

UNIVERSIDADE DE BRASÍLIA
Faculdade do Gama

Sistemas de Banco de Dados 2

Tecnologias de Banco de Dados (TI-BD)

Bancos de Dados No SQL

Lieverton Santos Silva - 17/0039251

Brasília, DF
2019

Introdução

O termo NoSQL (não SQL ou Not Only SQL - Não Somente SQL) é utilizado para referenciar bancos de dados não relacionais, logo que, a SQL é a principal linguagem para bancos de dados relacionais.

Bancos de dados NoSQL são criados para modelos de dados específicos e têm esquemas flexíveis para a criação de aplicativos modernos. Os bancos de dados NoSQL são amplamente reconhecidos por sua facilidade de desenvolvimento, funcionalidade e performance em escala. Eles usam vários modelos de dados, incluindo documento, grafo, chave/valor, memória e pesquisa.

Por que usar um Banco de Dados NoSQL?

Esse tipo de banco de dados são otimizados especificamente para aplicativos que exigem modelos de grande volume de dados, baixa latência e flexibilidade.

- **Flexibilidade:** os bancos de dados NoSQL geralmente fornecem esquemas flexíveis que permitem um desenvolvimento mais rápido e interativo. O modelo de dados flexível torna os bancos NoSQL ideais para dados semi estruturados e não estruturados.
- **Escalabilidade:** os bancos de dados NoSQL geralmente são projetados para serem escalados horizontalmente usando clusters distribuídos de hardware, em vez de escalá-los verticalmente adicionando servidores caros e robustos. Alguns provedores de nuvem lidam com essas operações nos bastidores como um serviço totalmente gerenciado.

- **Alta performance:** o banco de dados NoSQL é otimizado para modelos de dados específicos (como documento, chave/valor) e padrões de acesso que permitem maior performance do que quando se tenta realizar uma funcionalidade semelhante com banco de dados relacionais.
- **Altamente funcional:** os bancos de dados NoSQL fornecem APIs e tipos de dados altamente funcionais criados especificamente para cada um de seus respectivos modelos de dados.

Tipos de Bancos de Dados NoSQL

- **Chave/valor:** os bancos de dados de chave-valor são altamente particionáveis e permitem escalabilidade horizontal em escalas que outros tipos de bancos de dados não conseguem alcançar. Casos de uso, como jogos, tecnologia de publicidade e IoT, encaixam-se particularmente bem ao modelo de dados de chave-valor. O Amazon DynamoDB foi projetado para proporcionar uma latência consistente de um dígito de milissegundo para qualquer escala de cargas de trabalho. Essa performance consistente teve grande influência no motivo pelo qual o recurso Snapchat Stories, que inclui a maior carga de trabalho de gravação de armazenamento do Snapchat, mudou para o DynamoDB.
- **Documento:** no código do aplicativo, os dados costumam ser representados como um objeto ou um documento do tipo JSON porque esse é um modelo de dados eficiente e intuitivo para os desenvolvedores. Os bancos de dados de documentos facilitam para que os desenvolvedores armazenem e consultem dados usando o mesmo formato de modelo de documento que usam no código do aplicativo. A natureza flexível, semi estruturada e hierárquica dos documentos e dos bancos de dados de documentos permite que eles evoluem conforme as necessidades dos aplicativos. O modelo de documentos funciona bem com catálogos, perfis de usuários e sistemas

de gerenciamento de conteúdo, onde cada documento é único e evolui com o passar do tempo. O MongoDB é um banco de dados de documentos popular que fornece APIs eficientes e intuitivas para desenvolvimento flexível e iterativo.

- **Grafo:** a finalidade de um banco de dados baseado em grafo é facilitar a criação e a execução de aplicativos que funcionam com conjuntos de dados altamente conectados. Os casos típicos de uso de um banco de dados gráfico incluem redes sociais, mecanismos de recomendação e detecção de fraudes. Os bancos de dados baseados em grafos populares incluem o Neo4j e Giraph.
- **Em memória:** aplicativos de jogos e tecnologia de publicidade têm como exemplos de uso placares de líderes, armazenamentos de sessões e análises em tempo real que exigem tempos de resposta em microssegundos e podem ter grandes picos de tráfego a qualquer momento.

NoSQL (não relacional) x SQL (relacional)

Quando se compara as duas tecnologias é importante deixar claro que o NoSQL não veio para substituir os bancos de dados relacionais. Os bancos de dados NoSQL servem para uma gama de problemas que não necessariamente são os mesmos dos bancos de dados relacionais. É importante conhecer o problema para decidir a tecnologia mais apropriada. Por exemplo, não é uma boa ideia armazenar dados de usuário em um sistema NoSQL, como o nome, data de nascimento, endereço e telefone, tudo junto, mesmo que a informação seja apresentada em conjunto. Seria muito difícil registrar o número de usuários do sistema, devido o fato de estes não estarem numa mesma lista. A única forma de obter o número de usuário seria contá-los um a um. Logo seria mais viável utilizar um banco relacional. Por outro lado, os banco não

relacionais, são excelentes para lidar com o problema do Big Data, pois controlam bem o armazenamento de grandes quantidades de informação.

