Nota: Este material complementar representa uma cópia resumida do capítulo 1 do livro Sistemas de Gerenciamento Sistemas de Gerenciamento de Banco de Dados - Ramakrishnan Gehrke (3ª edição), versão disponível em pdf na internet e recortes de conteúdos disponibilizados gratuitamente na Internet.

Conceitos Fundamentais e Arquitetura de SGBDs Relacionais

A área de sistemas de gerenciamento de dados é um microcosmo da Ciência da Computação em geral. Os aspectos tratados e as técnicas utilizadas abrangem um amplo espectro, incluindo linguagens, orientação a objeto e outros paradigmas de programação, compilação, sistemas operacionais, programação concorrente, estruturas de dados, algoritmos, teoria, sistemas distribuídos e paralelos, interfaces do usuário, sistemas especialistas e inteligência artificial, técnicas estatísticas e programação dinâmica.

O que é um SGBD, em particular, um SGBD relacional?

- **Sistema de gerenciamento de banco de dados, ou SGBD**, é um software projetado para auxiliar a manutenção, gerenciamento, armazenamento, recuperacao e garantia da integridade de vastos conjuntos de dados. A alternativa para não se usar um SGBD é armazenar os dados em arquivos e escrever código específico do aplicativo para gerenciá-los. Na década de 1980-90

    - O modelo relacional consolidou sua posição como o paradigma dominante de SGBD
    - A linguagem de consulta SQL foi padronizada e o padrão atual, SQL 1999 adotado pelo (ANSI) e (ISO).
    - Diversos fabricantes (como o DB2 da IBM, Oracle 8, Informix2 UDS, Microsoft) estenderam seus sistemas com a capacidade de armazenar novos tipos de dados, como imagens e texto, e a possibilidade de consultas mais complexas.
- **banco de dados** é uma coleção de dados que, tipicamente, descreve as atividades de uma ou mais organizações relacionadas. Por exemplo, um banco de dados de uma universidade poderia conter informações sobre:
    - *Entidades* como alunos, professores, cursos e turmas.
    - *Relacionamentos* entre as entidades, como a matrícula dos alunos nos cursos, cursos ministrados pelos professores, e o uso de salas por cursos.

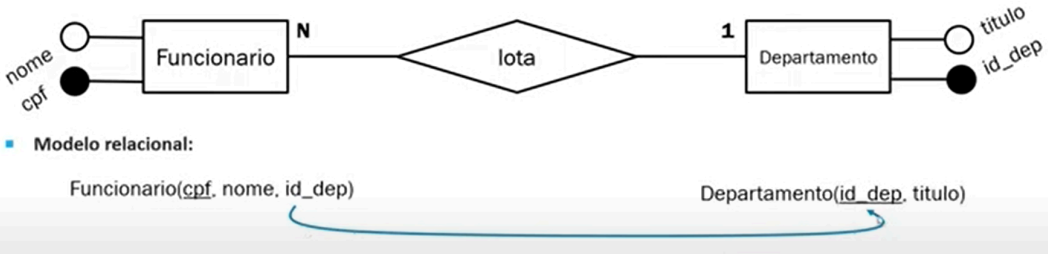
Por que devemos utilizar um SGBD para gerenciar dados?

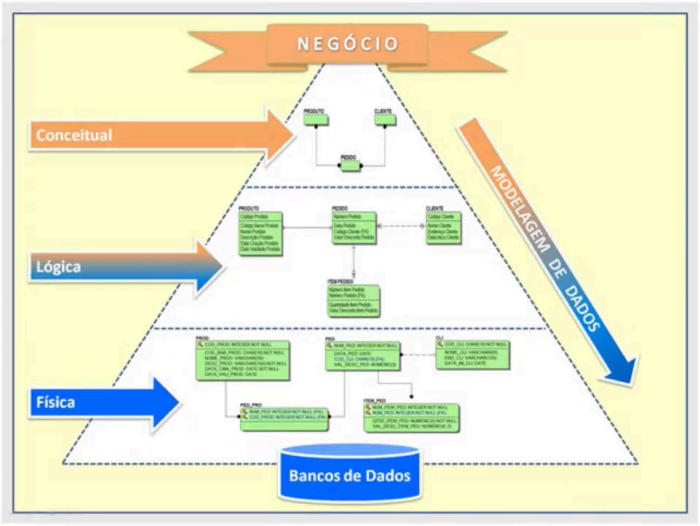
- **Organização e Estruturação dos Dados:**
    - Arquivos de Sistemas: Os dados são armazenados de forma geralmente desestruturada, muitas vezes em arquivos separados e sem uma padronização definida.
    - SGBD: Os dados são organizados em um esquema estruturado, seguindo um modelo de dados específico, como o modelo relacional, facilitando o armazenamento e a recuperação de informações. A aplicação não é exposta aos detalhes de representação e armazenamento do dado. O SGBD fornece uma visão abstrata dos dados.
- **Controle de Acesso e Segurança:**
    - Arquivos de Sistemas: O controle de acesso e a segurança dos dados são geralmente implementados de forma limitada, pelo SO, com poucos mecanismos de proteção.
    - SGBD: Oferece recursos avançados de controle de acesso e segurança, permitindo a definição de permissões granulares para diferentes usuários e garantindo a integridade e a confidencialidade dos dados.
- **Integridade dos Dados:**
    - Arquivos de Sistemas: A integridade dos dados muitas vezes depende da aplicação que os manipula, tornando mais propensos a erros e inconsistências.
    - SGBD: Mantém a integridade dos dados por meio de restrições e regras de acesso definidas que governam quais dados estão visíveis a diferentes classes de usuários.
- **Recuperação eficiente e Backup:**
    - Arquivos de Sistemas: A recuperação e o backup de dados geralmente exigem procedimentos manuais e podem ser complexos, aumentando o risco de perda de dados.
    - SGBD: Oferece recursos sofisticados para recuperação e armazenamento eficiente dos dados, permite a realização de cópias de segurança regulares e a recuperação rápida em caso de falhas ou desastres.

