

Introdução Banco de Dados I

Prof. Alex Luciano Roesler Rese, MSc.

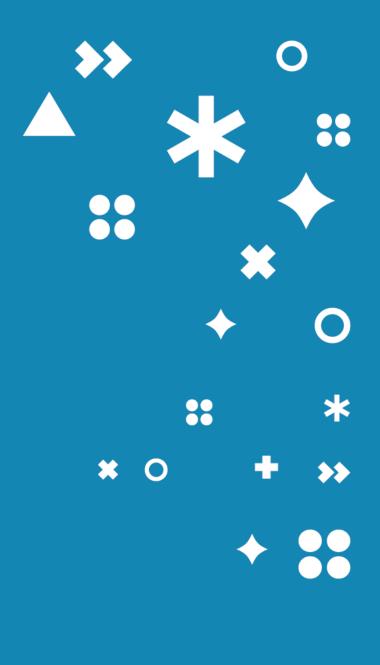
Adaptado: Prof. Lucas Debatin, MSc.



Conteúdo

- Sistemas Gerenciadores de Banco de Dados (SGBD)
- Modelo de Dados
- Trabalho M1

Sistemas
Gerenciadores de
Banco de Dados
(SGBD)

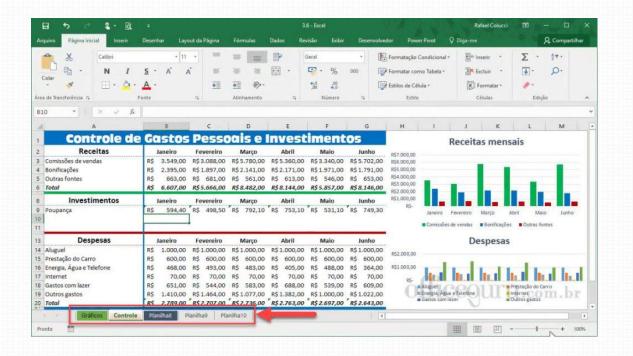


Conceito



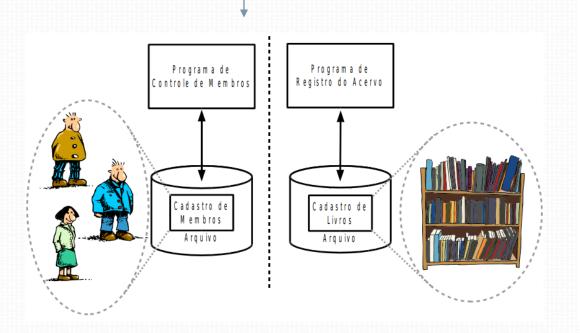
Motivação

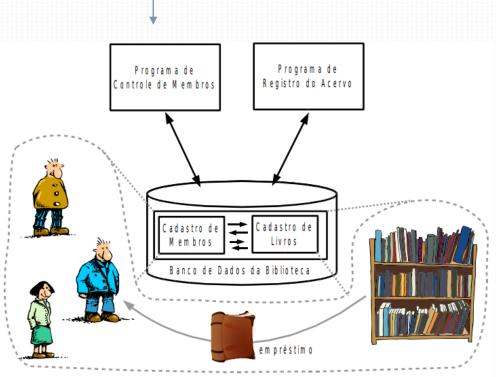
- Sistema de Arquivos X Sistema de Banco de Dados
- Antes dos SGBDs as aplicações utilizavam sistemas de arquivos do SO.





Sistemas Arquivos X SGDB



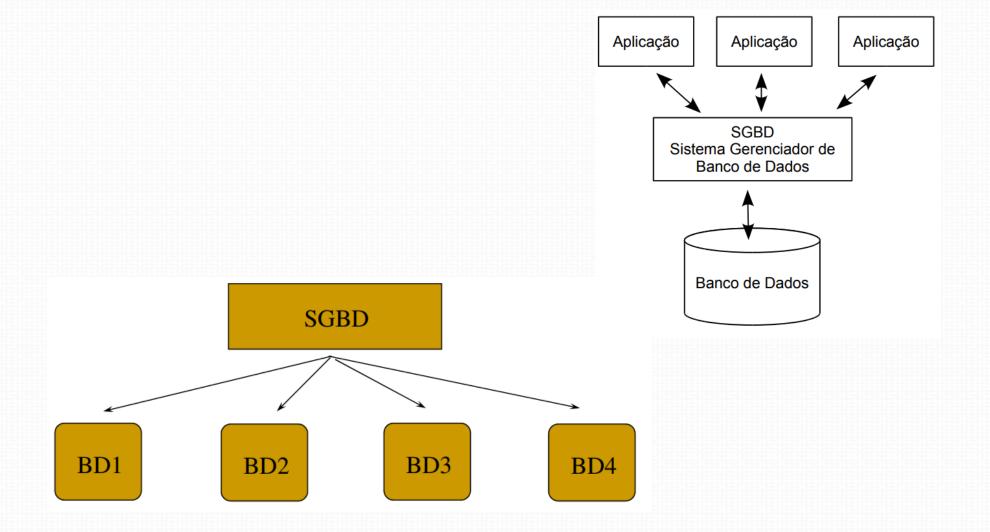




Conceito de SBD

 Um sistema de banco de dados (SBD) é composto por um programa de software chamado sistema gerenciador de banco de dados (SGBD) e por um conjunto de dados, chamado banco de dados (BD).

