

CENTRO UNIVERSITÁRIO UNA  
DIRETORIA DE EDUCAÇÃO CONTINUADA, PESQUISA E EXTENSÃO  
CURSO DE PÓS GRADUAÇÃO ENGENHARIA DE SOFTWARE

NoSQL  
Banco de Dados Não Relacional

ALUNO: Heitor Oliveira Silva  
PROFESSOR ORIENTADOR: Gilmar Luiz de Borba

Belo Horizonte  
2012 / 2º Semestre

## ÍNDICE

|   |    |
|---|----|
| 1. Resumo                                 | 3  |
| 1.1 Palavras Chave                        | 3  |
| 2. Introdução                             | 3  |
| 2.1 O que é banco de dados                | 4  |
| 2.2 SGBD                                  | 4  |
| 2.3 Linguagem SQL                         | 4  |
| 2.4 O Modelo relacional                   | 5  |
| 2.5 O Modelo não relacional               | 6  |
| 2.6 Problematização                       | 6  |
| 2.7 Objetivo Geral                        | 6  |
| 2.8 Objetivos Específicos                 | 6  |
| 3 Metodologia                             | 7  |
| 4 Análise dos Dados                       | 7  |
| 4.1 As Vantagens do Modelo Relacional     | 7  |
| 4.2 As Desvantagens do Modelo Relacional  | 8  |
| 4.3 O início do modelo não relacional     | 9  |
| 4.4 Tipos de soluções não relacionais     | 9  |
| 4.5 Desvantagens do modelo não relacional | 10 |
| 4.6 Relacional X Não relacional           | 10 |
| 5 Considerações finais                    | 11 |
| 6 Referencial Teórico                     | 13 |

## 1. Resumo

O modelo não relacional, chamado de *NoSQL (Not Only SQL)*, é uma tecnologia para escalabilidade e *storage* de dados que permite maior flexibilidade na estruturação do banco. É o escolhido por empresas como Facebook, Google e Twitter para tratar de alta concorrência e escalabilidade.

Este trabalho visa discutir e entender o modelo não relacional, trazendo dados, vantagens e desvantagens sobre o mesmo e também sobre o modelo relacional.

### 1.1 Palavras Chave

Banco de dados, *SQL*, *NoSQL*, Relacional, Não Relacional, escalabilidade, armazenamento, vantagens, desvantagens

## 2. Introdução

Bancos de dados são entidades que permitem o armazenamento seguro dos dados de maneira estruturada, também de forma a reduzir redundâncias. Nos primórdios da computação o armazenamento de dados não trazia os recursos atuais, como explica Rhoden (2009):

*“Na década de 60 a computação como um todo consistia no desenvolvimento de aplicações extremamente individuais que eram baseadas em arquivos. Todos os registros e relatórios gerados vinham de diversos arquivos, que com o tempo de uso e o aumento dos registros começavam apresentar problemas, tais como: complexidade na manutenção dos programas; complexidade no desenvolvimento de novos programas e o sincronismo dos dados a serem atualizados.”* (RHODEN, 2009, p.1).

Com as necessidades de armazenamento de um volume cada vez maior de dados e com o surgimento de novas tecnologias surgem os Bancos de Dados; mas o que é um banco de dados?

## 2.1 O que é banco de dados

Bancos de dados foram projetados oferecendo um mecanismo para armazenar, gerenciar e recuperar informações. Eles fazem isso através de tabelas, que contém colunas e linhas. O armazenamento foi pensado de maneira estruturada e permitindo a menor redundância possível.

*"Banco de dados é um repositório de dados, projetado para suportar armazenamento, recuperação e manutenção eficiente de dados. [...] Um banco de dados pode ser especializado em armazenar arquivos binários, documentos, imagens, vídeos, dados relacionais, dados multidimensionais, dados transacionais, dados analíticos ou dados geográficos." (SHARMA, 2010, p. 23)*

## 2.2 SGBD

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é o conjunto de programas de computador responsáveis pelo gerenciamento de uma base de dados. Seu principal objetivo é retirar da aplicação cliente a responsabilidade de gerenciar o acesso, a manipulação e a organização dos dados. O SGBD disponibiliza uma interface para que seus clientes possam incluir, alterar ou consultar dados previamente armazenados.

