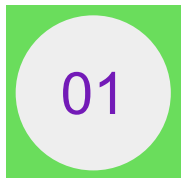
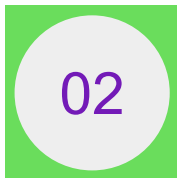


Banco de Dados





NoSQL



SQL



NoSQL



O que são bancos de dados NoSQL?

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.



Como funciona um banco de dados NoSQL (não relacional)?

Os bancos de dados NoSQL usam uma variedade de modelos de dados para acessar e gerenciar os dados. Esses tipos de banco de dados são **otimizados especificamente para aplicativos que exigem modelos de grande volume de dados, baixa latência e flexibilidade**. Esses requisitos são atendidos mediante o relaxamento de algumas restrições de consistência de dados dos outros bancos.

Considere o exemplo de modelagem do esquema para um banco de dados simples de livros:

- Em um banco de dados relacional, um registro de livro é normalmente disfarçado (ou “normalizado”) e armazenado em tabelas separadas, e os relacionamentos são definidos por restrições de chave primária e externa. Neste exemplo, a tabela Livros têm colunas para ISBN, Título do livro e Número da edição, a tabela Autores têm colunas para AuthorID e Nome do autor e, finalmente, a tabela Author-ISBN tem colunas para AuthorID e ISBN. O modelo relacional é projetado para permitir que o banco de dados imponha a integridade referencial entre as tabelas no banco de dados, normalizadas para reduzir a redundância e geralmente otimizadas para armazenamento.

Livros		
ISBN	Título Livro	Número Edição

Autores	
Author ID	Nome Autor

Author-ISBN	
Author ID	ISBN

-
- Em um banco de dados NoSQL, um registro de livro é normalmente armazenado como um documento **JSON**. Para cada livro, o item, o ISBN, o Título do livro, o Número de edição, o Nome do autor e o AuthorID são armazenados como atributos em um único documento. Neste modelo, os dados são otimizados para desenvolvimento intuitivo e escalabilidade horizontal.

Item	ISBN	Título Livro	Número Edição	Nome Autor	AuthorID
------	------	--------------	---------------	------------	----------

Por que você deve usar um banco de dados NoSQL?

Os bancos de dados NoSQL são ideais para muitos aplicativos modernos, como dispositivos móveis, Web e jogos, que exigem bancos de dados flexíveis, escaláveis, de alta performance e altamente funcionais para proporcionar ótimas experiências aos usuários.

- **Flexibilidade:** os bancos de dados NoSQL geralmente fornecem esquemas flexíveis que permitem um desenvolvimento mais rápido e iterativo. O modelo de dados flexível torna os bancos de dados NoSQL ideais para dados semiestruturados 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 e padrões de acesso que permitem maior performance do que quando se tenta realizar uma funcionalidade semelhante com bancos 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.

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, semiestruturada e hierárquica dos documentos e dos bancos de dados de documentos permite que eles evoluam 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.

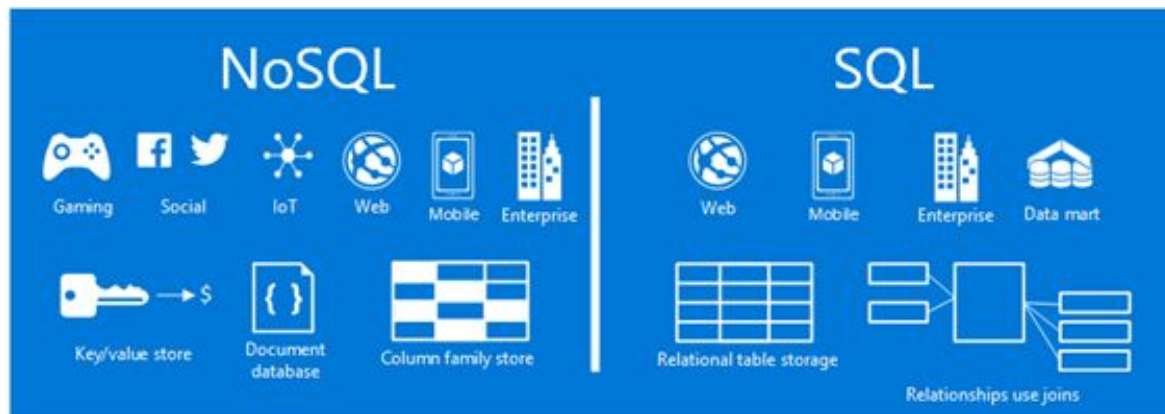


Bancos de dados SQL (relacional) vs. NoSQL (não relacional)



Bancos de dados SQL (relacional) vs. NoSQL (não relacional)

Durante décadas, o modelo de dados predominante usado para desenvolvimento de aplicativos foi o modelo usado por bancos de dados relacionais, como Oracle, DB2, SQL Server, MySQL e PostgreSQL. Somente em meados dos anos 2000 que outros modelos de dados começaram a ser adotados e ter um uso mais significativo. Para diferenciar e categorizar essas novas classes de bancos e modelos de dados, o termo “NoSQL” foi criado. Muitas vezes, o termo “NoSQL” é usado de forma intercambiável com “não relacional”.



