

Banco de dados NoSQL

Termos de Uso



Propriedade Growdev

Todo o conteúdo deste documento é propriedade da Growdev. O mesmo pode ser utilizado livremente para estudo pessoal.

É proibida qualquer utilização desse material que não se enquadre nas condições acima sem o prévio consentimento formal, por escrito, da Growdev. O uso indevido está sujeito às medidas legais cabíveis.

NoSQL



O termo NoSQL refere-se a uma categoria de bancos de dados projetados para lidar com grandes volumes de dados, alta velocidade de leitura e escrita e flexibilidade no armazenamento de diferentes tipos de dados.

O nome "NoSQL" originalmente significava "não apenas SQL", indicando que esses bancos de dados não dependem exclusivamente do modelo relacional tradicional baseado em tabelas e esquemas rígidos.

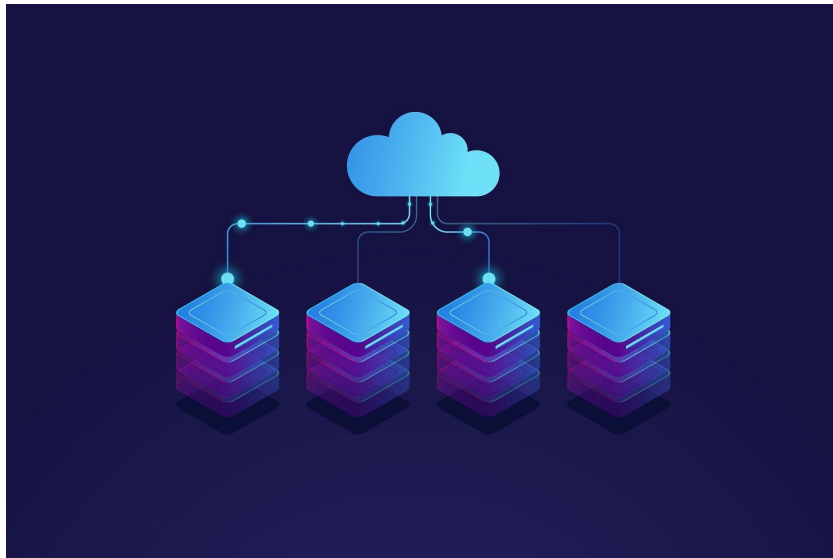


Porque usar NoSQL?

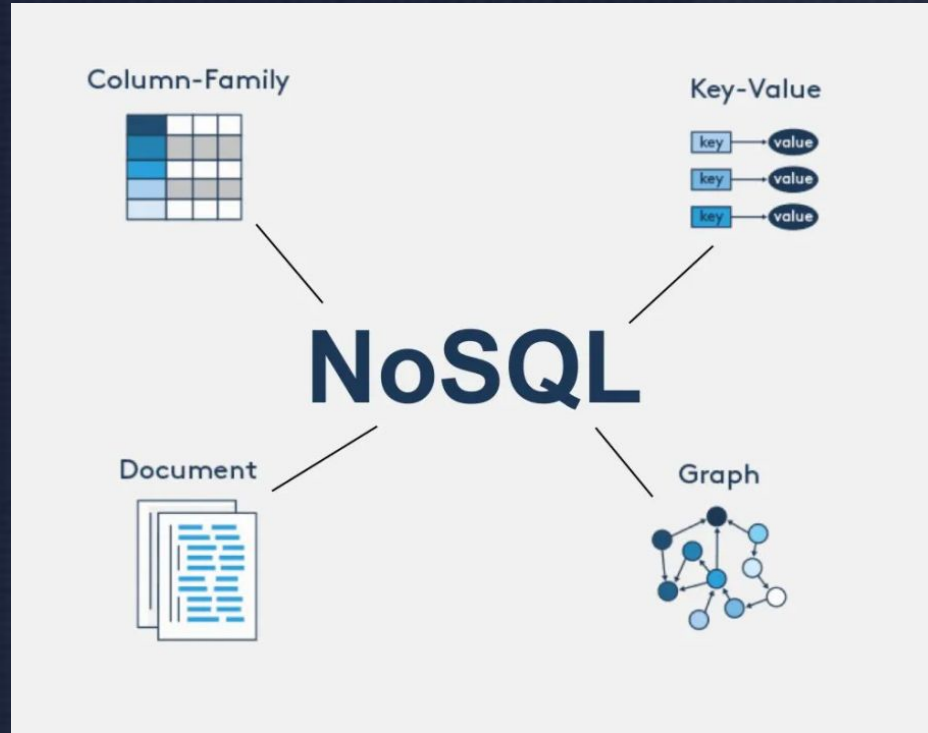


Bancos de dados relacionais possuem uma estrutura rígida de dados, através das tabelas. Porém, em muitos casos essa rigidez pode atrapalhar. O NoSQL pode ajudar:

