**Introdução: SQL e NoSQL — trabalhando com bancos relacionais e não relacionais**

Se você trabalha ou pretende trabalhar com dados, pode ser que já tenha ouvido falar sobre banco de dados. Os bancos de dados são peças fundamentais para a gestão de dados, em várias áreas, desde finanças até saúde e logística. Mas existem dois principais tipos que são utilizados pelo mercado: bancos **relacionais e não relacionais**. Neste artigo, vamos explicar as diferenças entre eles, a origem dos bancos, quando utilizar cada um deles e o uso no mercado.

## O que é um banco de dados relacional?

O **banco de dados relacional** é o tipo de banco que trabalha com tabelas relacionais, isto é, tabelas compostas por linhas e colunas, lembrando muito uma estrutura de tabela de Excel. Cada tabela representa uma entidade ou relação do mundo real. As linhas representam registros individuais nessa entidade, e as colunas representam os atributos ou características dos registros.

A principal característica do banco relacional é a **capacidade de estabelecer relacionamentos entre tabelas** por meio de chaves primárias e estrangeiras. Isso permite que os dados fiquem associados e que, futuramente, sejam consultados de maneira eficiente, garantindo a integridade relacional.

Este modelo relacional é bastante utilizado em sistemas de gerenciamento de bancos de dados (SGBDs) e, para trabalhar com ele, utilizamos SQL (Structured Query Language, linguagem de consulta estruturada), uma linguagem desenvolvida pela IBM na década de 70. Bons exemplos de SGBDs mais usados no mercado são o Oracle, MySQL e Microsoft SQL Server.

Caso queira saber um pouco mais sobre SQL, confira o artigo do Paulo Silveira [Saiba tudo sobre SQL - A linguagem padrão para trabalhar com banco de dados relacionais!](https://www.alura.com.br/artigos/o-que-e-sql)

## Quando utilizar um banco de dados relacional?

O uso de um banco de dados relacional é recomendado em várias situações, principalmente quando temos um cenário que exige uma **estrutura organizada** e **consistente**. E a linguagem SQL tem um papel fundamental para a manipulação e gerenciamento desses bancos. Algumas situações onde o banco de dados relacional costuma ser adequado são:

**1. Estrutura de dados definida**

Um banco de dados relacional é adequado quando os dados possuem uma estrutura definida e há relações claras entre as entidades.

**2. Integridade dos dados críticos**

É recomendado utilizar um banco de dados relacional quando a integridade dos dados é crucial, especialmente em áreas como finanças, estoque ou registros de pacientes.

**3. Consultas complexas e agregações**

Se você precisa realizar consultas complexas, como junções de tabelas, filtragem avançada, agrupamento e cálculos agregados, um banco de dados relacional é uma escolha adequada.

**4. Conformidade e segurança**

Se a conformidade com normas de segurança e regulamentações é fundamental, um banco de dados relacional fornece recursos avançados de segurança, como controle de acesso e criptografia.

## O que é NoSQL?

Agora, quando falamos de NoSQL, vale ressaltar que NoSQL não é uma linguagem. **NoSQL é um termo que referencia tipos de bancos de dados não relacionais**, ou seja, que não seguem o modelo de tabelas e relacionamentos utilizado pelos bancos de dados relacionais tradicionais. Para esses bancos de dados NoSQL, temos uma variedade de modelos, incluindo o **modelo colunar, modelo de grafos, chave-valor e modelo orientado a documentos**. Cada um desses modelos possui suas próprias características e é adequado para diferentes tipos de aplicação e necessidades de armazenamento de dados.

**1. Modelo Colunar**

Também conhecido como armazenamento de colunas, é uma abordagem em que os dados são armazenados como colunas em vez de linhas. Esse modelo é ideal para situações que envolvem grande quantidade de dados e exigem alta performance, pois permite que apenas as colunas relevantes sejam buscadas e lidas, economizando recursos de processamento. Uma das empresas que utilizam esse modelo é a [Netflix](https://medium.com/@andreciobezerra/cassandra-no-netflix-um-estudo-de-caso-de-uso-c4b4ba2ce855" \t "_blank), que utiliza o [Cassandra](https://cassandra.apache.org/_/index.html) para gravações de volume muito alto com baixa latência.

