

# NoSQL: UMA RELAÇÃO, NÃO RELACIONAL\*

Fabio Machado de Oliveira – Universidade Estadual do Norte Fluminense

Saymon Pires da Silva – Centro Universitário São Camilo/ES

Carlos Henrique Medeiros de Souza – Universidade Estadual do Norte Fluminense

**RESUMO:** A Web 2.0 modificou a forma como criamos conteúdo e trocamos informações, além de habilitar o desenvolvimento de uma série de novas aplicações e tecnologias, dentre elas o NoSQL (Not Only SQL). O grande volume de dados gerados por aplicações Web, e os requisitos diferenciados destas aplicações, forçam pesquisas em busca de dar aos bancos de dados uma nova forma de escalabilidade, sob demanda e elevado grau de disponibilidade, motivando novos paradigmas e tecnologias. A nova estrutura de dados, chamada NoSQL (Not Only SQL), foi proposta com o objetivo de atender aos requisitos das novas demandas em gerenciamentos de dados, semiestruturados ou não estruturados. Este artigo apresenta os fundamentos da tecnologia de banco de dados NoSQL. Foi apresentada a importância desta tecnologia neste novo cenário do ciberespaço. Destacou-se as principais características do NoSQL, pois como cada banco de dados desta plataforma pode ter diversos meios de implementação, seria difícil detalhar cada um deles. Não foi defendido o lado Relacional ou Não-Relacional neste estudo, o objeto principal está em demonstrar essa tecnologia que tem sua gênese sustentada por software livre.

**PALAVRAS-CHAVE:** NoSQL. Flexibilidade. Novas Tecnologias.

## INTRODUÇÃO

O número de informações e dados teve um crescimento considerável, a partir da Web 2.0. Como apontam os estudos de Costa (2010, p.1),

“...surgiram aplicações que necessitam de sistemas que consigam lidar com milhares de utilizadores e escalar milhões de *updates* e leituras a base de dados. Este volume de operações contrasta com o design das bases de dados e *data warehouses* tradicionais.”

Em tempos de grande volume de dados e de busca por velocidade no tráfego dos mesmos, algumas relações tiveram de ser renovadas para suportar este número de requisições.

Segundo Lóscio et al. (2010, p.1) “Os Sistemas Gerenciais de Banco de Dados(SGBD) surgiram em meados da década de 1970, e são utilizados de forma massiva na maioria das operações que envolvem bancos de dados.”

A internet junto com uma busca de diminuir a perda de espaço nos Datacenters, melhoria na performance e escalabilidade dos bancos de dados, provocaram o desenvolvimento de novas

---

\*

XIII EVIDOSOL e X CILTEC-Online - junho/2016 - <http://evidosol.textolivre.org>

perspectivas, algumas soluções foram encontradas, mas só em 1988 o NoSQL (Not Only SQL), que inicialmente chamado pelo seu autor de “NoRel” (termo que nos remete ao termo “Não Relacional”) é utilizado.

BRITO (2010, p.2) demonstra em seu texto que,

” O termo NoSQL surgiu em 1998, a partir de uma solução de banco de dados que não oferecia uma interface SQL, mas esse sistema ainda era baseado na arquitetura relacional [14]. Posteriormente, o termo passou a representar soluções que promoviam uma alternativa ao Modelo Relacional, tornando-se uma abreviação de Not Only SQL (não apenas SQL), sendo utilizado principalmente em casos em que o Modelo Relacional não apresentava performance adequada”.

Entendemos assim que não é o propósito da tecnologia tomar lugar ou deprecuar o antigo meio Relacional, reforçando essa ideia, BRITO (2010) diz que em 2004 a empresa Google lança o BibTable, que tem por característica exatamente suprir a falta de flexibilidade da estrutura encontrada no Modelo Relacional.

Antes de entender o NoSQL e interessante entendemos os Bancos Relacionais. De acordo com Date (2003) um banco de dados tem como essência ser um sistema de manutenção de registros (dados). O autor também explica que um sistema relacional é aquele onde temos os dados entregues aos usuários por meio de tabelas. Além disto, é dito que os operadores que são dispostos ao usuário são operadores que geram tabelas novas, a partir de tabelas mais antigas, com um operador de restrição, que extraia um subconjunto das linhas de uma tabela, ou até um operado de projeção, que traz um subconjunto de colunas, sendo assim, ao unirmos os subconjuntos se obtêm uma tabela.