- **Custo na Busca da Informação:**
    - Arquivos de Sistemas: O custo na busca da informação pode ser elevado para encontrar as informações desejadas e pela limitação de tamanho de memória principal, já que o arquivo possivelmente seria carregado na memória.
    - SGBD: Reduz significativamente o custo na busca da informação, pois oferece recursos de otimização de consulta, como índices e otimizadores de consultas, permitindo recuperar os dados de forma mais eficiente e rápida. Um SGBD planeja o acesso concorrente aos dados de maneira tal que os usuários podem achar que os dados estão sendo acessados por apenas um único usuário de cada vez.
- **Tempo Reduzido de Desenvolvimento de Aplicativo:**
    - SGBD: suporta funções importantes que são comuns a vários aplicativos que acessam os dados no SGBD. Isso, em conjunto com uma interface de alto nível aos dados, facilita o desenvolvimento rápido de aplicativos. Os aplicativos de SGBD tendem a ser mais robustos do que os aplicativos similares independentes porque muitas tarefas importantes são tratadas pelo SGBD (e não precisam ser depuradas e testadas no aplicativo).
- **Desvantagens do SGBD:**
    - Um SGBD é um software complexo e seu desempenho pode não ser adequado para determinados aplicativos especializados.
    - o aplicativo pode precisar manipular os dados de maneiras não suportadas pela linguagem de consulta.

Modelo de Dados

- **modelo de dados** é uma coleção de construtores de alto nível para descrição dos dados que ocultam vários detalhes de baixo nível de armazenamento. Os modelos mais comuns são: **Relacional** (que é utilizado em inúmeros sistemas, incluindo o DB2 da IBM, Informix, Oracle, Sybase, Access da Microsoft, FoxBase, Paradox, Tandem e Teradata), **Dimencional**, **Entidade-relacionamento**. Outros mais específicos são: **modelo hierárquico** (por exemplo, usado no SGBD IMS da IBM), o **modelo de rede** (usado no IDS e IDMS), o **modelo orientado a objetos** (por exemplo, usado no Objectstore e Versant) e o **modelo objeto-relacional** (por exemplo, usado nos produtos SGBD da IBM, Informix, ObjectStore, Oracle, Versant e outros). A maioria dos SGBDs atuais baseia-se no **modelo de dados relacional**
    - **modelo de dados semântico** é um modelo de dados de alto nível, mais abstrato, que facilita a um usuário alcançar uma boa descrição inicial dos dados de uma empresa. Um SGBD não é projetado para suportar todos esses construtores diretamente.
    - **modelo entidade-relacionamento (ER)**, é um modelo de dados semântico muito utilizado que nos permite denotar por meio de ilustrações as entidades e os relacionamentos entre elas.





Fonte: www.devmedia.com.br

Característica	Conceitual	Lógico	Físico
Nome de Entidade	✓	✓	
Relacionamentos de Entidade	✓	✓	
Atributos	✓	✓	
Chave Primária		✓	✓
Chave Estrangeira		✓	✓
Nome das Tabelas			✓
Nome das Colunas			✓
Tipo das Colunas			✓

Fonte: pt.stackoverflow.com

A modelagem de dados é um processo fundamental na criação de sistemas de informação eficientes e confiáveis. Esse processo é dividido em três níveis: conceitual, lógico e físico.

- **No nível conceitual,** é feita a representação abstrata do negócio, onde são definidas as entidades, seus atributos e relacionamentos. É importante entender o negócio, seus processos e as necessidades dos usuários para garantir que o modelo conceitual atenda às necessidades do negócio.
- **No nível lógico,** a modelagem é mais detalhada, as tabelas são criadas e as colunas, tipos de dados, chaves primárias e estrangeiras e restrições são definidas. Nesse nível, as regras de integridade e normalização são estabelecidas para garantir a consistência dos dados. A normalização ajuda a reduzir a redundância de dados, melhorando a eficiência do sistema.
- **No nível físico,** o modelo lógico é implementado em um banco de dados real. Questões como particionamento, índices e organização física dos dados são definidas nesse nível. É importante escolher o tipo de banco de dados que melhor se adapta às necessidades do negócio e garantir que o modelo físico suporte as necessidades de armazenamento e recuperação de dados.

