Banco de dados NoSQL: A importância de projeto

Alexandre Ribeiro Nascimento[1], Walter Edmundo Souza Guedes[2]
Curso de Sistema para Internet – Instituto Federal da Paraíba (IFPB)
– Campus João Pessoa 58015 – 435 – João Pessoa – PB – Brasil
alexandrepb@gmail.com, edsouzaguedes@gmail.com

Abstract. *NoSQL databases oriented to an object to meet a need to store data of increasingly complex types, difficult to represent non-relational model. This paper intends to identify the dimension of data modeling without NoSQL project as well as the implications of the absence of the project.*

Resumo. *Os bancos de dados NoSQL orientados a objeto surgiram para atender a uma necessidade relacionada ao armazenamento de tipos de dados cada vez mais complexos, de difícil representação no modelo relacional. Esse artigo tem o intuito de identificar a importância da modelagem de dados no projeto NoSQL bem como as implicações da ausência de projeto.*

1. Introdução

O Modelo Relacional (MR), surgiu na década de setenta e tonou-se ao longo dos anos o modelo padrão de bancos de dados. Durante as últimas décadas e ainda em dias atuais, o MR é líder de mercado mundial, conforme visto na Figura 1 dos 10 principais bancos de dados usados no último ano, 7 utilizam MR, além de ser os 5 primeiros bancos mais usados.

Figura 1. 10 Banco de dados mais usados.

Rank			DBMS	Database Model	Score		
Apr 2017	Mar 2017	Apr 2016			Apr 2017	Mar 2017	Apr 2016
1.	1.	1.	Oracle	Relational DBMS	1402.00	+2.50	-65.54
2.	2.	2.	MySQL	Relational DBMS	1364.62	-11.46	-5.49
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1204.77	-2.72	+69.72
4.	4.	↑ 5.	PostgreSQL	Relational DBMS	361.77	+4.14	+58.05
5.	5.	↓ 4.	MongoDB	Document store	325.43	-1.51	+12.98
6.	6.	6.	DB2	Relational DBMS	186.66	+1.74	+2.57
7.	7.	7.	Microsoft Access	Relational DBMS	128.18	-4.76	-3.79
8.	8.	8.	Cassandra	Wide column store	126.18	-3.01	-3.49
9.	↑ 10.	9.	Redis	Key-value store	114.36	+1.35	+3.12
10.	↓ 9.	10.	SQLite	Relational DBMS	113.80	-2.39	+5.83

Fonte: DB-Engines Raking, 2017

Segundo Wanzeller (2013), uma das grandes vantagens de fazer uso dos bancos de dados relacionais, deve-se aos relacionamentos entre tabelas que auxilia na integridade dos dados armazenados no banco. É notado que para desenvolvermos um banco de dados relacional eficiente, é preciso passar pela modelagem entidade-relacionamento e modelagem relacional. Cada uma dessas etapas, produzem documentos que auxiliam no desenvolvimento do projeto de banco de dados.

Com o crescente uso da internet e de novas tecnologias, os bancos de dados relacionais têm-se mostrado problemático em atendê-las (BOSCARIOLI *et al*, 2006).

Em virtude disso, tornasse fundamental a existência de um banco de dados que atenda a necessidade de manipular grandes quantidades de dados utilizando o menor tempo possível de processamento. Assim, surgem os chamados bancos de dados NoSQL que vem do termo Not only SQL ou, “não apenas SQL”. Portanto, esses bancos de dados surgiram com conceitos de não normalização relacional e sem restrições de integridade, como as definidas no modelo relacional, mas totalmente voltados à velocidade de acesso aos dados, escalabilidade para reduzir a carga de trabalho e flexibilidade no esquema de dados (DIANA, 2010).

Apesar da variedade de modelos, todos surgiram para suprir o alto consumo de dados gerado, principalmente pela internet. Assim, é alvo desse artigo identificar qual a real importância da modelagem de dados na criação de projeto NoSQL utilizando Banco de Dados Orientados a Objetos.

2. Comparação entre Banco de dados Relacional e NoSQL

É interessante destacar que existem diversos bancos de dados utilizando diversas formas de modelagem. Ainda temos os bancos de dados relacionais como os mais utilizados, porém os modelos NoSQL, que entre os principais temos: orientados a documentos, grafos, chave-valor e orientação à objetos, vem ganhando cada vez mais espaço, devido a sua forma escalar de trabalhar dados complexos. Apesar de possuírem características específicas, os bancos de dados existem para o armazenamento de dados, que uma vez guardados, possam ser consultados, excluídos e alterados. Tudo a fim de que gerem resultados úteis para as aplicações que deles necessitam (BOSCARIOLI *et al*, 2006). Vejamos os principais modelos.