Conceito de SBD





Conceito de SGBD

 Diversas aplicações geram e lidam com uma grande quantidade de dados.

• É importante garantir a integridade dos dados e disponibilizá-los da forma mais eficiente possível.

 SGBDs são responsáveis pelo armazenamento, gerenciamento e disponibilização dos dados.



Conceito de SGBD

 "Um sistema de gerenciamento de banco de dados, ou SGBD, é um software projetado para auxiliar a manutenção e utilização de vastos conjuntos de dados".

Ramakrishnan E Gehrke (2011)

Conceito de SGBD

- O SGBD provê:
 - Armazenamento e acesso a dados de forma eficiente, confiável, conveniente, segura e multiusuário.

- SGBDs desempenham muitas tarefas importantes que sobrecarregariam os programadores.
 - Veremos ao longo do curso o quanto eles facilitam nosso trabalho.

Objetivo

- Abstração de dados:
 - Simplifica a interação do usuário com o sistema.

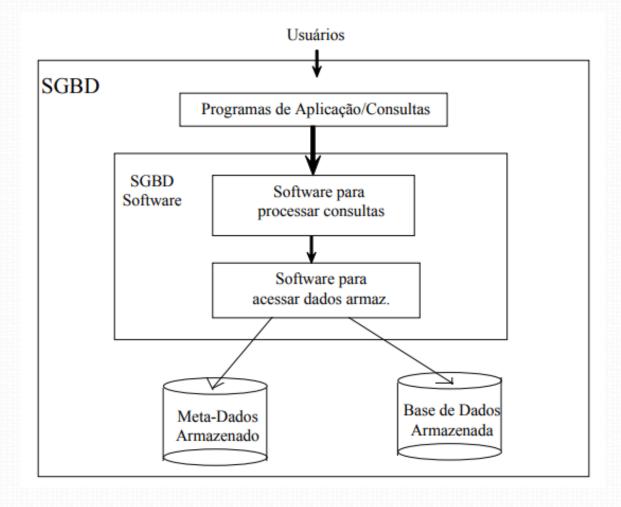
 Gerenciar o acesso e a correta manutenção dos dados armazenados em um banco de dados.

Objetivo

- Simplificar o desenvolvimento de aplicações caracterizadas por uso intensivo de dados.
 - Provendo serviços que diminuem o tempo de desenvolvimento.
 - Por meio de ferramentas o usuário pode:
 - Realizar entrada de dados;
 - Examinar dados;
 - Manipular dados de acordo com a aplicação.

6

Funcionamento



Tipos de Usúarios

 Projetista: Faz análise e cria o esquema (define os tipos ou classes);

 Desenvolvedor de aplicações: programador que usa o banco de dados (faz consultas, modifica dados);

 Administrador (DBA): Gerencia o carregamento de dados, segurança, backups, eficiência, entre outros.



Conceitos Importantes

- Modelo de Dados:
 - Representação dos dados (listas, tabelas, árvores, grafos, entre outros).
 - Relacional!
- Projeto de Banco de Dados:
 - · Conceitual, lógico e físico.
- Linguagem SQL:
 - Por exemplo, requisitar dados para o banco de forma conveniente:
 - "obter o salário de todos os funcionários que têm mais de 30 anos".



Top 10















Softwares Relacionais

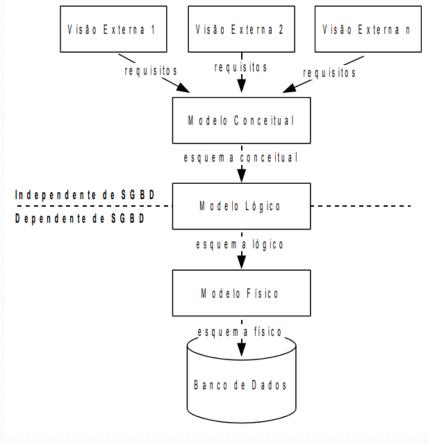
https://db-engines.com/en/ranking

	Rank				Score		
Mar 2023	Feb 2023	Mar 2022	DBMS	Database Model	Mar 2023	Feb 2023	Mar 2022
1.	1.	1.	Oracle 🚹	Relational, Multi-model 👔	1261.29	+13.77	+9.97
2.	2.	2.	MySQL #	Relational, Multi-model 👔	1182.79	-12.66	-15.45
3.	3.	3.	Microsoft SQL Server ☐	Relational, Multi-model 👔	922.01	-7.08	-11.77
4.	4.	4.	PostgreSQL [+	Relational, Multi-model 👔	613.83	-2.67	-3.10
5.	5.	5.	MongoDB 😷	Document, Multi-model 👔	458.78	+6.02	-26.88
6.	6.	6.	Redis 😷	Key-value, Multi-model 👔	172.45	-1.39	-4.31
7.	7.	7.	IBM Db2	Relational, Multi-model 👔	142.92	-0.04	-19.22
8.	8.	8.	Elasticsearch	Search engine, Multi-model 👔	139.07	+0.47	-20.88
9.	9.	1 0.	SQLite [1]	Relational	133.82	+1.15	+1.64
10.	10.	4 9.	Microsoft Access	Relational	132.06	+1.03	-3.37



Projeto de Dados

- Coleção de ferramentas para descrição de dados, seus relacionamentos, sua semântica e restrições de consistência:
 - Conceitual (DER);
 - Lógico;
 - Físico (MER e SQL).