O Modelo Relacional

O **modelo relacional** é um modelo de dados que se baseia no princípio de que todos os dados estão armazenados em tabelas (ou, matematicamente falando, relações)[wikipedia]. O conceito foi criado por Edgar Frank Codd em 1970. O construtor central para descrição de dados deste modelo é uma relação, que pode ser considerada um conjunto de **registros**. A descrição de dados é chamada **esquema** (**scheme**).
A informação sobre a **entidade** aluno pode ser armazenada em uma **relação** com o seguinte **esquema**:

Alunos (id-aluno: string, nome: string, login: string, idade: integer, média: real)

O esquema informa que cada registro na relação **Alunos** tem cinco campos, sendo os nomes e tipos dos campos conforme indicados. Cada **linha** na relação Alunos é um **registro** que descreve um aluno. O esquema pode, então, ser considerado um modelo para descrever um aluno.

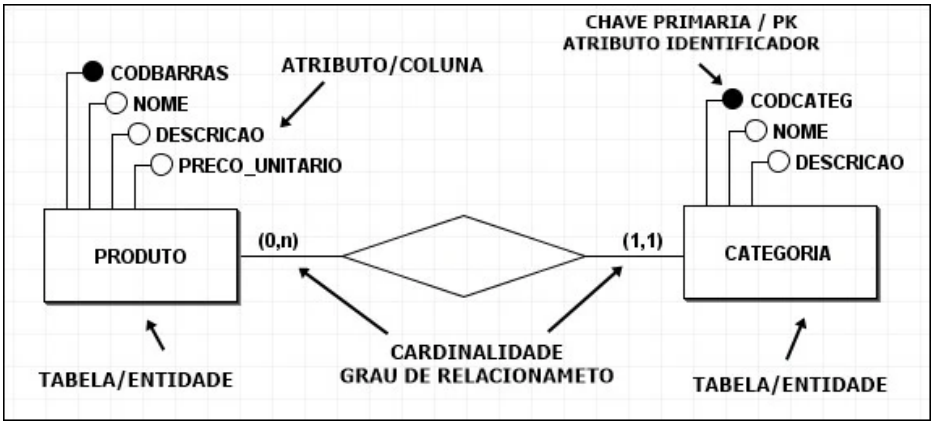
id-aluno	*nome*	*login*	*idade*	*média*
53666	Jones	jones@cs	18	3,4
53688	Smith	smith@ee	18	3,2
53650	Smith	smith@math	19	3,8
53831	Madayan	madayan@music	11	1,8
53832	Guldu	guldu@music	12	2,0

Figura 1.1 Uma instância da relação Alunos.

Podemos tornar a descrição de um conjunto de alunos mais precisa especificando as **restrições de integridade**, que são condições que os registros de uma relação devem satisfazer. Por exemplo, poderíamos especificar que todo aluno tenha um valor id-aluno único.

O modelo relacional se preocupa com a descrição de alguns itens:

- **Tabela/Entidade:** Local onde serão armazenados todos os dados informados ao banco. Dentro da tabela estão contidos as linhas e as colunas As tabelas se relacionam entre si, pode haver dezenas de relações dependendo de como o banco foi modelado
- **Coluna:** Também chamado de atributo, é a característica única da tabela. Cada coluna deve ter um tipo de dado, que será o formato do dado ela armazenará as informações (valor numérico, imagem, etc) inseridas no banco de dados.
- **Linha:** São os registros que foram armazenados na tabela. Também podem ser chamados de tupla ou dados.



Arquitetura do SGBD

https://www.youtube.com/watch?v=cpyplRQFjUw

Um dos principais objetivos de um SGBD é isolar os detalhes internos do banco de dados, do usuário (promover a abstração de dados) e promover a **independência dos dados** em relação a estratégia de **acesso** e a forma de **armazenamento**.

Níveis de Abstração em um SGBD - Arquitetura ANSI/SPARC

Os dados em um SGBD são descritos em três níveis de abstração, conforme ilustrado na Figura 1.2. A descrição do banco de dados consiste em um esquema em cada um desses três níveis de abstração: o conceitual, o físico e o externo.

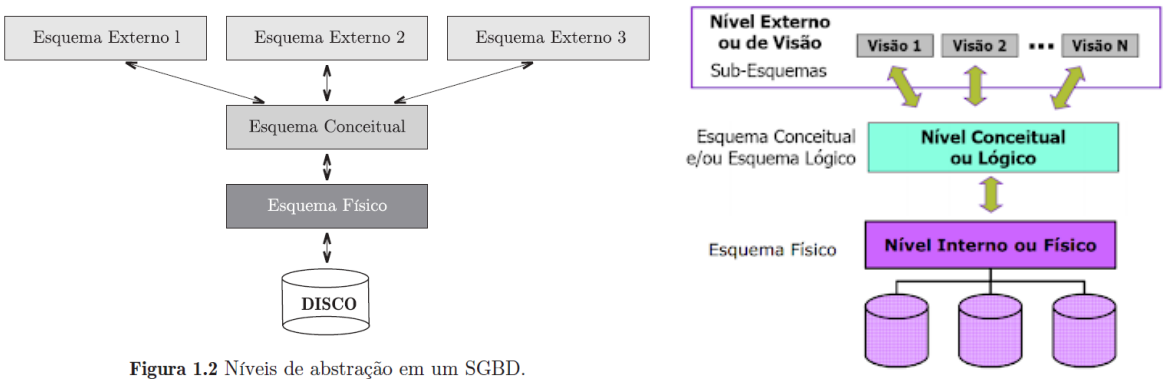


Figura 1.2 Níveis de abstração em um SGBD.