- **Flexibilidade** da estrutura de dados.
- Redefinição de como armazenar e acessar dados que não se encaixam em tabelas relacionais.
- Facilidade de **expandir** os dados através de múltiplos servidores.
- Melhor **desempenho** para certas aplicações, como **Big Data** e sistemas com grandes volumes de tráfego.



Tipos



Exemplos



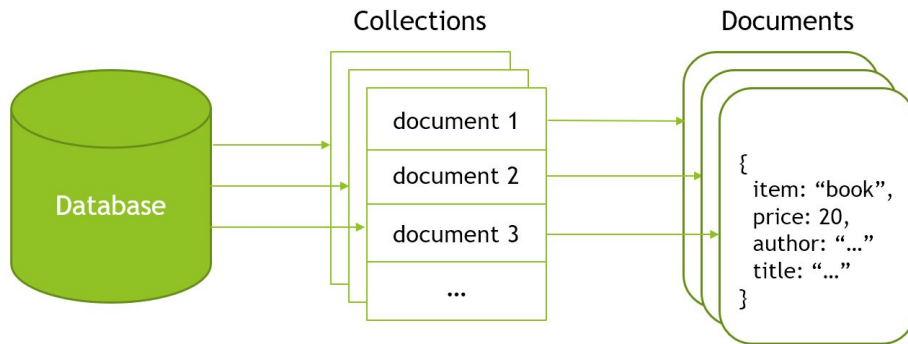
NoSQL – Document



Bancos NoSQL deste tipo armazenam dados em formato de documentos (geralmente JSON, BSON ou XML). Cada documento pode conter diferentes campos, permitindo flexibilidade estrutura de dados.

É mais adequado para aplicações que precisam armazenar dados complexos e aninhados, como perfis de usuários ou catálogos de produtos.

Exemplo: MongoDB, Couchbase.



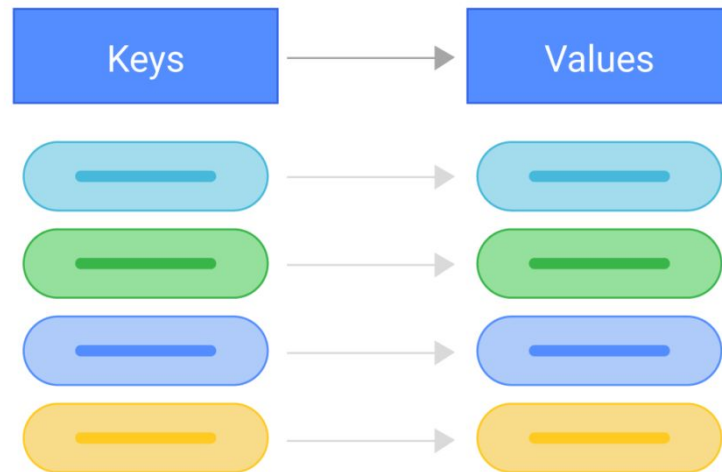
NoSQL - Key-value



Armazenam dados como pares de chave-valor, onde o registro é identificado por sua chave e pode ter uma estrutura definida como valor.

Possui característica mais simples, rápido e eficiente para operações de leitura e escrita. Geralmente usado para armazenar sessões e caches.

Exemplo: Redis, DynamoDB.