Linguagem SQL

- Linguagem de definição de dados (DDL);
- Linguagem de manipulação de dados (DML);
- Linguagem de consulta de dados (DQL);
- Linguagem de controle de dados (DCL);
- Linguagem de transação de dados (DTL).
- Resumo de SQL em um vídeo:
 - https://www.youtube.com/watch?v=ORRx7WaRhbA

Recursos do SQL

- Adição de novos arquivos;
- Inserção de dados;
- Recuperação de dados;
- Atualização dos dados;
- Eliminação dos dados;
- Criação de visões;
- Atribuição de privilégios;
- Programação;
- Entre outros.

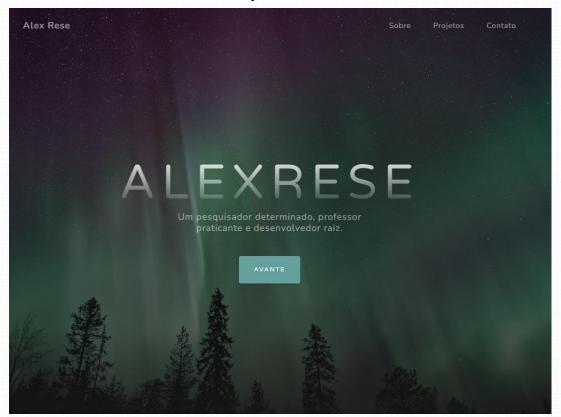
Vantagens do Uso de SGDBs

- Controle de acesso;
- Processamento eficiente de consultas;
- Backup e restauração;
- Recuperação de falhas;
- Relações complexas entre dados;
- Restrições de integridade;
- Administração de dados uniforme;
- Compartilhamento de dados por vários usuários;
- Padronização;
- Entre outras.



Quando não usar SGDBs

- Aplicação extremamente simples, sem previsão de mudança;
- Acesso de múltiplos usuários não é requerido.





Funções Básicas

- Flexibilidade:
 - É fácil mudar a estrutura quando os requisitos mudam.
- Integridade Semântica:
 - Garantia de dados sempre corretos na aplicação.
- Segurança:
 - Evitar violação de consistência dos dados;
 - Acesso e contra falhas.

Funções Básicas

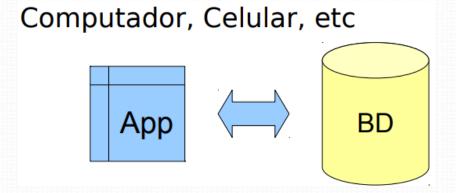
- Concorrência:
 - Evitar conflitos de acesso simultâneo a dados por transações;
 - Disponibilidade de atualizações.
- Independência:
 - Transparência da organização dos dados.

Arquiteturas

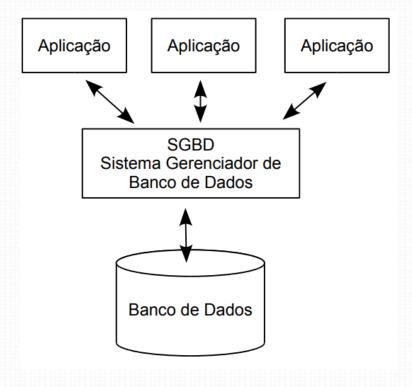
- O SGBD intermedia a manipulação dos dados para as aplicações.
- Como esta intermediação é feita e quais elementos estão envolvidos depende do contexto da aplicação.
- Aplicações podem demandar diversas configurações de arquiteturas locais, distribuídas ou híbridas.

Arquiteturas Local

- Não costumam ser classificados como SGBDs;
- O banco é um software ou biblioteca executado no mesmo dispositivo da aplicação.