Quando está relação ocorrer é tido então um banco relacional onde por meio de tabelas conseguimos extrair um dado com mais facilidade, ajudando então descobrir onde se deve guardar um novo dado sem que haja a o trabalho de entender como será salvo, pois com as tabelas e seus relacionamentos, tornam estas execuções mais simples e práticas, tal qual para a recuperação desta informação. A Figura 1 demonstra um exemplo de tabela em um banco de dados relacional.

Código	Tipo	Altura	Potência	Peso	Altura de elevação	Tag	Motores	Diâmetro
1	Britador		300 cv	2000 kg		BR-121	3	
2	Cimento							
3	Rompedor de maticos	4 metros		300 kg		RM-11	1	
4	Empilhadeira		10 cv	400 kg				
5	Transportador de correia				2 m			
6	Tubulação							30 cm

**Figura 1:** Tabelas de um modelo Relacional

**Fonte:** <http://www.itexto.net/devkico/?p=1199> – acesso em 10/09/2014

## NoSQL E SUAS PARTICULARIDADES

Segundo Lóscio et al. (2011) pequenas empresas e a comunidade de software livre foram os primeiros a adotar os bancos de dados não relacionais. Com a difusão desta tecnologia, o termo NoSQL (Not Only SQL) foi adotado para definir esta nova estrutura de banco, o termo que significa

“não apenas SQL”, tem relação direta com os novos SGBDs que não adotam o modelo relacional e são mais flexíveis se comparado as antigas propriedades ACID. A alta escalabilidade dos requisitos faz com que esta flexibilidade seja necessária para gerenciar o número grande de dados, assim garantindo uma melhor performance, mantendo a disponibilidade dos mesmos. O que vai de encontro com as aplicações da Web 2.0. Ele explica também que com a **Escalabilidade Horizontal** nos bancos não-relacionais (BNR) é possível medir o crescimento do volume de dados, com isto alterar a escalabilidade é melhorar o desempenho. Uma Alternativa muito conhecida para alcançar a escalabilidade horizontal é o Sharding, que consiste em dividir os dados em múltiplas tabelas a serem armazenadas ao longo de diversos nós de uma rede.

Apesar dos bancos de dados NoSQL não terem uma garantia da integridade dos dados como nos SGBDs tradicionais, por uso de técnica de ausência de esquemas ou esquemas flexíveis, esta sua característica facilita a escalabilidade e aumenta a disponibilidade. Com o suporte nativo a replicação, os dados tendem a serem encontrados em um número menor de tempo, esta replicação e feita de forma nativa, a partir de duas frentes de replicação, a Mestre-Slave, que em cada escrita no banco tem como resultado N escritas, sendo que N é o número de nós escravos. Após a escrita do nó mestre a replica é refeita e escrita nos nós filhos, isso cria uma leitura mais rápida, entretanto não é recomendado para um número grande de dados. Temos também o Multi-Master que não trata mais um nó mestre apenas, mais vários e a forma de replicação é equivalente a do Mestre-Slave, a vantagem e o desafogo no gargalo que ocorre na escrita dos dados, ainda assim a multiplicidade de nós mestres faz com que tenhamos conflitos de dados.

Duas outras principais características deste banco são:

Lóscio et al. (2011) entende que a Consistência eventual e uma API simples para acesso aos dados como estas últimas duas características;

Em bancos NoSQL a consistência nem sempre é mantida entre os diversos pontos de distribuição de dados, esta característica é dada pelo princípio do teorema CAP (Consistency, Availability e Partition Tolerance).

Costa; Costa, (2010) diz que num sistema de computação distribuída, ter simultaneamente as seguintes propriedades: Consistência, Disponibilidade e Tolerância a falhas; Sendo que os três requisitos influenciam o desenho e instalação de sistemas de base de dados distribuídas. Ele cita também que é importante definir o que são cada uma destas propriedades. Em suas palavras, “Reconhecer quais dos requisitos do teorema de CAP são importantes para o nosso sistema é o primeiro passo para construir com sucesso um sistema distribuído, escalável e com alta disponibilidade.” (Costa; Costa, 2010, p.3).

O que inicialmente parece ser algo menos organizado, pelo fato do objetivo de uma solução NoSQL ser prover uma forma eficiente de acesso aos dados, onde oferece alta disponibilidade e escalabilidade, ignorando a forma com que os dados são armazenados tendo em foco a recuperação dos mesmos. Assim sendo é necessário criar APIs que facilitem o acesso a estas informações, desta forma permitimos que qualquer aplicação possa utilizar os dados dos bancos de maneira rápida e eficiente.