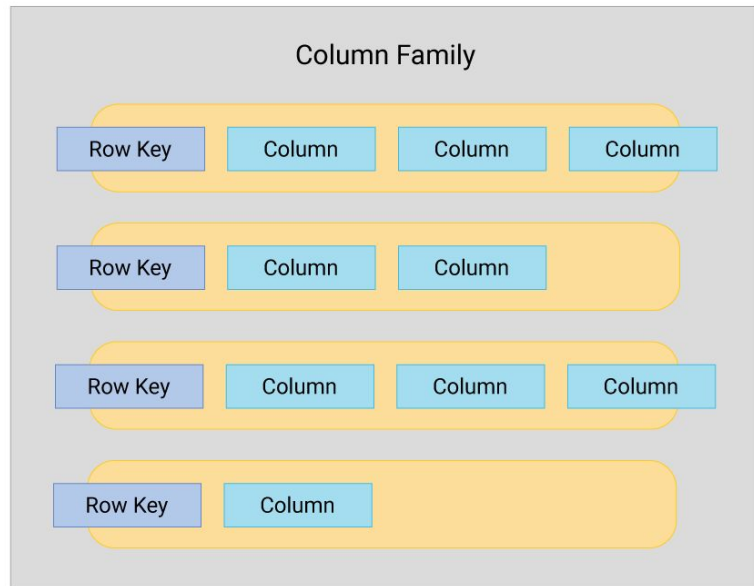
NoSQL - Column-family



Organizam os dados em famílias de colunas em vez de linhas. Parece semelhante ao esquema relacional, porém há uma grande diferença principalmente na forma de acesso.

Bancos column-wide são eficientes para análise de grandes volumes de dados (Big Data) e quando há necessidade de cálculos com poucas colunas.

Exemplo: Cassandra, HBase.



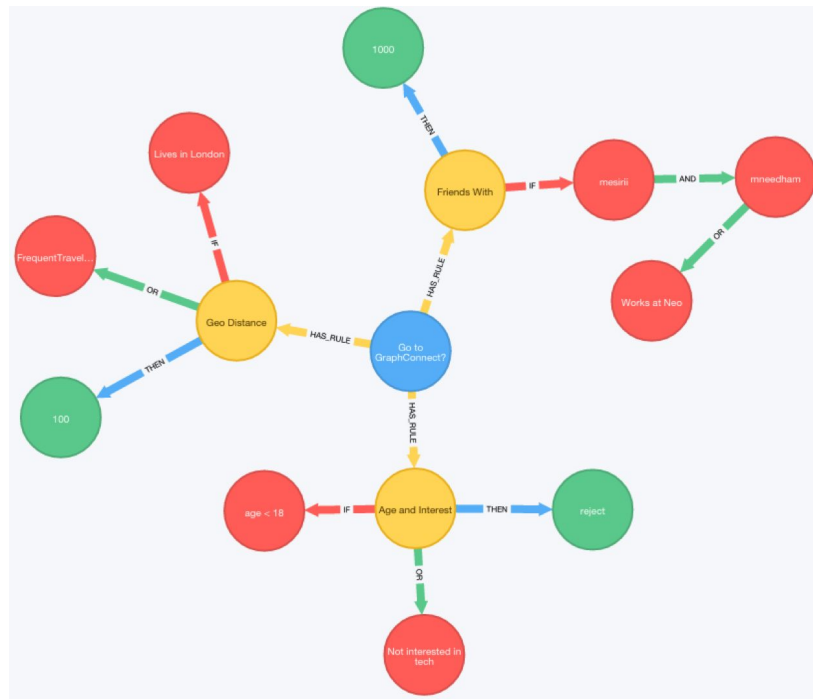
NoSQL – Graph



Como o nome sugere, representa dados em forma de grafos, com nós (entidades) e arestas (relacionamentos).

Ideal para modelar conexões complexas entre dados, como em redes sociais, sistemas de recomendação e análises de fraudes.

Exemplo: Neo4j, Amazon Neptune.



Características

Características



Os bancos de dados NoSQL se destacam por características específicas que os diferenciam dos bancos de dados relacionais (SQL):

- **Desempenho e escalabilidade**
 - Pode lidar com grandes quantidades de dados em múltiplos servidores.
- **Flexibilidade de dados**
 - Capacidade de armazenar dados estruturados, semiestruturados e não estruturados.
- **Alta disponibilidade**
 - Muitas soluções NoSQL oferecem replicação automática e tolerância a falhas.
- **Escalabilidade horizontal**
 - Em vez de aumentar a capacidade de um único servidor, NoSQL pode distribuir dados entre múltiplos servidores.

Consistência eventual



Muitos bancos NoSQL adotam o modelo de **consistência eventual**, onde os dados podem levar algum tempo para serem sincronizados entre diferentes servidores.

Isso é diferente do modelo ACID dos bancos relacionais, que prioriza consistência imediata.



