



Aula 01

Sistemas de Informação e Bancos de dados

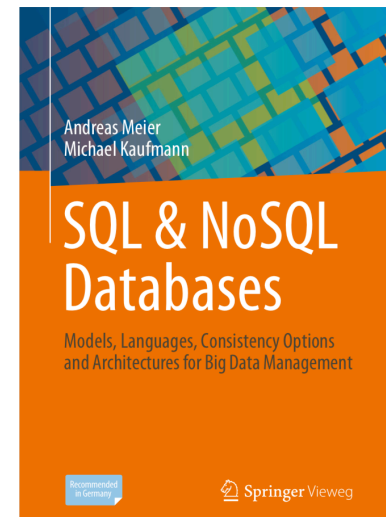
Bibliografia

NoSQL Databases
Christof Strauch
(notas de aula)

Profa A. Montini
(notas de aula)

&

Gestão de dados
Abordagens Relacionais
Big Data
Tecnologias NoSQL





Motivação

- ❖ Sociedade do Conhecimento - informação como um ativo produtivo
- ❖ Vivemos na Era dos Dados devido ao grande volume de dados gerados pela humanidade e pelos dados que trafegam rapidamente no universo digital.
- ❖ Esse volume de informação que pode ser processada em tempo real está revolucionando a forma de tomada de decisão.





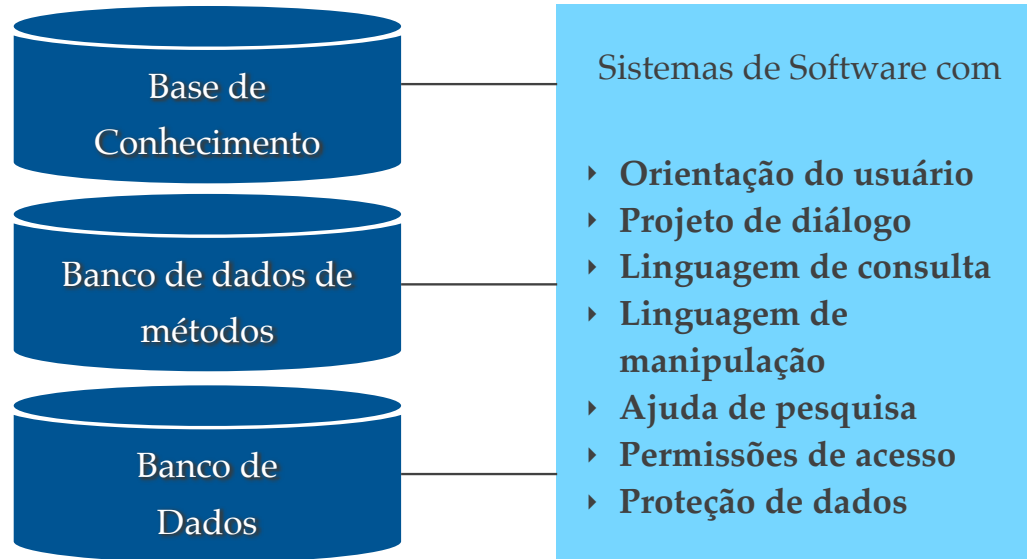
Principais aspectos da Informação

- ❖ **Representação:** é especificada por dados (sinais, signos, mensagens ou elementos de linguagem)
- ❖ **Processamento:** pode ser transmitida, armazenada, categorizada, encontrada, convertida em outra representação
- ❖ **Combinação:** pode ser combinada livremente
- ❖ **Idade:** Não está sujeita à ação do tempo
- ❖ **Originalidade:** pode ser copiada sem limites e não se distingue original e cópia
- ❖ **Imprecisão:** informação não é clara, frequentemente imprecisa e de diferentes validades (níveis de qualidade)
- ❖ **Meio:** não requer um meio fixo, é independente de localização



Dados como a base de informação como um fator produtivo

- ❖ **Base para tomada de decisão:** dados permitem decisões bem informadas, tornando-se vitais para funções organizacionais
- ❖ **Nível de qualidade:** disponíveis a partir de diferentes fontes, a qualidade depende da disponibilidade, correção e integridade
- ❖ **Necessidade de investimentos:** processos de coleta, armazenamento e processamento acarretam esforços e são dispendiosos
- ❖ **Grau de integração:** as funções organizacionais são conectadas por relações informacionais, logo o cumprimento das tarefas depende fortemente da integração de dados