2.1 Modelo Relacional

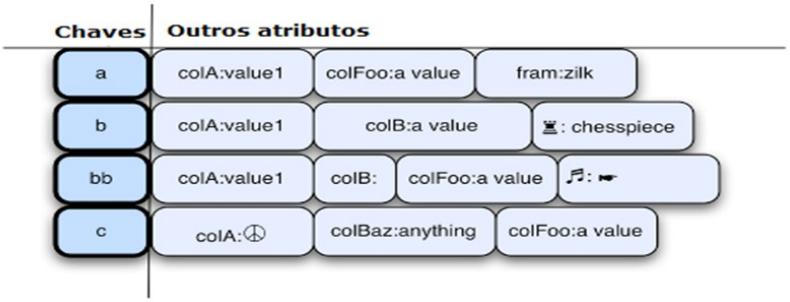
O modelo relacional tem por origem a teoria dos conjuntos e álgebra relacional. Sua modelagem de dados permite solucionar os vários problemas que se colocam no nível da concepção e implementação da base de dados (HEUSER, 1998).

A base do modelo relacional é a relação entre tabelas. Conforme verificado em Heuser (1998), uma tabela é composta por linhas e colunas, onde cada linha representa um registro exclusivo e cada coluna representa um campo no registro. É fortemente recomendado que uma tabela possua um campo de dados que a identifique, esse campo é chamando de atributo identificador, ou chave primária. Assim, uma relação entre tabelas é constituída por um ou mais atributos identificadores que ao relacionarem-se permite a integração entre tabelas, e o acesso a esses dados de forma referenciada. Porém, para trabalhar com essas tabelas, algumas restrições precisaram ser impostas para evitar aspectos indesejáveis, como: Repetição de informação, incapacidade de representar parte da informação e perda de informação. Exemplo de alguns Banco de Dados Relacional seria: *MySQL, MS-SQL-Sever* entre muitos outros.

2.2 Banco de dados chave-valor

É um dos modelos de banco de dados NoSQL mais elementares, onde sua estrutura é composta de uma chave que serve de campo identificador e o outro campo contendo o valor. Conforme vemos na Figura 2, a chave pode ser qualquer valor.

Figura 2. Chave-Valor



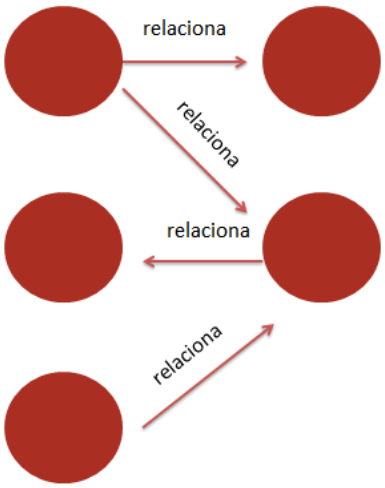
Se compararmos o modelo chave-valor com o modelo relacional, observamos que ambos se assemelham à uma tabela, sendo o campo chave o atributo identificador, porém podendo assumir qualquer valor escolhido pelo usuário (FREITAS, 2015).

O modelo de chave-valor tem como principal vantagem a facilidade na implementação e a simplicidade no armazenamento de dados. Porém, é desaconselhável, caso seja necessário executar consultas ou atualizações de parte de seus dados. Temos como banco de dados chave-valor: *Redis, Voldemort* entre outros.

2.3 Banco de dados de grafos

Entre os modelos NoSQL o modelo de grafos é o mais distinto. Seu alvo é a flexibilidade, e os relacionamentos entre os seus elementos ocorrem de forma natural, onde as entidades são descritas como vértices, que são representados pelos círculos vermelhos e os relacionamentos que são as arestas. As arestas não são apenas descritivas para o modelo, elas podem conter dados, e por ligar dois vértices a aresta contém a identificação de ambos os nós em suas pontas, garantindo assim a integridade referencial. Como vemos na Figura 3.

Figura 2. Ilustrativa – Grafo



O modelo de grafo tem como vantagem a rapidez que uma estrutura em grafo possui. Para ter uma pesquisa precisa é necessário que se percorra todo o grafo, trazendo assim uma possível desvantagem dependendo do tamanho da estrutura a ser percorrida (FREITAS, 2015). Como exemplos de banco de dados em grafo temos: *NEO4J* e *InfoGrid*

2.4 Banco de dados orientados a documentos

Um banco de dados orientados a documentos armazena os dados em conjunto de atributos e valores, porém a identificação deve ser única. Apesar de serem compostos por uma série de documentos, cada documento deve possuir sua identificação exclusiva (WANZELLER, 2013). Normalmente não dispõem de uma estrutura fixa, permitindo uma colocação continua no banco de dados, garantindo maior rapidez no registro dos dados. Alguns exemplos de banco de dados orientado a documentos são: *MongoDB* e *CouchDB*.