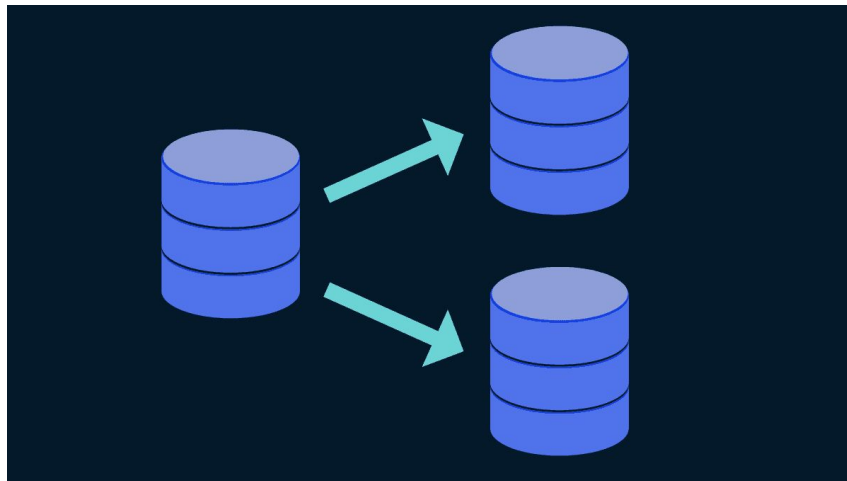
Replicação



Muitos bancos NoSQL usam replicação para garantir alta disponibilidade.

Os dados são copiados para múltiplos nós no cluster, permitindo que, se um nó falhar, outro assuma automaticamente.

A replicação também ajuda a melhorar a performance de leitura, permitindo que múltiplas cópias dos dados sejam acessadas simultaneamente.



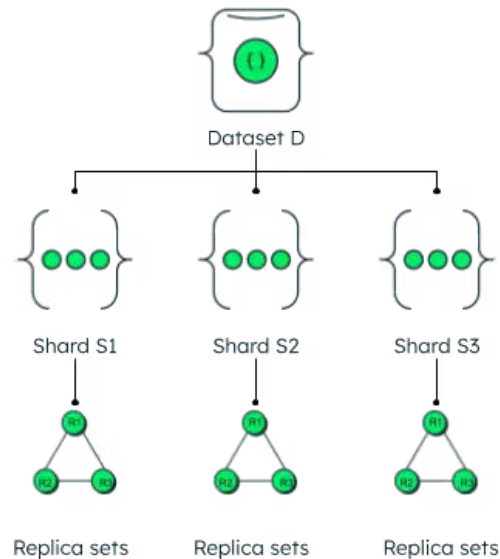
Sharding



Os dados são **divididos em partes menores**, chamadas partições ou **shards**, com base em uma chave de particionamento.

Cada shard contém uma porção dos dados e pode ser armazenado em servidores diferentes.

Isso permite **distribuir** tanto a carga de armazenamento quanto as operações de leitura e escrita.



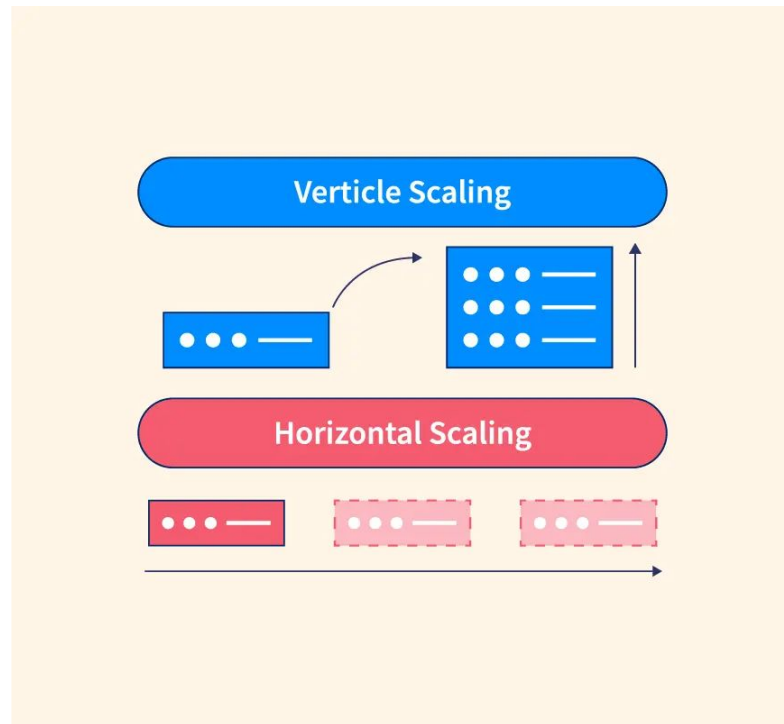
Escalabilidade



A escalabilidade é um dos principais motivos pelos quais os bancos de dados NoSQL têm se tornado populares, especialmente em cenários onde grandes volumes de dados e alta demanda de tráfego precisam ser gerenciados.

