1. **INTRODUÇÃO**

Ao decorrer da história da ciência da computação, houve múltiplas mudanças em diversos componentes desse campo - desde linguagens diferentes para se programar até novos processos e plataformas. Nesse contexto, algo que se mantém constante é a evolução dos modelos de bancos de dados visando sua optimização. Assim, durante cerca de duas décadas, o modelo vigente e mais utilizado foi o modelo de banco de dados relacional.

Contudo, com a popularização da internet e a maior acessibilidade da tecnologia, o mercado da Ciência da Computação se ampliou e, assim novos problemas em relação ao armazenamento de dados veio a superfície. Com isso iniciou-se o movimento “NoSQL”, que, no começo, interpretava-se como “Não SQL”, ou seja, um movimento contrário ao uso de SGBDs relacionais. Porém, ao passar do tempo a sigla passou a ser interpretada como “Não Apenas SQL”, ressaltando a necessidade de sempre pesquisar e escolher o melhor modelo para cada caso, mesmo que esse seja um modelo relacional.

No livro “NoSQL: como armazenar os dados de uma aplicação moderna”, David Paniz, exemplifica a necessidade do uso de um modelo NoSQL para o registro de discos de uma banda, em que cada disco teria diferentes atributos associados e que, caso armazenados em um modelo SQL, haveria tabelas com dezenas de colunas e diversos campos nulos. Dito isso, é importante ressaltar que, no geral, a aplicação do NoSQL não se trata de resolver problemas de resolução impossível no modelo relacional, mas sim sobre tornar tais soluções mais práticas e eficientes.

Em síntese, em um contexto em que são necessárias soluções mais maleáveis para abranger uma gama maior de aplicações de um banco de dados, o modelo NoSQL se tornou um aliado aos cientistas de dado quando se trata da optimização do armazenamento de dados por oferecer a adaptabilidade que lhes é necessária.

Nesse sentido, a seguir, este ensaio discorrerá brevemente sobre os fundamentos desse modelo, sua relação com OLTP e OLAP, e uma comparação técnica entre dois SGBDs.

1. **FUNDAMENTOS DE NoSQL**

**2.1. CONCEITO**

Bancos de dados NoSQL, não sistemas não relacionais com modelos de armazenamento de dados mais flexíveis, diferentemente do modelo tradicional de tabelas e tuplas. Nesse modelo, é possível acomodar dados não estruturados ou semiestruturados, o que permite maior adaptabilidade a mudanças nos dados, além de bancos de dados NoSQL serem optimizados para trabalhar em alta velocidade.

**2.2. HISTÓRICO E MOTIVAÇÃO**

Antes da grande popularização da internet nos anos 2000, os bancos de dados empresariais costumavam lidar com quantias de dados consideravelmente menores e menos detalhadas, porém, após tal popularização, as grandes empresas começaram a lidar com uma gama de usuários muito maior e os sites começaram a monitorar as atividades desses usuários de forma muito mais detalhada e, a cada momento, com maior velocidade – esse armazenamento de dados altamente variados, em grande volume e velocidade é o que conhecemos hoje como Big Data.

Nesse novo cenário, notou-se que o manejo de dados pelo modelo relacional SQL tornou-se inviável, visto que a maioria das databases SQL não eram pensadas para rodar em clusters – conjunto de maquinas mais baratas que funcionam em sincronia com um único sistema –, que começou a ser adotado por ser mais barato do que investir em um único servidor de alta performance e por ser mais seguro, visto que os danos pela interrupção do funcionamento de uma das máquinas em um cluster são consideravelmente menores do que os causados pela interrupção do funcionamento em um servidor inteiro, além do fato de a maioria dos SGBDs relacionais cobrarem a licença de uso por unidade de servidor.