2.5 Banco de dados orientado à objetos

O banco de dados orientado a objetos surgiu na de década de 80 da necessidade de armazenar dados complexos, uma vez que os bancos relacionais não atendem bem essas especificidades. Ele é baseado nos conceitos de orientação a objetos, muito conhecido em linguagens de programação. Uma vez que na programação o objeto só

existe enquanto os *softwares* os executam na memória dos equipamentos, o banco de dados orientados a objetos tem a capacidade de persistir, ou seja, armazena esses objetos em um banco de dados. Os pilares básicos da orientação a objetos são: encapsulamentos, herança e o polimorfismo, que garantem uma maior flexibilidade para trabalhar com os dados persistidos e a sua consistência (BOSCARIOLI *et al*, 2006). Temos como exemplo de banco de dados orientados a objetos o *DB4O*.

Como pôde ser visto, os Banco de Dados Relacional e NoSQL possuem semelhanças e algumas características que os de diferem. Porém, ambos têm a finalidade de armazenar dados que sejam reutilizados pelos aplicativos no momento desejado. Sabe-se que os modelos relacionais utilizam de relacionamentos entre as suas tabelas, garantindo uma integração entre os dados. Já os NoSQL, tem como princípio serem não relacional, permitindo assim trabalhar com grande quantidade de dados complexos de forma rápida.

3. Comparação entre Modelagem Relacional e NoSQL – BDOO

Foram abordadas as definições de Banco de Dados Relacional e NoSQL, mostrando as vantagens de cada modelo. Nesta seção apresentaremos a comparação da modelagem de dados em Banco de Dados Relacional e Banco de Dados Orientado a Objeto e a implicação no desenvolvimento do projeto utilizando cada um dos modelos.

O termo NoSQL tem como significado “Não apenas SQL”, ou seja, são banco de dados distintos que são unidos pelo não uso da modelagem relacional em suas estruturas.

Uma vez que existem diversos modelos de dados NoSQL, abordaremos a comparação do modelo relacional com o modelo orientado a objetos.

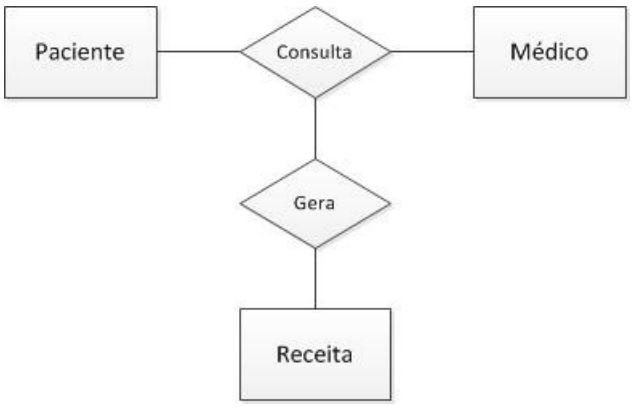
3.1 Modelos de Dados

Heuser (1998) explique que uma modelagem de dados é fundamental para a abstração de dados. Para construir um modelo de dados, é importante seguir conceitos predefinidos que auxiliem na construção de um banco de dados.

3.1.1 Modelo Entidade Relacionamento

É uma descrição do banco de dados que não reflete a implementação final do banco. Exibe, a nível do usuário, a modelagem das informações, utilizando os retângulos como entidades e os seus relacionamentos. O modelo entidade-relacionamento representa o banco de forma abstrata, conforme vemos na **Figura 3** não mostra uma representação real de um banco de dados (ITALIANO, *et al*, 2005).

Figura 3. Ilustrativa – Entidade Relacionamento

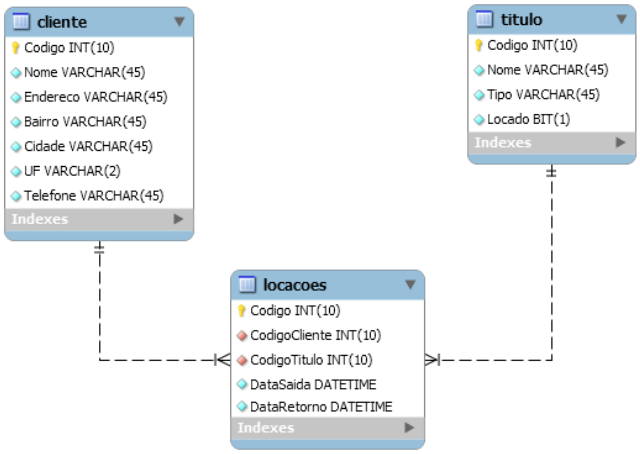


Hoje em dia, este modelo é amplamente utilizado na elaboração de um projeto de banco de dados, pois a sua estrutura é de fácil compreensão tanto por parte do usuário que auxiliam no projeto, como dos profissionais envolvidos.