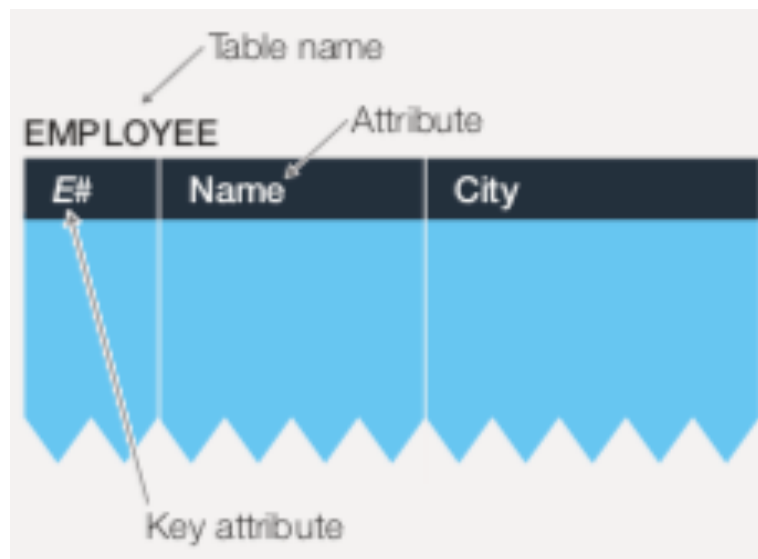




Bancos de Dados SQL

- ❖ Structured Query Language databases
- ❖ Modelo Relacional
- ❖ Simples e intuitivo
- ❖ Excelente para lidar com sistemas de processamento transacionais e sistemas com modelos de dados estáveis





EMPLOYEE		
E#	Name	Ort
E19	Stewart	Stow
E4	Bell	Kent
E1	Murphy	Kent
E7	Howard	Cleveland

Labels in the diagram:

- Column:** Points to the header row.
- Data value:** Points to the value 'Bell' in the row for E4.
- Record (row or tuple):** Points to the entire row for E4.

- ❖ **Chave de identificação:** a chave de uma tabela é um ou uma combinação mínima de atributos que representam unicamente um registro
- ❖ **Singularidade:** cada valor-chave identifica apenas um único registro em uma tabela
- ❖ **Minimalidade:** se uma chave é uma combinação de atributos, esta combinação deve ser a menor possível (se retirar um dos atributos, deixa de ser possível identificar um registro)