Esquema Conceitual

O esquema conceitual (chamado também de **esquema lógico**) descreve todas as relações que estão armazenadas no banco de dados. Essas relações contêm informações sobre **entidades**, como alunos e professores, e sobre **relacionamentos**, como a matrícula dos alunos nos cursos. Todas as **entidades alunos** podem ser descritas usando-se registros em uma **relação Alunos.**

De fato, cada coleção de entidades e cada coleção de relacionamentos podem ser descritas como uma relação, levando ao seguinte esquema conceitual:

Alunos(*id-aluno:* `string`, *nome*: `string`, *login:* `string`, *idade:* `integer`, *média:* `real`)
Professores(*id-prof:* `string`, *nomep:* `string`, *sal:* `real`)
Cursos(*id-curso:* `string`, *nomec:* `string`, *créditos:* `integer`)
Salas(*num-sala:* `integer`, *endereço:* `string`, *capacidade:* `integer`)
Matriculado(*id-aluno:* `string`, *id-curso:* `string`, *nota:* `string`)
Ministra(*id-prof:* `string`, *id-curso:* `string`)
Aula(*id-curso:* `string`, *num-sala:* `integer`, *horário:* `string`)

O processo de produzir um bom esquema conceitual é chamado **projeto conceitual de banco de dados**.

Esquema Físico

O esquema físico especifica os detalhes adicionais de armazenamento. O esquema físico resume como as relações descritas no esquema conceitual são realmente armazenadas em dispositivos de armazenamento secundário como discos e fitas. Devemos decidir quais organizações de arquivos utilizar para armazenar as relações e criar estruturas de dados auxiliares, chamadas **índices**, para acelerar as operações de recuperação de dados. O processo de produzir um bom esquema físico é chamado **projeto físico de banco de dados**.

Esquema Externo

Esquemas externos, que normalmente também são representados em termos do modelo de dados do SGBD, permitem que o acesso aos dados seja customizado (e autorizado) no nível dos usuários individuais ou em grupos. Qualquer banco de dados tem exatamente um esquema conceitual e um esquema físico porque ele tem apenas um conjunto de relações armazenadas, mas pode ter diversos esquemas externos, cada um adaptado a um grupo particular de usuários. Cada esquema externo consiste em uma coleção de uma ou mais **visões** e relações do esquema conceitual.

O projeto de esquema externo é orientado pelos requisitos do usuário final. Por exemplo, podemos permitir que os alunos localizem os nomes dos professores que ministram cursos, assim como as matrículas no curso. Isso pode ser feito definindo a seguinte visão:

InfoCurso(id-curso: string, nomep: string, matriculados: integer)

Um usuário pode tratar uma visão assim como uma relação e fazer perguntas sobre os registros da visão. Não incluímos InfoCurso no esquema conceitual porque podemos processar InfoCurso com base nas relações do esquema conceitual, e, além disso, armazená-la seria redundante.

Uma **linguagem de definição de dados** (**DDL** — data definition language), ex. SQL é utilizada para definir os esquemas externos e conceituais. As informações sobre os esquemas conceitual, externo e físico são armazenadas nos **catálogos de sistemas.**

Linguagens de manipulação de dados

https://www.youtube.com/watch?v=062Txe5Wsig
Existem três principais linguagens que modificam o banco de dados dentro do SGBD, que são:

DML (Linguagem de Manipulação de Dados): Serve para adicionar, modificar ou remover algum dado de uma tarefa. As principais palavras utilizadas são: CREATE, ALTER e DELETE.

DDL (Linguagem de Definição de Dados): Serve para adicionar ou modificar as estruturas de objetos do banco de dados. As principais palavras utilizadas são: ALTER, CREATE e DROP.

DCL (Linguagem de Controle de Dados): Utilizadas para controlar o acesso de usuários no SGBD. As principais palavras utilizadas são: GRANT, REVOKE e DENY.

Arquitetura simplificada de um SGBD

https://www.youtube.com/watch?v=9TouzGG4p_Y
A Figura 1.3 ilustra a estrutura (com alguma simplificação) de um SGBD típico com base no modelo de dados relacional. O SGBD aceita comandos SQL gerados de uma variedade de interfaces de usuário, produz planos de avaliação de consulta, executa estes planos no banco de dados, e retorna as respostas. (Esta é a simplificação: os comandos SQL podem ser embutidos em programas de aplicativo em linguagem hospedeira, como, por exemplo, programas Python, Java etc..

