

Muito se tem falado da utilização de NoSQL. Existem várias comparações de desempenho realizadas entre ele e o SQL, tendo como a velocidade o fator mais importante na decisão de qual banco utilizar. Há muitas opções de bancos de dados NoSQL com diferentes funcionalidades que podem ser muito úteis. Mas como decidir entre um banco de dados relacional e um não-relacional?

SQL

SQL é a sigla para “*Structured Query Language*” que significa, traduzindo para o português, “Linguagem de Consulta Estruturada”. Trata-se de uma linguagem de consulta a banco de dados relacionais. Com o SQL, você pode executar vários comandos para criar, alterar, gerenciar, consultar, dentre outras informações no seu banco de dados. Costumamos dizer que bancos SQL seguem uma modelagem relacional, pois estes se baseiam no fato de que todos seus dados sejam guardados em tabelas. Você pode conferir em nosso artigo os [principais SGBDs relacionais](#).

NoSQL

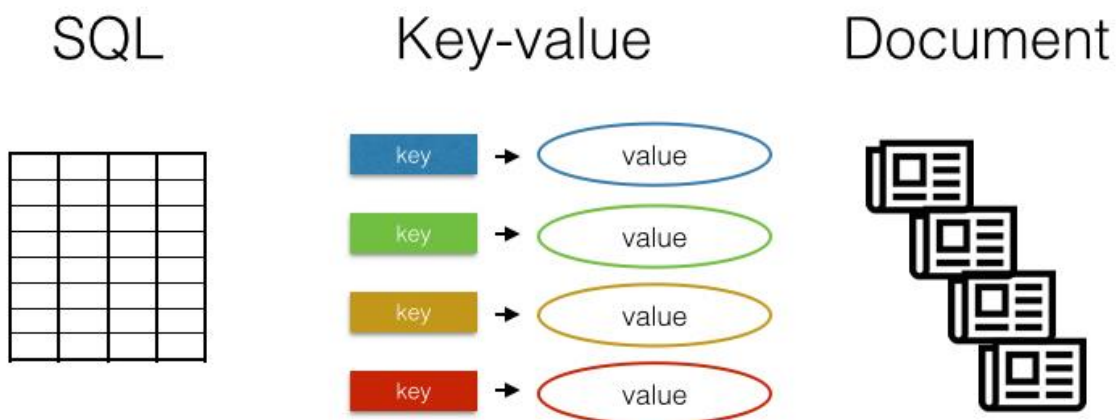
NoSQL (*Not Only SQL*) é o termo utilizado para banco de dados não relacionais de alto desempenho, onde geralmente não é utilizado o SQL como linguagem de consulta. O NoSQL foi criado para ter uma performance melhor e uma escalabilidade mais horizontal para suprir necessidades onde os bancos relacionais não são eficazes. No geral, temos 4 tipos de bancos de dados NoSQL:

- **Documento** – Os dados são armazenados como documentos. Os documentos podem ser descritos como dados no formato de chave-valor, como por exemplo, o padrão JSON. Um exemplo de banco de dados neste formato é o MongoDB;
- **Colunas** – Os dados são armazenados em linhas particulares de tabela no disco, podendo suportar várias linhas e colunas. Também

permitem sub-colunas. Um banco de dados dessa família, por exemplo, é o Cassandra;

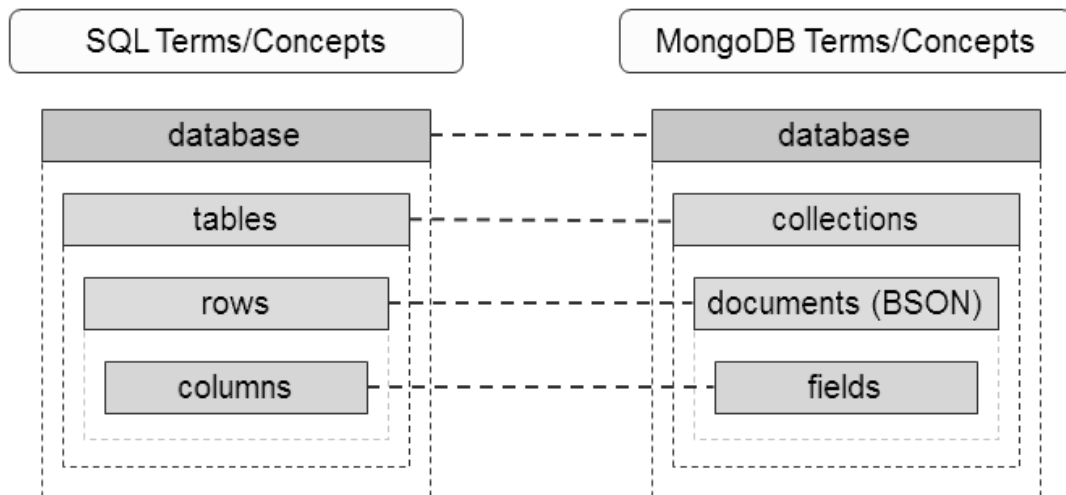
- **Grafos** – Os dados são armazenados na forma de grafos (vértices e arestas). O Neo4j é um banco que utiliza grafos;
- **Chave-valor** – Esta família de bancos NoSQL é a que aguenta mais carga de dados, pois o conceito dele é que um determinado valor seja acessado através de uma chave identificadora única. Um exemplo é o banco de dados Riak.

Resumindo: o conceito de modelo relacional (SQL) se baseia no fato de que todos os dados sejam guardados em tabelas. Ao modelo não-relacional (NoSQL) não se aplica o conceito de *schema*: uma chave de valor é que é utilizada para recuperar valores, conjunto de colunas ou documentos. Você pode conferir nosso artigo com os [principais SGBDs NoSQL](#).



Quais são as diferenças?

Uma das diferenças nos bancos de dados NoSQL é que toda a informação é agrupada e guardada no mesmo registro. Já no SQL você precisa ter o relacionamento entre várias tabelas para ter a informação, informação esta disposta no modelo entidade e relacionamento.



O SQL tem certa dificuldade em conciliar a demanda por escalabilidade. Quanto a escalabilidade do NoSQL, deve se levar em consideração a modelagem do sistema. Do que adianta termos um sistema super simples e querermos utilizar o NoSQL apenas pela escalabilidade? Todo o ganho de performance seria perdido quando rodássemos a aplicação.

Um ponto forte do SQL é quanto à consistência das informações. Já o NoSQL garante o último valor atualizado, isso se nenhuma atualização for realizada até o momento da consulta.

Quanto à segurança, ambos estão suscetíveis a ataques.

Como são feitas as consultas no NoSQL?

As suas chaves devem ser respeitadas, pois, são importantes para sabermos quais dados trazer. Defini-las e definir as formas como queremos consultar as informações armazenadas ali é importante, se pensarmos principalmente em performance. A aplicação precisa saber lidar com a falta de previsibilidade de um schema e assim sabermos qual informação não é importante e com isso pode ser desprezada.

Concluindo

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 de uso. Por isso, o mais comum em soluções escalares de sucesso é a utilização de uma arquitetura híbrida, aproveitando o melhor dos dois modelos.

Referências Bibliográficas

<https://www.treinaweb.com.br/blog/sql-vs-nosql-qual-usar>