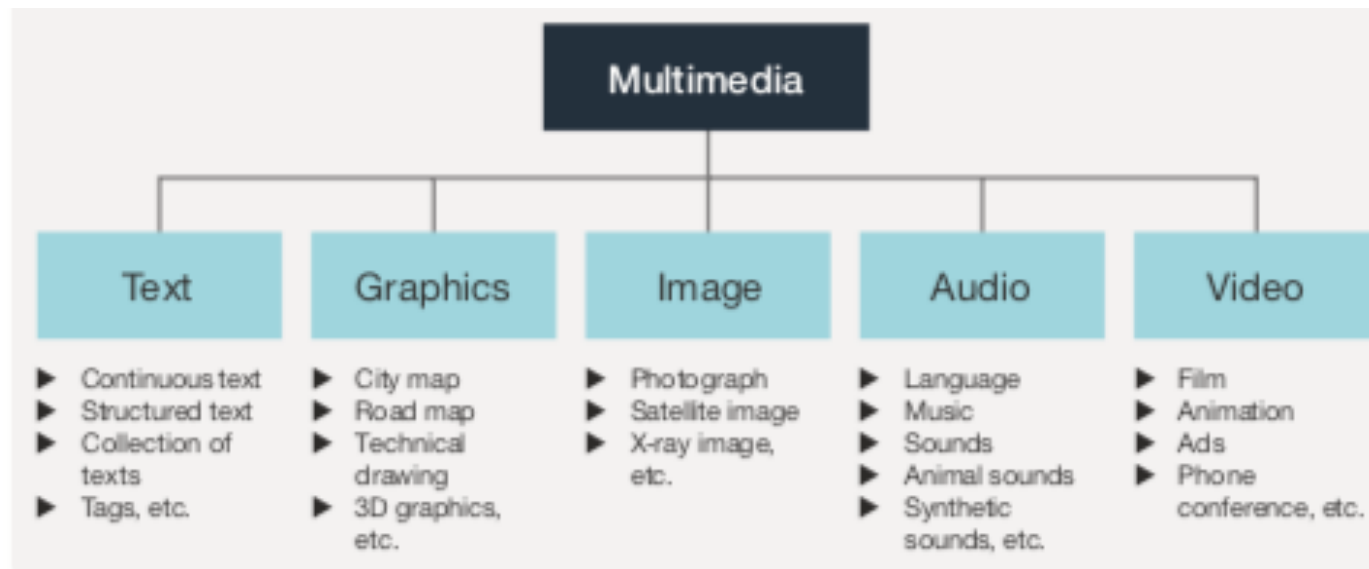
Propriedades de um SGBDR

- ❖ **Modelo:** o banco obedece o paradigma relacional
- ❖ **Esquema:** as definições das tabelas e atributos são armazenados no esquema. Contém definições das identificações de chaves e as regras de integridade definidas.
- ❖ **Linguagem:** SQL por definição.
- ❖ **Arquitetura:** Assegura independência de dados (dados e aplicações são separados) - alterações físicas não impactam, necessariamente, nas aplicações
- ❖ **Operação multi-usuário:** Vários usuários podem manipular as mesmas bases de dados ao mesmo tempo. O SGBDR assegura transações paralelas.
- ❖ **Garantia de consistência:** Provê ferramentas para assegurar a integridade dos dados
- ❖ **Segurança e proteção de dados:** Possui mecanismos de proteção contra destruição, perda ou acesso não autorizado



Big Data

- ❖ Conjunto de metodologias utilizadas para lidar com grandes **volumes** de dados, produzidos em alta **velocidade**, a partir de uma grande **variedade** de fontes. (3 V's)





Big Data

- ❖ Representa um desafio para empresas, governos, autoridades públicas, e ONGs
- ❖ Exemplo de aplicação
 - ❖ **Smart cities:** uso de tecnologias de Big Data nas cidades e aglomerações urbanas para o desenvolvimento sustentável dos aspectos sociais e ecológicos dos espaços humanos.
 - ❖ **Mobilidade**, o uso de **sistemas inteligentes** de abastecimento de **água** e **energia**, a promoção de **redes sociais**, a expansão da **participação política**, o incentivo ao **empreendedorismo**, a proteção do **meio ambiente** e o aumento da **segurança** e da qualidade de vida.
- ❖ Todo o uso de aplicativos de Big Data requer um gerenciamento bem-sucedido dos três Vs



Big Data e os 3 V's

- ❖ **Volume:** existem enormes quantidades de dados envolvidos, variando de tera a zetabytes (megabyte = 10^6 bytes, gigabyte = 10^9 bytes, terabyte = 10^{12} bytes, peta-byte = 10^{15} bytes, exabyte = 10^{18} bytes, zettabyte = 10^{21} bytes).
- ❖ **Variedade:** envolve o armazenamento de dados multimídia estruturados, semiestruturados e não estruturados (texto, gráficos, imagens, áudio e vídeo).
- ❖ **Velocidade:** os aplicativos devem poder processar e analisar fluxos de dados em tempo real, à medida que os dados são coletados.
- ❖ Como na definição do Grupo Gartner, o Big Data pode ser considerado um ativo de informações, e é por isso que às vezes outro V é adicionado:
- ❖ **Valor:** os aplicativos de Big Data destinam-se a aumentar o valor da empresa; portanto, são feitos investimentos em pessoal e infraestrutura técnica onde trarão alavancagem ou valor agregado.
- ❖ **Veracidade:** Como muitos dados são *vagos* ou *imprecisos*, são necessários algoritmos específicos para **avaliar a validade** e **avaliar a qualidade** do resultado. Grandes quantidades de dados não significam automaticamente melhores análises.