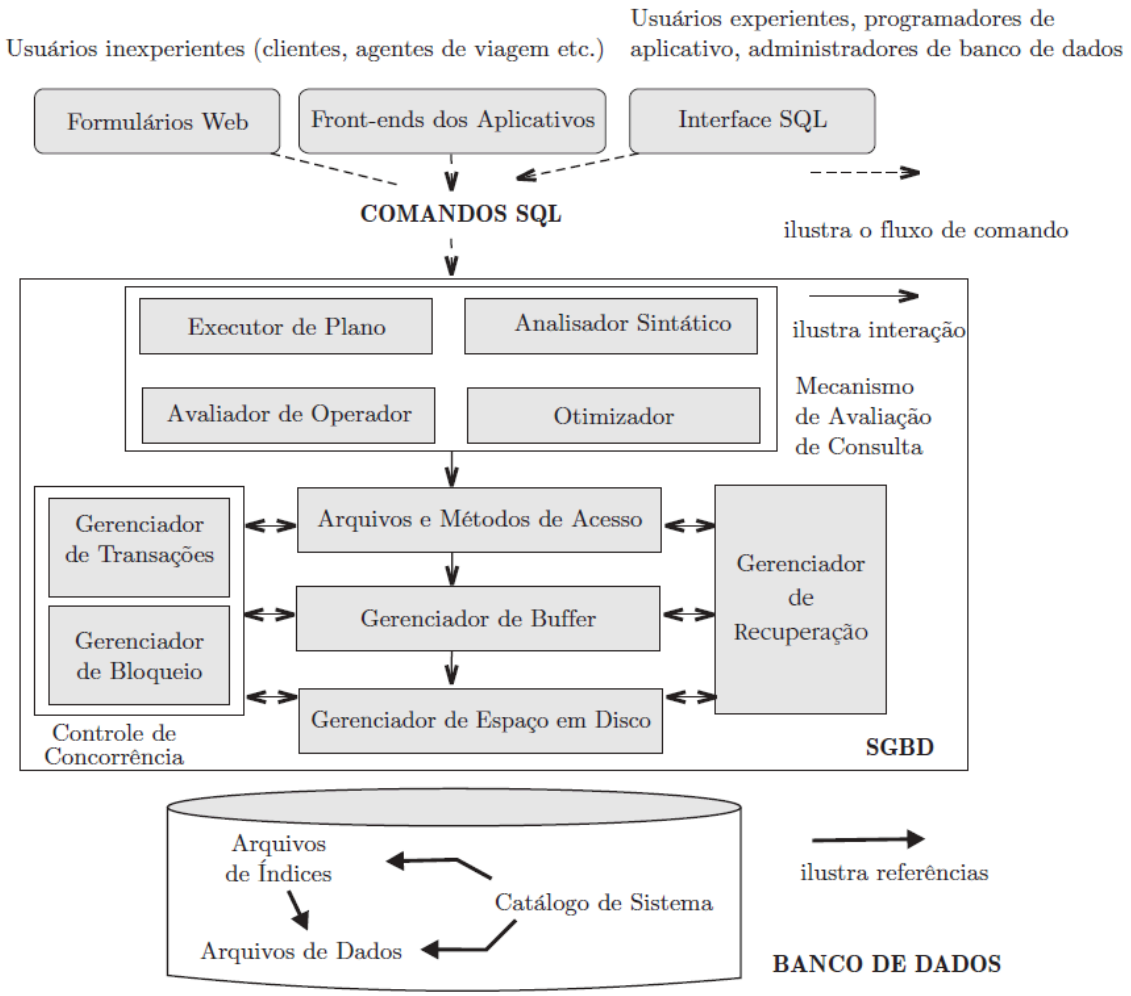


Figura 1.3 Arquitetura de um SGBD.

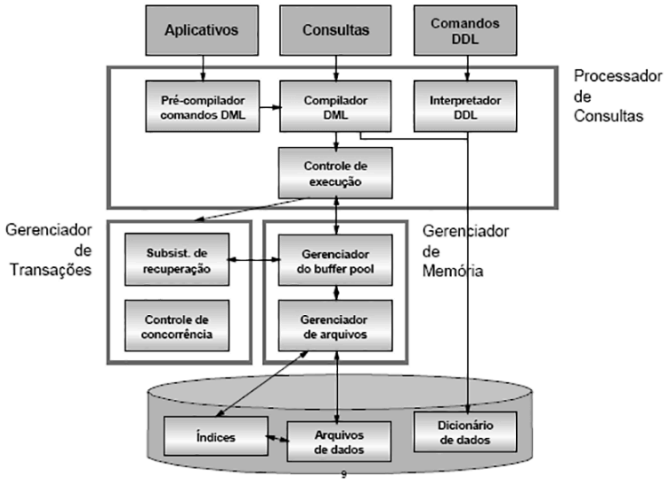
Quando um usuário emite uma consulta, a consulta sintaticamente analisada é apresentada a um **otimizador de consulta**, que usa a informação sobre como o dado é armazenado para produzir um plano de execução eficiente para processar a consulta. Um **plano de execução** é um projeto para processar uma consulta, normalmente representado como uma árvore de operadores relacionais (com anotações que contêm informações detalhadas adicionais sobre quais métodos de acesso usar etc.). Os operadores relacionais servem como blocos de construção para processar consultas elaboradas sobre os dados.