Arquiteturas Centralizada

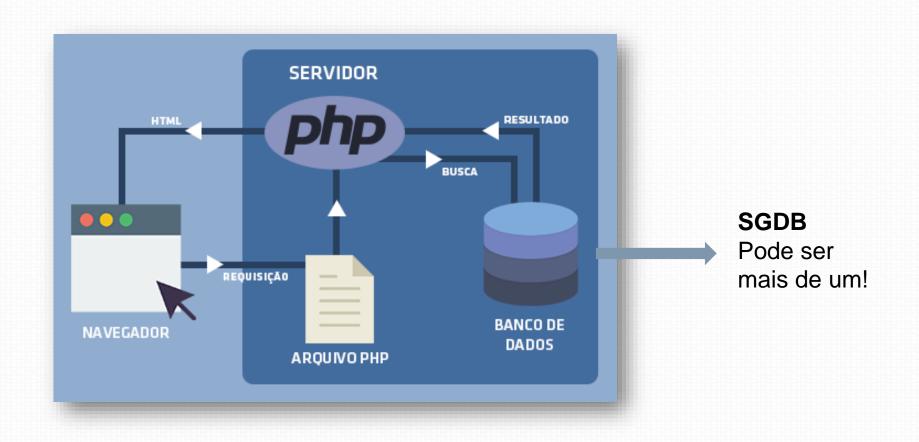




Arquiteturas Distribuída



Arquitetura Cliente-Servidor





Exemplo: Facebook

- Usuário abre o navegador e entra em facebook.com;
- Servidor Web do Facebook recebe a requisição do usuário;
- Servidor Web do Facebook obtém dados do feed de um SGBD interno;
- Servidor Web do Facebook obtém dados de propaganda de um outro SGBD interno;
- Servidor Web do Facebook monta a página e envia para o navegador exibir.

Uma breve história

Sem SGBDs (caos);

• SGBD;

Big data (novo caos).

Sem SGDB (até ~1980)

- Avanços em discos e fitas magnéticas permitem armazenamento de grandes quantidades de dados;
- Falta de um padrão no armazenamento de dados:
 - Cada um faz do seu jeito;
 - Métodos populares: hierárquico, rede.
- Manutenção cara, difícil integração entre sistemas, inconsistência, entre outros.

SGDB (Década de 1980)

- Modelo Relacional:
 - Modelo mais amplamente utilizado por SGBDs:
 - Oracle, MySQL, PostgreSQL, SQLite.
- Separação entre visão lógica e implementação física.
- Linguagem SQL.
- Implementações comerciais eficientes.

SGDB (90 em diante)

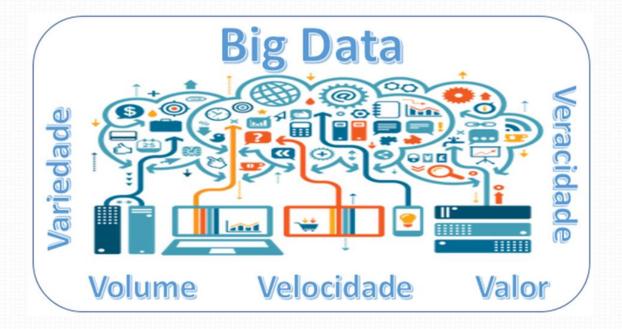
- Modelo relacional continua dominando o mercado:
 - São adequados para mais de 90% das aplicações.

 Mas a Internet, Celulares, Redes Sociais, Sensores, entre outros, causaram um revolução!

Big Data

• Demandam formas inovadoras e rentáveis de processamento da informação, para melhor percepção e tomada de decisão.

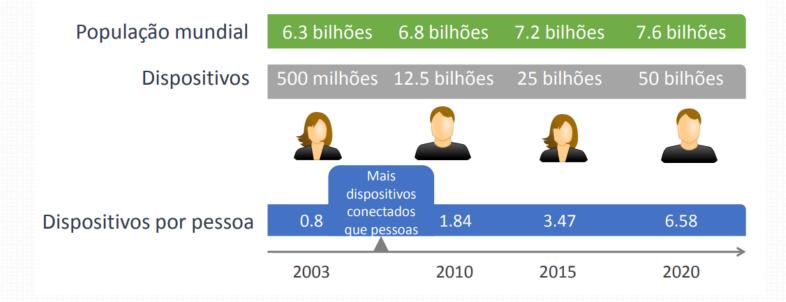
• Problema?





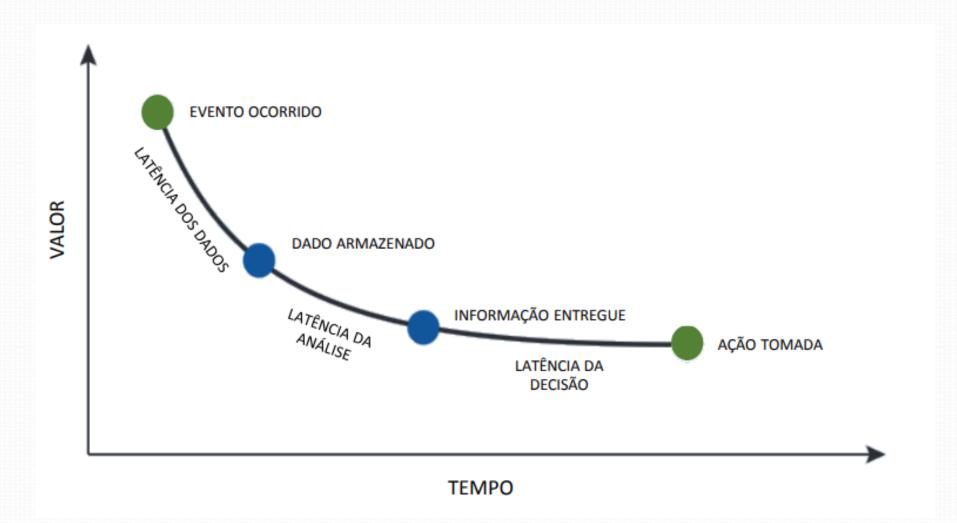
Big Data

• 1 bilhão de usuários conectados no Facebook (23/08/2015).