Bancos de dados relacionais

Bancos de dados NoSQL

Cargas de trabalho ideais

Bancos de dados relacionais são projetados para aplicativos transacionais e fortemente consistentes de processamento de transações online (OLTP) e são bons para processamento analítico online (OLAP).

Os bancos de dados do NoSQL são projetados para vários padrões de acesso aos dados que incluem aplicativos de baixa latência. Os bancos de dados de pesquisa NoSQL são projetados para análise de dados semiestruturados.

Modelo de dados

O modelo relacional normaliza dados em tabelas, compostas por linhas e colunas. Um esquema define estritamente tabelas, colunas, índices, relações entre tabelas e outros elementos do banco de dados. O banco de dados impõe a integridade referencial nos relacionamentos entre as tabelas.

Os bancos de dados NoSQL fornecem uma variedade de modelos de dados, como chave-valor, documento e gráfico, que são otimizados para performance e escala.

Propriedades ACID

Bancos de dados relacionais fornecem propriedades de atomicidade, consistência, isolamento e durabilidade (ACID):

- A atomicidade exige uma transação para executar completamente ou não é executada de forma alguma.
- A consistência exige que, quando uma transação é confirmada, os dados devem estar em conformidade com o esquema do banco de dados.
- O isolamento exige que as transações simultâneas sejam executadas separadamente umas das outras.
- A resiliência exige a capacidade de se recuperar de uma falha do sistema ou falta de energia inesperada para o último estado conhecido.

Os bancos de dados NoSQL geralmente fazem compensações relaxando algumas das propriedades ACID dos bancos de dados relacionais para um modelo de dados mais flexível que pode ser escalado horizontalmente. Isso torna os bancos de dados NoSQL uma excelente opção para casos de uso de baixa latência e alta taxa de transferência que precisam ser escalados horizontalmente além das limitações de uma única instância.

Performance

A performance normalmente depende do subsistema do disco. A otimização de consultas, índices e estrutura de tabela é necessária para alcançar máxima performance.

A performance geralmente é uma função do tamanho do cluster do hardware subjacente, da latência de rede e do aplicativo que faz a chamada.

Escala

Os bancos de dados relacionais geralmente escalam verticalmente o tamanho ao aumentar os recursos de computação do hardware, ou escalam horizontalmente o tamanho ao adicionar réplicas para cargas de trabalho somente leitura.

Os bancos de dados NoSQL normalmente são particionáveis porque os padrões de acesso podem escalar horizontalmente o tamanho usando arquitetura distribuída para aumentar a taxa de transferência que fornece performance consistente em escala quase ilimitada.

APIs

As solicitações para armazenar e recuperar dados são comunicadas usando consultas compatíveis com uma Structured Query Language (SQL – Linguagem de consultas estruturadas). Essas consultas são analisadas e executadas pelo banco de dados relacional.

APIs baseadas em objetos permitem que desenvolvedores de aplicativos armazenem e restaurem facilmente estruturas de dados. As chaves de partição permitem que os aplicativos procurem pares de chave-valor, conjuntos de colunas ou documentos semiestruturados que contenham objetos e atributos de aplicativos serializados.

Terminologia do SQL vs. do NoSQL

A tabela a seguir compara a terminologia usada pelos bancos de dados NoSQL selecionados com a terminologia usada pelos bancos de dados SQL.

SQL	MongoDB
Tabela	Coleta
Linha	Documento
Coluna	Campo
Chave primária	ObjectId
Índice	Índice
Visualização	Visualização
Tabela ou objeto aninhado	Documento incorporado
Matriz	Matriz

Qual banco de dados utilizar? SQL ou NoSQL?

Depende! Tudo vai depender do seu projeto, então sempre bom analisar o que precisará ser feito para decidir qual banco de dados é melhor para usar. Por exemplo, às vezes fazer consultas em diversas tabelas para conseguir retornar a informação que precisa, quando a base é exageradamente grande, pode exigir muito do seu banco de dados relacional. Pode ser que nesse caso faça sentido utilizar um não relacional (noSQL). Sempre vale uma análise pois cada caso é sempre um caso.

O NoSQL tem muitas vantagens para ser utilizado. Mas não é por isso que devemos utilizá-lo em todas as situações. Em muitos sistemas, você pode (e até deve) usar o modelo relacional. O NoSQL é mais indicado para aqueles sistemas que tenham necessidades maiores de armazenamento e desempenho. O NoSQL não veio para substituir o SQL, mas sim para oferecer mais uma alternativa de um banco de dados mais flexível no suporte de dados. Sendo assim, você pode usar ambas as soluções para diferentes casos.