Bancos de Dados NoSQL - Not Only SQL

- ❖ 2009 , revista Computerworld: “os defensores do NoSQL passaram a compartilhar como haviam derrubado a tirania de bancos de dados relacionais lentos e caros em favor de maneiras mais eficientes e baratas de gerenciar dados”
- ❖ Objetivos de Bancos NoSQL:
 - ❖ Atender demandas das aplicações, que necessitam de **alta disponibilidade e rápido desempenho** de processamento dos seus dados





Bancos de Dados NoSQL - Not Only SQL

- ❖ Caso Google: Processa 20 petabytes por dia em BigTable.
- ❖ Características:
 - ❖ **Escalabilidade horizontal:** aumentar o número de máquinas disponíveis; requer evitar bloqueios
 - ❖ **Ausência de esquema ou esquema flexível:** facilita alterações nas regras de persistência; não há garantia de integridade dos dados
 - ❖ **Suporte a replicação:** reduz o tempo gasto para a recuperação de informações
 - ❖ **API Simples:** torna o acesso a informação mais eficiente
 - ❖ **Nem sempre é consistente:** Nem sempre consegue se manter consistente



Bancos de Dados NoSQL - Not Only SQL

- ❖ Razões para desenvolver e usar Bancos NoSQL:
 - ❖ **Evitar um complexidade excessiva:** SGBDR mantém regras de consistência de dados (ACID) que podem ser desnecessárias para algumas aplicações
 - ❖ **Alta taxa de transferência:** Alguns bancos de dados NoSQL fornecem uma taxa de transferência de dados significativamente maior que os SGBDRs tradicionais.
 - ❖ Hypertable (column-based), que segue a abordagem Bigtable do Google, permite que o mecanismo de pesquisa local Zvent armazene um bilhão de células de dados por dia [Jud09].
 - ❖ O Google pode processar 20 petabytes por dia armazenados no Bigtable por meio da abordagem MapReduce [Com09b].



Bancos de Dados NoSQL - Not Only SQL

- ❖ Mais razões para desenvolver e usar Bancos NoSQL:
 - ❖ **Escalabilidade horizontal e execução em hardware:** volume de dados cada vez maior
 - ❖ Desafios em dimensionar dados, desempenho de servidores e projetos de esquema rígidos demais
 - ❖ **Evitar o custo de ORM:** a maioria dos bancos NoSQL são projetados para armazenar estruturas de dados simples
 - ❖ **Complexidade e custo da configuração de clusters de BD:** BDs NoSQL são projetados para que os clusters de PC possam ser expandidos de maneira fácil e barata sem a complexidade
 - ❖ **Comprometendo a confiabilidade para obter melhor desempenho:** há cenários em que os aplicativos estariam dispostos a comprometer a confiabilidade para obter melhor desempenho. Exemplo: dados da sessão HTTP
 - ❖ ...



ACID vs BASE

❖ ACID

- ❖ **Atomicidade:** a transação será executada totalmente ou não será executada
- ❖ **Consistência:** garante que o banco de dados passará de uma forma consistente para outra forma consistente
- ❖ **Isolamento:** a transação não sofrerá interferência por nenhuma outra transação concorrente
- ❖ **Durabilidade:** garante que o que foi salvo, não será mais perdido



ACID vs BASE

- ❖ BASE
 - ❖ **Basically Available:** basicamente disponível
 - ❖ **Soft state:** estado leve
 - ❖ **Eventually consistent:** eventualmente consistente
- ❖ Uma aplicação funciona basicamente o tempo todo (basicamente disponível), não tem de ser consistente todo o tempo (estado leve) e o sistema torna-se consistente no momento devido (eventualmente consistente).



ACID vs BASE

ACID	BASE
Consistência forte	Fraca consistência
Isolamento	Foco em Disponibilidade
Concentra-se em "commit"	Melhor esforço
Transações aninhadas	Respostas aproximadas
Disponibilidade	Mais simples e mais rápido
Conservador (pessimista)	Agressivo (otimista)
Evolução difícil (por exemplo, esquema)	Evolução mais fácil