*"Enquanto o banco de dados é um repositório de dados, um sistema gerenciador de banco de dados ou simplesmente SGBD, é um conjunto de ferramentas que controla acesso, organiza, armazena, gerencia, recupera e mantém dados em um banco de dados." (SHARMA, 2010, p. 23).*

## 2.3 Linguagem SQL

Em bancos de dados relacionais a interface é constituída pelas *APIs (Application Programming Interface)* ou drivers do SGBD, que executam comandos na linguagem *SQL (Structured Query Language)*.

Conforme nos ensina Sharma (2010, p. 115), "*Structured Query Language (SQL)* é uma linguagem de alto nível que permite usuários manipularem dados relacionais."

O *SQL* foi desenvolvido originalmente no início dos anos 70 nos laboratórios da IBM

em San Jose, que tinha por objetivo demonstrar a viabilidade da implementação do modelo relacional proposto por Edgar F. Codd. O nome original da linguagem era *SEQUEL*.

"Originalmente, a linguagem foi chamada '*Structured English Query Language*' ou *SEQUEL*, mas foi alterada para SQL pois *SEQUEL* era uma marca registrada de uma empresa inglesa." (SHARMA, 2010, p.115)

## **2.4 O modelo relacional**

"O modelo relacional é simples e elegante. Tem uma fundação matemática sólida baseada em conjuntos de teoria e cálculo de predicados e é o mais utilizado modelo de dados para bancos de dados hoje." (SHARMA, 2010, p. 29)

O primeiro conceito de banco de dados relacional foi proposto por Edgar Frank Codd, em 1936, através de um artigo.

"Uma das diretivas para a pesquisa de Codd era o fato de que programadores *IMS* estavam gastando muito tempo realizando manutenção em aplicações *IMS* quando mudanças lógicas ou físicas aconteciam." (SHARMA, 2010, p. 29)

Este modelo existe há mais de 35 anos, baseado em conjuntos de relações, é simples e permite consultas com linguagens de alto nível.

"A relação se representa mediante uma tabela, esta tabela representa ao que no modelo entidade-relação chamávamos entidade. Esta tabela contém os atributos (colunas) e as tuplas (filas)." (LANGA, 2007).

## 2.5 O modelo não relacional

O modelo não relacional, chamado de *NoSQL (Not Only SQL)*, é uma tecnologia para escalabilidade<sup>1</sup> e *storage* de dados que permite maior flexibilidade na estruturação do banco. De acordo com Gustavo Lima (2010), o termo *NoSQL* foi primeiramente utilizado em 1998 como o nome de um banco de dados relacional de código aberto que não possuía um interface *SQL*.

## 2.6 Problematização

"Nas grandes aplicações web é cada vez mais comum a quantidade de informações ser enorme, e ainda temos uma certeza: amanhã teremos mais dados para armazenar." (STEPPAT, 2009)

O modelo não relacional facilita ou resolve problemas enfrentados no uso do modelo relacional?

## 2.7 Objetivo Geral

Discutir e entender o uso do modelo não relacional.

## 2.8 Objetivos Específicos

Discutir e entender as principais características do modelo relacional.

Discutir e entender as principais características do modelo não relacional.

Comparar modelo relacional com não relacional.

Expor as principais características do ambiente em que o modelo não relacional é usado.

Apresentar as principais limitações do modelo não relacional.

---

<sup>1</sup> Escalabilidade é a característica de um sistema que indica sua habilidade de manipular uma porção crescente de trabalho de forma uniforme, ou estar preparado para crescer. GOMES, Diego 31/01/2010

### 3. Metodologia

Para realização deste trabalho, foram usados livros e trabalhos acadêmicos, além de sites e blogs mantidos por personalidades da área.

A pesquisa está escrita de forma explicativa e descritiva, para servir como guia introdutório de entendimento e consulta para qualquer pessoa da área de TI que tem interesse em conhecer ou aplicar o modelo não relacional em algum tipo de projeto.

### 4. Análise dos Dados

Banco de dados é um tópico que hoje em dia está presente em qualquer curso superior de tecnologia. Ele é apresentado geralmente através do modelo relacional, pois o mercado o usa amplamente.