**2. Orientado a Documentos**

Nesse modelo, os dados são armazenados em documentos no formato JSON. Cada documento é identificado por uma chave única e pode conter diversas informações, como atributos e subdocumentos. Esse modelo é interessante para aplicações que exigem flexibilidade na estrutura dos dados e que lidam com grande volume de informações. Na [Expedia](https://www.mongodb.com/customers/expedia), a empresa utiliza o modelo flexível do [MongoDB](https://www.mongodb.com/" \t "_blank) que facilita o armazenamento de qualquer combinação de cidades, datas e destinos.

**3. Chave-valor**

Os dados são armazenados em pares de chave-valor, o que significa que cada dado é identificado por uma chave única. Esse modelo é ideal para aplicações que exigem alta performance em leitura e gravação de dados, como em aplicações de cache ou armazenamento de sessões de usuários. Este modelo usado pelo [Twitter](https://redis.io/docs/manual/patterns/twitter-clone/" \t "_blank), que utiliza o [Redis](https://redis.io/) para implementar recursos em tempo real como contadores de retweets, curtidas e seguidores.

**4. Modelo de Grafos**

Neste modelo os dados são usados para armazenar dados interconectados, como em redes sociais ou sistemas de recomendação. Com o modelo de grafos, é possível fazer buscas detalhadas nas relações entre os dados, mesmo em bancos com centenas de milhares de relacionamentos. O [Medium](https://neo4j.com/videos/how-medium-uses-neo4j/?" \t "_blank), por exemplo, utiliza o [Neo4j](https://neo4j.com/) para criar grafos que representam as conexões entre usuários e artigos, permitindo a montagem de um sistema de recomendação.

Caso queira saber mais sobre NoSQL. O Vinicius Dias fala sobre o tópico no Alura+ [O que é NoSQL?](https://cursos.alura.com.br/extra/alura-mais/o-que-e-nosql--c1142)

## Quando utilizar bancos de dados não relacionais?

Os bancos não relacionais oferecem uma flexibilidade e escalabilidade muito vantajosa, principalmente quando se trata de grandes conjuntos de dados. Mas como as operações dos bancos NoSQL dependem do tipo de modelo escolhido, para utilizá-lo, precisamos entender a necessidade de nosso negócio, como:

**1. Aplicações que trabalham com cache**

Em cenários onde o desempenho de leitura e gravação é fundamental, como em um sistema que precise de armazenar dados frequentemente acessados de forma rápida (sistema de cache) em tempo real, os modelos chave-valor dos bancos NoSQL, como o Redis, são frequentemente utilizados devido à sua alta velocidade de acesso e recuperação.

**2. Sistemas de catálogos ou estruturas flexíveis**

Se a aplicação requer flexibilidade na estrutura e na consulta de dados, o modelo **orientado a documentos**, como MongoDB, pode ser uma boa escolha pela sua capacidade de conter informações de um objeto em um único documento.

Por estas e outras razões, é muito complicado comparar um modelo com outro, já que dependemos bastante do problema que precisamos resolver. Este assunto também já foi debatido em um artigo da Danielle Oliveira, sobre [Cassandra ou MongoDB, qual a melhor escolha para o meu projeto?](https://www.alura.com.br/artigos/cassandra-ou-mongodb-qual-a-melhor-escolha-para-o-meu-projeto), onde ela conta sobre os dois bancos NoSQL bastante utilizados no mercado.

Em resumo, tanto os bancos de dados relacionais quanto os não relacionais têm seus pontos fortes e fracos, e a escolha entre um ou outro dependerá das necessidades específicas de cada aplicação. Os bancos de dados relacionais são ideais para aplicações que exigem consistência e integridade de dados, enquanto os bancos de dados não relacionais são mais adequados para aplicações que exigem alta escalabilidade e flexibilidade no esquema de dados. Por isso, ter conhecimento dessas ferramentas é interessante para qualquer profissional de dados.