O código que implementa os operadores relacionais situa-se acima da camada de **arquivos e métodos de acesso**. Essa camada suporta o conceito de um **arquivo**, que, em um SGBD, é uma coleção de páginas ou uma coleção de registros. Os **arquivos heap**, ou arquivos de páginas não ordenadas, assim como os índices, são suportados. Além de controlar as páginas de um arquivo, essa camada organiza as informações dentro de uma página. O código da camada de arquivos e métodos de acesso situa-se acima do **gerenciador de buffer**, que carrega as páginas do disco para a memória principal conforme necessário em resposta às solicitações de leitura.

Os componentes de SGBD associados com o **controle de concorrência e recuperação** incluem o **gerenciador de transações**, que assegura que as transações solicitem e liberem bloqueios de acordo com um protocolo adequado de bloqueio e planeja a execução das transações; o **gerenciador de bloqueio**, que controla as requisições por bloqueio e concede o direito de bloqueio nos objetos de banco

de dados quando eles se tornam disponíveis; e o **gerenciador de recuperação**, que é responsável por manter um log e restaurar o sistema a um estado consistente após a ocorrência de uma falha. O **gerenciador de espaço em disco**, **gerenciador de buffer** e as **camadas de arquivo e métodos de acesso** devem interagir com esses componentes.

SGBDs Centralizados



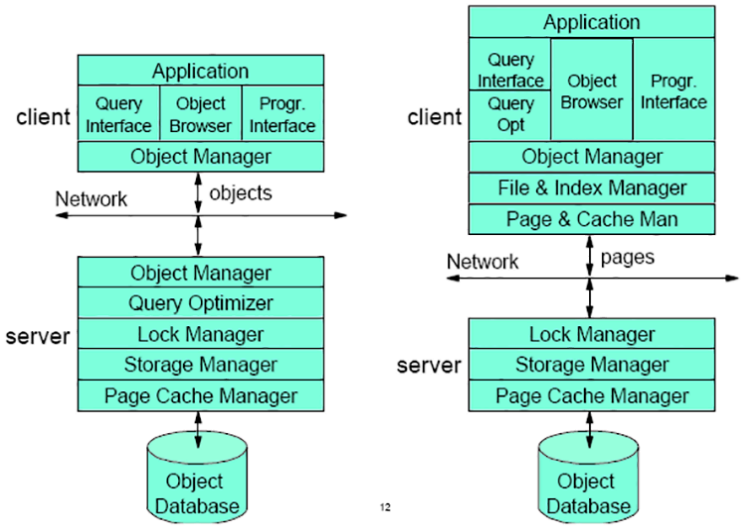
Na arquitetura centralizada, existe um computador com grande capacidade de processamento, o qual é o hospedeiro do SGBD e emuladores para os vários aplicativos. Esta arquitetura tem como principal vantagem a de permitir que muitos usuários manipulem grande volume de dados. Sua principal desvantagem está no seu alto custo, pois exige ambiente especial para mainframes e soluções centralizadas.

Os componentes de um SGBD são a estrutura lógica básica para que o sistema consiga realizar a integração com um tipo específico de banco de dados.

- **Pré-compilador DML:** É o primeiro passo de um SGBD. No pré-compilador, o SGBD começa a identificar a sintaxe do DML (Linguagem de Manipulação de Dados) e realiza as chamadas proceduralmente.
- **Compilador DML:** Aqui o compilador realiza a tradução do DML para a linguagem de baixo nível e as operações são feitas.
- **Interpretador DDL:** O interpretador de DDL (Linguagem de Definição de Dados) funciona para gerenciar as tabelas e seus dados.
- **Gerenciador de autorização:** Como o próprio nome já diz, é o gerenciador que verifica se a pessoa usuária atual tem permissão para realizar tal tarefa.
- **Gerenciador de integridade:** Verifica a integridade do banco de dados quando uma operação é realizada.
- **Gerenciador de transação:** Controla tanto o acesso simultâneo de várias pessoas usuárias, como também a integridade antes e depois de uma operação no banco de dados.
- **Gerenciador de arquivos:** Gerencia o espaço de disco do banco de dados e a sua estrutura de arquivos.
- **Gerenciador de buffer:** É o responsável pela memória em cache e também coordena a transferência de dados do banco de dados principal com o secundário.