3.1.2 Modelo Relacional

Heuser (1998) concluir que o modelo relacional, tem como ponto forte o relacionamento, pois os dados são organizados através de tabelas que se inter-relacionam. Como vemos na Figura 4, uma tabela é composta de linhas e colunas mantendo assim o dado contido da relação coluna *versus* linha, ficando isolado dos outros dados.

Figura 4. Ilustrativa – Modelo Relacional



Outro fato importante para obter-se uma modelagem relacional bem elaborada é a normalização. Pois através desta etapa se reduz uma tabela grande em tabelas menores e é definido os relacionamentos entre elas, bem como as demais tabelas existentes no banco de dados. Obtendo assim a redução da redundância e a dependência transitiva dos dados (HEUSER, 1998).

Observa-se através do modelo relacional uma imagem coesa da base de dados, sendo os documentos gerados nesta etapa de grande importância para a compreensão e atualizações futuras do banco de dados.

3.1.3 Modelo Orientado a Objeto

A orientação a objeto deixa o mundo real muito próximo do desenvolvedor, pois pode-se transcrever quase tudo para um objeto. Uma vez que no modelo relacional é dada a importância aos relacionamentos e a forma de armazenamento, na orientação a objeto as atenções se voltam para o objeto em si.

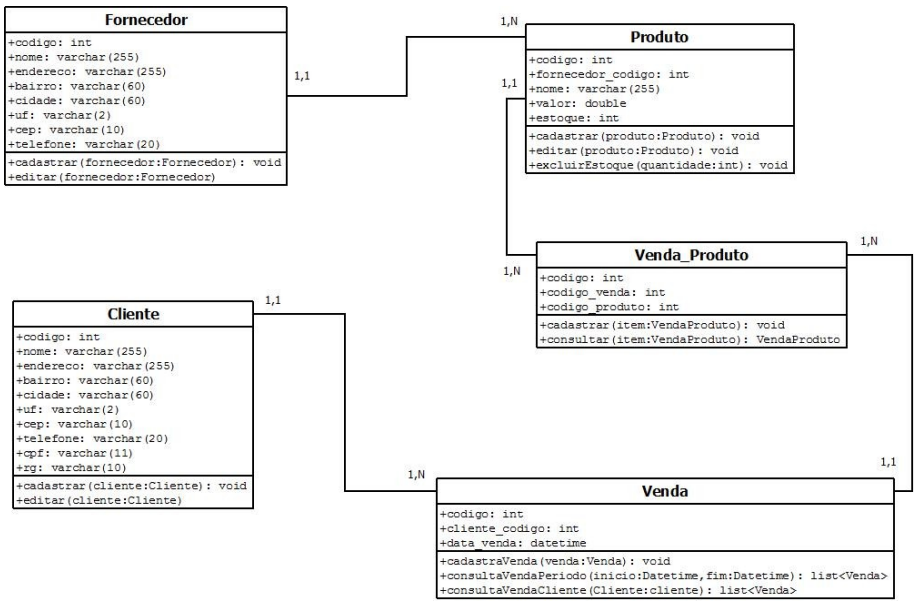
Conforme Deitel, *et al*, (2006) a orientação a objetos (OO), possuem alguns conceitos como: herança, agregação, polimorfismo e outros.

- Agregação: Consiste em um objeto obter outros objetos dentro de si;
- Herança: *"A herança é uma forma de reutilização de software em que o programador cria uma classe que absorve dados e comportamentos de uma classe existente e os aprimora com novas capacidades"* (DEITEL, 2006, p.502).

- Polimorfismo: Tratam-se de classes diferentes dispondo de métodos que possuem a mesma assinatura e se comportam de forma diferente.

Para obter-se a modelagem de uma tabela OO utilizam-se dos conceitos citados acima, bem como do uso do diagrama de classes conforme vemos na Figura 5.

Figura 5. Ilustrativa – Diagrama de Classes



O diagrama de classes é obtido através da UML (*Unified Modeling Language*). É uma linguagem que padroniza o modelo orientado a objetos, com finalidade de documentar, classificar e desenvolver sistemas, não importando a linguagem de programação (BENEGA, 2010).