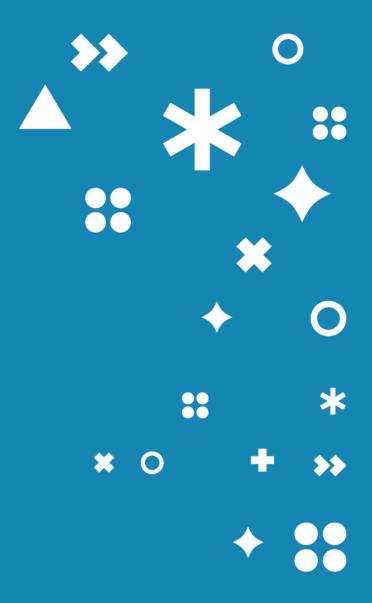




Big Data

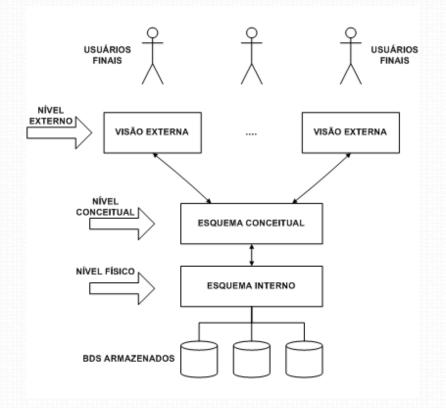






Abstração de Dados

 Supressão de detalhes da organização e armazenamento dos dados, destacando recursos essenciais para um melhor conhecimento desses dados.



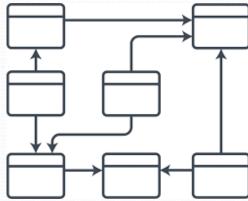




• Oferece os meios necessários para a abstração de dados.

Descrição formal da estrutura de um banco de dados.

- Para construir um banco de dados usa-se uma linguagem de modelagem de dados:
 - Linguagem gráfica e linguagem textual.



 Uma coleção de conceitos que podem ser usados para descrever a estrutura de um banco de dados".

Elmasri e Navathe (2011)



- Estrutura de um banco de dados:
 - Tipos, relacionamentos ou restrições que se aplicam aos dados.
- É o principal critério para a classificação dos softwares.







Tipos

- Hierárquico;
- Rede;
- Orientado a Objetos;
- Relacional;
- Não relacional.

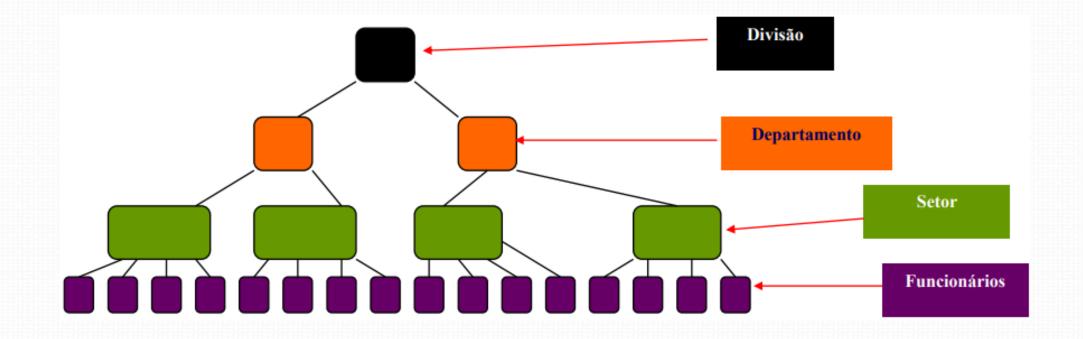


• É um dos mais antigos métodos de organização e armazenamento de dados.

• Implementado em um esforço conjunto da IBM e North American Rockwell em 1965.

Muito utilizado na década de 1970 e no início da década de 1980.

- Permite organizar dados em uma estrutura hierárquica (árvore) com sentido unidirecional:
 - Do pai para o filho, sempre começando pela raiz.