SGBDs Cliente-Servidor

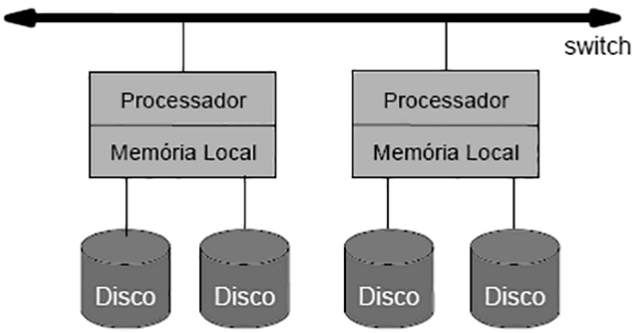
Na arquitetura Cliente-Servidor, o cliente (front_end) executa as tarefas do aplicativo, ou seja, fornece a interface do usuário (tela, e processamento de entrada e saída). O servidor (back_end) executa as consultas no DBMS e retorna os resultados ao cliente. Apesar de ser uma arquitetura bastante popular, são necessárias soluções sofisticadas de software que possibilitem: o tratamento de transações, as confirmações de transações (commits), desfazer transações (rollbacks), linguagens de consultas (stored procedures) e gatilhos (triggers). A principal vantagem desta arquitetura é a divisão do processamento entre dois sistemas, o que reduz o tráfego de dados na rede.



SGBDs Paralelos

Combinam técnicas de gerência de dados e processamento paralelo para aumentar desempenho e confiabilidade:

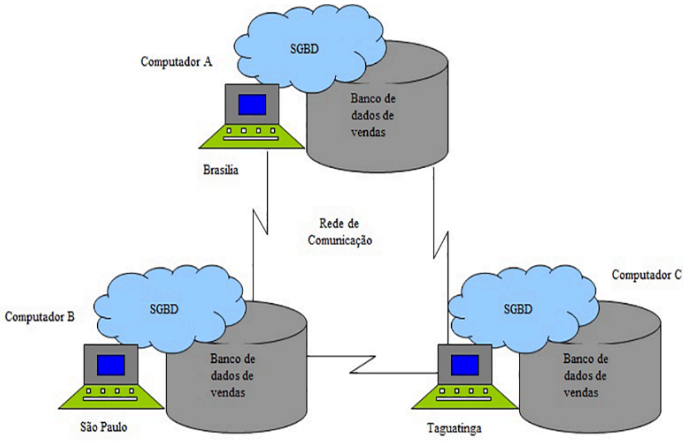
Particionamento do BD em discos controlados por multiprocessadores resulta em aumento da taxa de transferência de dados da memória secundária para memória principal (I/O bandwidth) paralelização do processamento interno de consultas resulta em diminuição do tempo de resposta paralelização do processamento de transações resulta em aumento da capacidade do sistema (throughput)



Banco de Dados Distribuidos

Banco de dados distribuído (BDD) é uma coleção de vários bancos de dados logicamente inter-relacionados, distribuídos por uma rede de computadores. Existem dois tipos de banco de dados

distribuídos, os homogêneos e os heterogêneos. Os homogêneos são compostos pelos mesmos bancos de dados, já os Heterogêneos são aqueles que são compostos por mais de um tipo de banco de dados.



Vantagens de bancos de dados distribuídos

- **Disponibilidade:** Em uma arquitetura distribuída, uma falha em um sítio não torna todo o banco de dados inoperável; apenas algumas partes ficam inacessíveis, enquanto outras permanecem acessíveis.
- **Autonomia Local:** Cada departamento pode controlar seus próprios dados, pois está mais familiarizado com eles.
- **Melhor Performance:** Dados localizados próximos à maior demanda melhoram a performance. Bancos de dados distribuídos permitem balanceamento de carga entre servidores, de modo que uma elevada carga em um módulo não afeta os demais.
- **Econômico:** É mais barato criar uma rede de pequenos computadores com o mesmo poder computacional que um único computador maior.
- **Modularidade:** Sistemas podem ser modificados, adicionados ou removidos do banco de dados distribuído sem afetar os outros módulos.

Desvantagens de banco de dados distribuídos

- **Complexidade:** Trabalho extra para DBAs garantir transparência na distribuição, manter múltiplos sistemas e adaptar o design do banco de dados. Além das dificuldades normais, é necessário considerar fragmentação, alocação e replicação dos dados.
- **Implantação Mais Cara:** A complexidade e infraestrutura extensiva aumentam os custos de trabalho.
- **Segurança:** Fragmentos remotos devem ser seguros, assim como a infraestrutura, exigindo encriptação dos links de rede.
- **Dificuldade em Manter a Integridade:** Reforçar a integridade em uma rede pode sobrecarregar os recursos da rede.
- **Inexperiência:** Trabalhar com bancos de dados distribuídos é difícil e requer especialistas.
- **Falta de Padrões:** Ausência de metodologias e ferramentas para converter um SGBD centralizado em distribuído.

Consultas em um SGBD

Os sistemas de banco de dados relacionais permitem que uma rica variedade de perguntas, **consultas**, sejam formuladas facilmente. Um recurso muito atrativo do modelo relacional é o suporte a linguagens de

consulta poderosas. O **cálculo relacional** é uma linguagem de consulta formal baseada na lógica matemática, e as consultas nesta linguagem têm um significado intuitivo e preciso. A **álgebra relacional** é outra linguagem de consulta formal, baseada em uma coleção de operadores para manipular relações, que é equivalente em poder ao cálculo.

Um SGBD preocupa-se em processar as consultas de forma tão eficiente quanto possível. Obviamente, a eficiência da avaliação da consulta é determinada em grande parte por como os dados são armazenados fisicamente. Os **índices** podem ser usados para acelerar muitas consultas — de fato, uma boa escolha de índices para as relações correspondentes pode acelerar cada consulta da lista anterior.

