

# Capítulo 2

## NoSQL

---

### **NoSQL**

#### **O que é o NoSQL?**

O NoSQL é um movimento que promove soluções de armazenamento de dados não relacionais.

É composto por diversas ferramentas que, de forma particular e específica, resolvem problemas como:

- Tratamento de grandes volumes de dados
- Execução de consultas com baixa latência
- Modelos flexíveis de armazenamento de dados, como documentos XML ou JSON.

As tecnologias NoSQL não têm como objetivo substituir as bases de dados relacionais, mas apenas propor algumas soluções que em determinados cenários são mais adequadas. Desta forma, é possível trabalhar com tecnologias NoSQL e base de dados relacionais dentro de uma mesma aplicação.

- **É útil em:**

Cenários onde sistemas de base de dados tradicionais não são suficientes ou adequados às necessidades específicas, tais como:

- baixa latência
- grandes volumes de dados,
- escalabilidade
- ou estruturas em que as conexões entre os dados são tão importantes quanto o próprio dado.

- **Para que serve:**

As ferramentas NoSQL fornecem meios mais eficientes de armazenamento de grandes volumes de dados e/ou mecanismos de pesquisa de baixa latência, factores importantes que precisam ser considerados durante a escolha de uma solução de armazenamento de dados.

#### **Porquê o NoSQL?**

- **Vantagens de base de dados NoSQL sobre base de dados relacionais:**

As razões para as empresas a adoptarem num ambiente de base de dados NoSQL invés de uma base de dados relacional tem quase tudo a ver com os seguintes drivers do mercado e os requisitos técnicos:

Crescimento de Big Data	Localização real independente
<p>Big Data é um dos principais motivos do crescimento e popularidade do NoSQL para os negócios. A diversidade de tecnologias que recolhem dados e distribuem para vários dispositivos, multiplicam o crescimento dos dados.</p> <p>Na verdade, uma das primeiras razões para usar NoSQL é porque teremos um projeto de Big Data, normalmente caracterizado por:</p> <ul style="list-style-type: none"> <li>- <u>Alta velocidade (de dados)</u> - Dados que vêm rapidamente, possivelmente de diferentes locais.</li> <li>- <u>Variedade de dados</u> - armazenamento de dados estruturados, semi-estruturados e não estruturados.</li> <li>- <u>Volume de dados</u> - dados que envolvem muitos terabytes ou petabytes em tamanho.</li> <li>- <u>Complexidade de dados</u> - dados que são armazenados e geridos em diferentes locais ou em data-centers.</li> </ul>	<p>O termo “localização independente” significa a capacidade de ler e gravar numa base de dados, independentemente de onde a operação I / O ocorre fisicamente e ter qualquer funcionalidade de escrita propagada para fora a partir desse local, para que ele esteja disponível para os utilizadores e máquinas em outros sites. Tal funcionalidade é muito difícil de projetar em base de dados relacionais. Algumas técnicas podem ser implementadas, como arquiteturas de mestre / escravo, podem satisfazer a necessidade da localização de operações de leitura independentes, mas a escrita de dados em todos os lugares é uma questão diferente, especialmente quando esses volumes de dados são elevados.</p> <p>Outros cenários onde a independência do local é uma vantagem são muitas e incluem atendimento aos clientes em muitas regiões geográficas diferentes e que precisam para manter os dados locais a esses sites para acesso rápido.</p>
Disponibilidade de dados contínua	Capacidade transacionais modernas
<p>No mercado de hoje, onde a concorrência é apenas um clique de distância, o tempo de inatividade (downtime) pode ser mortal para a reputação de uma empresa. As falhas de hardware podem e irão ocorrer, felizmente ambientes de base de dados NoSQL são construídas com uma arquitetura distribuída para que não haja pontos únicos de falha e implementação de métodos redundantes dos dados. Se um ou mais servidores da base de dados, ou "nós" for abaixo, os outros nós do sistema são capazes de continuar com as operações sem perda de dados, mostrando, assim, tolerância a falhas verdadeira.</p> <p>Desta forma, ambientes de banco de dados NoSQL são capazes de fornecer disponibilidade contínua seja em locais simples, através de data-centers ou na nuvem. Quando implementado de forma adequada, base de dados NoSQL pode fornecer alto desempenho em escala maciça, que nunca vão abaixo.</p> <p>Isto é extremamente benéfico como atualizações do sistema ou modificações sem ter que colocar a base de dados offline. Este facto atrai a atenção de empresas que prestam serviços a clientes que esperam que as suas aplicações estejam sempre disponíveis e onde o tempo de inatividade equivale perdas de imenso dinheiro.</p>	<p>O conceito de transações parece estar a mudar na era da Internet, e tem sido demonstrado que as transações ACID já não são uma exigência em sistemas de bases de dados orientadas.</p> <p>À primeira vista, essa afirmação parece exagerada, como integridade transacional é uma característica da maioria cada sistema de dados - especialmente aqueles com requisitos de informação que exigem precisão e segurança.</p> <p>No entanto, o que isto se refere não é ao comprometimento dos dados, mas sim às aplicações modernas que garantem a consistência transacional entre sistemas amplamente distribuídos.</p> <p>O "C" no ACID refere-se a consistência de dados em sistemas de gestão de base de dados relacional que é aplicada através de chaves estrangeiras / restrições de integridade referencial. Este tipo de consistência não é utilizada em sistemas de gestão de dados progressivas, tais como base de dados NoSQL, porque não há operações JOIN, pois isso exigiria uma aplicação mais rígida da consistência. Em vez disso, a "consistência" que diz respeito à base de dados NoSQL é encontrado no teorema de CAP, o que significa a consistência imediata ou eventual de dados em todos os nós que participam numa rede de dados distribuída.</p>
Modelos de dados flexíveis	Melhor Arquitetura
<p>Uma das principais razões que as empresas optem por um sistema de base de dados NoSQL de um Sistema de Gestão de Base de Dados Relacional (SGBDR) é devido ao modelo de dados mais flexível que é encontrado na maioria das bases de dados NoSQL.</p> <p>O modelo de dados relacional é baseado em relacionamentos definidos entre tabelas, que são definidas por uma determinada estrutura de coluna, todos estes guardados num schema - tudo muito rigoroso e uniforme.</p> <p>Os problemas começam a surgir com o modelo relacional em torno de escalabilidade e desempenho ao tentar gerir os grandes volumes de dados que estão a tornar-se um facto da vida num moderno ambiente de IT e de negócios.</p> <p>Um modelo de dados NoSQL - muitas vezes referida como sem schema - pode suportar muitos desses casos de uso e outros que não se encaixam bem num SGBDR.</p> <p>Uma base de dados NoSQL é capaz de aceitar todos os tipos de dados - estruturados, semi-estruturados, não estruturados e - muito mais facilmente do que uma base de dados relacional que contam com um esquema predefinido. Esta característica de uma base de dados relacional pode ser um obstáculo à flexibilidade porque um schema predefinido rigidamente determina como são organizados os dados de uma base de dados. Muitas das aplicações de negócios de hoje, na verdade, têm a capacidade de fazer cumprir as regras sobre o uso de dados tornando-se uma plataforma de base de dados sem schema uma opção viável.</p> <p>Finalmente, os factores de performance entram em jogo com um modelo de dados SGBDR, especialmente quando estão envolvidas imensas entradas e se tentam atualizá-las, o que pode ter implicações reais sobre o desempenho. No entanto, um modelo de dados NoSQL lida facilmente com tais situações e proporciona um desempenho muito rápido para operações de leitura e escrita.</p>	<p>Outro motivo para usar uma base de dados NoSQL é porque é necessário uma arquitetura mais adequada para uma aplicação particular. É fundamental que as organizações adotem uma plataforma NoSQL que lhes permite manter os seus dados num contexto das suas aplicações. Algumas, mas nem todas as soluções NoSQL fornecem arquiteturas modernas que podem resolver o tipo de aplicações que requerem altos níveis de escalabilidade, a distribuição de dados e disponibilidade contínua.</p>
Inteligência de negócios e análises	
<p>Um driver estratégico da implementação de um ambiente de base de dados NoSQL é a capacidade de extrair os dados que estão sendo captados a fim de derivar conhecimentos que coloca o seu negócio em uma vantagem competitiva. Extraíndo inteligência de negócios de volumes de dados é uma tarefa muito difícil de alcançar com os sistemas de base de dados relacionais tradicionais. Bases de dados modernas NoSQL não só fornecem armazenamento e gestão de dados de aplicações de negócios, mas também entregar análises de dados integrados que fornecem compreensão instantânea de conjuntos de dados complexos e facilitar a tomada de decisão flexível.</p>	