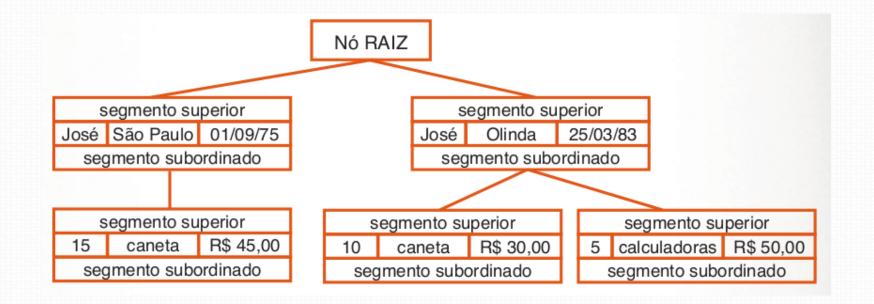




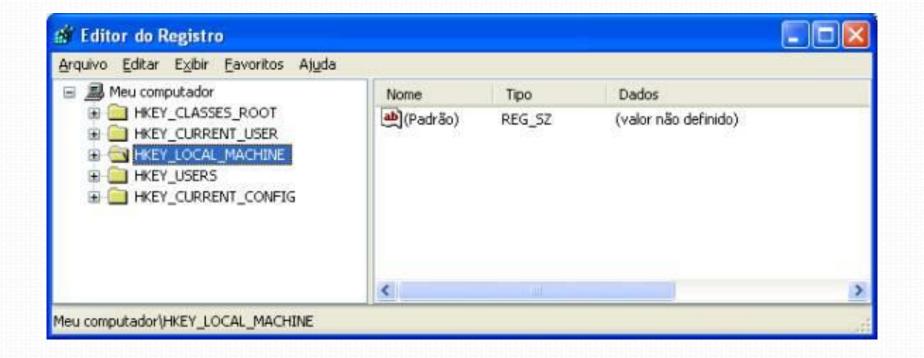
- Vantagem:
 - Os dados podem ser acessados e atualizados rapidamente.
 - Se conhecer todas as informações.
- Desvantagens:
 - A forma de armazenamento influencia o acesso, causando dependência de informações.
 - Somente é possível encontrar um funcionário, informando sua divisão, departamento e setor.
 - Cada filho pode ter apenas um pai.



- Principais Softwares:
 - IMS, ADABAS e o System 2000

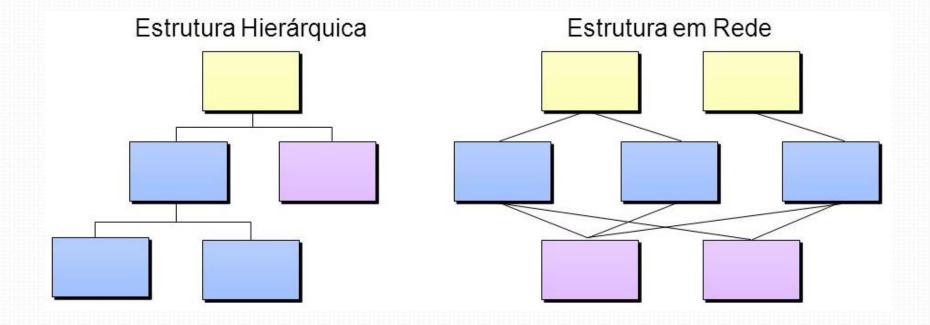


• É utilizado no Registro do Windows.



• Esse modelo foi utilizado principalmente no final da década de 60 e durante a década de 70.

- Também possui uma estrutura hierárquica.
 - Parece com uma teia de aranha ou uma rede interligada de registros

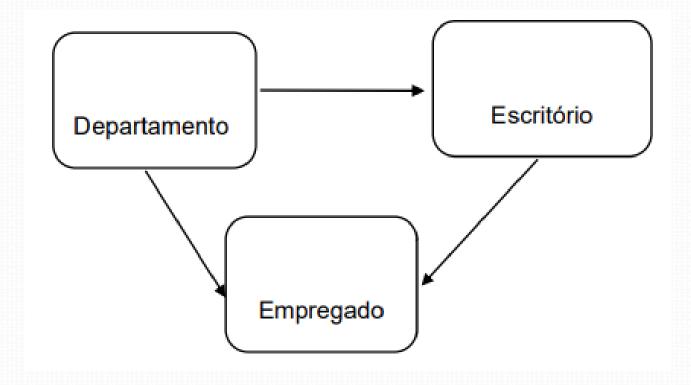


• A diferença mais importante, para o modelo hierárquico, é que cada filho pode ter mais de um pai.

• Além disso, são considerados mais flexíveis, pois as conexões podem ser feitas entre diferentes tipos de dados.

- Permite organizar os dados em uma estrutura formada por várias listas, que definem uma complexa rede de ligações:
 - Similar a um grafo direcionado.

• Deve armazenar as relações entre nós que a rede compõem.

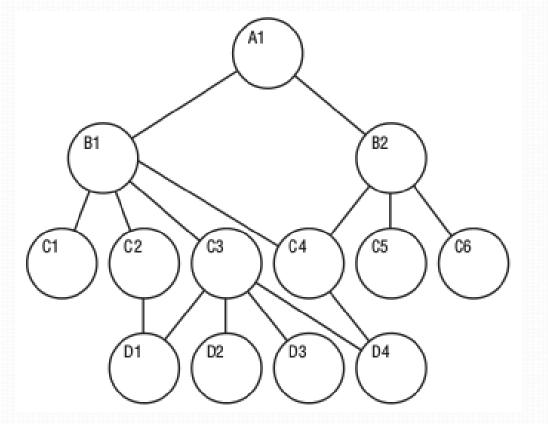




- Vantagem:
 - Representação natural dos relacionamentos muitos para muitos.

- Desvantagem:
 - Se o banco possuir muitos tipos de entidades, pode resultar em esquemas muito complexos de relacionamentos.

- Principais softwares:
 - IDMS e o Total.



• São utilizados em computadores mainframe



Surgiu no início da década de 80.

 O modelo orientado a objeto é baseado no paradigma da programação orientada a objetos.

• As funcionalidades de orientação a objetos são integradas às do

banco de dados.