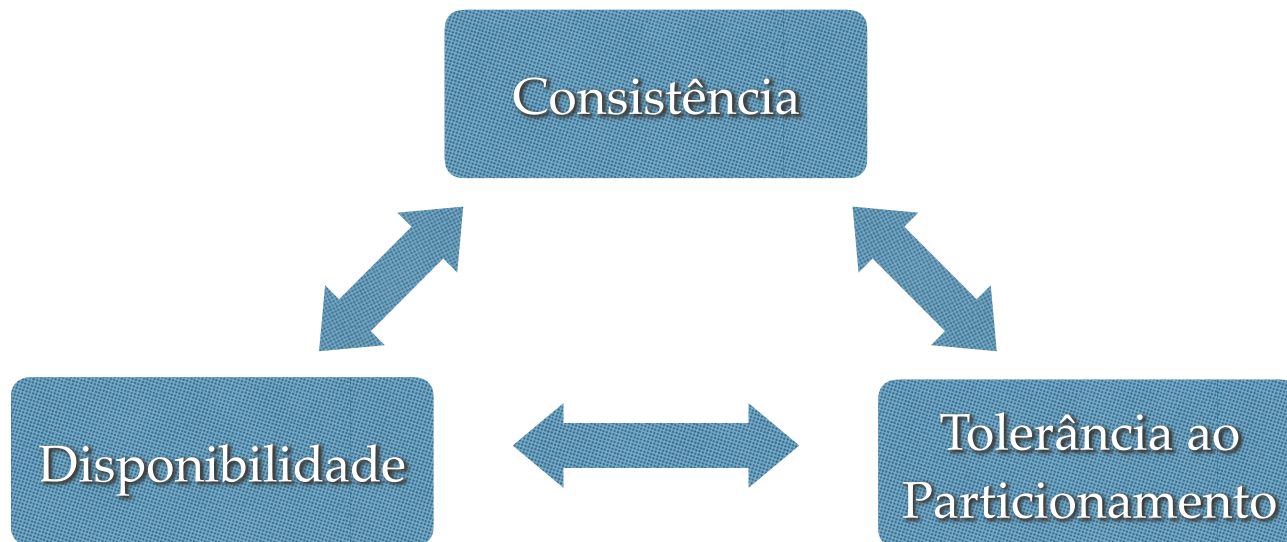


Teorema CAP

- ❖ **Consistência:** exemplo - um sistema é consistente se depois da atualização de um dado, todos os usuários que têm acesso podem acessá-lo (atualizado) em tempo real.
- ❖ **Disponibilidade (availability):** concepção e implementação de um sistema que assegure que ele esteja ativo durante um determinado período
- ❖ **Tolerância ao particionamento:** capacidade de um sistema continuar operando mesmo depois de uma falha na rede