Através da UML podemos melhorar a comunicação entre os envolvidos no projeto, destacar partes importantes, além de oferecer uma visão múltipla do sistema através de vários conceitos e diagramas (FOWLER, 2005).

Portanto, vemos que a OO possui uma grande variedade de recursos e mecanismos para a elaboração de um projeto. Porém não se trata de documentos específicos para a modelagem do bando de dados.

4. Conclusão

Ao analisar o modelo relacional e foi possível perceber que o mesmo possui etapas a serem observadas no desenvolvimento de um banco de dados. E em cada uma dessas etapas são gerados uma série de documentos e representações do banco de dados que ajudam na compreensão dos envolvidos, produz documentos que servem de referência em fase de projeto, bem como no futuro gerenciamento do banco de dados.

Também pode-se destacar que, por existir há mais tempo, esse modelo relacional domina o mercado, como também tem bem sido amplamente abordado em estudos.

Entretanto, observa-se que o modelo NoSQL surgiu da necessidade da existência de um banco de dados que suportasse trabalhar com uma grande quantidade de dados de forma rápida e não escalar. Uma vez que o modelo relacional apresenta essa desvantagem, devido sua forma de relacionar os seus dados. Assim surgiram diversos modelos de dados NoSQL, entre eles o abordado que é o modelo OO.

Com a comparação do modelo relacional com o modelo OO foi verificado que através da UML, que é a linguagem que padroniza a OO, pode-se obter diversas formas de documentar um banco de dados OO, entre elas, a já utilizada é o diagrama de classes. Porém no estudo para o desenvolvimento deste artigo não foi encontrado nenhum padrão especifico para a construção de um banco de dados OO, e sim conceitos da programação OO que podem ser utilizados na íntegra ou adaptados para a construção da documentação do banco de dados.

Portanto, conclui-se que por se tratar de um conceito novo, porém de vital importância, em face da demanda e da velocidade das novas tecnologias, faz-se necessária a ampliação do estudo da documentação dos bancos de dados OO.

As principais contribuições obtidas neste artigo foram a percepção da falta de modelos específicos para bancos de dados OO e a relevância da documentação para a elaboração de um projeto de banco de dados, onde essa documentação servirá de referência para futuras alterações.

Referências

BENEGA, Thiago Vicente. Monografia: Padrões de Projeto em Modelagem Orientada a Objetos Persistida em Banco de Dados Relacional, Faculdade Farias Brito. Fortaleza, 2010.

BOSCARIOLI, Clodis, BEZERRA, Anderson, BENEDICTO, Marcos de, DELMIRO, Gilliard. Uma reflexão sobre Banco de Dados Orientados a Objetos – Paraná – 2006.

CODD, EDGARD F. A Relational Model of Data for Large Shared Data Banks. Disponível em: www.unilivros.com.br/pdf/_codd_acm_1970.pdf (idioma: inglês). Acesso: janeiro/2017

DB-Engines Ranking. Disponível: http://db-engines.com/en/ranking. Acesso: janeiro/2017.

Imasters. Disponível: https://imasters.com.br/banco-de-dados/bancos-de-dados-nosql-uma-visao-geral/?trace=1519021197&source=single. Acesso: fevereiro/2017.

DEITEL, H. M.. C++ como programar. 5. ed. São Paulo: Pearson Education do Brasil, 2006

DIANA, Mauricio de, GEROSA, Marco Aurélio. NoSQL na Web 2.0: Um Estudo Comparativo de Bancos Não-Reacionais para Armazenamento de Dados Web 2.0 – São Paulo – 2010.

FREITAS, Myller Claudino. Dissertação de Mestrado: Mapeamento Conceitual entre Modelo Relacional e Estruturas NoSQL: Um estudo de caso com documentos. Dissertação de Mestrado. CIn, UFPE. Recife, 2015.

FOWLER, Martin. UML Essencial: Um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. São Paulo: Bookman, 2005. (Livros didáticos).

HEUSER, Carlos Alberto. Projeto de banco de dados. 4. ed. Porto Alegre: Sagra Luzzato, 2004. (Livros didáticos).

ITALIANO, Isabel Cristina, TAKAI, Osvaldo Kotaro, FERREIRA, João Eduardo. Apostila. Introdução a Banco de Dados. IME-USP. São Paulo, 2005.

WANZELLER, Diogo Araújo Pacheco. Monografia de conclusão de curso: Investigando o Uso de Bancos de Dados Orientados a Documentos para Gerenciar Informações da Administração Pública. Brasília, 2013.