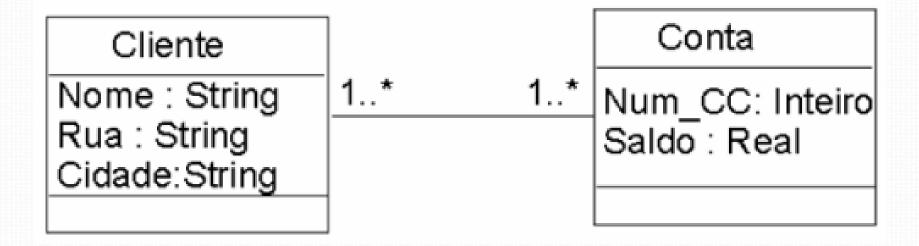


 Define o banco de dados como uma coleção de objetos, ou elementos de software reutilizáveis, com recursos e métodos associados.

 Capaz de lidar com muitos tipos de dados, incluindo gráficos, fotografias, áudio e vídeo.



- Há alguns tipos de bancos de dados orientados para objetos:
 - Um banco de dados multimídia incorpora mídia, como imagens, que não podem ser armazenadas em um banco de dados relacional.
 - Um banco de dados de hipertexto permite que qualquer objeto seja vinculado a qualquer outro objeto.

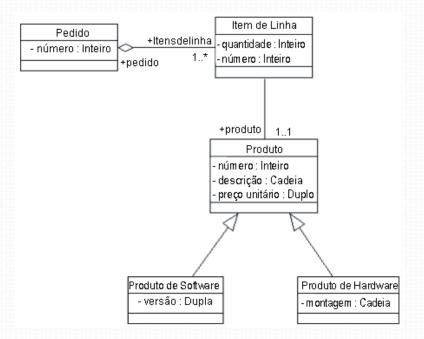


- Vantagem:
 - Capacidade de misturar e combinar objetos reutilizáveis.

- Desvantagens:
 - Possuem um alto custo para desenvolver;
 - A maioria das organizações estão abandonando esse modelo.



- Principais softwares:
 - ONTOS, Jasmine, Caché.





 Podem ser utilizados em organizações de saúde, por exemplo, pois é necessário armazenar, controlar e recuperar tomografias, raios-X, eletrocardiogramas e muitas outras formas de dados cruciais.



 Formalmente definido por Edgard Frank Codd, no laboratório da IBM em 1970.

- Conectam dados em tabelas diferentes, usando um campo chave.
 - "Entidade" e "relação".

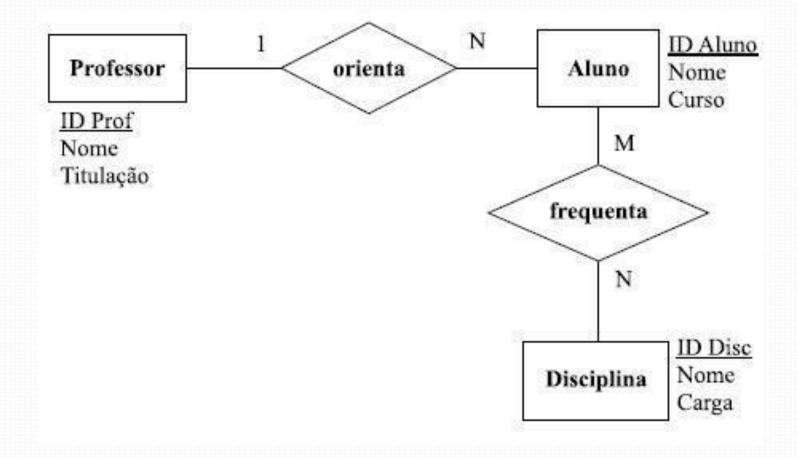
• São mais flexíveis do que os modelos hierárquicos ou rede.



• A ligação entre as tabelas são chamadas de relações.

- As tuplas designam uma linha ou registro;
 - Cada tabela tem um campo chave que identifica unicamente cada linha.

As colunas são referidas como atributos ou campos.



Vantagens:

- O banco de dados pode ser modificado sem precisar redefinir a sua estrutura inteira;
- Possui um padrão tanto no projeto quanto na linguagem de manipulação de dados (SQL);
- Propriedades ACID, integridade referencial e normalização dos dados;
- Oferece maior consistência e confiabilidade, mas exige o relacionamento entre várias tabelas para o acesso à informação.

- Desvantagens:
 - A busca de dados pode levar mais tempo se comparado com outros modelos;
 - Com o aumento da quantidade e do fluxo de informações, o modelo relacional sofre com limitações de escalabilidade:
 - Big Data.









- Principais softwares:
 - MySQL, SQLite, Oracle, PostgreSQL, SQL Server, IBM DB2, entre outros.















- São adequados para mais de 90% das aplicações:
 - Sites,
 - Sistemas em geral,
 - Entre outros.

- Final da década de 90.
- Surgiu como uma maneira de lidar com o crescente volume de dados, que é a principal limitação do modelo relacional.
- O NoSQL atende aos requisitos do ambiente de computação distribuída em larga escala, o que permite escalabilidade, alta disponibilidade, alto desempenho e confiabilidade.