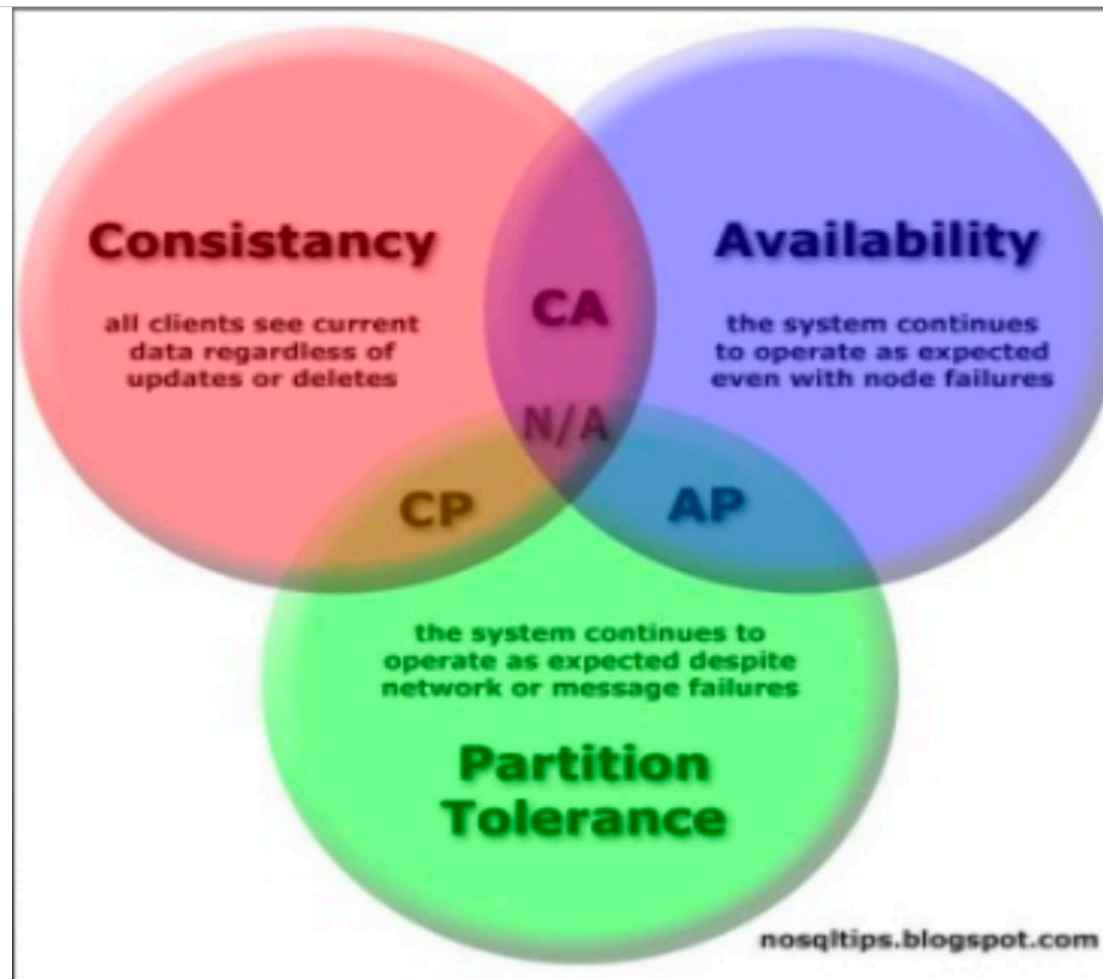


Teorema CAP





Teorema CAP





Sistemas CA

- ❖ **Os sistemas com consistência forte e alta disponibilidade** não sabem lidar com a possível falha de uma partição.
- ❖ **Em caso de falha de partição**, o sistema inteiro pode ficar indisponível até o membro do cluster voltar



Sistemas CP

- ❖ Os sistemas com consistência forte e tolerância de particionamento abrem a mão da disponibilidade
- ❖ MongoDB, HBase, BigTable



Sistemas AP

- ❖ Para sistemas que jamais podem ficar offline
- ❖ Nesses casos, é preciso sacrificar um pouco a **consistência**
- ❖ MongoDB, Cassandra, Voldemort



NoSQL - modelos

- ❖ Bancos de Dados NoSQL suportam diferentes modelos de dados
 - ❖ **Modelo baseado em Chave/Valor**
 - ❖ **Modelo baseado em Colunas**
 - ❖ **Modelo baseado em Documentos**
 - ❖ **Modelo baseado em Grafos**
 - ❖ **Modelo orientado a Objetos**



Banco de dados Key Value

- ❖ Tipo mais simples de BD NoSQL
- ❖ Armazenamento: nome de atributo (chave) => valor
 - ❖ Estes pares chave-valor estão na forma de linhas de matrizes associativas (hash)
- ❖ Persistência de dados em dispositivos secundários ou apenas na memória (bd de cache) - permitem recuperação eficiente de dados
 - ❖ **Armazenamento e recuperação, Armazenamento em cache, Estrutura de dados simples, Escalabilidade**



Banco de dados Key Value

- ❖ **Aplicações:** armazenamento de dados de sessão, perfil de usuário sem necessidade de esquema, preferências de usuário, carrinhos de compras (**apps**)
- ❖ **Não usar quando:** necessitar identificar relações entre os dados, operar múltiplas chaves, requisitos de negócio mudam frequentemente