*"Os Bancos de Dados Relacionais são atualmente o tipo de banco de dados mais utilizado no mundo. Felipe N.R. Machado e Maurício P. Abreu, no livro Projeto de Banco de Dados – Uma Visão Prática, os definem como conjuntos de dados vistos segundo um conjunto de Tabelas e as operações sobre elas (Tabelas) são feitas por linguagens que manipulam a álgebra relacional. Os BDRs utilizam os conceitos da Teoria dos Conjuntos e a idéia de que não é relevante ao usuário saber onde estão os dados nem como estão os dados. O usuário apenas manipula os dados dispostos em linhas e colunas de tabelas através de um conjunto de operadores e funções de alto nível constantes na Álgebra Relacional.*

*Acredito que o Modelo de Banco de Dados Relacional deve ser utilizado sempre que não houver nenhuma forte razão específica que justifique escolher outro modelo. Como disse anteriormente, o modelo relacional é, e de longe, o mais utilizado e o que menos apresenta problemas em aplicações gerais." (KRAUSE, 2009)*

#### 4.1 As vantagens do modelo relacional

Segundo Sharma (2010, p. 29), os objetivos de Edgar F. Codd ao propor o modelo relacional eram:

- Armazenar os dados em uma estrutura simples (tabelas)
- Acessá-los através de uma linguagem de manipulação de dados de alto nível
- Ser independente de armazenamento físico

O modelo relacional, através de um SGBD, oferece aos usuários processos de validação, verificação e garantias de integridade dos dados, controle de concorrência, recuperação de falhas, segurança, controle de transações, otimização de consultas, dentre outros.

“Uma vantagem adicional da visão relacional é que ela constitui uma base sólida para o tratamento de derivabilidade, redundância e consistência das relações” (Codd, 1969, p.1).

A utilização de tais recursos facilitou a vida dos desenvolvedores de aplicações, possibilitando que estes pudessem se preocupar exclusivamente com o foco da aplicação.

## 4.2 As desvantagens do modelo relacional

Ao trabalhar com banco de dados, se a aplicação tem um crescimento contínuo, e a vida da aplicação é longa, chega um momento em que a capacidade do servidor de banco de dados não é suficiente para continuar provendo o serviço, atendendo as requisições em tempo hábil, e então uma medida deve ser tomada para não deixar a máquina parada.

“Atualmente, o volume de dados de determinadas organizações, como é o caso do Google, atinge o nível de petabytes, o que corresponde a 1 125 899 906 842 624 bytes. Em cenários nos quais o gerenciamento de grande volume de dados tem se mostrado problemático, a utilização de SGBDs relacionais, ou em outras palavras do próprio Modelo Relacional, já não se mostra tão eficiente.” (BRITO, 2010, pág. 2)

Segundo Edmar Ferreira (2010), uma solução que pode ser adotada é "apelar para a Escalabilidade Vertical e fazer um *upgrade* na máquina que está rodando nosso banco de dados", mas esta é uma abordagem limitada, pois o servidor chegaria ao seu pico novamente.

A outra opção é fazer uma Escalabilidade Horizontal, na qual mais máquinas para prover o mesmo serviço trabalham em paralelo. Com esta opção, é necessário uma série de configurações nos servidores e alterações de código para que o ambiente se comporte como o esperado.

"[...] vamos perceber que é bem mais complicado do que com os nossos servidores web. Na maioria dos casos não podemos simplesmente ligar mais uma máquina rodando o banco e esperar que tudo funcione." (FERREIRA, 2010).



### 4.3 O início do modelo não relacional

As dificuldades fizeram com que desenvolvedores e projetistas pensassem em alternativas de modelagem para os banco dados e também procurassem outras formas de armazenar os dados, propondo soluções que eram como a prática antiga de gerenciamento de arquivos.

Segundo Brito (pág. 2), 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. Posteriormente, o termo passou a representar soluções que promoviam uma alternativa ao Modelo Relacional, tornando-se uma abreviação de *Not Only SQL*, sendo utilizado principalmente em casos em que o Modelo Relacional não apresentava performance adequada.

Posteriormente, o modelo não relacional tornou-se um conceito de banco de dados estruturado que permite acesso a dados através de uma chave simples.