Outras vantagens estão relacionadas às características dos sistemas NoSQL, como a escalabilidade mais barata e esquemas dinâmicos. Vale a pena ressaltar que grande parte dos SGBDs NoSQL são de código aberto. Porém são sistemas novos no mercado, necessitam de mais esforços para manutenção, além da comunidade ser menor e ter menos profissionais especializados.

Principais SGBD's NoSQL

MongoDB

É um document store e o atual líder no segmento de banco de dados NoSQL. Como é o requisito para os engines NoSQL, o MongoDB não usa um esquema relacional, em vez disso, usa documentos semelhantes ao formato JSON (Java Script Object Notation), para armazenar dados. O documento é semelhante a um registro, com campos e valores. O MongoDB suporta esquemas dinâmicos, é um software livre e de código aberto.

O MongoDB também fornece os recursos necessários para um ambiente de produção: balanceamento de carga, replicação, indexação, consulta e pode atuar como um sistema de arquivos (com balanceamento de carga e tolerância a falhas).

Cassandra

Originalmente desenvolvido no Facebook, Cassandra é um mecanismo de banco de dados descentralizado, distribuído e orientado a coluna. É otimizado para clusters, especialmente aqueles em vários datacenters, e graças à sua atualização assíncrona e design sem mestre, o Cassandra fornece acesso de

baixa latência a clientes. Como o MongoDB, também é gratuito e de código aberto.

Cassandra é um banco de dados orientado por coluna, o que significa que suas linhas realmente contêm o que geralmente pensamos como dados verticais, ou o que é tradicionalmente realizado em colunas relacionais. A vantagem do design de banco de dados orientado por coluna é que alguns tipos de pesquisas de dados podem se tornar muito rápidos, uma vez que os dados desejados podem ser armazenados consecutivamente em uma única linha (compare isso com ter que pesquisar e ler de múltiplas linhas não consecutivas para atingir o mesmo valor de campo em banco de dados orientado a linha). No início de 2010, o Cassandra desbancou o MySQL como banco de dados do Twitter, demonstrando sua importância cada vez mais crescente.

Redis

O Redis é a implementação key/value store mais popular e amplamente utilizada. O Redis mantém seus pares de valores-chave na memória, tornando seu acesso rápido. Se a durabilidade dos dados pode ser sacrificada (principalmente com dados não críticos, ou em situações de somente leitura), ser capaz de renunciar a escritas de dados significa que esses dados somente em memória possuem um desempenho incrivelmente rápido. Ao longo dos anos, muitas APIs foram desenvolvidas para uma variedade incrivelmente ampla de linguagens de programação.

Amazon DynamoDB

O Amazon DynamoDB é um serviço de banco de dados NoSQL em nuvem oferecido pela AWS (Amazon Web Service). O DynamoDB é rápido e flexível para todas as aplicações que precisam de latência constante abaixo de 10 milissegundos em qualquer escala. O serviço é um banco de dados em nuvem totalmente gerenciado e é compatível com os modelos de armazenamento de documentos e de chave-valor. O modelo de dados flexível, o desempenho confiável e a escalabilidade automática da capacidade de throughput fazem

desse serviço uma boa opção para aplicações móveis, web e de jogos e tecnologia de anúncios.

Neo4j

O Neo4j é o sistema de gerenciamento de banco de dados baseado em grafos sendo o sistema mais popular desta categoria em uso atualmente. O Neo4j também possui uma implementação de código aberto.

Os dados no Neo4j podem ser acessados e atualizados através da Cypher Query Language, uma linguagem semelhante à linguagem SQL.

Os bancos de dados baseados em grafos têm vantagens em alguns casos de uso, incluindo determinados cenários de mineração de dados e reconhecimento de padrões, uma vez que as associações entre instâncias de dados são explicitamente declaradas.

Referências

VAISH, G. **Getting Started with NoSQL: 1ª Edição**. Birmingham, Reino Unido, Packt Publishing. 2013.

MATOS, David. Top 6 NoSQL Databases. **Ciência e Dados**, 2019. Disponível em: <http://www.cienciaedados.com/top-6-nosql-databases/#targetText=O%20levantamento%2C%20feito%20em%20Fevereiro,em%20compara%C3%A7%C3%A3o%20aos%20bancos%20relacionais>. Acesso em: 8, set. 2019.

Introdução aos bancos de dados NoSQL. **DevMedia**, 2019. Disponível em: <https://www.devmedia.com.br/introducao-aos-bancos-de-dados-nosql/26044>. Acesso em: 8, set. 2019.

O que é NoSQL?. **AWS**, 2019. Disponível em: <https://aws.amazon.com/pt/nosql/>. Acesso em: 8, set. 2019.