Um SGBD possibilita aos usuários **criar, modificar e consultar** dados através de uma **linguagem de manipulação de dados** (**DML** — data manipulation language), SQL por exemplo. Assim, a linguagem de consulta é apenas uma parte da DML, que também fornece construtores para **inserir, excluir e modificar** dados.

Transações

Quando vários usuários acessam (e possivelmente modificam) um banco de dados de forma concorrente, o SGBD deve comandar cuidadosamente suas solicitações para evitar conflitos. Além disso, o SGBD deve proteger os usuários dos efeitos de falhas do sistema assegurando que todos os dados (e o status dos aplicativos ativos) sejam restaurados para um estado consistente quando o sistema for reiniciado após uma falha. Por exemplo, se um agente de viagens solicitar uma reserva, e o SGBD responder que a reserva foi realizada, a reserva não deve ser perdida se o sistema falhar. Por outro lado, se o SGBD ainda não respondeu à solicitação, mas está fazendo as alterações necessárias nos dados quando a falha ocorrer, as alterações parciais devem ser desfeitas quando o sistema voltar a funcionar.

Uma **transação** é uma execução de um programa de usuário em um SGBD. (A execução de um mesmo programa diversas vezes gerará diversas transações.) Esta é a unidade básica de alteração reconhecida pelo SGBD: transações parciais não são permitidas, e o efeito de um grupo de transações é equivalente a uma execução serial de todas as transações.

Execução Concorrente de Transações

Uma importante tarefa de um SGBD é planejar os acessos concorrentes aos dados de forma que cada usuário possa seguramente ignorar o fato de que há outros usuários acessando os dados concorrentemente. Por exemplo, se um programa que deposita um valor em uma conta é submetido ao SGBD ao mesmo tempo em que um outro programa debita dinheiro da mesma conta, qualquer um desses programas poderia ser executado primeiro pelo SGBD, mas seus passos não podem ser intercalados de maneira que eles interfiram entre si.

Um **protocolo de bloqueio** é um conjunto de regras que devem ser seguidas por cada transação (e forçadas pelo SGBD) para assegurar que, mesmo que ações de diversas transações possam ser intercaladas, o efeito geral seja idêntico à execução de todas as transações em alguma ordem serial. Um **bloqueio** é um mecanismo utilizado para controlar o acesso aos objetos do banco de dados. Dois tipos de bloqueios são normalmente suportados por um SGBD: **bloqueios compartilhados** em um objeto podem ser mantidos por duas transações diferentes ao mesmo tempo, mas um **bloqueio exclusivo** em um objeto assegura que nenhuma outra transação mantenha nenhum bloqueio nesse objeto.

Considere duas transações T1 e T2 tais que T1 deseja modificar um objeto de dado e T2 deseja ler o mesmo objeto. Intuitivamente, se a solicitação de T1 por um bloqueio exclusivo no objeto for atendida primeiro, T2 não pode ter sua execução continuada até que T1 libere esse bloqueio, pois a solicitação de T2 por um bloqueio compartilhado não será atendida pelo SGBD até então. Assim, todas as ações de T1 serão completadas antes que qualquer uma das ações de T2 sejam iniciadas.

Transações Incompletas e Falhas de Sistema

As transações podem ser interrompidas antes de executadas por completo por uma variedade de razões, por exemplo, uma falha de sistema. Um SGBD deve assegurar que as alterações realizadas por tais transações incompletas sejam removidas do banco de dados.

Para fazer isso, o SGBD mantém um log de todas as escritas no banco de dados. Uma propriedade crucial do log é a de que cada ação de escrita deve ser registrada no log (em disco) antes que a alteração correspondente seja refletida no banco de dados propriamente dito — caso contrário, se o sistema falhar exatamente após a alteração ser feita no banco de dados, mas antes da alteração ser registrada no log, o SGBD será incapaz de detectar e desfazer essa alteração.

Essa propriedade é chamada **Write-Ahead Log** (Gravação Antecipada do Log) ou **WAL**. Para assegurar essa propriedade, o SGBD deve ser capaz de forçar seletivamente a escrita de uma página da memória no disco. O log também é utilizado para assegurar que as alterações realizadas por uma transação completada com sucesso não sejam perdidas por causa de uma falha no sistema. O tempo necessário para a recuperação de uma falha pode ser reduzido forçando periodicamente o registro de alguma informação no disco; esta operação periódica é chamada ponto de verificação (checkpoint).

Referências:

- Fundamentos de Sistemas de Gerenciamento de Banco de Dados - Elmasri & Navathe (7ª edição)
- Sistemas de Gerenciamento Sistemas de Gerenciamento de Banco de Dados - Ramakrishnan Gehrke (3ª edição)
- https://github.com/brunocampos01/banco-de-dados/tree/master
- https://www.youtube.com/playlist?list=PLPCQv3jj4oas-E5LCctT4SLGlMhISIgz8
- http://www.anisio.eti.br/banco-de-dados-menuvertical/item/44-arquitetura-de-banco-de-dados