### 4.4 Tipos de soluções não relacionais

Agora em período de maturidade da tecnologia, podemos escolher alguns tipos de soluções não relacionais, pois mesmo compartilhando algumas características em comum, há pontos chave importantes que as colocam mais adequadas para cenários específicos.

O primeiro tipo é o chave / valor, que é o tipo de banco de dados *NoSQL* mais simples, o conceito dele é uma chave e um valor para essa chave. Mas ele é o que aguenta mais carga de dados. Esses tipos de bancos de dados são o que tem a maior escalabilidade.

O segundo tipo é baseado em colunas, inspirado no *BigTable*<sup>2</sup> do Google, suportam várias linhas e colunas, além de permitir subcolunas.

---

<sup>2</sup> BigTable é um sistema distribuído de armazenamento para gerenciar dados estruturados e foi projetado para escalar o máximo possível, chegando a petabytes.

O terceiro tipo, baseado em documentos *XML*<sup>3</sup> ou *JSON*<sup>4</sup>, os dados podem ser localizados pelo seu id único ou por qualquer registro que tenha no documento.

O quarto tipo é o de grafos, que com uma complexibilidade maior, esses bancos de dados guardam objetos, e não registros como os outros tipos de NoSQL. A busca desses itens é feita pela navegação desses objetos.

#### **4.5 Desvantagens do modelo não relacional**

Mesmo sendo um modelo ideal para resolver alguns problemas, certos pontos devem ser considerados ao escolher trabalhar com esta tecnologia.

De acordo com Adamo (28/06/2011), a maioria dos sistemas *NoSQL* ainda são imaturas e precisam da implementação de muitos recursos. Também é importante considerar que a maioria dos sistemas *NoSQL* são open source, o que pode causar um problema ao precisar de suporte técnico especializado, serviço este que toda grande empresa requisita com frequência.

Os sistemas *NoSQL* têm poucas ferramentas de análise e de *query*<sup>5</sup>, tornando necessário uma bagagem técnica maior e uma maturidade alta do profissional, tanto para desenvolver quanto gerenciar, e encontrar profissionais capacitados em bancos de dados não relacionais é difícil.

#### **4.6 Relacional x Não Relacional**

“Quando se analisa a possibilidade de se optar por uma estratégia *NoSQL* em detrimento de um SGBD tradicional, é preciso levar em consideração algumas questões básicas, como, por exemplo, os critérios de escalonamento, consistência de dados e disponibilidade.” (BRITO, 2010, pág. 3)

---

<sup>3</sup> XML (*Extensible Markup Language*) é um formato para criação de documentos com dados organizados de forma hierárquica

<sup>4</sup> JSON (*Javascript Object Notation*) é um formato leve para troca de dados computacionais

<sup>5</sup> *Query* é uma forma de questionamento, que no contexto de tecnologia da informação e banco de dados, é uma linguagem de consulta aos dados armazenados

O escalonamento é o ponto de principal vantagem do não relacional em relação ao modelo relacional, pois os SGBD's possuem uma estruturação menos flexível e adaptada para cenários em que o escalonamento é importante.

Com relação a consistência de dados, o modelo relacional tem esta característica como seu ponto mais forte, pois as regras presentes propiciam um maior grau de rigor. Já o modelo não relacional tem consistência realizada de modo eventual.

*“Outro fator importante a ser considerado é a ausência de uma linguagem de consulta, como é o caso do SQL para os SGBDs relacionais. Não existe, em qualquer abordagem NoSQL, nada que se aproxime da simplicidade e expressividade oferecida pelo SQL. Adicionalmente, perde-se toda a funcionalidade oferecida pela linguagem, tais como funções, rotinas, etc. Além disso, deixa-se de utilizar a mais simples restrição de integridade sobre o banco, o que pode tornar a aplicação mais pesada.” (BRITO, 2010, pág. 6)*

O desempenho do modelo não relacional será sempre superior, pois um dos motivos é que os dados ficam guardados em memória *RAM*<sup>6</sup>, diferente do modelo relacional, que armazena dados em disco rígido.

Segundo Faure (21/12/2010), é simples observar que bancos de dados orientados a chave-valor vão fornecer desempenho superior porque têm menos funcionalidades e há menos etapas para acontecer durante o uso, como autenticação e gestão de índices.

