados reside. O primeiro é chamado de computador cliente, que executa uma software cliente, o segundo é chamado de servidor de banco de dados. A figura ilusta os módulos básico de uma SGBD

2.4.2 Utilitários do sistema de banco de dados

Ajudam o DBA a gerenciar o sistema:

1. **Carga:** É usado para carregar os arquivos de dados existentes no banco de dados.

2) **Backup:** Cria uma cópia de segurança do banco de dados.

3) **Reorganização do armazenamento do banco de dados:** Pode ser usado para reorganizar um conjunto de arquivos do banco de dados em diferentes organizações de arquivo e criar novos caminhos de acesso para melhorar o desempenho.

4) **Monitoração de desempenho:** Monitora o uso do banco de dados e oferece estatísticas ao DBA.

2.4.3 Ferramentas, ambientes de aplicação e facilidades de comunicações

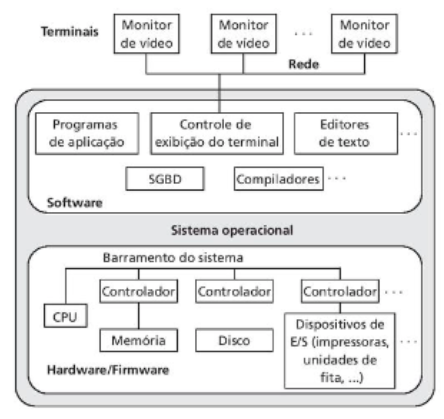
Ferramentas que estão frequentemente disponíveis aos projetistas de banco de dados:

As ferramentas CASE são usadas na fase de projeto dos sistemas de banco de dados, além dessa ferramenta uma muito útil se chama **sistema de dicionario de dados** que além de armazenar informações de catálogo sobre esquemas e restrições o dicionário de dados armazena também decisões do projeto, padrões de uso, descrições do programa de aplicação e informações do usuário. Os **ambientes de desenvolvimento de aplicação** são muito populares, e ofrecem uma ambiente para desenvolver aplicações de banco de dados e incluem estruturas que ajudam em muitas facetas dos sistemas, incluindo projeto de banco de dados, desenvolvimento GUI, consulta e atualização e desenvolvimento de programas de aplicação. Finalmente, há também o **software de comunicações**, cuja a função é permitir que os usuários em locais remotos do sistema de banco de dados acessem-no por meio de terminais de computador, estações de trabalho ou computador pessoais.

**2.5 Arquiteturas centralizadas e cliente/servidor para SGBDs**

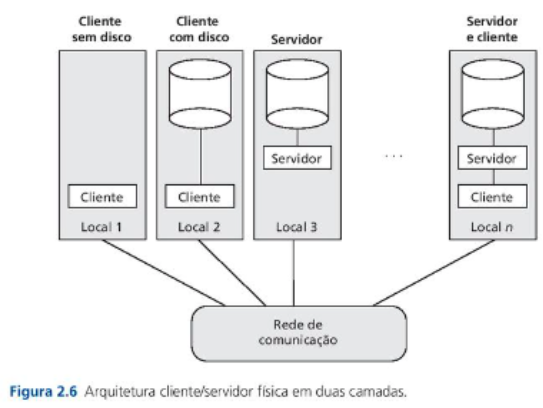
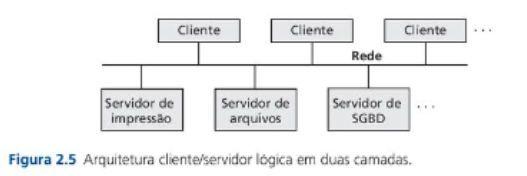
2.5.1 Arquitetura de SGBDs centralizada

Esse tipo de arquitetura é antiga, em que a funcionalidade de execução de programas de aplicação e processamento de interface do usuário eram realizados em uma máquina, a figura abaixo representa os componentes físicos em uma arquitetura centralizada.



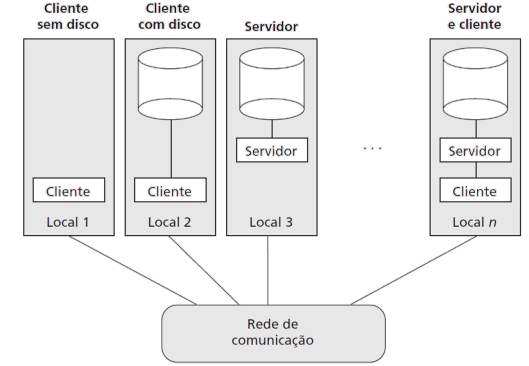
2.5.2 Arquiteturas cliente/servidor básicas

A arquitetura cliente/servidor foi desenvolvida para lidar com ambientes de computação em que uma grande numero de PCs são conectados por uma rede. A ideia é definir servidores especializados com funcionalidades específicas, por exemplo, pode-se conectar uma serie de PCs como clientes a uma servidor de arquivos que mantém os arquivos das máquinas clientes.As **máquinas clientes** oferecem ao usuário interfaces apropriadas para utilizar esses servidores.