Bancos NoSQL priorizam a **escalabilidade horizontal**, que consiste em adicionar mais servidores ao cluster para lidar com cargas maiores de dados ou tráfego.

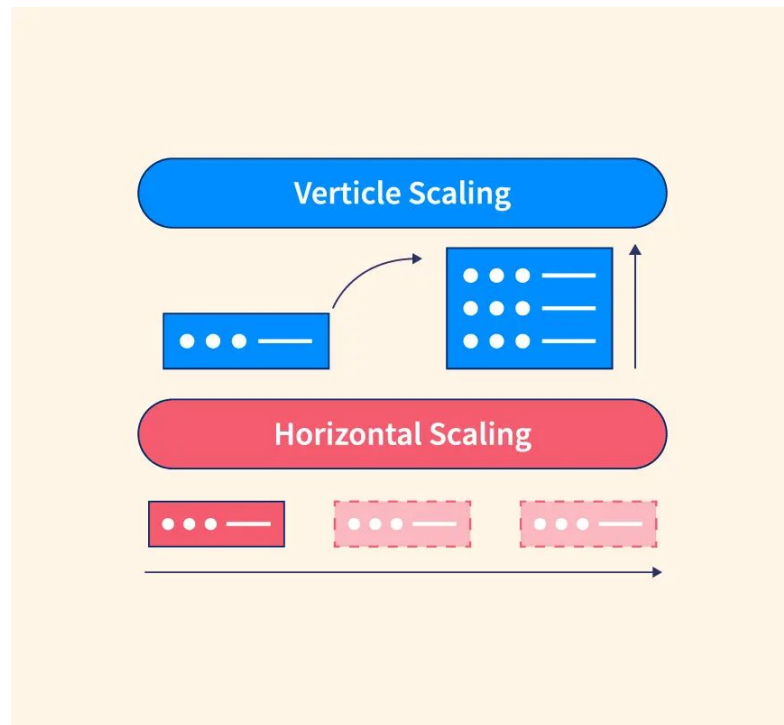
Isso é mais econômico e escalável, pois não depende de um único servidor robusto.



Escalabilidade



- **Custo Reduzido:** é mais barato adicionar servidores do que investir em máquinas robustas e caras.
- **Alta Disponibilidade:** com a replicação e o particionamento, os bancos NoSQL podem continuar operando mesmo se partes do sistema falharem.
- **Suporte a Big Data:** com capacidade de distribuição, consegue lidar com grandes volumes de dados.
- **Performance em Alta Escala:** BDs conseguem manter alta performance mesmo com milhões de usuários ou petabytes de dados.



Desvantagens



- **Consistência Eventual:** Essa característica, apesar de ser uma vantagem, pode ser impeditiva para certas aplicações.
- **Menos suporte a transações:** Faltam transações ACID completas em alguns bancos NoSQL.
- **Curva de aprendizado:** Para desenvolvedores acostumados com bancos relacionais, a transição pode exigir tempo e adaptação.
- **Falta de Padrões:** O ecossistema NoSQL não tem tantos padrões consolidados como os bancos relacionais.

Casos de uso



- **Facebook:** Usando Graph Databases (ex. Cassandra) para armazenar e gerenciar relacionamentos.
- **Netflix:** Uso de NoSQL para personalização de recomendações e escalabilidade em sua plataforma global.
- **Instagram:** Migrando para NoSQL para lidar com grandes volumes de dados de usuários e fotos.
- **IoT (Internet das Coisas):** Bancos de dados de chave-valor, como Redis, são usados para gerenciar dados de sensores em tempo real.
- **Análise de Dados:** Bancos de column-wide, como Cassandra, são utilizados em sistemas de big data para processar grandes volumes de dados rapidamente.

Fixando ...

NoSQL se refere a uma categoria de bancos de dados que não utilizam o modelo relacional tradicional.

É uma excelente alternativa para bancos de dados relacionais quando se trata de escalabilidade, desempenho e flexibilidade.

Escolha certa depende do caso de uso: Nem sempre é a melhor solução para todas as situações.



Parabéns!
Nos vemos na
próxima etapa!