Lóscio et al. (2011, p.5) Define as que a função da API Simples em NoSQL como:

“... o objetivo da solução NoSQL é prover uma forma eficiente de acesso aos dados, oferecendo alta disponibilidade e escalabilidade, ou seja, o foco não está em como os dados são armazenados e sim como poderemos recuperá-los de forma eficiente. Para isto, é necessário que APIs sejam desenvolvidas para facilitar o acesso a estas informações, permitindo que qualquer aplicação possa utilizar os dados do banco de forma rápida e eficiente.”

## CONCLUSÃO

Notasse então que a flexibilidade de um NoSQL não se dá apenas no quesito programático, mais dá a seu usuário a flexibilidade de poder de fato se mover dentre suas características de modo a alcançar o melhor resultado para seus problemas. Neste trabalho foi descrito brevemente a nova necessidade das empresas de TI em todo mundo em procurar uma em relação com banco de dados na web. A alternativa mais procurada e aceita é a opção NoSQL, no que tange os quesitos como: a necessidade de disponibilidade grandes quantidades de dados e necessidade de compartilhamento em tempo real.

Como aumentou consideravelmente o número de dados e informações, a agilidade e flexibilidade do NoSQL, permitiu as novas aplicações gerencie seus dados de melhor forma. É importante saber que a solução NoSQL não veio com o intuito de substituir o modelo relacional em âmbito geral, e sim permitir a melhor utilização de banco de dados que nem sempre é possível nos bancos relacionais.

Foi visto as opções que o NoSQL pode nos oferecer, adaptando-se a necessidade de cada um, permitindo que as aplicações tenham vantagens de: alta disponibilidade, escalabilidade, esquema flexível, alta performance e gerenciamento de dados semiestruturados, com essas vantagens sendo mais eficaz que os bancos de dados tradicionais, para empresas como Facebook, Google e Amazon que aderiram a solução.

Com tantas tecnologias de bancos de dados disponíveis e com a crescente evolução das mesmas, é de se esperar que cada vez mais o venham surgindo opções de banco de dados não relacionais. Sendo assim enquanto uma base de dados Relacional tem um modelo específico, inflexível e tabular, o NoSQL oferece maior flexibilidade existindo uma solução viável para os novos tipos de estruturas.

## REFERÊNCIAS

BRITO, Ricardo W. *Banco de Dados NoSQL x SGBDs Relacionais: Analise Comparativa*. 2010.

COSTA, Rui; COSTA, Teresa; *SQL e NoSQL*. 2010.

LOSCIO et. al., *NoSQL no desenvolvimento de aplicação Web colaborativas*. 2011.

SILVA, Tiago Pasqualini da; *Cassandra – Uma sistema de armazenamento NoSQL altamente escalável*. 2010.

QUEIROZ, Gilberto Ribeiro de; MONTEIRO, Antônio Miguel Vieira; CÂMARA, Gilberto. *BANCOS DE DADOS GEOGRÁFICOS E SISTEMAS NOSQL: ONDE ESTAMOS E PARA ONDE VAMOS*. 2012.

TANAKA, Luiz Carlos; CAMARGO, Felipe Melo; GOTARDO, Reginaldo; *SISTEMA GERENCIADOR DE BANCO DE DADOS: SGBD eXist XML*. Revista Eletronica de sistemas de informação e Gestao Tecnologica. Vol.02; São Paulo; 2012.

TOTH, Renato Molina. *Abordagem NoSQL – uma real alternativa*. Sao Paulo; 2011.

RAOULT, B. *Architecture of the new MARS server*. Disponível em: <[www.ecmwf.int/publications](http://www.ecmwf.int/publications)>. Acesso: 05 outubro 2014.

GRAY, J.; LIU, D. T.; NIETO-SANTISTEBAN, M.; SZALAY, A.; DEWITT, D. J.; HEBER, G. *Scientific Data Management in the Coming Decade*. ACM SIGMOD, v. 34, n. 4, December 2005.

CUDRE-MAUROUX, P.; KIMURA, H.; LIM, K.- T.; ROGERS, J.; SIMAKOV, R.; SOROUSH, E.; VELIKHOV, P.; WANG, D. L.; BALAZINSKA, M.; BECLA, J.; DEWITT, D.; HEATH, B.; MAIER, D.; MADDEN, S.; PATEL, J.; STONEBRAKER, M.; ZDONIK, S. *A demonstration of scidb: a science-oriented dbms*. Proc. VLDB Endow., v. 2, n. 2, p.1534-1537, August, 2009.