## Tipos de Base de Dados NoSQL

Existem quatro tipos genéricos de base de dados NoSQL, cada um com seus próprios atributos específicos:

Base de Dados orientada a Grafos	Base de Dados Key-Value
Baseada na teoria dos grafos, estas bases de dados são projetadas para dados cujas relações estão bem representados como um grafo e tem elementos que estão interligados, com um número indeterminado de relações entre eles. Exemplo: Neo4j e Titan.	Base de dados projetada para armazenar dados de uma forma sem esquema ( <i>schema-less</i> ) e uma das menos complexas opções de NoSQL. Num armazenamento chave-valor, a totalidade dos dados consiste num par chave-valor indexado, daí o nome. Exemplos: Cassandra, Dyanmo DB, Azure Table Storage (ATS), Riak, Berkeley DB.
Base de Dados Colunares	Base de Dados orientada a Documentos
Base de dados orientada a colunas, onde a informação deixa ser armazenada em linhas e são concebidas para armazenar tabelas de dados como secções de colunas de dados. Embora esta simples descrição soe como o inverso de uma base de dados padrão, as base de dados colunares oferecem um desempenho muito elevado e uma arquitetura altamente escalável. Exemplos: HBase, BigTable e HyperTable.	Expande a ideia básica de armazenamento "key-value" onde "documentos" são mais complexos, na medida em que contêm dados e a cada documento é atribuída uma chave única, que é usado para recuperar o documento. Estes são projetados para armazenamento, recuperação e getão de informações orientado a documentos, também conhecido como dados semi-estruturados. Exemplos: MongoDB e CouchDB.

A tabela a seguir apresenta alguns dos principais atributos que devem ser considerados ao avaliar os bancos de dados NoSQL.

Tipo de BD NoSQL	Performance	Escalabilidade	Complexidade	Funcionalidade
Key-Value	Alta	Alta	Nenhuma	Variável (Alta)
Colunares	Alta	Alta	Baixa	Mínima
Documentos	Alta	Variável (Alta)	Baixa	Variável (Baixa)
Grafos	Variável	Variável	Alta	Teoria de Grafos