O conceito desse tipo de arquitetura assume uma estrutura básica, composta por muitos PCs/estações de trabalho. Uma **cliente** nesse estrutura normalmente é uma máquina que oferece capacidades de interface com o usuário e processamento local, quando o cliente requer acesso a alguma funcionalidade adicional como acesso ao banco de dados, ele se conecta a uma servidor que oferece a funcionalidade necessária. Uma **servidor** é uma sistema com hardware e software que pode oferecer serviços às máquinas clientes como acesso a arquivo, impressão, arquivamento ou acesso a banco de dados. É comum que o software cliente e servidor seja executado em máquinas separadas.

Existem dois tipo de arquitetura SGBD, nesse estrutura cliente/servidor: duas camadas e três camadas.



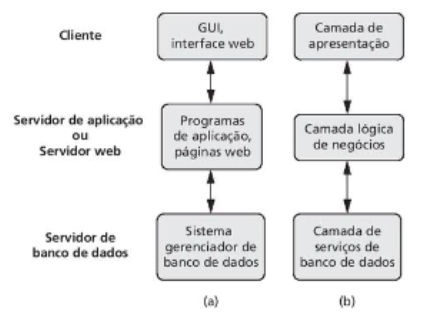
2.5.3 Arquiteturas cliente/servidor de duas camadas para SGBDs

Em sistemas de banco de dados relacionais os componentes do sistema movidos inicialmente para o lado do cliente foram a interface com o usuário e os programas de aplicação. Nesse tipo de arquitetura o servidor frequentemente é chamado de **servidor consulta** ou **servidor de transação** pois oferece a funcionalidade de consulta e de transação relacionadas ao processamento da SQL. Já os programas da interface com o usuário e os programas de aplicação podem ser executados no lado do cliente, quando é necessário acessar o SGBD o programa estabelece uma conexão com o SGBD, assim quando a conexão é criada, o programa cliente pode se comunicar com o SGBD.

Resumindo como os componentes de software são distribuídos por dois sistemas, cliente e servidor, essa arquitetura é chamada de “duas camadas”. Suas vantagens são a sua simplicidade e compatibilidade transparente com os sistemas existentes.

2.5.4 Arquitetura de três camadas e n camadas para aplicações web

Muitas aplicações web ultiizam essa arquitetura pois ela acrescenta uma camada intermediária entre o cliente e o servidor de BD. Essa **camada intermediária** é chamada de **servidor de aplicação** ou **servidor web**, esse servidor desempenha uma papel intermediario pela execução de programas de aplicação e armazenamento de regras de procedimentos ou restrições, usados para acessar os dados do servidor de banco de dados, ela também melhora a segurança do banco de dados, verificando as credenciais de uma cliente, antes de encaminhar a solicitação a uma servidor. Esse servidor web, aceita e processa solicitações do cliente, enviando consultas e comandos do banco de dados ao servidor, e atua como uma canal para passar dados processados do servidor aos clientes.



**2.6 Classificação dos sistemas gerenciadores de banco de dados**

Existem vários critérios para classificar os SGBDs:

1)Modelo de dados no qual o SGBD é baseado:

O principal modelo de dados usado atualmente em muitos SGBDs comerciais é o modelo de dados relacional, e os sistemas baseados nele são conhecidos como sistemas SQL.

2)Número de usuários suportados pelo sistema:

Sistemas monousuários, admitem apenas uma usuário de cada vez, e são usados principlamente em computadores pessoais, já sistemas multiusuários que incluem a maioria dos SGBDs admitem múltiplos usuários simultâneos.

3)Número de locais sobre os quais o banco de dados está distribuído

Uma SGBD é centralizado se os dados estiverem armazenados em uma unico comutador, podem atender diverso usuários, mas o SGBD e o banco de dados residem integralmente em uma único computador.

Uma SGBD é distribuido(SGBDD) é quando se tem uma banco de dados real e o software distribuido por vários locais, conectados por uma rede de computadores.

Os SGBDD homogêneos usam o mesmo software de SGBD em todos os locais, ao passo que SGDBDDs heterogêneos podem usar uma software de SGBD diferente em cada local.Também é possível desenvolver uma software **mediador** para acessar vários bancos de dados autonomos preexistentes, armazenados sob SGBDs heterogêneos. Isso leva a uma SGBD federado.

4) Custo

5)Opções de tipos de caminho de acesso

6)Uso geral ou uso especial:

Quando é importante o desempenho, uma SGBD de uso especial pode ser projetado e construido para uma aplicação específica, esse sistema não pode ser usado para outras aplicações sem mudanças relevantes.

7)Modelo de dados:

O **modelo de dados relacional básico** representa um banco de dados como

uma coleção de tabelas, em que cada tabela pode ser armazenada como um

arquivo separado.

O **modelo de dados de objeto** define um banco de dados em termos de objetos, suas propriedades e operações. Objetos com a mesma estrutura e comportamento pertencem a uma classe, e as classe são organizadas em hierarquias e cada classe pode ter vários métodos

O **modelo de dados de chave-valor** associa uma chave única a cada valor e

fornece acesso muito rápido a um valor a partir de sua chave.

O **modelo de dados de documento** é baseado em JSON e armazena os dados

como documentos.

O **modelo de dados de grafo** armazena objetos como nós de um grafo e relações entre objetos como arestas direcionadas do grafo.

Os **modelos de dados baseados em coluna** armazenam as colunas de linhas

agrupadas em páginas de disco para obter acesso rápido e permitem múltiplas

versões dos dados.

O **modelo hierárquico** representa os dados como estruturas hierárquicas em forma de árvore.