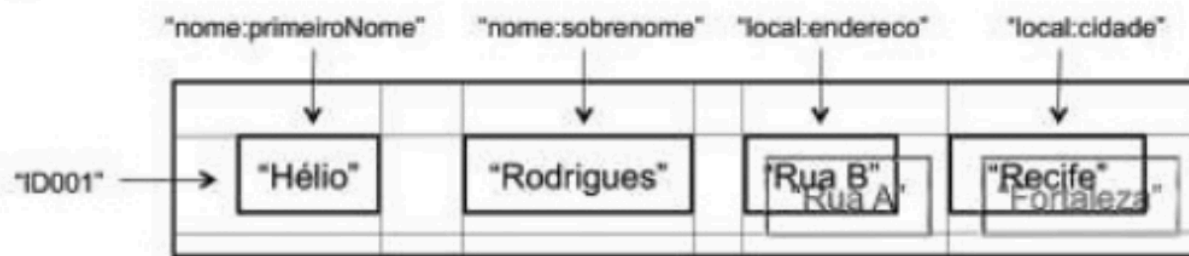
Exemplos de Key-value databases





Column-based databases

- ❖ Agrupa células em colunas de dados, não em linhas
- ❖ Os dados são indexados por triplas: linha, coluna, timestamp
 - ❖ As linhas e as colunas são identificadas por chaves e o timestamp é o que permite identificar as diferentes versões de um mesmo dado
- ❖ Aumenta consideravelmente o desempenho de consultas analíticas
 - ❖ Reduz as premissas de IO e diminui a quantidade de dados carregadas a partir do disco





Column-based databases

- ❖ As principais técnicas aplicadas: particionamento, indexação, compressão e diferentes estratégias para realização de joins.
- ❖ Aplicações: CMS, blogs, contadores, serviços que controlam uso expirado, sistemas para gravações pesadas (ie. logs)

people_id	people_name	people_age
101	Mary	54
102	Jhon	35
103	Paul	22

people_id	people_name	people_age
101	Mary	54
102	Jhon	35
103	Paul	22

Exemplos de Column-based databases





Document-oriented databases

- ❖ Gestão de documentos, ou dados semi-estruturados
- ❖ Uma especialização de key-value databases
- ❖ Para este modelo, um documento é um objeto, cujos valores são dados estruturados ou semi-estruturados (XML, JSON, BSON...)
- ❖ Aplicações: plataformas de e-commerce, CMS, plataformas analíticas, blogs
- ❖ Não usar quando: o trabalho envolver consultas complexas ou múltiplas operações complexas!



Exemplos de Document-oriented databases





Graph-based databases

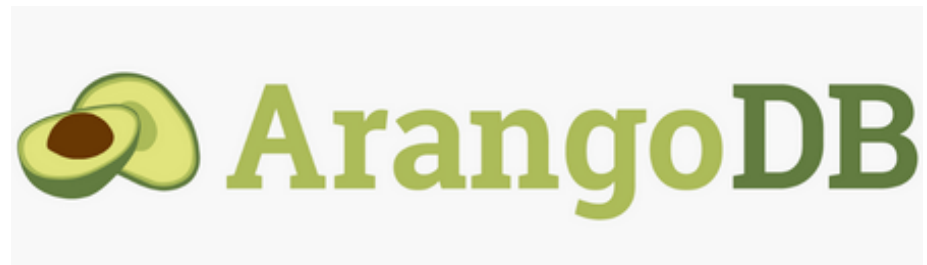
- ❖ Consiste em nós e arestas direcionais (relacionamentos) - possuem rótulos
- ❖ Baseado na teoria de grafos => Entidade (nós) - Atributo - Valor
 - ❖ Nós (vértices do grafo), relacionamentos (arestas), propriedades (atributos do nó)



Graph-based databases

- ❖ Os sistemas de BD podem oferecer funcionalidades para:
 - ❖ **Conectividade:** um grafo é conectado quando todos os nós possuem caminho
 - ❖ **Menor caminho:** é a menor distância entre dois nós
 - ❖ **Vizinho mais próximo:** Em grafos ponderados, pode determinar o vizinho pelo caminho mais eficiente
 - ❖ **Correspondência:** encontrar conjunto de arestas sem nós comuns
- ❖ Aplicações: detecção de fraude, pesquisa baseada em Grafos, redes e operações de TI, redes sociais, etc

Exemplos de Graph-based databases





OODBMS

- ❖ Excelente para integrar a sistemas desenvolvidos com paradigma OO
- ❖ Gera pouco impacto nas transações, pois o código de persistência muito próximo do programa
- ❖ As regras definidas no modelo de classes são tratadas normalmente no banco de dados



Exemplos de OODBMS