## **5. Considerações Finais**

Apesar dos modelos relacionais e não relacionais serem bem diferentes um do outro, com pontos positivos e negativos distintos, as diferenças não são o motivo que define que nunca será possível que os modelos coexistam em um ambiente.

As diferenças de ambos complementam um ao outro, fazendo com que um entregue funcionalidades que o outro não pode, ocupando partes diferentes e importantes no cenário de um banco de dados.

O *NoSQL* pode, por exemplo, ser usado como uma camada de cache para um

---

<sup>6</sup> Memória *RAM* (*random access memory*) é um tipo de memória que permite leitura e escrita, utilizada como memória primária em sistemas eletrônicos digitais

MySQL<sup>7</sup>, que é relacional. Desta forma, a consistência de dados é preservada e a alta performance de acesso é acrescentada no ambiente, fazendo com que um sistema ou site possa atender mais usuários.

Também podemos pensar em um sistema de *chat*, em que a conversa é gerida pelo *NoSQL* mas a autenticação e os dados do usuário são geridos por um banco de dados relacional.

Acredito que o modelo não relacional veio para nos permitir alternativas e / ou complementos para soluções de banco de dados, e de maneira alguma irá acabar com o uso do modelo relacional. Ele irá facilitar e / ou resolver problemas que enfrentamos ao trabalhar com o modelo relacional e deve ser estudado por todo profissional que está de alguma forma envolvido com bancos de dados e armazenamento de informações.

---

<sup>7</sup> MySQL é um sistema gerenciador de banco de dados, que usa a linguagem SQL

## 6. Referencial Teórico

ADAMO, David. "NoSQL Database - Advantages and Disadvantages".

<http://tekedia.com/12083/nosql-database-advantages-and-disadvantages/> Acessado em: 01/08/2012.

BRITO, Ricardo. "Bancos de Dados NoSQL x SGBDs Relacionais: Análise Comparativa".

<http://www.infobrasil.inf.br/userfiles/27-05-S4-1-68840-Bancos%20de%20Dados%20NoSQL.pdf>. Acesso em: 22/02/2012.

CODD, Edgar F. "A Relational Model of Data For Large Shared Data Banks".

<http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>. Acesso em 03/10/2012.

FAURE, Frédéric. "SQL + NoSQL = YES!".

<http://highscalability.com/blog/2010/12/21/sql-nosql-yes.html>. Acesso em: 10/08/2012

FERREIRA, Edmar. "Introdução ao NoSQL".

<http://escalabilidade.com/2010/03/08/introducao-ao-nosql-parte-i/>

<http://escalabilidade.com/2010/04/06/introducao-ao-nosql-parte-ii/>

Acesso em: 01/03/2012/

GOMES, Diego. "O que é escalabilidade?".

<http://escalabilidade.com/2010/01/31/o-que-e-escalabilidade/>. Acesso em 07/11/2012.

KRAUSE, Maico. "Entrevista sobre banco de dados relacionais a TI Digital".

<http://my.opera.com/maicokrause/blog/entrevista-sobre-banco-de-dados-relacionais-a-ti-digital>. Acesso em: 05/03/2012.

LANGA, Sara Alvarez. "Estrutura do Modelo Relacional".

<http://www.criarweb.com/artigos/estrutura-modelo-relacional.html>. Acesso em: 20/02/2012.

LIMA, Gustavo. "NoSQL, A Evolução das Bases de Dados".

<http://blog.corujadeti.com.br/nosql-a-evolucao-das-bases-de-dados/>. Acesso em: 01/03/2012.

RHODEN, Rafael B. "Banco de Dados: Relacional x Multidimensional".

<http://pt.scribd.com/doc/22742853/Artigo-Banco-de-Dados-Relacional-vs-Multidimensional>. Acesso em 03/10/2012.

SHARMA, Neeraj. "Database Fundamentals".

[http://public.dhe.ibm.com/software/dw/db2/express-c/wiki/Database\\_fundamentals.pdf](http://public.dhe.ibm.com/software/dw/db2/express-c/wiki/Database_fundamentals.pdf). Acesso em 10/10/2012

STEPAT, Nico. "Bancos de Dados não relacionais e o movimento no-sql".

<http://blog.caelum.com.br/bancos-de-dados-nao-relacionais-e-o-movimento-nosql/>.

Acesso em: 01/03/2012.