Assim, tornou se evidente que outra opção era necessária para melhor se adaptar a nova realidade. E então, iniciou-se a idealizar o modelo NoSQL, um modelo de banco de dados orientado a rodar em clusters e mais eficiente no manejo de dados com relações mais complexas. Desse modo, sendo um modelo com melhor escalabilidade e maior eficiência no tratamento de dados em ampla quantidade e variação.

Ironicamente, o termo NoSQL surgiu primeiramente como o nome de um banco de dados relacional open-source, o Strozzi NoSQL, pois esse banco de dados não utilizava SQL como linguagem de query. Porém, mais adiante, esse termo iniciou a representar o movimento contrário ao uso de bancos de dados relacionais, contudo, após mais reflexões, tornou a representar o movimento que defende o uso variado de modelos de database visando o melhor desempenho para cada caso específico.

**2.3. CATEGORIAS DE NoSQL**

Existem 4 categorias de bancos de dados não relacionais, entre elas estão:

* **Chave-valor:** os dados são armazenados em pares, uma chave e um valor, cada um sendo um registro único, a chave identifica o registro e o valor pode ser uma estrutura de dados simples ou complexa.
* **Colunar:** todos os registros são parte de uma mesma tabela mas cada um pode ter diferentes colunas, os dados são organizados verticalmente e de forma contígua em cada linha.
* **Documento:** cada registro tem uma coleção específica, porém a coleção armazena os dados em formato não estruturado ou semiestruturado, como JSON ou XML.
* **Grafo:** cada registro é um nó em um grafo, e são interligados por relacionamentos.

**2.4. VANTAGENS E DESVANTAGENS**

Existem múltiplas vantagens no uso de bancos de dados não relacionais, dentre elas a escalabilidade horizontal – é um modelo criado e pensado para funcionar em clusters, ou seja, sempre que necessário há a possibilidade de adicionar uma nova máquina ao sistema –, flexibilidade – permite o armazenamento de dados com estruturas variadas –, alta disponibilidade – ao utilizar replicação de dados, garante que o sistema continue disponível mesmo após a falha de um servidor –, alto desempenho – são otimizados para processar grandes volumes de dados em alta velocidade – e custos reduzidos – em sua maioria são modelos open-source, o que elimina o custo com licenciamento.

Contudo, essa tecnologia, assim como todas as outras, tem suas desvantagens, dentre elas estão a eventual inconsistência – a maioria sacrifica o ACID (atomicidade, consistência, isolamento e durabilidade) pelo BASE (basically available, soft state e eventually consistente, ou seja, disponibilidade, maleabilidade e se torna consistente ao longo do tempo) –, consultas mais complexas – podem ser mais difíceis de executar e um pouco mais lentas – e falta de padrão universal – não há um padrão definido para esse formato de banco de dados, nem para sua linguagem de query.

1. **OLTP vs OLAP**
2. **SGBD1 vs SGBD2**
3. **CONCLUSÃO**

**FONTES**

NoSQL Distilled – Martin Fowler e Pramod J. Sadalage

NoSQL: como armazenar os dados de uma aplicação moderna – David Paniz

<https://nelsonfrugeri-tech.medium.com/entenda-de-uma-vez-por-todas-os-conceitos-b%C3%A1sicos-dos-banco-de-dados-chave-valor-5ba44c1a9ab8>

<https://www.purestorage.com/br/knowledge/what-is-a-columnar-database.html#:~:text=Voc%C3%AA%20pode%20estar%20familiarizado%20com,coluna%20e%20n%C3%A3o%20por%20linhas>.

<https://aws.amazon.com/pt/nosql/#:~:text=de%20dados%20NoSQL-,O%20que%20s%C3%A3o%20bancos%20de%20dados%20NoSQL?,a%20come%C3%A7ar%20a%20us%C3%A1%2Dlo>.

<https://www.homehost.com.br/blog/banco-de-dados/nosql/#:~:text=Desafios%20e%20Limita%C3%A7%C3%B5es%20do%20NoSQL,de%20dados%2C%20como%20sistemas%20financeiros>.