- O termo NoSQL refere-se a armazenamentos de dados que não usam a linguagem SQL e uma estrutura fixa de "tabelas" (linhas e colunas).
- Podem ser armazenados como pares chave/valor simples, documentos JSON ou como um grafo que consiste em bordas e vértices.
 - Permitem uma maior flexibilidade.

Key	Document
1001	{ "CustomerID": 99, "OrderItems": [
1002	{ "CustomerID": 220, "OrderItems": [

CustomerID	Column Family: Identity	CustomerID
001	First name: Mu Bae Last name: Min	001
		002
002	First name: Francisco	
	Last name: Vila Nova Suffix Jr.	003
003	First name: Lena Last name: Adamcyz Title: Dr.	

CustomerID	Column Family: Contact Info
001	Phone number: 555-0100 Email: someone@example.com
002	Email: vilanova@ contoso.com
003	Phone number: 555-0120

Opaque to data store

Key	Value
AAAA	11010011110101001101011111
AABAB	10011000010110011010111110
DFA766	00000000001010101010101010
FABCC4	1110110110101010100101101



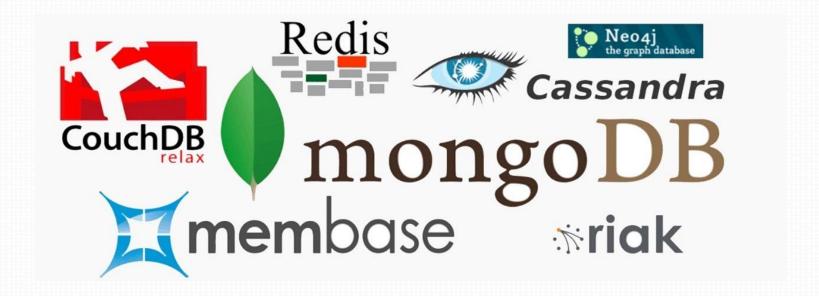
- Vantagens:
 - Não exige que todas as entidades tenham a mesma estrutura;
 - Flexibilidade;
 - Facilidade em adicionar novos dados;
 - Escalabilidade horizontal natural;
 - Excelente maneira de lidar com o problema do Big Data.



- Desvantagens:
 - Tecnologia nova, atualização rápida da tecnologia, comunidade nova, suporte falho, falta de documentação;
 - Ausência das propriedades ACID e integridade referencial;
 - Algumas aplicações necessitam desses controles.
 - Não possui padrões de linguagens e estruturas.



- Principais softwares:
 - Apache Cassandra, Amazon DynamoDB, MongoDB, HBase, Apollo, entre outros.





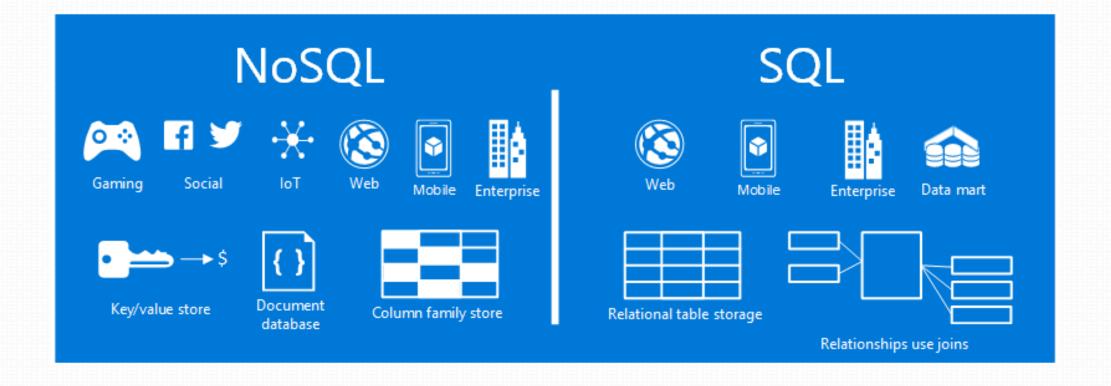
- É utilizado em:
 - Redes sociais;
 - Grandes e-commerces;
 - Entre outros.











New SQL

- O NoSQL foi projetado especialmente para arquiteturas distribuídas, com conceitos da não normalização e a não padronização de uma linguagem.
 - Com a sua utilização desenfreada alguns problemas foram enfrentados, tais como falta do uso de transações, a falta do SQL e a estrutura complexa por falta de uma boa modelagem.
 - Isso deu abertura a nova proposta: o NewSQL.
- Une o melhor do desempenho e escalabilidade do NoSQL, mas sem abrir mão dos benefícios da linguagem SQL, nem das propriedades ACID.

New SQL

Características:

- Linguagem SQL como meio de interação entre o SGBD e a aplicação;
- Suporte para transações ACID;
- Controle de concorrência não bloqueante, para que as leituras e escritas não causem conflitos entre si;
- Arquitetura que forneça um maior desempenho por nó de processamento;
- Arquitetura escalável, com memória distribuída e com capacidade de funcionar em um aglomerado com um grande número de nós.

New SQL

- Softwares:
 - MemSQL;
 - VoltDB;
 - SQLFire;
 - MariaDB.

